IEC 61968-100

Edition 2.0 2022-02

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Application integration at electric utilities – System interfaces for distribution management –
Part 100: IEC implementation profiles for application integration

Intégration d'applications pour les services électriques – Interfaces système pour la gestion de la distribution –
Partie 100: Profils de mise en œuvre IEC pour l'intégration d'application



IEC 61968-100:2022-02(en-fr)

THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2022 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, …). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

A propos de l'IEC
La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC
Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC - webstore.iec.ch/advsearchform
La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études, …). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished
Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc
Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org
Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 300 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 19 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

IEC 61968-100

Edition 2.0 2022-02

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Application integration at electric utilities – System interfaces for distribution management – Part 100: IEC implementation profiles for application integration

Intégration d'applications pour les services électriques – Interfaces système pour la gestion de la distribution – Partie 100: Profils de mise en œuvre IEC pour l'intégration d'application

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

ICS 33.200

ISBN 978-2-8322-1071-4

Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.

® Registered trademark of the International Electrotechnical Commission
Marque déposée de la Commission Electrotechnique Internationale

CONTENTS

FOREWORD 14
INTRODUCTION 16
1 Scope 17
1.1 General 17
1.2 About this document 17
1.3 What is not covered by this document 17
1.4 Future considerations 17
1.4.1 Choice of Encoding Mechanisms 17
1.4.2 Choice of Web Service Technologies 17
2 Normative references 18
3 Terms, definitions and abbreviated terms 18
3.1 Terms and definitions 18
3.2 Abbreviated terms 19
4 Overview 19
4.1 General 19
4.2 Message format 19
4.3 Profiles 20
4.4 Message archetypes 20
4.5 Header verb and noun 20
4.6 Example IEC 61968-100 messages 21
4.7 Message exchange patterns 21
4.7.1 General 21
4.7.2 Request/response message exchange pattern 21
4.7.3 Unsolicited event message exchange pattern 21
4.7.4 Consequential event message exchange pattern 22
4.8 Basic request/response message types 22
5 Message archetypes 23
5.1 General 23
5.2 Request message 23
5.2.1 General 23
5.2.2 Root element 23
5.2.3 Structure 23
5.2.4 Header element 24
5.2.5 Application-specific data 24
5.2.6 Side-effects of a request message 24
5.3 Response message 24
5.3.1 General 24
5.3.2 Root element 24
5.3.3 Structure 24
5.3.4 Header element 25
5.3.5 Application-specific data and status notifications 25
5.3.6 Simple acknowledgement message 25
5.4 Event message 26
5.4.1 General 26
5.4.2 Root element 26
5.4.3 Structure 26

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

5.4.4 Header element 26
5.4.5 Application-specific data 26
5.5 Fault message 27
5.5.1 General 27
5.5.2 Root element 27
5.5.3 Structure 27
5.5.4 Reply element 27
6 Elementary message exchange patterns 27
6.1 General 27
6.2 Request/response message exchange pattern 27
6.2.1 General 27
6.2.2 Response messages 28
6.3 Unsolicited event message exchange pattern 29
6.4 Consequential event message exchange pattern 29
6.5 Simple acknowledgement message exchange patterns 30
6.5.1 General 30
6.5.2 Request/response message exchange pattern with simple acknowledgement messages 31
6.5.3 Unsolicited event message exchange pattern with simple acknowledgement message 32
6.5.4 Consequential event message exchange pattern with simple acknowledgement messages 32
7 Message details 33
7.1 General 33
7.2 Message encoding 34
7.3 Schema definition and validation 34
7.4 Top-level structure 34
7.4.1 General 34
7.4.2 Structure of request messages 34
7.4.3 Structure of response messages 37
7.4.4 Structure of event messages 39
7.5 The <Header> element 42
7.5.1 General 42
7.5.2 Message header subelements 42
7.5.3 Message header for request messages 44
7.5.4 Message header for response messages 46
7.5.5 Message header for event messages 48
7.6 The <Reply> element 50
7.6.1 General 50
7.6.2 Structure 51
7.6.3 Reply subelements 51
7.6.4 The <Reply><Status> subelement 51
7.6.5 Combining the <Reply><Status><ObjectID>, <OperarionSet>, <UpdateID> and <TransactionID> elements 58
7.6.6 The <Reply><Transaction> element 59
7.7 The <Payload> element 59
7.7.1 General 59
7.7.2 Payload subelements 59
7.7.3 The <Payload><OperationSet> element 60

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.7.4 The <Payload><ObjectUpdate> element 66
7.7.5 AbstractPayloadType data 68
7.7.6 Multiple abstract payloads 68
7.8 The <CompressedPayload> element 68
7.8.1 General 68
7.8.2 Structure 68
7.8.3 CompressedPayload elements 69
7.8.4 The <CompressedPayload><CompressionType> subelement 69
7.8.5 The <CompressedPayload><Base64> subelement 69
7.9 The Transaction Type 69
8 Constraints and conventions 70
8.1 General 70
8.2 Application-specific data 70
8.2.1 General 70
8.2.2 The AbstractPayloadType 71
8.2.3 Application-specific customisation 71
8.2.4 RDF data 71
8.3 Object management requests and notifications 71
8.3.1 General 71
8.3.2 Create request message 72
8.3.3 Change request message 72
8.3.4 Delete request message 73
8.3.5 Update request message 73
8.3.6 Object management notifications 74
8.4 Get requests 74
8.4.1 Making a get request 74
8.4.2 Specifying multiple constraints in a request message 75
8.4.3 Responding to a get request 76
8.5 Cancelling transactions 76
8.5.1 General 76
8.5.2 Message exchange pattern 76
8.5.3 Transactions IDs 77
8.5.4 Receiving a transaction ID 78
8.5.5 Sending a cancellation request message 78
8.5.6 Responding to a cancellation request message 78
8.6 Object identification 78
8.6.1 General 78
8.6.2 Rules for object identification 79
8.6.3 Identifying objects using an <mRID> element 79
8.6.4 Identifying objects using a <Name> element 79
8.6.5 Referencing objects 80
8.7 The ObjectIdentificationRevisionSets profile 81
8.7.1 General 81
8.7.2 Structure 81
8.7.3 Using the ObjectIdentificationRevisionSets profile 83
8.7.4 Adding one or more new identifiers 83
8.7.5 Removing one or more identifiers 84
8.7.6 Possible Errors Pertaining To ObjectIdentificationRevisionSets Requests 84

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

9 Transporting messages 84
9.1 General 84
9.2 Transporting over SOAP web services 85
9.2.1 General 85
9.2.2 Encoding of IEC 61968-100 messages inside SOAP messages 85
9.2.3 Example SOAP message 85
9.2.4 Mapping of IEC 61968-100 messages to SOAP messages 85
9.2.5 WSDL files 86
9.3 Transporting over JMS 87
9.3.1 General 87
9.3.2 Message exchange patterns 88
9.3.3 Application-level acknowledgement messages 88
9.3.4 JMS headers 88
Annex A (normative) XSD files 90
A.1 General 90
A.2 The Message.xsd file 90
A.3 The `AbstractPayloadType.xsd` file 102
A.4 The LegacyObjectIdentificationRevisionSets.xsd File 103
A.5 The `RDF.xsd` file 108
Annex B (normative) Template files 110
B.1 General 110
B.2 The GenericMessageWsdlTemplate.txt file 110
B.3 The `StrongMessageXsdTemplate.txt` file 116
B.4 The `StrongMessageWsdlTemplate.txt` file 128
Annex C (informative) Example XSD profiles inheriting from AbstractPayloadType 135
C.1 GetMeterReadings profile 135
C.2 BinaryData profile 136
Annex D (informative) Example IEC 61968-100:2021 messages 138
D.1 General 138
D.2 Messages dealing with object instantiation 138
D.2.1 General 138
D.2.2 Sequence diagram 138
D.2.3 Meter instantiation request (request message) 139
D.2.4 Meter instantiation response (step 4, success case) 141
D.2.5 Meter instantiation response (step 4, failure case) 141
D.3 Messages dealing with the event notification of object instantiation 142
D.3.1 General 142
D.3.2 Sequence diagram 142
D.3.3 Publication of a meter instantiation (event message) 142
D.3.4 Publication of a meter instantiation (simple acknowledgement message) 146
D.4 Messages dealing with requests for information concerning an object instantiation 146
D.4.1 General 146
D.4.2 Sequence diagram 146
D.4.3 Request for information concerning a meter instantiation (request message) 147
D.4.4 Request for information concerning a meter instantiation (response message) 148
D.5 Messages utilizing an operationset to enforce transactional integrity 149

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.5.1 General 149
D.5.2 Sequence diagram 149
D.5.3 Using an operationset to enforce transactional integrity (request message) 150
D.5.4 Using an operationset to enforce transactional integrity (response message) 153
D.6 Messages dealing with the publication of data 153
D.6.1 General 153
D.6.2 Sequence diagram 153
D.6.3 Publication of meter readings (event message) 154
D.6.4 Publication of meter readings (simple acknowledgement message) 155
D.7 Messages dealing with an on-demand data request resulting in a single response 156
D.7.1 General 156
D.7.2 Sequence Diagram 156
D.7.3 On-demand meter reading message (request message) 157
D.7.4 On-demand meter reading message (response message) 158
D.8 Messages dealing with an on-demand data request resulting in multiple responses 159
D.8.1 General 159
D.8.2 Sequence diagram 159
D.8.3 Request for an on-demand meter ping (request message) 160
D.8.4 Request for an on-demand meter ping (response message 1) 161
D.8.5 Request for an on-demand meter ping (response message 2) 162
D.8.6 Request for an on-demand meter ping (simple acknowledgement of message 2) 163
D.8.7 Request for an on-demand meter ping (response message 3) 164
D.8.8 Request for an on-demand meter ping (simple acknowledgement of message 3) 165
D.9 Messages dealing with historical data request 165
D.9.1 General 165
D.9.2 Sequence diagram 165
D.9.3 Historical meter read (request message) 166
D.9.4 Historical meter read (response message) 167
D.10 Messages dealing with historical data request (responses spread over several messages) 168
D.10.1 General 168
D.10.2 Historical meter read (response message 1 of 4) 169
D.10.3 Historical meter read (response message 2 of 4) 169
D.10.4 Historical meter read (response message 3 of 4) 170
D.10.5 Historical meter read (response message 4 of 4) 171
D.11 Messages dealing with multiple statuses within a response 172
D.11.1 General 172
D.11.2 Sequence diagram 173
D.11.3 Historical meter read (response message) 174
D.12 Messages dealing with a consequential event message 176
D.12.1 General 176
D.12.2 Sequence diagram 176
D.12.3 Request for a meter connect (request message) 177
D.12.4 Request for a meter connect (response message) 178

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.12.5 Request for a meter connect (event message) 179
D.12.6 Request for a meter connect (simple acknowledgement message) 180
D.13 Using two operationsets to sequence multiple operations with transactional integrity 180
D.13.1 General 180
D.13.2 Using two OperationSets to read two meters before disconnecting them (request message) 181
D.13.3 Using two OperationSets to read two meters before disconnecting them (response message) 182
D.14 Messages dealing with object deletion 184
D.14.1 General 184
D.14.2 Sequence diagram 184
D.14.3 Request to delete a meter (request message) 185
D.14.4 Request to delete a meter (response message) 185
D.15 Messages dealing with cancellation of a single transaction 185
D.15.1 Meter control request message 185
D.15.2 Response message containing a single transaction id 186
D.15.3 Cancellation request (request message) 187
D.15.4 Cancellation request (response message) 187
D.16 Messages dealing with cancellation of multiple transactions 188
D.16.1 Response message containing multiple transaction IDs 188
D.16.2 Cancellation request (request message) 189
D.16.3 Cancellation request (response message) 190
D.17 Messages dealing with failed XSD validation 190
D.17.1 General 190
D.17.2 Message that fails XSD validation (request message) 191
D.17.3 Message that fails XSD validation (response message) 191
D.18 Messages dealing with malformed requests 192
D.18.1 General 192
D.18.2 Message that is malformed (request message) 192
D.18.3 Message that is malformed (response message) 193
Annex E (informative) Example IEC 61968-100:2021 messages for updating objects 194
E.1 General 194
E.2 CreateMeterConfig.xml 194
E.3 ModifyFormNumber.xml 195
E.4 RemoveFormNumber.xml 196
E.5 AddMeterMultiplier.xml 196
E.6 ModifyMeterMultiplierKind.xml 197
E.7 ModifyMeterMultiplierValue.xml 198
E.8 RemoveMeterMultiplierValue.xml 199
E.9 RemoveMeterMultiplier.xml 200
E.10 AddMeterSeal.xml 201
E.11 AddMeterSealKind.xml 201
E.12 ModifyMeterSealCondition.xml 202
E.13 RemoveMeterSealCondition.xml 203
E.14 RemoveMeterSeal.xml 204
E.15 ModifyProgramId.xml 205
E.16 RemoveProgramId.xml 205
E.17 AddChannel.xml 206

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

E.18 ModifyChannelReadingType.xml 207
E.19 RemoveChannel.xml 208
E.20 AddConnectDisconnectFunction.xml 209
E.21 RemoveConnectDisconnectFunction.xml 210
E.22 DeleteMeterConfig.xml 211
E.23 DeleteMeterConfig.xml 211
E.24 CreateMeterConfigs.xml 212
E.25 MultipleObjectChanges.xml 214
E.26 MultipleObjectChangesWithOperationSet.xml 216
Annex F (informative) Examples of ObjectIdentificationRevisionSets Messages 220
F.1 General 220
F.2 Add an mRID for the meter object 220
F.3 Modify name two of the meter object 222
F.4 Add name three of the meter object 223
F.5 Remove Name One of the Meter Object 225
F.6 Modify the mRID of the meter object 226
F.7 Remove the mRID of the meter object 228
F.8 Add an mRID and modify name two of the meter object 229
F.9 Modify name two of the meter object and add name one of the customer object 230
F.10 Add name two of the customer object to carry a new IdentifiedObject.name 233
Annex G (informative) Example IEC 61968-100:2021 messages illustrating the cross-product rule 235
G.1 General 235
G.2 Specifying "OR" operations over the selection criteria 235
G.3 Specifying "AND" operations over the selection criteria 235
G.4 Specifying a combination of logical "AND" and "OR" operations 236
G.5 Example of multiple AbstractPayloadType profiles in a single message 237
Annex H (informative) Validation of received messages 239
H.1 General 239
H.2 Example message validation using Altova XMLSpy 239
H.3 Example message validation using a Java validator 240
Annex I (informative) Significant changes between IEC 61968-100:2013 and IEC 61968-100:2021 241
Bibliography 247

Figure 1 – Structure of a request message 24
Figure 2 – Structure of a ResponseMessage 25
Figure 3 – Structure of an EventMessage 26
Figure 4 – Structure of a FaultMessage 27
Figure 5 – Request/response message exchange pattern 28
Figure 6 – Unsolicited event message exchange pattern 29
Figure 7 – Consequential event message exchange pattern 30
Figure 8 – Request/response message exchange pattern with simple acknowledgement messages 31
Figure 9 – Unsolicited message exchange pattern with simple acknowledgement messages 32

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 10 – Consequential event message exchange pattern with simple acknowledgement messages 33
Figure 11 – Structure of the <CancelRequestMessage> element 35
Figure 12 – Structure of the <ChangeRequestMessage> element 35
Figure 13 – Structure of the <CreateRequestMessage> element 35
Figure 14 – Structure of the <DeleteRequestMessage> element 36
Figure 15 – Structure of the <ExecuteRequestMessage> element 36
Figure 16 – Structure of the <GetRequestMessage> element 36
Figure 17 – Structure of the <UpdateRequestMessage> element 37
Figure 18 – Structure of the <CancelResponseMessage> element 37
Figure 19 – Structure of the <ChangeResponseMessage> element 38
Figure 20 – Structure of the <CreateResponseMessage> element 38
Figure 21 – Structure of the <DeleteResponseMessage> element 38
Figure 22 – Structure of the <ExecuteResponseMessage> element 39
Figure 23 – Structure of the <GetResponseMessage> element 39
Figure 24 – Structure of the <UpdateResponseMessage> element 39
Figure 25 – Structure of the <CanceledEventMessage> element 40
Figure 26 – Structure of the <ChangedEventMessage> element 40
Figure 27 – Structure of the <CreatedEventMessage> element 41
Figure 28 – Structure of the <DeletedEventMessage> element 41
Figure 29 – Structure of the <ExecutedEventMessage> element 41
Figure 30 – Structure of the <UpdatedEventMessage> element 42
Figure 31 – Structure of the <Header> element for request messages 45
Figure 32 – Structure of the <Header> element for response messages 47
Figure 33 – Structure of the <Header> element for event messages 49
Figure 34 – Structure of the <Reply> element 51
Figure 35 – Structure of the <Reply><Status> element 52
Figure 36 – Example of error indications in a <Reply> element 58
Figure 37 – Structure of the <Payload><OperationSet> element for request messages 61
Figure 38 – Structure of the <Payload><OperationSet> element for response messages 62
Figure 39 – Structure of the <Payload><OperationSet> element for event messages 62
Figure 40 – Structure of the <Payload><OperationSet><Operation> element for request messages 64
Figure 41 – Structure of the <Payload><OperationSet><Operation> element for response messages 65
Figure 42 – Structure of the <Payload><OperationSet><Operation> element for event messages 66
Figure 43 – Structure of the <Payload><ObjectUpdate> element 67
Figure 44 – Structure of the <CompressedPayload> element 69
Figure 45 – Structure of the TransactionType 70

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 46 – Message Exchange Pattern for a Cancellation Request 77
Figure 47 – General Naming Scheme 80
Figure 48 – Structure of an <ObjectID> element 80
Figure 49 – ObjectIdentificationRevisionSets profile 82
Figure 50 – Example SOAP message 85
Figure A.1 – Contents of Message.xsd file 102
Figure A.2 – Contents of AbstractPayloadType.xsd file 102
Figure A.3 – Contents of LegacyObjectIdentificationRevisionSets.xsd file 108
Figure A.4 – Contents of the RDF.xsd file 109
Figure B.1 – Contents of GenericMessageXsdTemplate.txt file 115
Figure B.2 – Contents of StrongMessageXsdTemplate.txt file 128
Figure B.3 – Contents of StrongMessageWsdlTemplate.txt file 134
Figure C.1 – Example XSD profile illustrating inheritance from `AbstractPayloadType` 136
Figure C.2 – Example XSD profile suitable for the transmission of binary data 137
Figure D.1 – Example sequence diagram for a meter instantiation request 139
Figure D.2 – Example meter instantiation request message 141
Figure D.3 – Example meter instantiation response message (success case) 141
Figure D.4 – Example meter instantiation response message (failure case) 141
Figure D.5 – Example sequence diagram for the publication of a meter instantiation 142
Figure D.6 – Example publication of a meter instantiation (event message) 146
Figure D.7 – Example publication of a meter instantiaion (simple acknowledgement message) 146
Figure D.8 – Example sequence diagram for requests for information concerning an object instantiation 147
Figure D.9 – Example request for information concerning a meter instantiation (request message) 148
Figure D.10 – Example request for information concerning a meter instantiation (response message) 149
Figure D.11 – Example sequence diagram for an OperationSet request 150
Figure D.12 – Example OperationSet message to enforce transactional integrity (request message) 153
Figure D.13 – Example OperationSet message to enforce transactional integrity (response message) 153
Figure D.14 – Example sequence diagram for the publication of meter readings 154
Figure D.15 – Example publication of routine meter readings (event message) 155
Figure D.16 – Example publication of routine meter readings (simple acknowledgement message) 156
Figure D.17 – Example sequence diagram for the retrieval of on-demand meter readings 157
Figure D.18 – Example on-demand meter readings request message 158
Figure D.19 – Example on-demand meter readings response message 159
Figure D.20 – Example sequence diagram for an on-demand meter ping request 160
Figure D.21 – Example on-demand meter ping request (request message) 161
Figure D.22 – Example on-demand meter ping request (response message) 162
Figure D.23 – Example on-demand meter ping request (response message 2) 163

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure D.24 – Example on-demand meter ping request (simple acknowledgement message 2) 164
Figure D.25 – Example on-demand meter ping request (response message 3) 164
Figure D.26 – Example on-demand meter ping request (simple acknowledgement message 3) 165
Figure D.27 – Example sequence diagram for the retrieval of historical meter readings 166
Figure D.28 – Example historical meter read request (request message) 167
Figure D.29 – Example historical meter read request (response message) 168
Figure D.30 – Example historical meter read request (response message 1 of 4) 169
Figure D.31 – Example historical meter read request (response message 2 of 4) 170
Figure D.32 – Example historical meter read request (response message 3 of 4) 171
Figure D.33 – Example historical meter read request (response message 4 of 4) 172
Figure D.34 – Example sequence diagram for the retrieval of historical meter readings (request message) 173
Figure D.35 – Example historical meter read request, mixed success and failure cases (request message) 174
Figure D.36 – Example historical meter read request, mixed success and failure cases (response message) 176
Figure D.37 – Example sequence diagram for the requesting a meter connect 177
Figure D.38 – Example meter connect (request message) 178
Figure D.39 – Example meter connect (response message) 179
Figure D.40 – Example meter connect (event message) 180
Figure D.41 – Example meter connect (simple acknowledgement message) 180
Figure D.42 – Using two OperationSets to read two meters before disconnecting them (request message) 182
Figure D.43 – Using two OperationSets to Read two meters before disconnecting them (response message) 183
Figure D.44 – Example sequence diagram for the deletion of a meter 184
Figure D.45 – Example meter deletion (request message) 185
Figure D.46 – Example meter deletion (response message) 185
Figure D.47 – Example create(EndDeviceControls) request message 186
Figure D.48 – Example response message containing a single transaction ID 187
Figure D.49 – Example cancellation request message 187
Figure D.50 – Example response to a cancellation request message 188
Figure D.51 – Example response message containing several transaction IDs 189
Figure D.52 – Example cancellation request message 190
Figure D.53 – Example response to a cancellation request message 190
Figure D.54 – Example message that fails XSD validation (request message) 191
Figure D.55 – Example message that fails XSD validation (response message) 192
Figure D.56 – Example malformed message (request message) 192
Figure D.57 – Example malformed message (response message) 193
Figure E.1 – CreateMeterConfig.xml 195
Figure E.2 – ModifyFormNumber.xml 196
Figure E.3 – RemoveFormNumber.xml 196
Figure E.4 – AddMeterMultiplier.xml 197

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure E.5 – ModifyMeterMultiplierKind.xml 198
Figure E.6 – ModifyMeterMultiplierValue.xml 199
Figure E.7 – RemoveMeterMultiplierValue.xml 200
Figure E.8 – RemoveMeterMultiplier.xml 200
Figure E.9 – AddMeterSeal.xml 201
Figure E.10 – AddMeterSealKind.xml 202
Figure E.11 – ModifyMeterSealCondition.xml 203
Figure E.12 – RemoveMeterDealCondition.xml 204
Figure E.13 – RemoveMeterSeal.xml 204
Figure E.14 – ModifyProgramId.xml 205
Figure E.15 – RemoveProgramId.xml 206
Figure E.16 – AddChannelId.xml 207
Figure E.17 – ModifyChannelReadingType.xml 208
Figure E.18 – RemoveChannel.xml 209
Figure E.19 – AddConnectDisconnectFunction.xml 210
Figure E.20 – RemoveConnectDisconnectFunction.xml 211
Figure E.21 – DeleteMeterConfig.xml 211
Figure E.22 – DeleteMeterConfig.xml 212
Figure E.23 – CreateMeterConfigs.xml 214
Figure E.24 – MultipleObjectChanges.xml 216
Figure E.25 – MultipleObjectChangesWithOperationSet.xml 219
Figure F.1 – Add an mRID for the Meter Object 221
Figure F.2 – Modify name two of the meter object 223
Figure F.3 – Add name three of the meter object 224
Figure F.4 – Remove name one of the meter object 226
Figure F.5 – Modify the mRID of the meter object 227
Figure F.6 – Remove the mRID of the meter object 228
Figure F.7 – Add an mRID and modify name two of the meter object 230
Figure F.8 – Add name one of the customer object and modify name two of the meter object 232
Figure F.9 – Add name two of the customer object to carry a new IdentifiedObject.name 234
Figure G.1 – Example get(MeterConfig) request illustrating a logical "OR" operation 235
Figure G.2 – Example get(MeterReadings) request illustrating a logical "AND" operation 236
Figure G.3 – Example request message illustrating combined logical "AND" and "OR" operations 237
Figure G.4 – Example of multiple AbstractPayloadType profiles in a single message 238
Figure H.1 – Example message suitably modified to specify the locations of the relevant XSD files 240
Figure H.2 – Example of local modifications made to the Message.xsd file 240

Table 1 – List of top-level root elements in request messages 34
Table 2 – List of top-level root elements in response messages 37
Table 3 – List of top-level root elements in event messages 40

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table 4 – Elements in the Message <Header> element 42
Table 5 – Verbs used in request messages 45
Table 6 – Verbs used in event messages 49
Table 7 – Subelements within the <Reply> element 51
Table 8 – Subelements within the <Reply><Status> element 53
Table 9 – Values of the <Reply><Status><Level> element 54
Table 10 – Normative values for the <Reply><Status><Code> element 54
Table 11 – Subelements within the <Payload> element 60
Table 12 – Subelements within the <Payload><OperationSet> element 61
Table 13 – Subelements within the <Payload><OperationSet><Operation> element 63
Table 14 – Subelements within the <Payload><ObjectUpdate> element 67
Table 15 – Subelements within the <Payload><ObjectUpdate><UpdateAction> element 68
Table 16 – Subelements within the <CompressedPayload> element 69
Table 17 – Values of the <CompressedPayload><CompressionType> element 69
Table 18 – Elements of a transaction type 70
Table 19 – Subelements within an <ObjectID> element 81
Table 20 – Elements of an ObjectIdentificationRevisionSets profile 83
Table 21 – Generic WSDL template variable names 86
Table 22 – Strongly-typed message XSD template variable names 87
Table 23 – Strongly-typed WSDL template variable names 87
Table F.1 – Initial naming states of the example meter object 220
Table F.2 – Initial naming states of the example customer object 220
Table F.3 – Add an mRID for the meter object 221
Table F.4 – Modify name two of the meter object 222
Table F.5 – Add name three of the meter object 224
Table F.6 – Remove name one of the meter object 225
Table F.7 – Modify the mRID of the meter object 227
Table F.8 – Remove the mRID of the meter object 228
Table F.9 – Add an mRID and modify name two of the meter object 229
Table F.10 – Modify name two of the meter object 231
Table F.11 – Add name one of the customer object 231
Table F.12 – Add name two of the customer object to carry a new IdentifiedObject.name 233
Table I.1 – Significant changes introduced in IEC 61968-100:2021 since IEC 61968-100:2013 242
Table I.2 – Detailed changes introduced from IEC 61968-100:2013 to IEC 61968-100:2021 243
Table I.3 – Elements removed from IEC 61968-100:2013 244

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

APPLICATION INTEGRATION AT ELECTRIC UTILITIES – SYSTEM INTERFACES FOR DISTRIBUTION MANAGEMENT –

Part 100: IEC implementation profiles for application integration

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as closely as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

IEC 61968-100 has been prepared by IEC technical committee 57: Power systems management and associated information exchange. It is an International Standard.

Future standards in this series will carry the new general title as cited above. Titles of existing standards in this series will be updated at the time of the next edition.

This IEC 61968-100:2022 edition cancels and replaces the IEC 61968-100:2013 edition published in 2013. This edition constitutes a technical revision.

a) This edition includes the following significant technical changes with respect to the previous edition: IEC 61968-100:2022 has been refined to remove ambiguities and redundancies. Informative and normative clauses are marked as such and, in the latter case, appropriate verbal forms of language are used;
b) The messages defined by IEC 61968-100:2022 are in general not backwards compatible with those of IEC 61968-100:2013. Annex I lists the significant technical changes introduced since the publication of IEC 61968-100:2013 and provides appropriate suggestions for migrating to IEC 61968-100:2022.

The text of this International Standard is based on the following documents:

Draft	Report on voting
57/2446/FDIS	57/2455/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

A list of all parts in the IEC 61968 series, published under the general title *Application integration at electric utilities – System interfaces for distribution management*, can be found on the IEC website.

NOTE

The following print types are used:

- Literal text such as program fragments or XML elements in `Courier New 10`.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

INTRODUCTION

This document specifies how implementations of IEC 61968 may be made interoperable using common enterprise integration technologies. In particular, it describes how message payloads defined in IEC 61968 (Parts 3 to 9 and Part 13), IEC 61970 and IEC 62325 are to be exchanged between systems using transport mechanisms such as web services and the Java Message Service.

This document may also find usage beyond information exchanges defined for these International Standards such as for the integration of market systems or for general enterprise integration.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

APPLICATION INTEGRATION AT ELECTRIC UTILITIES – SYSTEM INTERFACES FOR DISTRIBUTION MANAGEMENT –

Part 100: IEC implementation profiles for application integration

1 Scope

1.1 General

This part of IEC 61968 defines how messages may be exchanged between cooperating systems in order to facilitate the transfer of application-specific data. Such application-specific data include but are not limited to the message payloads defined in IEC 61968 (Parts 3 to 9 and Part 13), IEC 61970 and IEC 62325.

1.2 About this document

This document provides normative definitions for:

- a set of message archetypes (Clause 5);
- a set of message exchange patterns that both sending and receiving systems are expected to implement (Clause 6);
- the exact format of the messages that are to be transmitted over the various integration technologies including a precise description of the information that each message must contain (Clause 7);
- a set of constraints and conventions to which applications must adhere in order to facilitate message exchange using IEC 61968-100 (Clause 8);
- the details of how IEC 61968-100 messages should be implemented using various underlying transport mechanisms (Clause 9).

1.3 What is not covered by this document

Security considerations lie outside the scope of IEC 61968-100. This document defers to the IEC 62351 series for definitions and practices relating to the secure transmission of messages.

1.4 Future considerations

1.4.1 Choice of Encoding Mechanisms

This document prescribes XML as the normative encoding mechanism for all messages defined by this document.

Future editions of IEC 61968-100 may specify additional normative encoding methods including support for JSON (RFC 4627) documents whose semantics are defined by the IEC CIM and whose syntax is defined by an IETF JSON schema.

1.4.2 Choice of Web Service Technologies

This document provides normative definitions for the use of SOAP Web Services (9.2) and Java Message Service (9.3) for the transport of messages.

Future editions of IEC 61968-100 may specify additional normative web service technologies such as REST.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62325-301:2018, *Framework for energy market communications – Part 301: Common information model (CIM) extensions for markets*

IEC TS 61968-2:2011, *Application integration at electric utilities – System interfaces for distribution management – Part 2: Glossary*

IEC 61968-9:2013, *Application integration at electric utilities – System interfaces for distribution management – Part 9: Interfaces for meter reading and control*

IEC 61968-11:2013, *Application integration at electric utilities – System interfaces for distribution management – Part 11: Common information model (CIM) extensions for distribution*

IEC 61970-301:2020, *Energy management system application program interface (EMS-API) – Part 301: Common information model (CIM) base*

IEC 61970-552:2016, *Energy management system application program interface (EMS-API) – Part 552: CIMXML Model exchange format*

ISO IEC 21320-1:2015, *Information technology – Document Container File – Part 1: Core*

ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*

Java Message Service (version 1.0.2 November 9, 1999). Available from https://docs.oracle.com/cd/E19957-01/816-5904-10/816-5904-10.pdf

RFC 1952, GZIP file format specification version 4.3. Available from https://tools.ietf.org/html/rfc1952.

RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace. Available from https://tools.ietf.org/html/rfc4122

RFC 4648, The Base16, Base32, and Base64 Data Encodings. Available from https://tools.ietf.org/html/rfc4648

SOAP (Simple Object Access Protocol). Available from http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this specification, the terms and definitions given in IEC TS 61968-2:2011 apply.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

3.2 Abbreviated terms

The following abbreviated terms are used within this document:

AMI	Advanced Metering Infrastructure
CIM	Common Information Model
CIS	Customer Information System
HES	Head End System
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
JMS	Java Message Service
JSON	JavaScript Object Notation
MDMS	Meter Data Management System
mRID	CIM master Resource IDentifier
RDF	Resource Description Framework
REST	REpresentational State Transfer
RFC	Request For Comments
SOAP	Simple Object Access Protocol
URL	Uniform Resource Locator
WS	Web Services
WSDL	Web Services Definition Language
XML	eXtensible Markup Language
XSD	XML Schema Definition

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

4 Overview

4.1 General

This clause provides an overview of this document. Details of the concepts described in this clause are provided in the subsequent clauses of this document.

4.2 Message format

An IEC 61968-100 message is an XML document. As such, it conforms to the `Message.xsd` XML Schema Definition (XSD) file that also forms part of this document (Clause A.2) and is capable of being successfully validated against it (Annex H).

Annex D, Annex E, Annex F and Annex G provide informative examples of IEC 61968-100 messages. These examples illustrate that an IEC 61968-100 message comprises both header information and application-specific data.

The header information is contained within a `<Header>` element inside the message. This header provides meta-data that describe how the remainder of the message is to be interpreted and processed. For example, the header shows whether the message is a command, a request for data, a reply to a previous message or an asynchronous notification.

The application-specific data are located within the <Reply> or <Payload> elements, depending on the purpose of the message. The <Reply> element conveys the status such as success, warning or error indications associated with a previous request back to the requesting system. The <Payload> element contains application-specific data conforming to profiles defined in other IEC standards.

This document makes no attempt to define these application-specific data. They are left to be defined by the relevant IEC working groups or the local organisation. However, this document imposes certain constraints and conventions to which the messages conveying such data shall adhere (Clause 8).

4.3 Profiles

The application-specific data generally conform to a profile. A profile is an XML schema definition (XSD) that has been defined by the corresponding IEC working group or local organisation. A profile is typically but not necessarily derived from the Common Information Model (CIM) as defined in standards such as IEC 62325-301, IEC 61968-11 and IEC 61970-301.

A profile generally falls into one of the following categories:

- It may correspond to a class, possibly with other associated classes, that belong to the CIM and which are related to one another by means of inheritance or association.

  EXAMPLE 1 An IEC 61968-9:2013 MeterReadings profile holds information relating to the data read from a smart meter. MeterReading is a class defined in the CIM.

  EXAMPLE 2 An IEC 61968-9:2013 CustomerConfig profile holds information pertaining to customer management. Customer is a CIM class but CustomerConfig is not.

- A profile may describe the linkages among CIM objects rather than the objects themselves.

  EXAMPLE 3 An IEC 61968-9:2013 MasterDataLinkageConfig profile specifies the relationship between various CIM objects such as Customer, Meter, UsagePoint and so on. However, MasterDataLinkageConfig is not itself a CIM class.

- A profile may describe a set of query parameters by which a request for one or more existing CIM objects may be made. In this document these are referred to as application-specific query parameters.

  EXAMPLE 4 An IEC 61968-9:2013 GetMeterReadings profile allows query parameters to be defined by which data read from one or more meters may be retrieved.

- A local organisation may define its own set of profiles.These have meaning only within that organisation.

4.4 Message archetypes

Messages are categorised according to whether they are request messages, response messages or event (asynchronous notification) messages. These categories are termed the message archetypes (Clause 5).

4.5 Header verb and noun

The message archetype can be inferred from the header verb which is carried in the <Header><Verb> element. The verb is one of a fixed set of enumerated verbs that are listed in clauses 7.5.3.3, 7.5.4.3 and 7.5.5.3.

A profile has a name by which it is identified in the message header – for example, MeterReadings, CustomerConfig or MasterDataLinkageConfig. This name is termed the header noun and is located in the <Header><Noun> element. (The precise rules for specifying the noun are more complicated than this brief description. Subclauses 7.5.3.4, 7.5.4.4 and 7.5.5.4 provide the full details).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

The header verb and header noun together determine the purpose of the message. They are designated by the terminology `verb(noun)`.

4.6 Example IEC 61968-100 messages

Subclause D.7.3 illustrates a simple IEC 61968-100 message that shows the `<Header>` and `<Payload>` elements. In this example, the application-specific data in the `<Payload>` element are defined according to IEC 61968-9:2013 and form a request for retrieving meter read data.

Subclause D.7.4 depicts the corresponding response message and shows an IEC 61968-100 message with `<Header>`, `<Reply>` and `<Payload>` elements. The `<Reply>` element contains a status notification, in this case a notification of success. The `<Payload>` element contains the meter read data that are being returned as the result of the previous request. These application-specific data are also defined according to IEC 61968-9:2013.

4.7 Message exchange patterns

4.7.1 General

This document defines several basic kinds of message exchange patterns (Clause 6). Any exchange of IEC 61968-100 messages between cooperating systems is composed of one or more of these message exchange patterns.

4.7.2 Request/response message exchange pattern

The request/response message exchange pattern (6.2) is used when a requesting system needs to request data from or that an action be taken by a responding system. The responding system processes the request and sends one or more response messages back to the requesting system.

The `<Header><Verb>` in the request message must be one of `cancel`, `create`, `delete`, `change`, `execute`, `get` or `update`. The header verb in the response message is always `reply`.

EXAMPLE 1 An IEC 61968-9:2013 `create(CustomerConfig)` message represents a request from one system to another to instantiate a CIM Customer object using the data supplied in the payload. The noun in this message is CustomerConfig and the CustomerConfig profile is used in the payload section. On receipt of such a message, the receiving system processes it and then sends a `reply(CustomerConfig)` message back to the request originator. This response message shows the status of the create request.

Messages that illustrate a similar example may be found in Clause D.2.

In distinction to the conventional request/response paradigm, this document permits multiple response messages to be associated with a single request. This may be used in cases where larger response payloads need to be separated into multiple messages or where there are multiple operations occuring in parallel and independently of one another.

EXAMPLE 2 An IEC 61968-9:2013 get(MeterReadings) request denotes that the given message is a request for specific meter reading data. The name of the profile in the request message is GetMeterReadings. The name of the profile in the corresponding reply is MeterReadings. The receipt of a get(MeterReadings) request can cause a head end system (HES) to request data from meters in its AMI network. There may be some delay as the data from the meters are returned back to the HES. The HES may decide to wait until data from all meters are received before bundling these together into a single reply(MeterReadings) message back to the request originator. Alternatively, it may elect to send data back to the requestor as each meter responds separately over the AMI network. In the latter case, the single request message may result in multiple reply(MeterReadings) messages being generated and sent to the requestor.

4.7.3 Unsolicited event message exchange pattern

The unsolicited event message exchange pattern (6.3), sometimes known as ‘fire and forget’, consists of a single message sent from one system to another and for which no response is expected. This may occur when the message is sent as the result of some autonomous action within a system. Such a message is termed an unsolicited event message.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

The <Header><Verb> in an unsolicited event message belongs to one of the set canceled, created, changed, deleted, executed or updated.

EXAMPLE A HES may send out an IEC 61968-9:2013 created(EndDeviceEvents) message as a result of a meter suffering a power outage or in order to send a high temperature alarm.

A full example may be found in Clause D.6.

4.7.4 Consequential event message exchange pattern

The consequential event message exchange pattern (6.4) comprises the request/response followed by the unsolicited event message exchange pattern. In this case, the associated event message is termed a consequential event message. The complete sequence of messages comprises the original request, a single response that denotes the status of that request and then zero or more consequential event messages.

EXAMPLE A meter data management system (MDMS) may send an IEC 61968-9:2013 create(EndDeviceControls) message to a HES to request an operation be performed on meters connected to its AMI network. The HES verifies the request and returns a reply(EndDeviceControls) message back to the MDMS. This response message denotes whether the create command can be carried out successfully or whether it contains errors – for example, references to non-existent meters – or other informational conditions. It then sends the appropriate commands over the AMI network to the meters. When these react to the commands, they respond back to the HES with the results of the command. The HES in turn sends one or more created(EndDeviceEvents) messages back to the MDMS to inform it that the meters have changed state.

A full example may be found in Clause D.12.

4.8 Basic request/response message types

An IEC 61968-100 message appearing in a request/response message exchange pattern falls into one of a set of categories according to its <Header><Verb>.

- A message with a verb of create is a command to instantiate a new object on the receiving system. The act of instantiating an object may, as a side-effect, also cause the receiving system to perform some external action.

  EXAMPLE 1 A customer information system (CIS) may send information to a MDMS using an IEC 61968-9:2013 create(CustomerConfig) message. The MDMS then uses the supplied information to instantiate the new customer in its local database.

  EXAMPLE 2 An MDMS may send an IEC 61968-9:2013 create(MeterReadSchedule) message to a HES to request it to perform some future meter reading operations.

- A message where the verb is change is a command to replace an existing object in its entirety on the receiving system. This may cause some external actions to be performed.

  EXAMPLE 3 An IEC 61968-9:2013 change(CustomerConfig) message may be used to instruct a CIS to replace the information in its local database.

- A message where the verb is delete is a command to completely remove an existing object from the receiving system. This may cause some external actions to be performed.

  EXAMPLE 4 An IEC 61968-9:2013 delete(CustomerConfig) message may be used to instruct a CIS to remove the corresponding customer entries from its local database.

  EXAMPLE 5 An MDMS may send an IEC 61968-9:2013 delete(MeterReadSchedule) message to a HES to request it to remove a meter reading schedule.

- A message with a verb of get is used to query a receiving system thereby causing one or more objects to be returned. The message supplies query parameters in its <Payload> section and the receiving system returns the requested data accordingly.

  EXAMPLE 6 An IEC 61968-9:2013 get(CustomerConfig) message may cause a CIS to return information about the requested customers from its local database.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

- A message with a verb of `execute` is used to group several commands together into one message. Transactional integrity may be optionally enforced such that either all the commands succeed or none of them do. The verb execute is used only in conjunction with the noun OperationSet (7.5.3, 7.7.3).
- A message with a verb of `cancel` is a command for cancelling an operation in progress.

  EXAMPLE 7 An IEC 61968-9:2013 `cancel(EndDeviceControls)` message may be used to cancel a pending operation resulting from a previous `create(EndDeviceControls)` message.

- A message with a verb of `update` is a command to make partial modifications to an existing object (8.3.5). The `update` verb is new to this edition of IEC 61968-100.

  EXAMPLE 8 An IEC 61968-9:2013 update(CustomerConfig) message may be used to instruct a CIS to update selected information such as a customer's telephone number in its local database.

- A message with a verb of `reply` is used for one system to inform another that it has received a command or request in a message containing one of the other verbs. The response message may include any requested data as well as any application status notifications.

  EXAMPLE 9 An IEC 61968-9:2013 `reply(MeterReadings)` message may be sent from a HES to a MDMS to provide meter readings in response to a previous IEC 61968-9:2013 `get(MeterReadings)` request. It may also include error notifications – for instance, if the original request specifies invalid meters.

5 Message archetypes

5.1 General

This document defines four broad kinds of message: request messages, response messages, event messages and fault messages (5.2, 5.3, 5.4, 5.5). These message kinds are termed the message archetypes.

Every IEC 61968-100 message shall conform to one of these archetypes.

5.2 Request message

5.2.1 General

An IEC 61968-100 request message shall be used for sending a request from a requesting system to a responding system.

A request message shall be used solely within the request/response (6.2) or the consequential event (6.4) message exchange patterns.

5.2.2 Root element

The name of the root element of a request message depends on the value of the `<Header><Verb>` element in the message (7.5.3.3) and shall be taken from one of the root element names listed in Table 1.

This rule shall not apply in the case of strongly-typed web services (9.2.5).

5.2.3 Structure

The structure of a request message is depicted in Figure 1.

NOTE The generic name RequestMessage appearing in Figure 1 denotes one of the request element names listed in Table 1.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

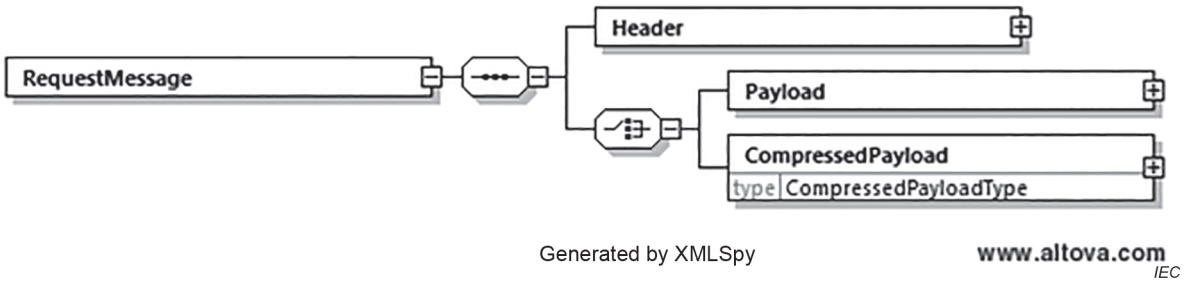


Figure 1 – Structure of a request message

A request message shall contain a `<Header>` subelement (7.5) together with either a `<Payload>` subelement (7.7) or a `<CompressedPayload>` subelement (7.8).

5.2.4 Header element

Subclause 7.5.2 specifies the `<Header>` element as it pertains to request messages.

5.2.5 Application-specific data

Any application-specific data that are associated with the request shall be carried in the `<Payload>` or `<CompressedPayload>` subelement (7.7, 7.8).

5.2.6 Side-effects of a request message

The reception of an request message by a responding system may cause that system to carry out some application-specific actions. Such actions or side-effects are non-normative and are beyond the scope of this document. They are left to be defined by the relevant working groups and application developers.

5.3 Response message

5.3.1 General

An IEC 61968-100 response message shall be used for sending a reply from a responding system to a requesting system that has previously sent a request message.

A response message shall be used within the request/response (6.2) or the consequential event (6.4) message exchange patterns. A response message may also be used for the acknowledgement of a message receipt where required by the underlying transport (6.5).

5.3.2 Root element

The name of the root element of a response message depends on the value of the `<Header><Verb>` element of the original request message (7.5.3.3) and shall be taken from one of the root element names listed in Table 2.

This rule shall not apply in the case of strongly-typed web services (9.2.5).

5.3.3 Structure

The structure of a response message is depicted in Figure 2.

NOTE The generic name ResponseMessage appearing in Figure 2 denotes one of the response element names listed in Table 2.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

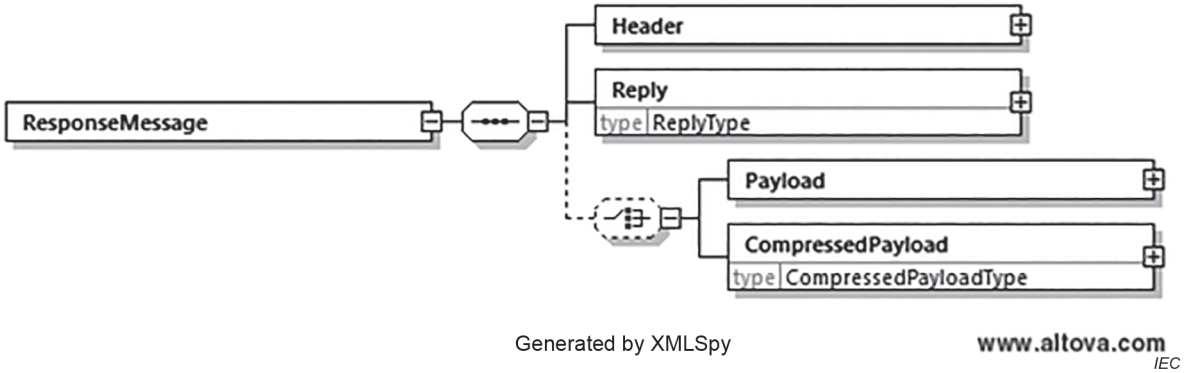


Figure 2 – Structure of a ResponseMessage

A response message shall contain both `<Header>` and `<Reply>` subelements (7.5, 7.6). It may also contain a `<Payload>` subelement (7.7) or a `<CompressedPayload>` subelement (7.8).

5.3.4 Header element

Subclause 7.5.2 specifies the `<Header>` element as it pertains to response messages.

5.3.5 Application-specific data and status notifications

A response message shall contain status indications. It may also contain application-specific data.

The application-specific data, if present, shall be placed in the `<Payload>` (7.7) or `<CompressedPayload>` subelement (7.8).

Any status indications shall be placed in the `<Reply><Status>` elements (7.6.4).

In the case of a response message returning notifications of success and no informational notifications, a `<Reply>` element containing an explicit indication of success shall be included in the message (7.6.4.3).

EXAMPLE 1 D.4.3 provides an example of a response message being used to convey application-specific data.

EXAMPLE 2 D.11.3 provides an example of a response message being used to convey both application-specific data and error notifications.

5.3.6 Simple acknowledgement message

A simple acknowledgement message is a specialised response message that is used solely to convey the receipt of a message (6.5).

The following elements shall be set in a simple acknowledgement message:

- The `<Header><Verb>` shall be `reply`.
- The `<Header><Noun>` shall have the same value as that of the original message for which the simple acknowledgement is being sent.
- A single `<Reply><Status>` element shall be included in the simple acknowledgement message (7.6.4). This shall contain `<Reply><Status><Level>` and `<Reply><Status><Code>` subelements with the values `OK` and `0.3` respectively.
- It may also contain an optional `<Reply><Status><Reason>` element containing some arbitrary text.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

No application-specific data shall be conveyed inside a simple acknowledgement message.

5.4 Event message

5.4.1 General

An IEC 61968-100 event message shall be used for sending data from one system to another when no response is expected or allowed. It shall be used for the publication of data or for the notification of an occurrence or incident.

An event message shall be sent either in the context of the unsolicited event (6.3) or the consequential event (6.4) message exchange patterns.

5.4.2 Root element

The name of the root element of an event message depends on the value of the `<Header><Verb>` element in the message (7.5.5.3) and shall be taken from one of the enumerated values listed in Table 3.

This rule shall not apply in the case of strongly-typed web services (9.2.5).

5.4.3 Structure

The structure of an event message is depicted in Figure 3.

NOTE The generic name EventMessage appearing in Figure 3 denotes one of the element names listed in Table 3.

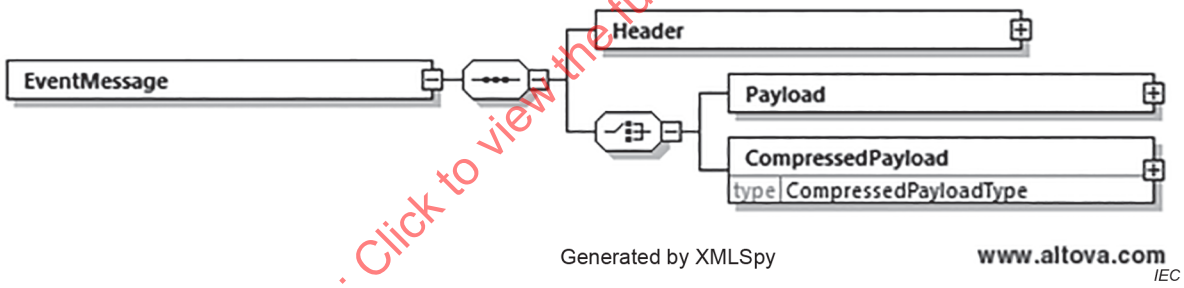


Figure 3 – Structure of an EventMessage

An event message shall contain a `<Header>` subelement (7.5) together with either a `<Payload>` subelement (7.7) or a `<CompressedPayload>` subelement (7.8).

5.4.4 Header element

Sublause 7.5.2 specifies the `<Header>` element as it pertains to event messages.

5.4.5 Application-specific data

The application-specific data shall be placed in the `<Payload>` element (7.7.3).

EXAMPLE D.3.3 provides an example of an event message.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

5.5 Fault message

5.5.1 General

An IEC 61968-100 fault message shall be used solely in the case where a recipient system has received a malformed request message; that is, a request message that cannot be parsed as a valid XML document. In such a case, the responding system shall return a fault message to the requesting system.

A fault message shall not be used for any other kind of error detected by the recipient system.

XML schema validation errors shall invoke a response message containing a `<Reply><Status><Code>` element with an appropriate value (7.6.4.7).

EXAMPLE D.18.3 provides an example of a fault message.

5.5.2 Root element

A fault message shall contain `<FaultMessage>` as its root element. This is depicted in Figure 4.

5.5.3 Structure

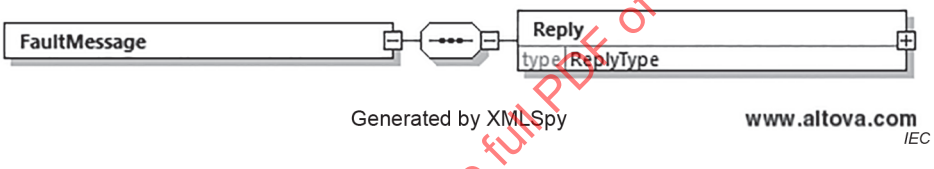


Figure 4 – Structure of a FaultMessage

5.5.4 Reply element

Subclause 7.6.4.5 specifies the `<Reply>` element as it pertains to fault messages

6 Elementary message exchange patterns

6.1 General

This clause describes the set of IEC 61968-100 elementary message exchange patterns. Any exchange of IEC 61968-100 messages between cooperating systems shall be composed of one or more of these elementary message exchange patterns.

6.2 Request/response message exchange pattern

6.2.1 General

A request/response message exchange pattern shall consist of a single request message (5.2) sent from a requesting system to a responding system followed by one or more response messages (5.3) sent from the responding system to the requesting system.

EXAMPLE D.1 provides an example of a message exchange based on the request/response message exchange pattern.

The request/response message exchange pattern is depicted in Figure 5.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

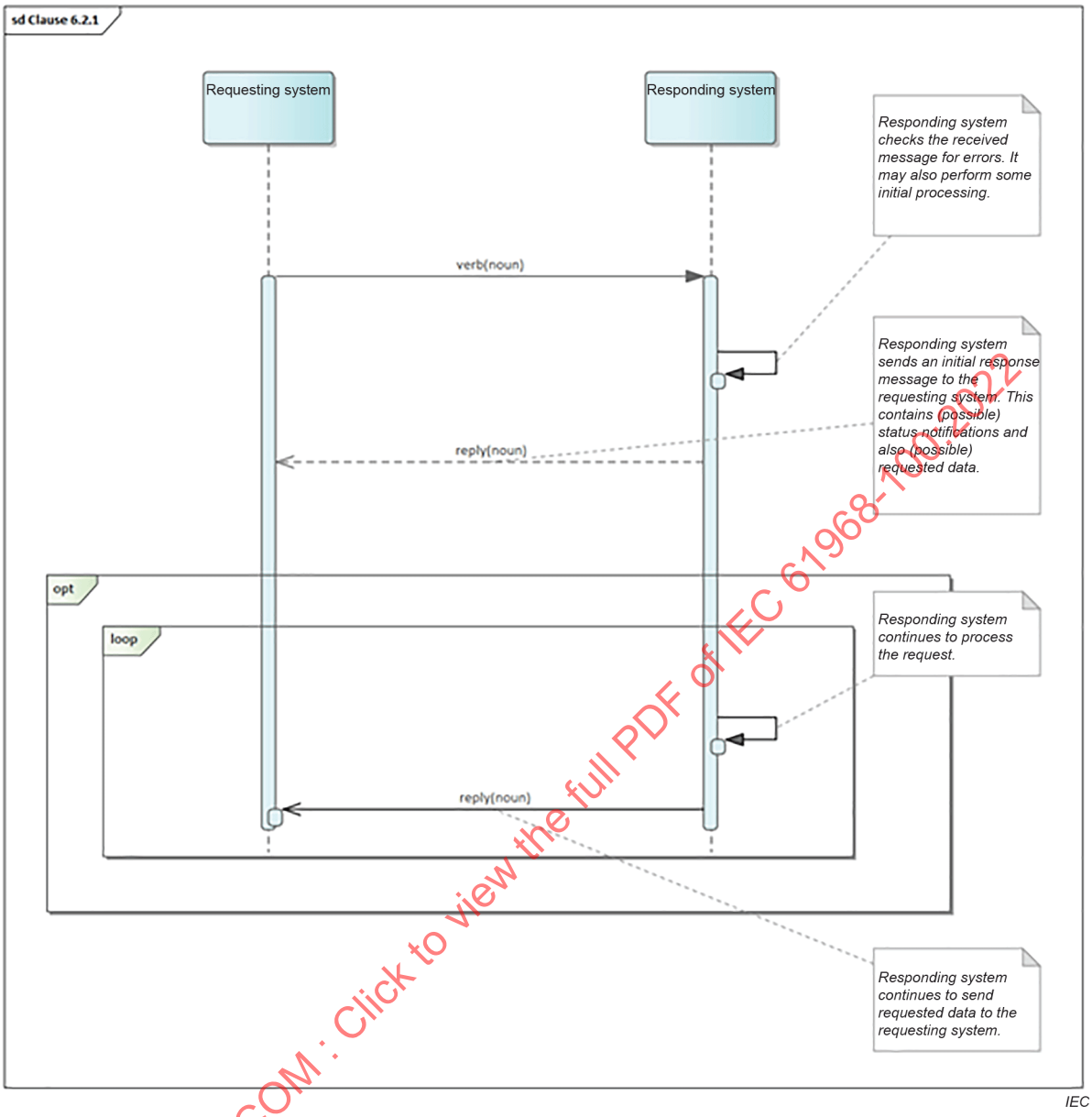


Figure 5 – Request/response message exchange pattern

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

6.2.2 Response messages

The response message (5.3) of a request/response message exchange pattern is used to convey status indications (7.6.4) and application-specific response data (7.7).

A single request message may cause the responding system to generate either a single response message or several response messages at its discretion. Possible reasons for a responding system to generate multiple response messages to a single request include the following:

- The data to be returned are too large to fit inside a single message and hence must be split up across multiple messages.
- The data to be returned are not all immediately available. That is, timing issues may determine the way in which the requested data are returned to the requesting system.

EXAMPLE An example of a single request causing multiple response messages to be generated is shown in D.10.3.

6.3 Unsolicited event message exchange pattern

An unsolicited event message exchange pattern shall be used to send a notification from one system to another in those cases when no application-level response is expected or required. An unsolicited event message exchange pattern shall consist of a single event message (5.4).

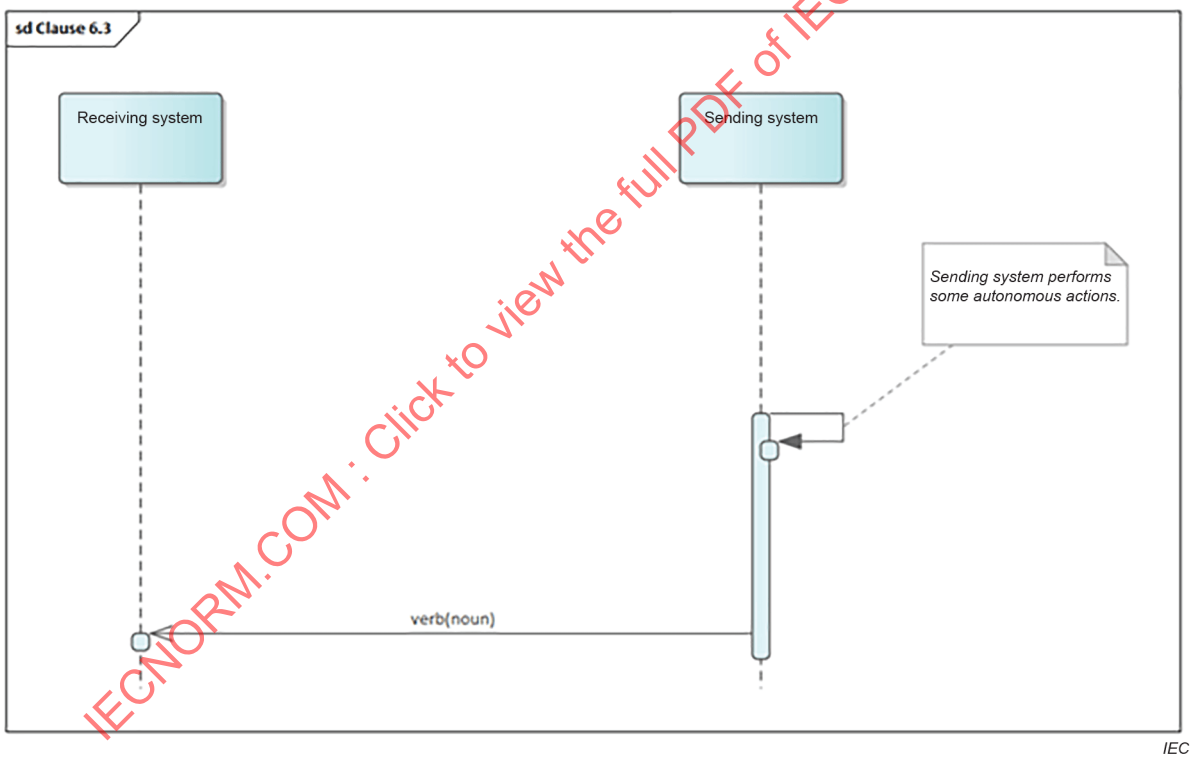
An unsolicited event message shall be sent as a result of some autonomous action in a system and not as the direct result of a previous request.

The unsolicited event message exchange pattern is depicted in Figure 6.

EXAMPLE D.6 provides an example of an unsolicited event message.

If a sending system sends a message to a receiving system that is intended to cause some operation to take place and the sending system needs to know whether that operation succeeds or fails, then it shall use the request/response message exchange pattern and not the unsolicited event message exchange pattern.

An unsolicited event message should be sent only when the underlying operation has completed.



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 6 – Unsolicited event message exchange pattern

6.4 Consequential event message exchange pattern

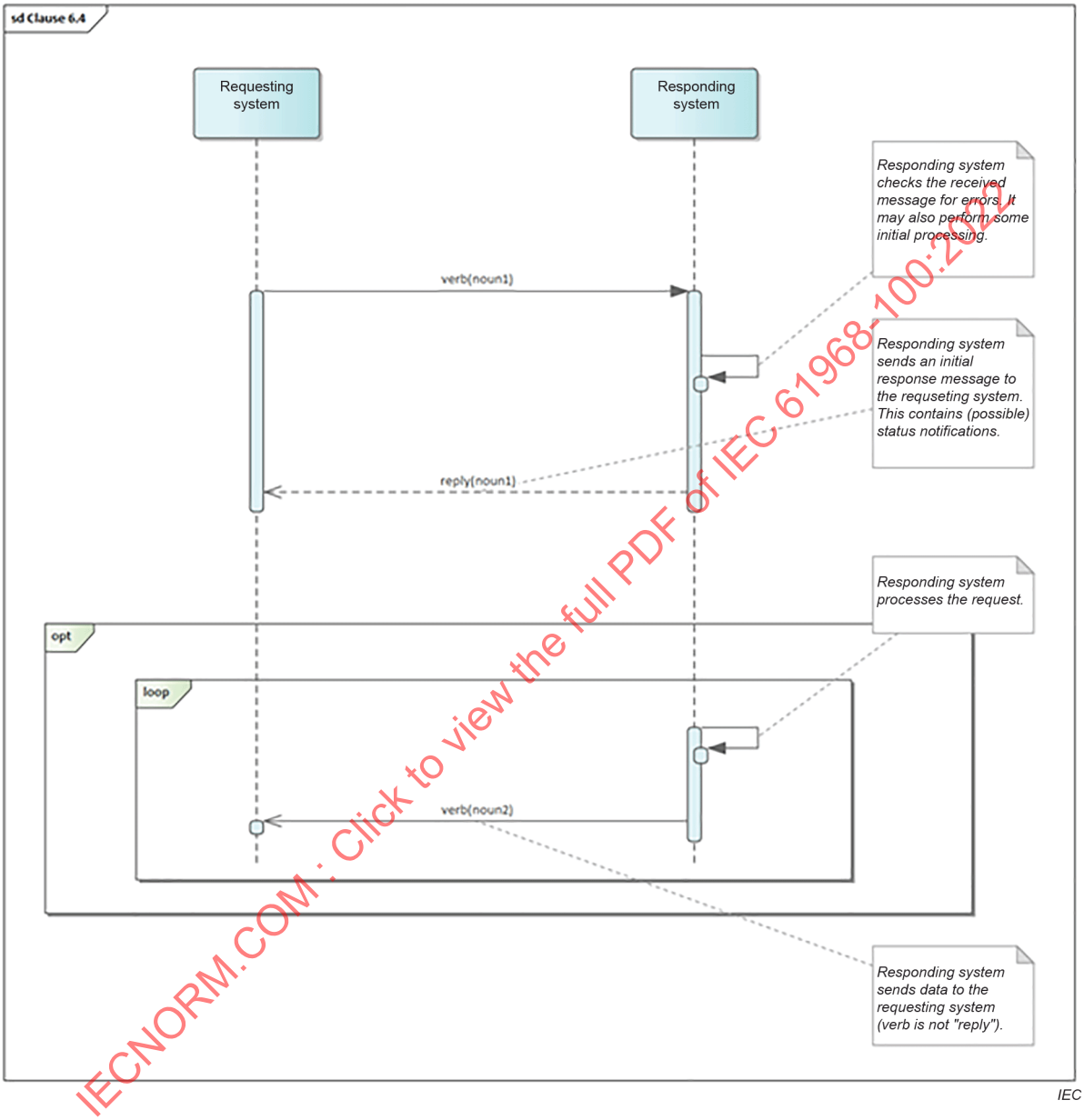
The consequential event message exchange pattern shall consist of a request message (5.2) followed by a single response message (5.3) followed by zero or more event messages (5.4).

A consequential event message exchange pattern differs from the unsolicited event message exchange pattern in that the event being communicated is the result of a previous request message.

The consequential event message exchange pattern is depicted in Figure 7.

EXAMPLE D.12 provides an example of a message exchange based on the consequential event message pattern.

NOTE The messages being conveyed in a request/response message exchange pattern share the same `<Header><Noun>` element (7.5.2). For the consequential event message pattern, however, the `<Header><Noun>` element of the event messages is different to those of the initial request/response. Furthermore, the `<Header><Verb>` element of the event messages is not "reply".



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 7 – Consequential event message exchange pattern

6.5 Simple acknowledgement message exchange patterns

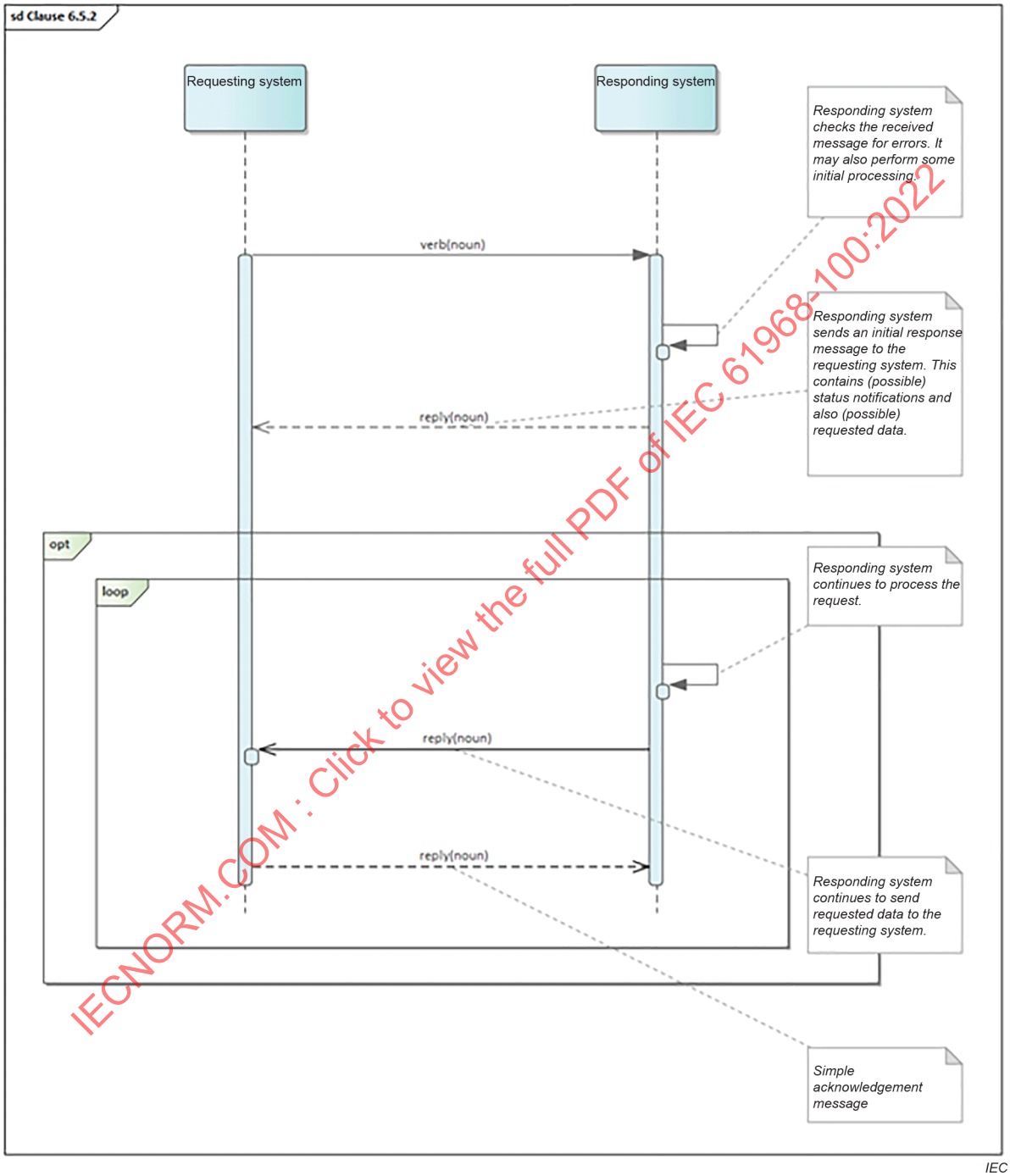
6.5.1 General

A responding system shall send a simple acknowledgement message in either one of the following circumstances:

- The responding system has received a request message where the `<Header><AckRequired>` element is present and has the value `true` (7.5.2).
- The responding system has received a response message (5.2) or an event message (5.4) and the underlying transport requires that a transport-level acknowledgement of the response or event message be sent (9.2.4).

6.5.2 Request/response message exchange pattern with simple acknowledgement messages

The request/response message exchange pattern of Figure 5 is modified as shown in Figure 8 when a simple acknowledgement message is to be sent.



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 8 – Request/response message exchange pattern with simple acknowledgement messages

NOTE If the requesting system detects an error in the reply message received from the responding system, it may replace the simple acknowledgement message with a status code of 0.3 by a reply message containing a status code corresponding to the error encountered.

6.5.3 Unsolicited event message exchange pattern with simple acknowledgement message

The unsolicited event message exchange pattern of Figure 6 is modified as shown in Figure 9 when a simple acknowledgement message is to be sent.

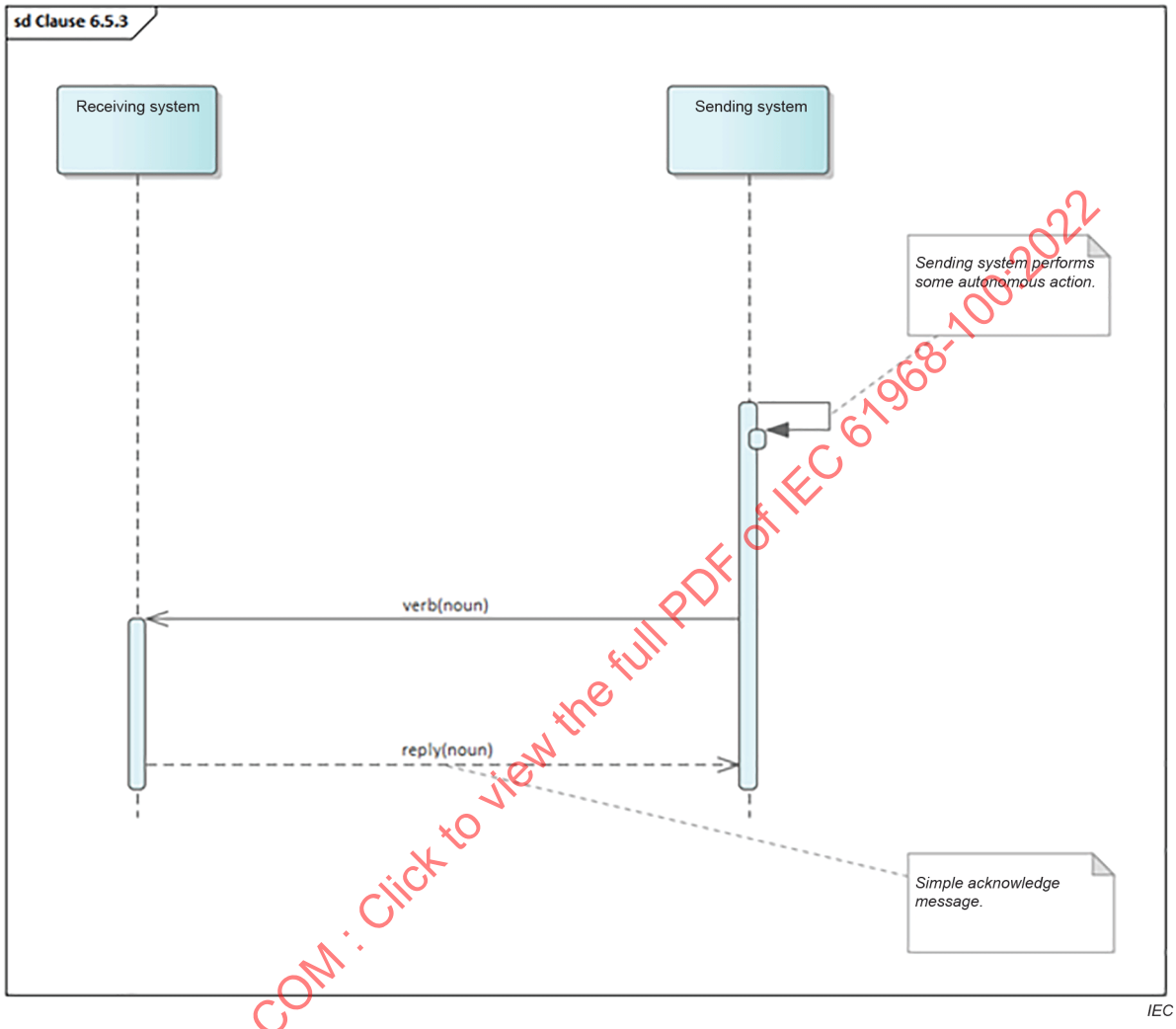


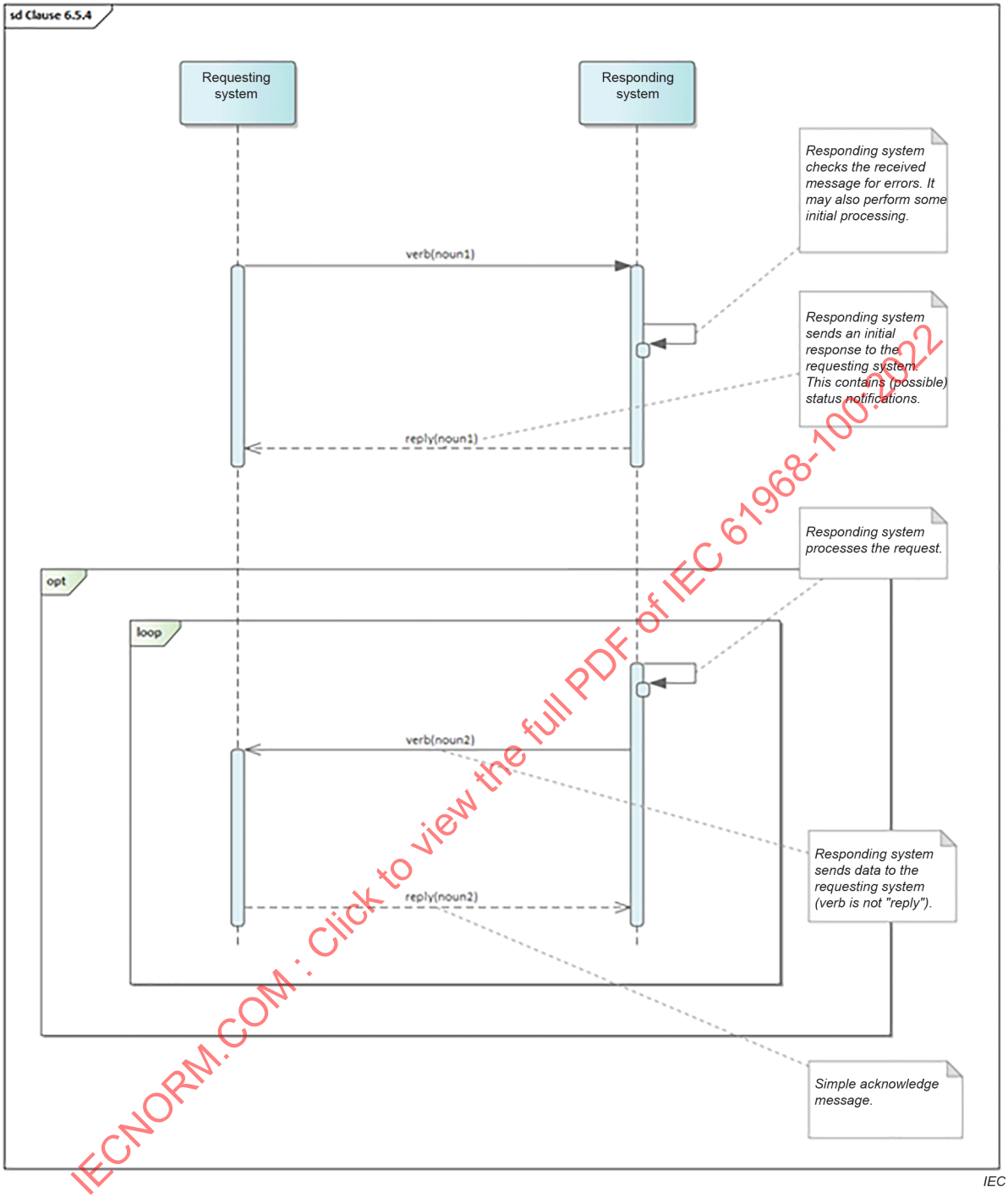
Figure 9 – Unsolicited message exchange pattern with simple acknowledgement messages

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

NOTE If the requesting system detects an error in the unsolicited event message received from the responding system, it may replace the simple acknowledgement message with a status code of 0.3 by a reply message containing a status code corresponding to the error encountered.

6.5.4 Consequential event message exchange pattern with simple acknowledgement messages

The consequential event message exchange pattern of Figure 7 is modified as shown in Figure 10 when simple acknowledgement messages are to be sent.



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 10 – Consequential event message exchange pattern with simple acknowledgement messages

NOTE If the requesting system detects an error in the consequential event message received from the responding system, it may replace the simple acknowledgement message with a status code of 0.3 by a reply message containing a status code corresponding to the error encountered.

7 Message details

7.1 General

This clause describes in detail the format of all IEC 61968-100 messages.

7.2 Message encoding

An IEC 61968-100 message shall be composed as an XML document.

7.3 Schema definition and validation

The `Message.xsd` XML schema definition file listed in Clause A.2 defines the precise syntactic structure of any IEC 61968-100 message.

An IEC 61968-100 message shall be capable of successful XML schema validation against this XSD file (Annex H).

NOTE Other standards may impose additional constraints beyond IEC 61968-100.

7.4 Top-level structure

7.4.1 General

This clause specifies the top-level structure of the IEC 61968-100 message kinds according to their respective root elements.

Other root element names shall be used in the case of strongly-typed web services (9.2.5).

7.4.2 Structure of request messages

7.4.2.1 General

This clause describes the structure of the request messages (5.2) according to their root elements.

Table 1 lists the possible message root elements and their respective `<Header><Verb>` elements (7.5.3.3).

Table 1 – List of top-level root elements in request messages

`<Header><Verb>` Element	Corresponding Root Element	See Clause
`cancel`	`<CancelRequestMessage>`	7.4.2.2
`change`	`<ChangeRequestMessage>`	7.4.2.3
`create`	`<CreateRequestMessage>`	7.4.2.4
`delete`	`<DeleteRequestMessage>`	7.4.2.5
`execute`	`<ExecuteRequestMessage>`	7.4.2.6
`get`	`<GetRequestMessage>`	7.4.2.7
`update`	`<UpdateRequestMessage>`	7.4.2.8

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.4.2.2 Structure of the `<CancelRequestMessage>` element

The top-level structure of a request message (5.2) whose root element is `<CancelRequestMessage>` is shown in Figure 11.

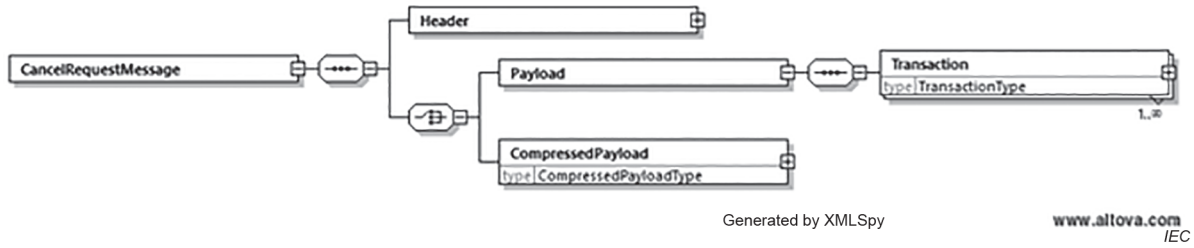


Figure 11 – Structure of the <CancelRequestMessage> element

7.4.2.3 Structure of the <ChangeRequestMessage> element

The top-level structure of a request message whose root element is <ChangeRequestMessage> is shown in Figure 12.

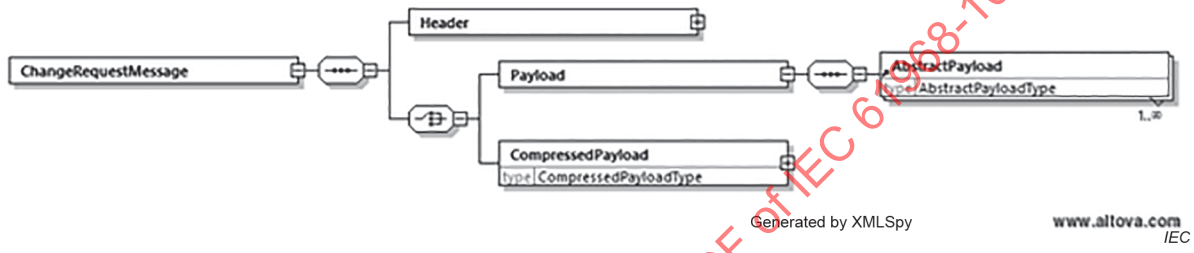


Figure 12 – Structure of the <ChangeRequestMessage> element

7.4.2.4 Structure of the <CreateRequestMessage> element

The top-level structure of a request message whose root element is <CreateRequestMessage> is shown in Figure 13.

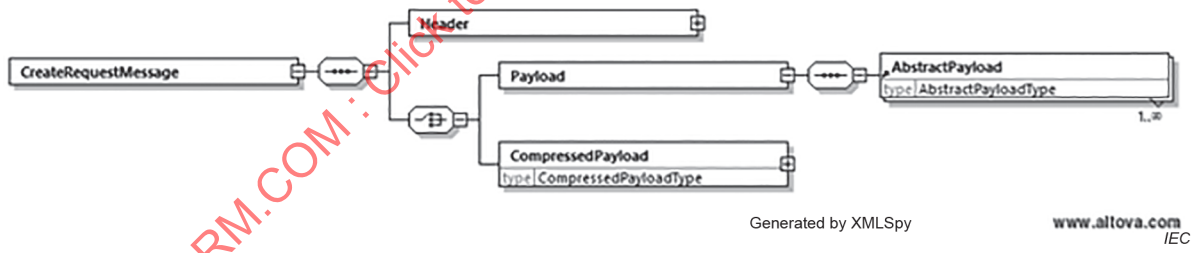


Figure 13 – Structure of the <CreateRequestMessage> element

7.4.2.5 Structure of the <DeleteRequestMessage> element

The top-level structure of the request message whose root element is <DeleteRequestMessage> is shown in Figure 14.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

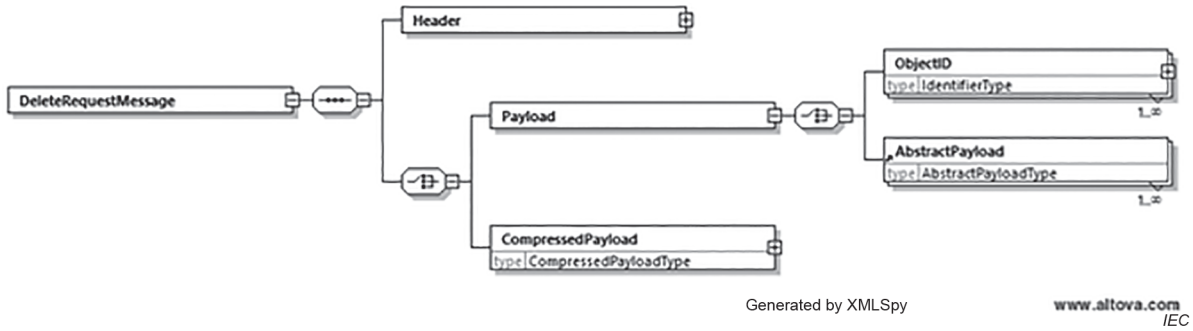


Figure 14 – Structure of the <DeleteRequestMessage> element

7.4.2.6 Structure of the <ExecuteRequestMessage> element

The top-level structure of a request message whose root element is <ExecuteRequestMessage> is shown in Figure 15.

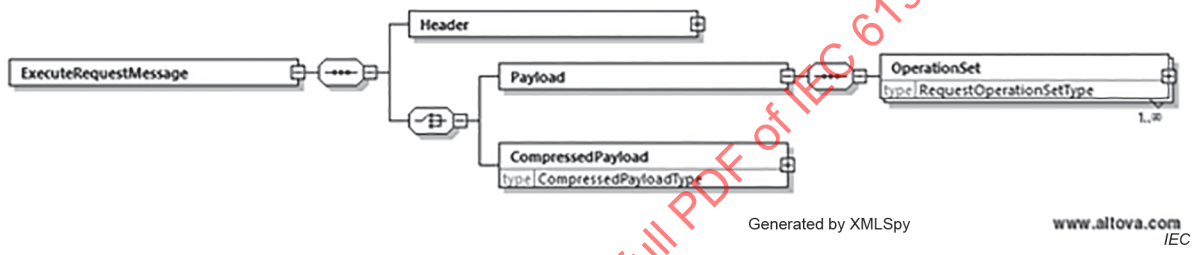


Figure 15 – Structure of the <ExecuteRequestMessage> element

7.4.2.7 Structure of the <GetRequestMessage> element

The top-level structure of a request message whose root element is <GetRequestMessage> is shown in Figure 16.

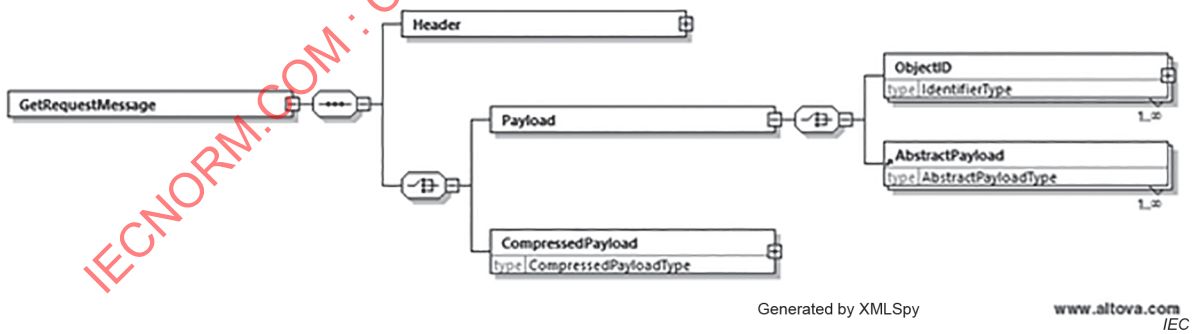


Figure 16 – Structure of the <GetRequestMessage> element

7.4.2.8 Structure of the <UpdateRequestMessage> element

The top-level structure of the request message whose root element is <UpdateRequestMessage> is shown in Figure 17.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

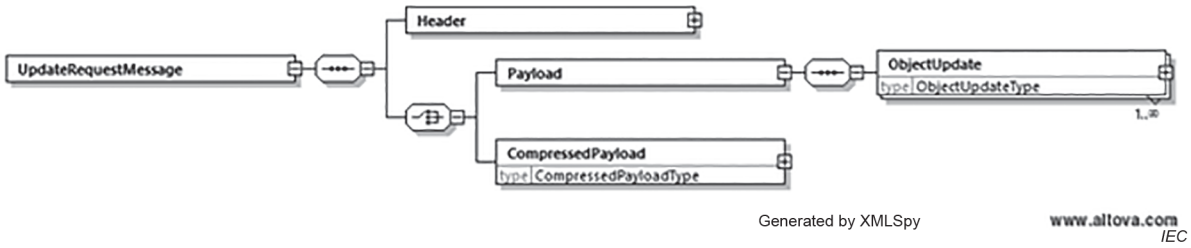


Figure 17 – Structure of the <UpdateRequestMessage> element

7.4.3 Structure of response messages

7.4.3.1 General

This clause describes the top-level structures of the response messages (5.3) according to their root elements.

Table 2 lists the possible message root elements as defined by the <Header><Verb> elements of the original request message (7.5.3.3).

Table 2 – List of top-level root elements in response messages

<Header><Verb> Element in Request Message	Corresponding Root Element	See Clause
cancel	<CancelResponseMessage>	7.4.3.2, 8.5
change	<ChangeResponseMessage>	7.4.3.3, 8.3.3
create	<CreateResponseMessage>	7.4.3.4, 8.3.2
delete	<DeleteResponseMessage>	7.4.3.5, 8.3.4
execute	<ExecuteResponseMessage>	7.4.3.6, 7.7.3
get	<GetResponseMessage>	7.4.3.7, 8.4
update	<UpdateResponseMessage>	7.4.3.8, 8.3.5

7.4.3.2 Structure of the <CancelResponseMessage> element

The structure of the response message whose root element is <CancelResponseMessage> is shown in Figure 18.

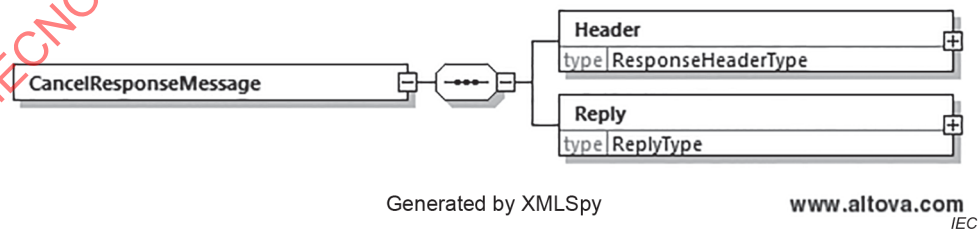


Figure 18 – Structure of the <CancelResponseMessage> element

7.4.3.3 Structure of the <ChangeResponseMessage> element

The structure of the response message whose root element is <ChangeResponseMessage> is shown in Figure 19.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

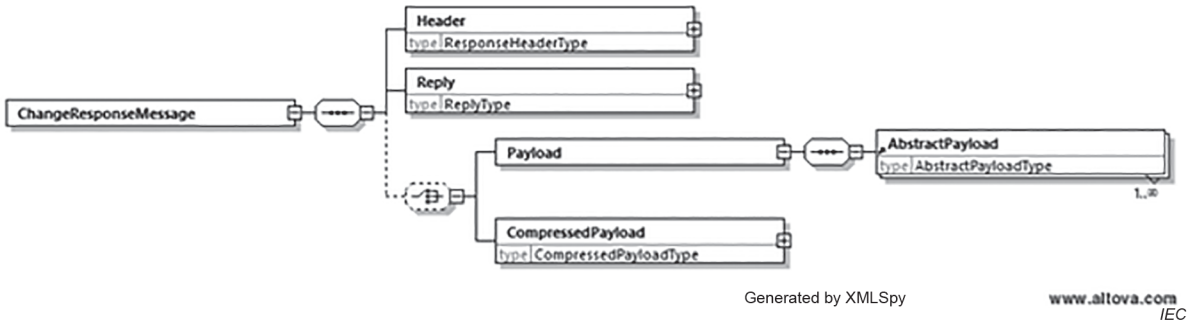


Figure 19 – Structure of the <ChangeResponseMessage> element

7.4.3.4 Structure of the <CreateResponseMessage> element

The structure of the response message whose root element is <ChangeResponseMessage> is shown in Figure 20.

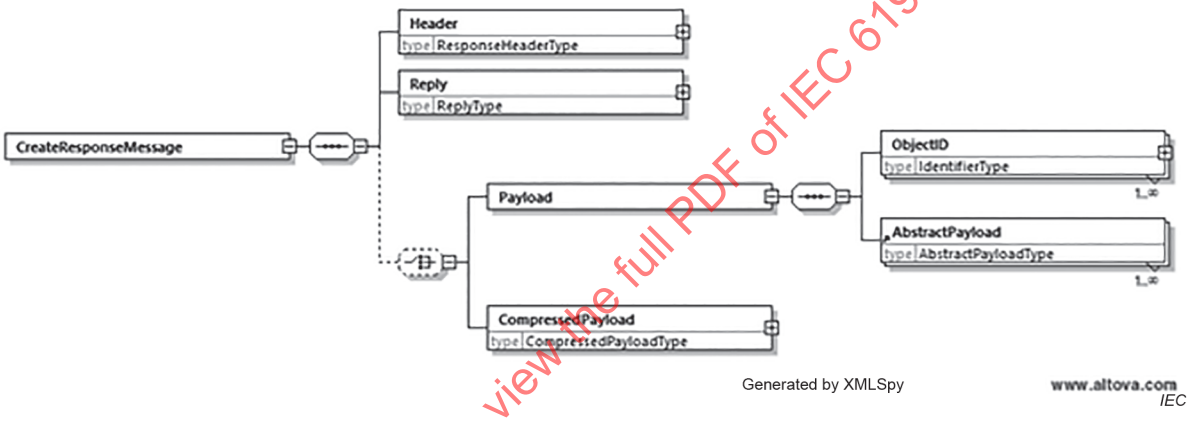


Figure 20 – Structure of the <CreateResponseMessage> element

7.4.3.5 Structure of the <DeleteResponseMessage> element

The structure of the response message whose root element is <DeleteResponseMessage> is shown in Figure 21.

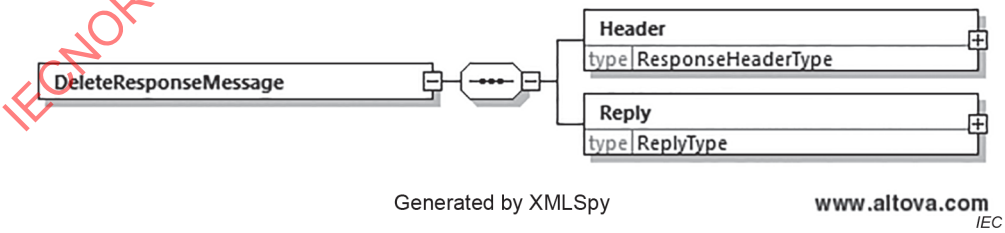


Figure 21 – Structure of the <DeleteResponseMessage> element

7.4.3.6 Structure of the <ExecuteResponseMessage> element

The structure of the response message whose root element is <ExecuteResponseMessage> is shown in Figure 22.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

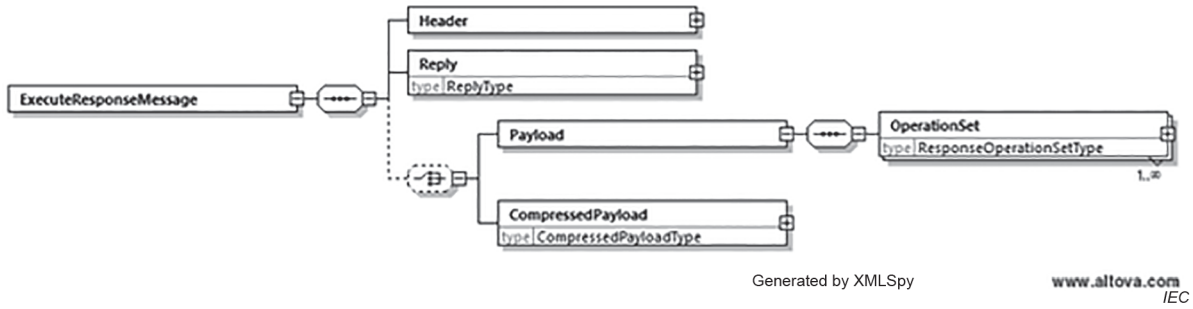


Figure 22 – Structure of the <ExecuteResponseMessage> element

7.4.3.7 Structure of the <GetReponseMessage> element

The structure of the response message whose root element is <GetResponseMessage> is shown in Figure 23.

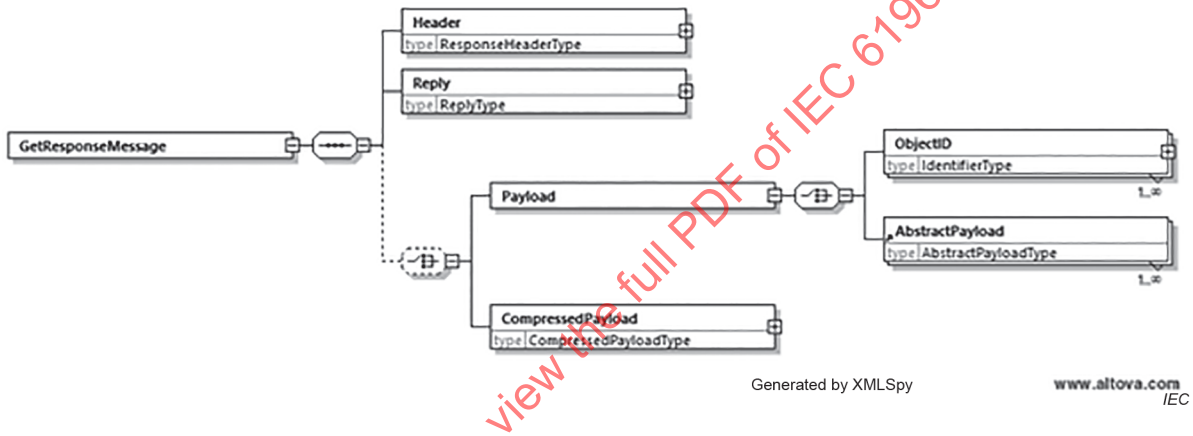


Figure 23 – Structure of the <GetResponseMessage> element

7.4.3.8 Structure of the <UpdateResponseMessage> element

The structure of the response message whose root element is <UpdateResponseMessage> is shown in Figure 24.

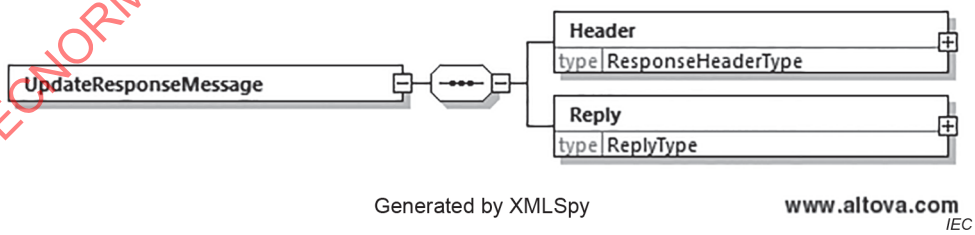


Figure 24 – Structure of the <UpdateResponseMessage> element

7.4.4 Structure of event messages

7.4.4.1 General

This clause describes the top-level structures of the event messages (5.4) according to their root elements.

Table 3 lists the possible message root elements and their respective <Header><Verb> elements (7.5.5.3)

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table 3 – List of top-level root elements in event messages

<Header><Verb> Element	Corresponding Root Element	See Clause
canceled	<CanceledEventMessage>	7.4.4.2
changed	<ChangedEventMessage>	7.4.4.3
created	<CreatedEventMessage>	7.4.4.4
deleted	<DeletedEventMessage>	7.4.4.5
executed	<ExecutedEventMessage>	7.4.4.6
updated	<UpdatedEventMessage>	7.4.4.7

7.4.4.2 Structure of the <CanceledEventMessage> element

The structure of the event message whose root element is <CanceledEventMessage> shown in Figure 25.

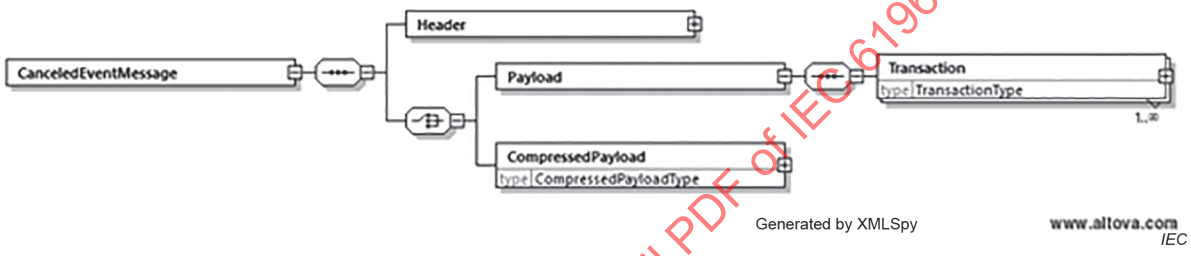


Figure 25 – Structure of the <CanceledEventMessage> element

7.4.4.3 Structure of the <ChangedEventMessage> element

The structure of the event message whose root element is <ChangedEventMessage> is shown in Figure 26.

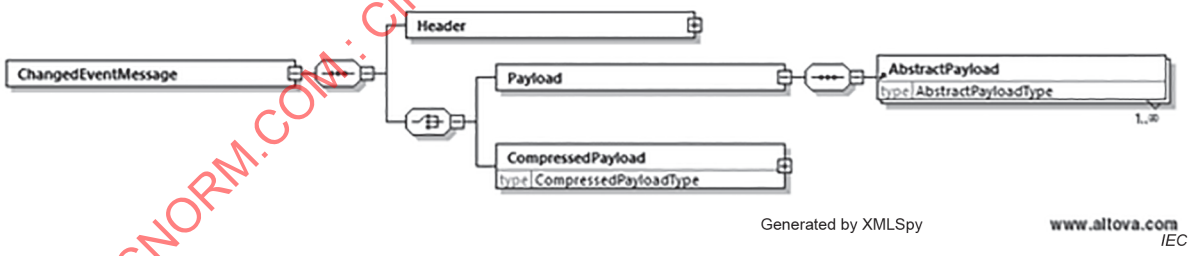


Figure 26 – Structure of the <ChangedEventMessage> element

7.4.4.4 Structure of the <CreatedEventMessage> element

The structure of the event message whose root element is <CreatedEventMessage> is shown in Figure 27.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

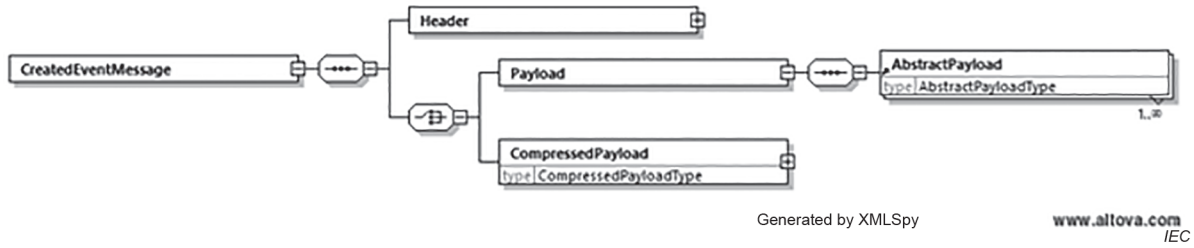


Figure 27 – Structure of the <CreatedEventMessage> element

7.4.4.5 Structure of the <DeletedEventMessage> element

The structure of the event message whose root element is <DeletedEventMessage> is shown in Figure 28.

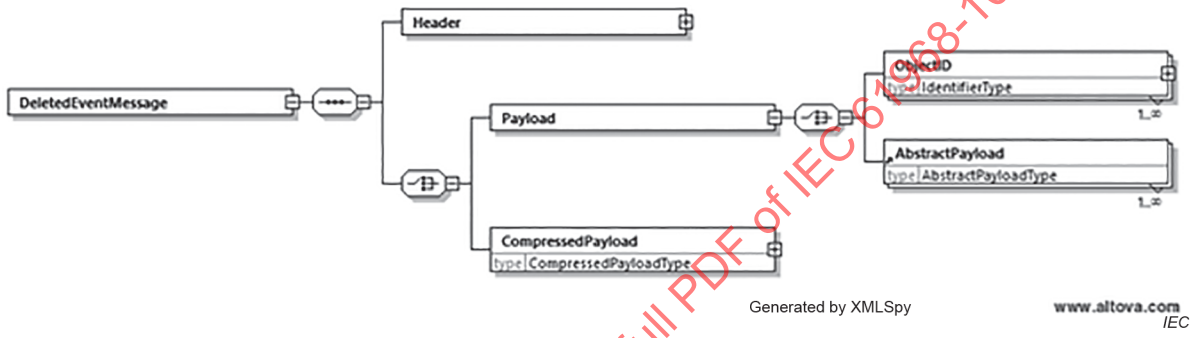


Figure 28 – Structure of the <DeletedEventMessage> element

7.4.4.6 Structure of the <ExecutedEventMessage> element

The structure of the event message whose root element is <ExecutedEventMessage> is shown in Figure 29.

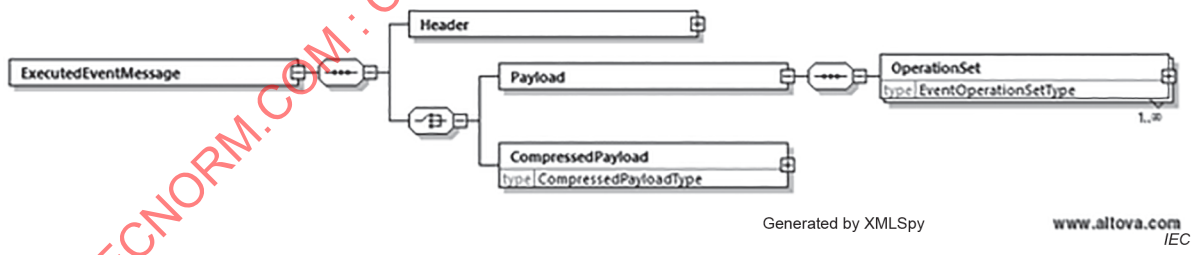


Figure 29 – Structure of the <ExecutedEventMessage> element

7.4.4.7 Structure of the <UpdatedEventMessage> element

The structure of the event message whose root element is <UpdatedEventMessage> is shown in Figure 30.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

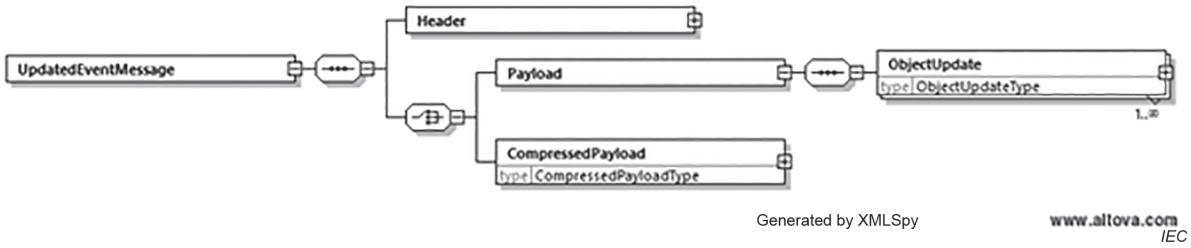


Figure 30 – Structure of the <UpdatedEventMessage> element

7.5 The <Header> element

7.5.1 General

Any given IEC 61968-100 message other than a fault message (5.5) shall contain a <Header> element.

The <Header> element comprises information that specifies how the remainder of the message is to be processed.

The precise structure of the <Header> element shall depend on whether it is being used in the context of a request message (7.5.3), a response message (7.5.4) or an event message (7.5.5).

7.5.2 Message header subelements

This clause describes the various elements that may be present in a message <Header> and their associated semantics.

NOTE Of the elements listed in Table 4, <Source>, <User> and <Property> are optional elements that may facililtate systems integration. The values assigned to these elements are non-normative and have meaning only within the local implementation (8.2.3).

Table 4 – Elements in the Message <Header> element

Header element	Range of Cardinality		XSD Data Type	Meaning
<Verb>	1	1	VerbType	The <Verb> and <Noun> together shall define the interpretation of the data supplied in the <Reply> and <Payload> elements (7.5.3, 7.5.4, 7.5.5).
<Noun>	1	1	xs:string	
<Timestamp>	1	1	xs:dateTime	This element shall indicate when the message was produced. (It is not the time the message was transmitted because the actual send may occur later due to transactions or other client side queueing of messages). The timestamp shall conform to ISO 8601. It shall include either the letter "Z" to denote UTC or an explicit timezone offset.
<MessageID>	1	1	UUID	This element shall be a 128-bit universally unique identifier (UUID) generated according to RFC 4122. It has no normative meaning other than to provide a unique identifier for the message. This element together with the <Timestamp> element is useful for tracing or logging messages as they are processed by the requesting, responding or other systems.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Header element	Range of Cardinality		XSD Data Type	Meaning
<CorrelationID>	1	1	UUID	This element shall be a 128-bit universally unique identifier (UUID) generated according to RFC 4122 (7.5.3.5, 7.5.4.5, 7.5.5.5).
<ReplyAddress>	0	1	xs:string	This element shall be present only in request messages (5.2). If a request message containing this element is received by a responding system, the latter shall ensure that any resulting response messages or consequential event messages are directed to the specified address. If this element is not provided, any resulting response messages or consequential event messages are directed to a default address defined by the responding system. The address is typically a URL in the case of a web services transport (9.2) or the name of a queue or topic in the case of a JMS transport (9.3)
<AckRequired>	0	1	xs:boolean	This element shall be present only in request messages (5.2). The purpose of this element is for the requesting system to be able to determine that the request message has been delivered successfully. If this element is present and has the value true, an explicit simple acknowledgement message (5.3.6, 6.5) shall be sent immediately by the responding system as soon as it receives a request. If this element is either omitted or is present and has the value false, no simple acknowledgement message is required to be sent. Such a message is an application-level acknowledgement and is independent of the underlying transport.
<MessageSequence>	0	1	MessageSequenceType	This element shall be present only in response and consequential event messages (7.5.4.6, 7.5.5.6).
<SequenceNumber>	0	1	xs:unsignedInt	If present, a responding system shall set this element to the value 1 for the first response or consequential event message to be transmitted. It shall be incremented for each subsequent message sent thereafter.
<LastMessage>	0	1	xs:boolean	See 7.5.4.6 and 7.5.5.6.
<Source>	0	1	xs:string	If present, this element shall be used to identify the origin of the informational content. The values assigned to this element are non-normative and have meaning only within the local organisation.
<User>	0	1	UserType	If present, this element shall identify the user and (optionally) the associated organisation. This may be useful for auditing purposes. The values assigned to this element are non-normative and have meaning only within the local organisation.
<UserID>	1	1	xs:string	This element shall identify the user associated with the creation of the message. The values assigned to this element are non-normative and have meaning only within the local implementation.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Header element		Range of Cardinality		XSD Data Type	Meaning
	<Organisation>	0	1	xs:string	If present, this element shall identify the name of the organisation with which the user is associated. The values assigned to this element are non-normative and have meaning only within the local implementation.
<Comment>		0	1	xs:string	This element may contain any text. It has no normative meaning.
<Property>		0	many	MessagePropertyType	This element consists of a set of name/value pairs that may be used to customise the system behaviour in ways going beyond those defined by this International Standard. This may be used to provide some implementation-specific functionality. The values assigned to these elements are non-normative and have meaning only within the local implementation. In the interests of interoperability, usage of the <Property> element should be avoided whenever possible.
	<Name>	1	1	xs:string	Name of the property
	<Value>	0	1	xs:string	Associated value of the property

7.5.3 Message header for request messages

7.5.3.1 General

This clause defines the usage of the <Header> subelements within request messages (5.2).

7.5.3.2 Structure

Figure 31 depicts the structure of the <Header> element as defined by the Message.xsd file for request messages (5.2).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

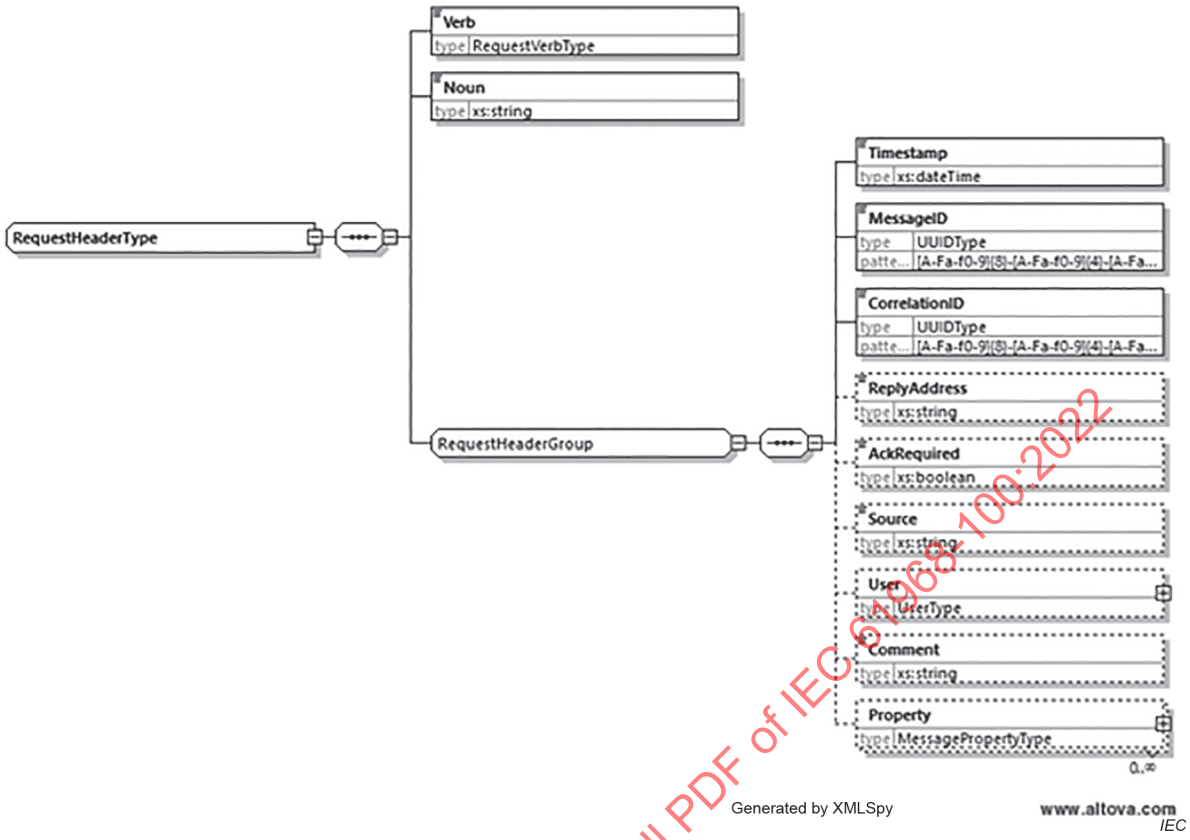


Figure 31 – Structure of the <Header> element for request messages

NOTE In Figure 31 RequestHeaderGroup denotes a named group of XML elements that are referenced in different parts of the Message.xsd file. It does not denote an XML element.

7.5.3.3 The <Header><Verb> element

The value of the <Header><Verb> element shall be one of the values listed in Table 5.

Table 5 – Verbs used in request messages

Verb	Meaning	See clause
cancel	Specifies a request to cancel one or more previously requested transactions in the responding system.	8.5
change	Specifies a request to replace in their entirety the data pertaining to one or more existing objects in the responding system with application-specific data provided in the <Payload> section of the message.	8.3.3
create	Specifies a request to instantiate one or more new objects in the responding system with application-specific data provided in the <Payload> section of the message.	8.3.2
delete	Specifies a request to delete in their entirety one or more existing objects in the responding system.	8.3.4
execute	Specifies a request to execute a composite transaction in the responding system as specified in the <Payload><OperationSet> section of the message.	7.7.3
get	Specifies a request to retrieve a set of objects from the responding system as specified in the <Payload> section of the message.	8.4
update	Specifies a request to modify in part or in whole the data pertaining to one or more existing objects on the responding system.	8.3.5

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.5.3.4 The <Header><Noun> element

The <Header> element shall contain a <Noun> subelement.

If the <Verb> is cancel, the <Noun> subelement shall be the same as that contained in the original request message that is being cancelled (8.5).

If the <Verb> is change or create, the <Noun> subelement shall specify the name of the element substituted for the <Payload><AbstractPayload> element (7.7.5, 8.3).

If the <Verb> is delete, then the <Noun> subelement shall specify the name of the element substituted for the <Payload><AbstractPayload> element that was specified when the object was originally instantiated (8.3.2.1).

If the <Verb> is execute, the <Noun> subelement shall be OperationSet (7.7).

If the <Verb> is get, the <Noun> subelement shall specify the name of the element substituted for the <Payload><AbstractPayload> element of the expected response message (5.3, 7.7.5, 8.4).

If the <Verb> is update, the <Noun> subelement shall specify the name of the element substituted for the <Payload><ObjectUpdate><AbstractPayload> element.

7.5.3.5 The <Header><CorrelationID> element

The <Header> element shall contain a <CorrelationID> subelement. This allows a requesting system to correlate subsequent response and consequential event messages with the original request message.

A new <CorrelationID> subelement shall be generated for each separate request message.

7.5.4 Message header for response messages

7.5.4.1 General

This clause defines the usage of the <Header> element within response messages (5.3).

7.5.4.2 Structure

Figure 32 depicts the structure of the <Header> element as defined by the Message.xsd file for response messages (5.3).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

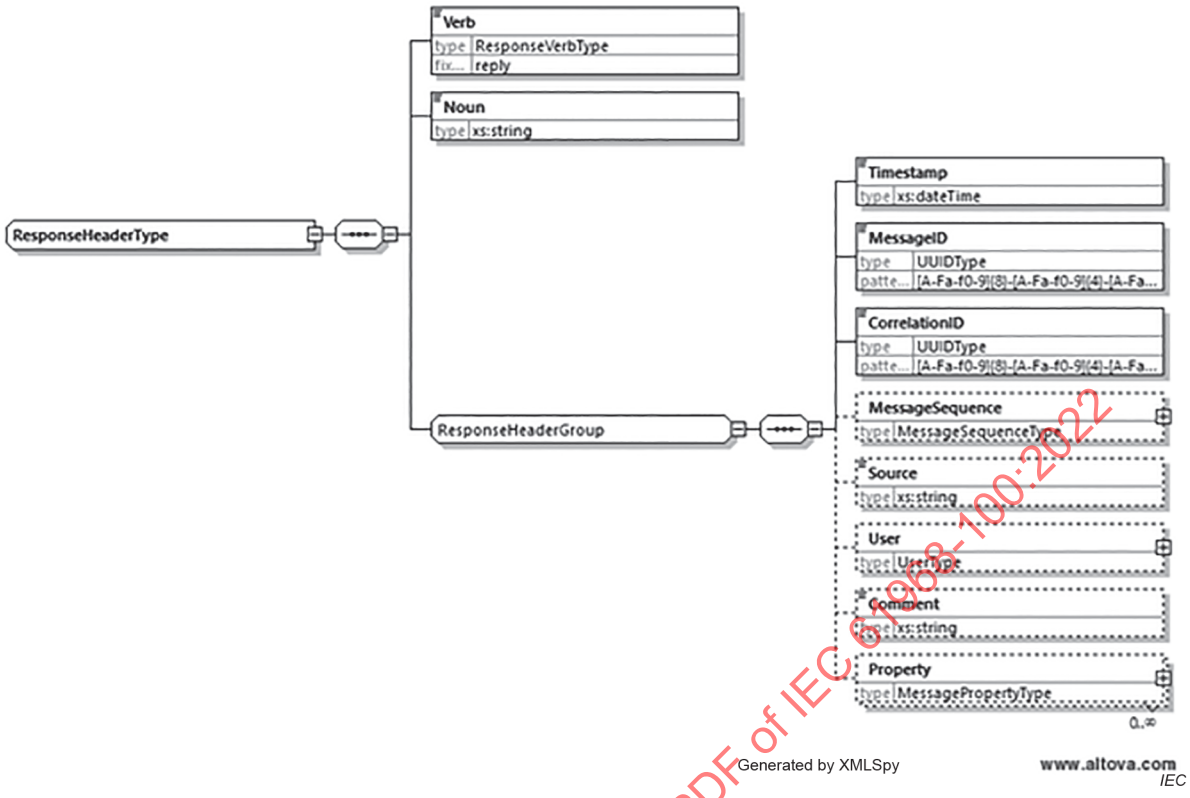


Figure 32 – Structure of the <Header> element for response messages

NOTE In Figure 32 ResponseHeaderGroup denotes a named group of XML elements that are referenced in different parts of the Message.xsd file. It does not denote an XML element.

7.5.4.3 The <Header><Verb> element

The value of the <Header><Verb> element shall be reply for a response message.

7.5.4.4 The <Header><Noun> element

The <Header> element shall contain a <Noun> subelement.

The <Noun> element contained in a response message shall be identical to the <Noun> element in the original request message for which this response message is being generated.

7.5.4.5 The <Header><CorrelationID> element

The <Header> in the response message shall contain a <CorrelationID> element that is identical to the <CorrelationID> element in the original request message for which this response message is being generated (7.5.5.5).

This allows a requesting system to match received response messages against the original request message.

In the case of multiple response messages (6.2.2), each response message shall contain the same value for their respective <CorrelationID> elements.

7.5.4.6 The <Header><MessageSequence> element

A single request message may cause multiple response messages to be generated (4.7.2, 6.2.2).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

The <MessageSequence> element shall be included when more than one response message (5.3) is to be sent and when the responding system is able either to identify the sequence of messages or to determine the last message of such a sequence.

The precise rules that a responding system shall enact when generating a response message in a possible set of multiple messages are as follows:

1) If the responding system is generating exactly one message then the <Header><MessageSequence> element shall be omitted.
2) Otherwise, if the responding system is generating a message that it knows to be the last in a sequence of such response messages then the <Header><MessageSequence> element shall be included and the <Header><MessageSequence><LastMessage> element shall be assigned the value true.
3) Otherwise, the responding system is generating a message that it either knows not to be the last in a sequence of such response messages or which may be the last but the responding system is unaware of this fact. The <Header><MessageSequence> element shall be included and the <Header><MessageSequence><LastMessage> element shall be assigned the value false.

These rules allow a receiving system to sequence received messages properly and to determine when it has received all such messages even when they are delivered out of order.

7.5.5 Message header for event messages

7.5.5.1 General

This clause defines the usage of the <Header> element within event messages (5.4).

7.5.5.2 Structure

Figure 33 depicts the structure of the <Header> element as defined by the Message.xsd file for event messages (5.4).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

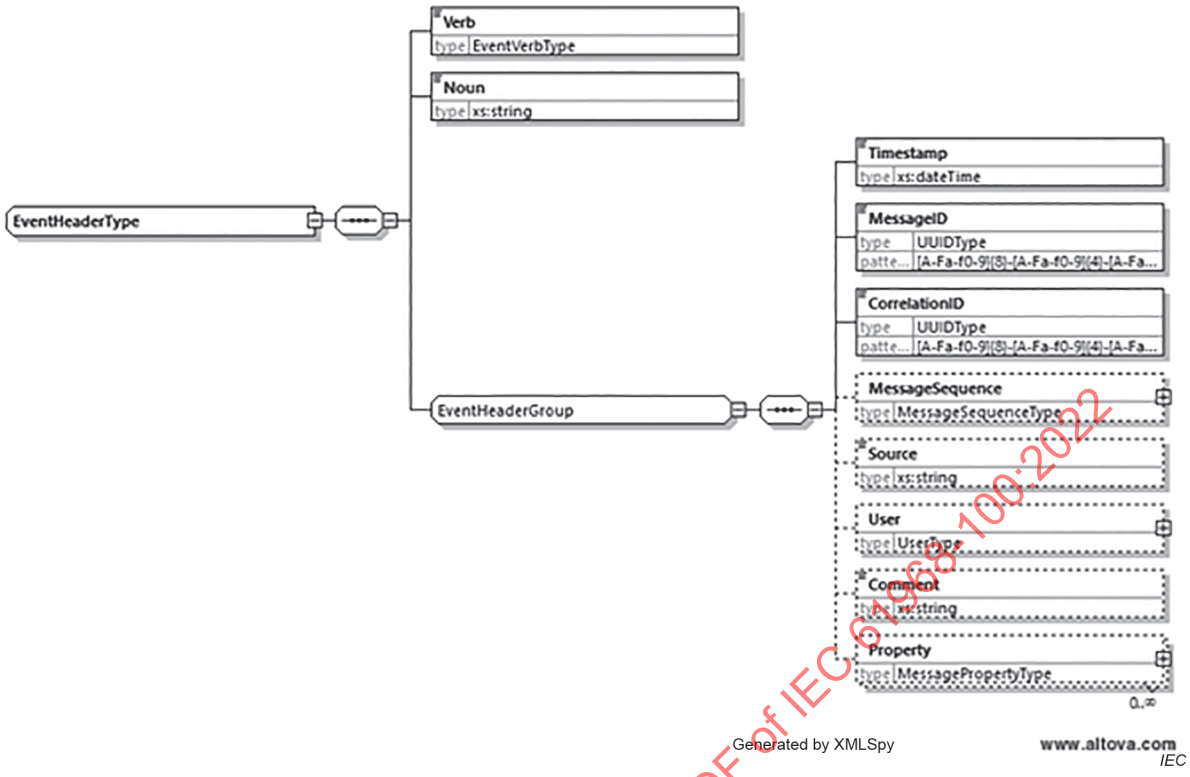


Figure 33 – Structure of the <Header> element for event messages

NOTE In Figure 33 EventHeaderGroup denotes a named group of XML elements that are referenced in different parts of the Message.xsd file. It does not denote an XML element.

7.5.5.3 The <Header><Verb> element

The <Header><Verb> shall belong to one of the enumerated set listed in Table 6.

Table 6 – Verbs used in event messages

Verb	Meaning
canceled	Used to publish a notification that one or more actions have been aborted.
changed	Used to publish a notification that the data pertaining to one or more existing objects have been replaced in their entirety.
created	Used to publish a notification of the instantiation of one or more objects.
deleted	Used to publish a notification that one or more existing objects have been deleted in their entirety.
executed	Used to publish a notification that the execution of a complex transaction that uses the <Payload><OperationSet> element has taken place.
updated	Used to publish a notification that the data pertaining to one or more objects have been modified in part or in whole.

7.5.5.4 The <Header><Noun> element

The <Header> element shall contain a <Noun> subelement.

If the <Verb> is canceled, the <Noun> subelement shall be the same as that contained in the original request message that is being cancelled.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

If the <Verb> is changed, created or deleted, the <Noun> subelement shall specify the name of the profile appearing in the AbstractPayloadType section of the <Payload> element.

If the <Verb> is executed, the <Noun> subelement shall be OperationSet.

If the <Verb> is updated, the <Noun> subelement shall specify the name of the profile appearing in the AbstractPayloadType section of the <Payload><ObjectUpdate> element.

7.5.5.5 The <Header><CorrelationID> element

7.5.5.5.1 Unsolicited event message

An unsolicited event message (6.3) shall contain a <CorrelationID> element. This <CorrelationID> subelement shall be created afresh for each new unsolicited event message.

7.5.5.5.2 Consequential event message

A consequential event message (6.4) shall contain a <CorrelationID> element. The value of this element shall be set to that of the original request message that is directly responsible for the creation of the consequential event.

7.5.5.6 The <Header><MessageSequence> element

A single request message may cause multiple consequential event messages to be generated (6.4).

The <MessageSequence> element shall be included when more than one consequential event (5.4) message is to be sent and when the responding system is able either to identify the sequence of messages or to determine the last message of such a sequence.

The precise rules that a responding system shall enact when generating a consequential event message in a possible set of multiple messages are as follows:

1) If the responding system is generating exactly one message then the <Header><MessageSequence> element shall be omitted.
2) Otherwise, if the responding system is generating a message that it knows to be the last in a sequence of such consequential event messages then the <Header><MessageSequence> element shall be included and the <Header><MessageSequence><LastMessage> element shall be assigned the value true.
3) Otherwise, the responding system is generating a message that it either knows not to be the last in a sequence of such consequential event messages or which may be the last, but the responding system is unaware of this fact. The <Header><MessageSequence> element shall be included and the <Header><MessageSequence><LastMessage> element shall be assigned the value false.

These rules allow a receiving system to sequence received messages properly and to determine when it has received all such messages even when they are delivered out of order.

7.6 The <Reply> element

7.6.1 General

The <Reply> element shall be used only in the context of the response message or fault message archetypes (5.3, 5.5).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.6.2 Structure

The structure of the <Reply> element is shown in Figure 34.

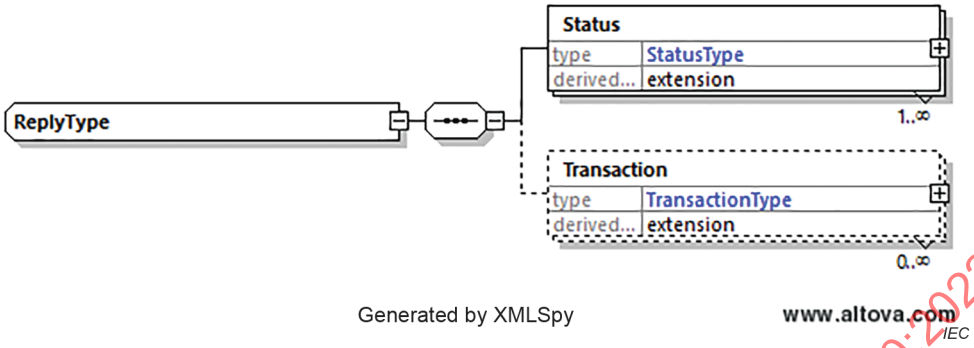


Figure 34 – Structure of the <Reply> element

7.6.3 Reply subelements

Table 7 describes the various subelements that may be present in a <Reply> element and their associated semantics.

Table 7 – Subelements within the <Reply> element

Element	Range of Cardinality		XSD Data Type	Meaning
<Status>	1	many	StatusType	See clause 7.6.4
<Transaction>	0	many	TransactionType	See clause 7.6.4.10

7.6.4 The <Reply><Status> subelement

7.6.4.1 General

The <Reply><Status> element shall be used for sending informational and error notifications from a responding system back to a requesting system.

A <Reply> element shall contain at least one <Status> subelement. Each separate informational or error notification shall be assigned to a distinct <Reply><Status> instance.

Each <Reply><Status> element shall contain a single <Level> and a single <Code> subelement (7.6.4.6, 7.6.4.7).

7.6.4.2 Replying with a notification of success

A notification that the original request message has been carried out successfully in its entirety shall be denoted by a <Reply><Status><Code> element of 0.0 and a <Reply><Status><Level> of OK. Such a success notification shall be sent only if there are no informational or error notifications to be conveyed (7.6.4.3).

7.6.4.3 Replying with notifications of success, error or other information

A <Reply><Status><Code> element other than 0.0 denotes an informational or error notification.

NOTE In the case of a response consisting of multiple messages, a requesting system can determine the sucessful cases only when all the response messages have been received.

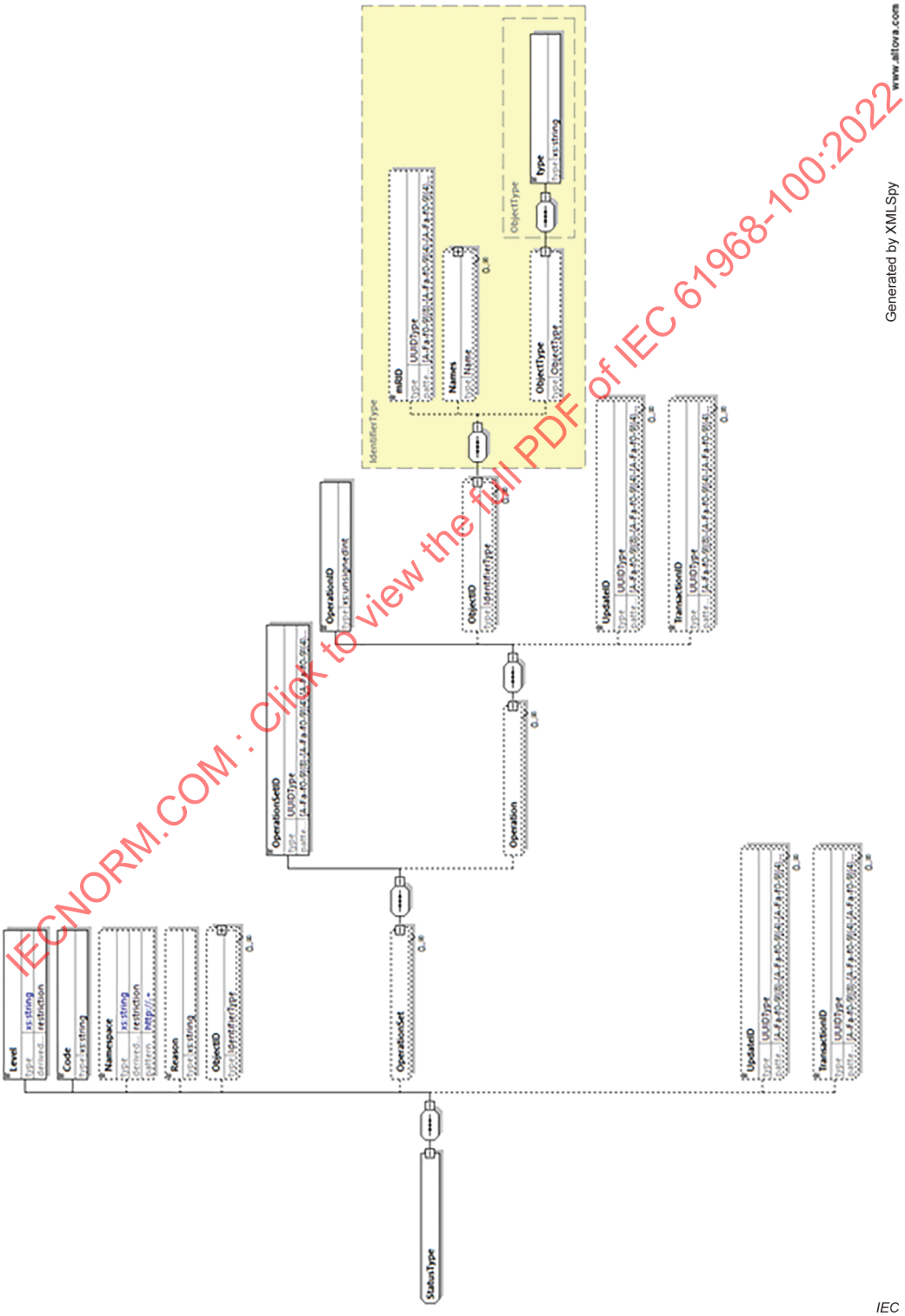
IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

EXAMPLE D.9.4 shows an example response message for when there are no explicit indications of success. A single <Reply><Status> element is included.

EXAMPLE D.11.3 shows an example response message where an explicit error notification is included.

7.6.4.4 Structure

The structure of the XML schema corresponding to the <Reply><Status> element is shown in Figure 35.



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 35 – Structure of the <Reply><Status> element

7.6.4.5 <Reply><Status> subelements

Table 8 describes the various subelements that may be present in a message <Reply><Status> subelement and their associated semantics.

Table 8 – Subelements within the <Reply><Status> element

Element	Range of Cardinality		XSD Data Type	Meaning
<Level>	1	1	enumeration	See 7.6.4.6.
<Code>	1	1	xs:string	See 7.6.4.7.
<Namespace>	0	1	URIType	See 7.6.4.8.
<Reason>	0	1	xs:string	If present, this element shall contain free-form text clarifying the error or informational notification. This element has no normative meaning.
<ObjectID>	1	many	IdentifierType	If present, these elements shall specify the objects to which this error or status notification pertains (7.6.4.9, 8.6.5).
<OperationSet>	1	many		This element shall be used only if the response is being sent as the result of an OperationSet request.
<OperationSetID>	1	1	UUIDType	This element shall contain the value of the <Payload><OperationSet> <OperationSetID> from the original request. This allows a requesting system to determine the precise operation set for which the notification is being generated (7.7.3).
<Operation>	0	many		
<OperationID>	1	1	xs:unsignedInt	If the response is being sent as the result of an OperationSet request, this element shall contain the value of the <Payload><OperationSet> <Operation><OperationID> from the original request. This allows a requesting system to determine the exact operation for which the notification is being generated (7.7.3).
<ObjectID>	1	many	IdentifierType	
<mRID>	0	1	UUIDType	If the response is being sent as the result of an OperationSet request, these elements shall contain the values of the corresponding subelements of the <Payload> <OperationSet><Operation> <ObjectID> element or corresponding element in the AbstractPayload element in the original request. This allows a requesting system to determine the exact operation for which the error notification is being generated.
<Names>	0	many	Name	
<ObjectType>	0	1	ObjectType	
<type>	1	1	xs:string	
<UpdateID>	1	many		If the response is being sent as the result of an update operation within an OperationSet request, this element shall contain the value of the <Payload><OperationSet> <Operation><ObjectUpdate><UpdateID> element from the original request. This allows a requesting system to determine the exact update operation for which the error notification is being generated (7.7.3, 8.3.5).
<TransactionID>	1	many		If the response is being sent as the result of an OperationSet request to cancel a request in progress, this element shall contain the relevant transaction IDs.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Element	Range of Cardinality		XSD Data Type	Meaning
<UpdateID>	0	1	UUIDType	If the response is being sent as the result of a request to update an object, this element shall contain the value of the <Payload><ObjectUpdate> <UpdateID> element from the original request. This allows a requesting system to determine the exact update operation for which the error notification is being generated (7.7.3, 8.3.5).
<TransactionID>	1	many	UUIDType	If the response is being sent as the result of a request to cancel an operation in progress, this element shall contain the relevant transaction IDs (7.9, 8.5).

7.6.4.6 The <Reply><Status><Level> element

The <Reply><Status><Level> element is an enumeration and shall be assigned one of the values listed in Table 9. Further explanatory information may be present in the <Reply><Status><Code> and <Reply><Status><Reason> elements.

Table 9 – Values of the <Reply><Status><Level> element

Value	Meaning
OK	This value shall indicate that the operation completed successfully.
WARNING	This value shall indicate that the operation completed successfully but that there may have been issues that require investigation.
ERROR	This value shall indicate that the operation failed.
UNKNOWN	This value shall indicate that the responding system cannot ascertain whether the operation completed successfully.

7.6.4.7 The <Reply><Status><Code> element

The <Reply><Status><Code> element shall denote the specific status code.

Unless an application-dependent value is assigned to the <Reply><Status><Code> element (7.6.4.8), the <Reply><Status><Code> element shall be assigned one of the normative set of values listed in Table 10. This table also shows the corresponding values of the <Reply><Status><Level> element.

Table 10 – Normative values for the <Reply><Status><Code> element

Value	Meaning	Corresponding <Level> Value	Usage Notes
0.0	Message reports no errors	OK	In the case of multiple response messages being generated (6.2.2), the other messages may report errors or status codes.
0.3	Simple acknowledgement	OK	Reserved for simple acknowledgement messages (5.3.6).
1.1	Mandatory IEC 61968-100 elements are missing or invalid	ERROR	
1.2	Message failed XML schema validation	ERROR	
1.3	Unsupported IEC 61968-100 namespace specified	ERROR	Used when a specific edition of IEC 61968-100 (as denoted by the namespace in the request message) is not supported by the responding system.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Value	Meaning	Corresponding <Level> Value	Usage Notes
1.4	Message cannot be parsed as an XML document	ERROR	See 5.5.
2.1	Unsupported parameter	ERROR	
2.2	Invalid message header element	ERROR	
2.3	Invalid message reply element	ERROR	
2.4	Invalid message payload element	ERROR	Used to specify when a payload element specific to part 100 (that is, not application-specific) is invalid (for example, <CompressedPayload><CompressionKind>). For application-specific data errors in the payload, use code 5.2.
2.5	Referenced object not uniquely identified	ERROR	Used when the object reference does not uniquely identify a single object. (for example, a <Names><name> reference is used when there are multiple objects with that name).
3.1	System limit exceeded	ERROR	Used for messages that exceed a limitation in system capacity (for example, when message size limitations or infrastructure limitations such as memory or disk space are exceeded).
3.2	Application limit exceeded	ERROR	Used for messages that exceed an application imposed limitation such as too many items of a certain type.
4.1	Request timed out	ERROR	Used for cases where the request cannot be completed in the allowed time.
4.2	Request aborted	ERROR	Used for cases where a system process or user on the responding system actively aborts a request prior to its execution. For requests that are cancelled by a requesting system see 8.5.
4.3	Service not available	ERROR	Used for cases where another application or service is not available for processing this request.
5.0	General application error	ERROR	An application may use this code to denote that some error has occurred during processing.
5.1	Unable to cancel transaction	ERROR	Used only for responses to a cancel transaction request (8.5).
5.2	Unsupported application-specific data specified	ERROR	Used for cases where the responding system does not support the application data specified. For example, a responding system is requested to create an object of a type that the system does not support or an unsupported IEC 61968-9 ReadingType or EndDeviceControls code is specified. Can also be used when a combination of application-specific data elements is invalid.
5.3	Referenced object does not exist	ERROR	The referenced object for this code can be either an application-specific data object or a general IEC 61968-100 object such as a TransactionID or an OperationID. Usage of the appropriate element under the <StatusCode> element (<ObjectID>, <OperationSetID>, etc.) can be used to specify more information about the referenced object.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Value	Meaning	Corresponding <Level> Value	Usage Notes
5.4	Referenced object already exists	ERROR	The referenced object for this code can be either an application-specific data object or a general IEC 61968-100 object such as a TransactionID or an OperationID. Usage of the appropriate element under the <StatusCode> element (<ObjectID>, <OperationSetID>, etc.) can be used to specify more information about the referenced object.
5.5	The object being modified is not uniquely identified	ERROR	Used for an <ObjectIdentificationRevisionSets> message that attempts to identify an object and the <mRID> does not uniquely identify a single object (8.7.6).
5.6	Mandatory application-specific data missing	ERROR	An application may use this error code to denote that some data needed to carry out the specified request are missing.
5.7	Unable to enforce transactional integrity	ERROR	One or more Operations in an OperationSet were aborted or rolled back in order to maintain transactional integrity. Used for cases where an OperationSet requires enforcement of transactional integrity.
5.8	The <mRID> being added is not unique	ERROR	Used for an <ObjectIdentificationRevisionSets> message that attempts to add an <mRID> that already exists for another object (8.7.6).
5.9	Removing the last remaining <mRID> is prohibited	ERROR	Used for an <ObjectIdentificationRevisionSets> message that attempts to remove the last <mRID> remaining identifier of an object (8.7.6).
5.10	The <mRID> or <Name> being removed does not exist	ERROR	Used for an <ObjectIdentificationRevisionSets> message that attempts to remove a non-existent <mRID> or <Name> of a given object (8.7.6).
5.11	The <ObjectType> element does not match the type of the object	ERROR	Used when an <mRID> or <Name> is paired with an <ObjectType> and no such object of that type with the given identifier exists.
5.12	Invalid object relationship	ERROR	Used to denote where two or more objects are related together in a application-specific payload where the relationship is invalid. Use the <ObjectID> element in the reply to specify the specific objects in question. For example, a request to remove an IEC 61968-9 Meter object from a UsagePoint would generate this error if the specified Meter were not currently installed at that UsagePoint.
6.0	Business rule violation	ERROR, WARNING or UNKNOWN	An application may use this status code to denote that some business rule has been violated.
6.1	Request rejected per business rule	ERROR	Used to denote where a business rule prohibits the execution of a request. For example, a request to disconnect a meter may be rejected owing to the existence of a business rule prohibiting disconnect operations during the winter season.
6.2	Request placed on hold per business rule	WARNING	Used to denote where a business rule requires that the requested operation be placed on hold prior to execution. For example, a request to disconnect a meter may be placed on hold owing to the existence of a business rule requiring review and approval by an authorized user before execution.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Value	Meaning	Corresponding <Level> Value	Usage Notes
6.3	Request released from business rule hold	OK	Used to denote that a request has been released from a prior business rule hold and can now be executed. For example, a request to disconnect a meter that had previously been placed on hold has been reviewed and approved and is now available for execution.
6.4	Request rescheduled per business rule	WARNING	Used to denote where a business rule requires that the requested operation be rescheduled for a different date and/or time. For example, a request to disconnect a meter that is sent at 1:00 AM may need to be rescheduled to 8:00 AM owing to the existence of a business rule requiring such operations to be executed during normal business hours.
7.0	Security violation	ERROR	An application may use this status code to denote that an attempt to break some security measure has been detected.
99.x	Some other status not listed elsewhere in this table	OK, ERROR, WARNING or UNKNOWN	This code series can be used to extend the set of status codes as candidates for inclusion in the next edition of IEC 61968-100. This code series should not be used for application-specific codes.

7.6.4.8 The <Reply><Status><Namespace> element

The value for the <Reply><Status><Code> element may be assigned an application-dependent value that is not listed in Table 10.

For the purposes of disambiguation, such a value shall be associated with an application-specific namespace. This namespace shall be assigned to the <Reply><Status><Namespace> element.

For the purposes of disambiguating status codes defined by standards published by the IEC TC57, the namespace shall be of the form:

http://iec.ch/TC57/*<year-of-publication>/<publication-id>#.*

EXAMPLE An example of such a name space is as follows: http://iec.ch/TC57/2013/61968-9#.

If no <Reply><Status><Namespace> element is specified, then the <Reply><Status><Code> element shall be one of the values listed in Table 10.

For other, project-specific implementations, the <Reply><Status><Namespace> element has no normative set of values. It is left for the relevant application developer to define suitable values.

7.6.4.9 The <Reply><Status><ObjectID> element

The <Reply><Status><ObjectID> element, if present, shall identify the object to which the notification pertains (8.6.5).

There may be multiple <ObjectID> subelements within a single <Reply><Status> element. Each <ObjectID> instance shall denote a separate object with which the status is associated.

EXAMPLE In the example shown in Figure 36, the <Reply><Status><ObjectID> elements indicate the objects that caused the requests to fail. Error notifications are returned only for those cases that are deemed to have failed in some way (7.6.4.3).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<Reply>
  <Status>
    <Level>ERROR</Level>
    <Code>2.4</Code>
    <Namespace>http://iec.ch/TC57/2013/61968-9#</Namespace>
    <Reason>no such meter</Reason>
    <ObjectID>
      <Names>
        <name>M1003</name>
      </Names>
     <ObjectType>
       <type>Meter</type>
     </ObjectType>
    </ObjectID>
    <ObjectID>
      <Names>
        <name>M1004</name>
      </Names>
      <ObjectType>
        <type>Meter</type>
      </ObjectType>
    </ObjectID>
  </Status>
  <Status>
    <Level>ERROR</Level>
    <Code>2.12</Code>
    <Namespace>http://iec.ch/TC57/2013/61968-9#</Namespace>
    <Reason>invalid usagepoint</Reason>
    <ObjectID>
      <Names>
        <name>UP2</name>
      </Names>
      <ObjectType>
        <type>UsagePoint</type>
      </ObjectType>
    </ObjectID>
  </Status>
</Reply>
```

Figure 36 – Example of error indications in a <Reply> element

7.6.4.10 The <Reply><Status><OperationSet> element

The <Reply><Status><OperationSet> element, if present, shall identify the operation set to which the notification pertains (7.7.3.6).

7.6.4.11 The <Reply><Status><UpdateID> element

The <Reply><Status><UpdateID> element, if present, shall identify the object update to which the notification pertains (7.7.4).

7.6.4.12 The <Reply><Status><TransactionID> element

The <Reply><Status><TransactionID> element, if present, shall identify the transaction to which the notification pertains (7.9).

7.6.5 Combining the <Reply><Status><ObjectID>, <OperarionSet>, <UpdateID> and <TransactionID> elements

A status notification may contain the following:

- zero or more <ObjectID> elements (7.6.4.9);
- zero or more <OperationSet> elements (7.6.4.10);

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

- zero or more <UpdateID> elements (7.6.4.11);
- zero or more <TransactionID> elements (7.6.4.12).

A <Reply><Status> element that contains none of these identification elements shall be interpreted as referring to the entire request.

If a <Reply><Status> element contains more than one identification using the same element name – for example, two <ObjectID> elements – these shall be interpreted as referring to any such identified parts of the original request (logical "OR").

If a <Reply><Status> element contains more than one identification using different element names– for example, an <ObjectID> element and a <TransactionID> element – these shall be interpreted as referring to those parts of the original request that match all the specified criteria (logical "AND").

NOTE This is similar to the cross-product rule (8.4.2).

7.6.6 The <Reply><Transaction> element

The <Reply><Transaction> element is used for sending transaction identifiers from a responding system to a requesting system (7.9, 8.5).

7.7 The <Payload> element

7.7.1 General

The purpose of the <Payload> element is to convey application-specific data.

7.7.2 Payload subelements

Table 11 describes the various subelements that may be present in a <Payload> element, and their associated semantics.

Only one of the <ObjectID>, <Transaction>, <OperationSet>, <ObjectUpdate> or <AbstractPayloadType> subelements may be present (7.4.2, 7.4.3, 7.4.4).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table 11 – Subelements within the <Payload> element

Element	Range of Cardinality		XSD Data Type	Meaning
<ObjectID>	1	many	IdentifierType	This element shall be present only in request messages (5.2) or event messages (5.4). This element shall be used to supply object IDs or get or delete operations in cases where they are not otherwise specified using a type-specific payload (8.6.5).
<Transaction>	1	many	TransactionType	This element shall be present only in request messages (5.2) or event messages (5.4). This element shall be used to supply transaction IDs for cancel operations (8.5.3).
<OperationSet>	1	many	RequestOperationSetType, ResponseOperationSetType or EventOperationSetType	See 7.7.3
<ObjectUpdate>	1	many	UpdatePayloadType	See 7.7.4
< ... >	1	many	AbstractPayloadType	See 7.7.5

7.7.3 The <Payload><OperationSet> element

7.7.3.1 General

An OperationSet is used to combine several Operations together in a single message (7.7.3.6). This can be useful when it is necessary to ensure that these Operations are carried out in a specific order or for when it is necessary to ensure transactional integrity across the various Operations.

The precise structure of the <Payload><OperationSet> subelement depends on whether it is being used in the context of a request message (7.7.3.3), a response message (7.7.3.4) or an event message (7.7.3.5).

EXAMPLE Clauses D.5 and D.13 contain example messages illustrating the usage of OperationSets.

7.7.3.2 OperationSet subelements

Table 12 specifies the various subelements that may be present in a <Payload><OperationSet> element and their associated semantics.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table 12 – Subelements within the <Payload><OperationSet> element

Element	Range of Cardinality		XSD Data Type	Meaning
<OperationSetID>	1	1	UUIDType	This element shall be used to correlate the <Payload> <OperationSetID> with the <OperationSetID> element in the reply.
<EnforceMsgSequence>	0	1	xs:boolean	This element may be present only in request (5.2) or event messages (5.4). If true, then the receiving system shall process the <Payload><OperationSet> <Operation> subelements in strict ascending order of the <Payload><OperationSet> <Operation><OperationID> sublements. If this element is missing or has the value false, the responding system is free to carry out the specified Operations in the OperationSet in whatever order it deems appropriate.
<EnforceTransactionalIntegrity>	0	1	xs:boolean	This element may be present only in request (5.2) or event messages (5.4). The Operations in an OperationSet may also constitute a single "ACID" (atomic, consistent, isolated and durable) transaction in such a way that they shall be either all carried out successfully or the failure of any one Operation shall cause them all to be rolled back. Transactional integrity functionality may be enabled or disabled according to the setting of this element (true or false respectively).
<Operation>	1	many	OperationType	See 7.7.3.6

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.7.3.3 Structure for request messages

For request messages (5.2), the contents of the <Payload><OperationSet> subelement are depicted in Figure 37.

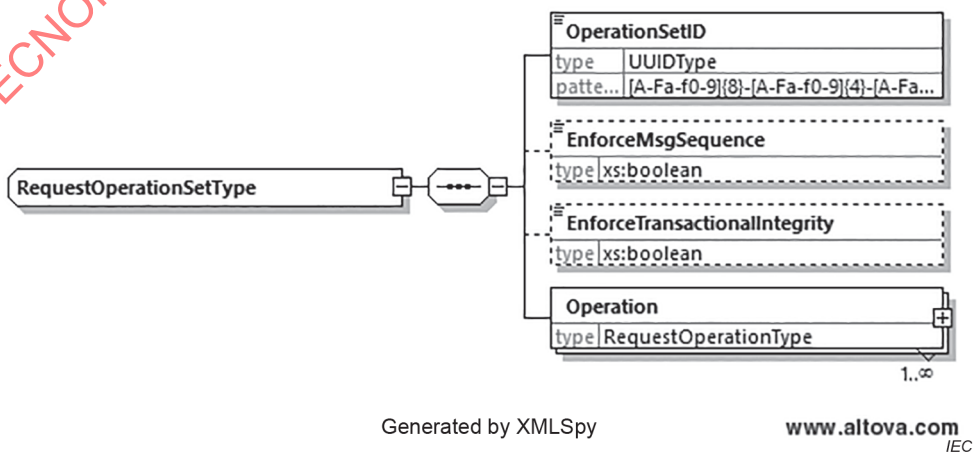


Figure 37 – Structure of the <Payload><OperationSet> element for request messages

7.7.3.4 Structure for response messages

For response messages (5.3), the contents of the <Payload><OperationSet> subelement are depicted in Figure 38.

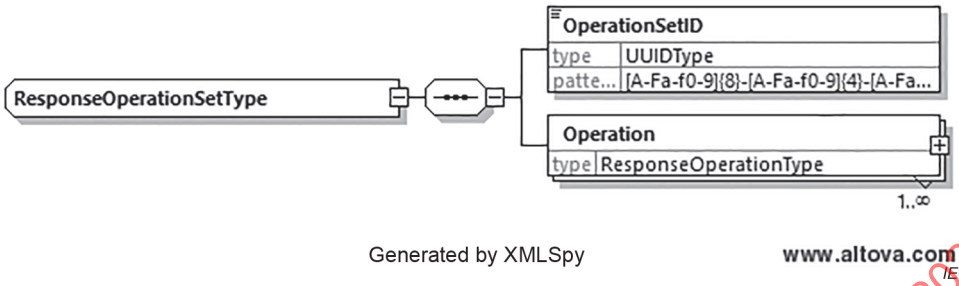


Figure 38 – Structure of the <Payload><OperationSet> element for response messages

7.7.3.5 Structure for event messages

For event messages (5.4), the contents of the <Payload><OperationSet> subelement are depicted in Figure 39.

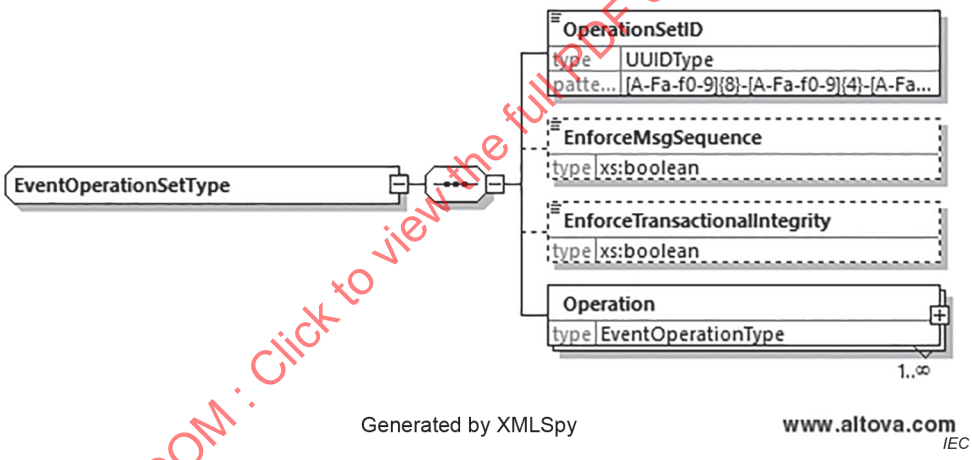


Figure 39 – Structure of the <Payload><OperationSet> element for event messages

7.7.3.6 The <Payload><OperationSet><Operation> element

7.7.3.6.1 General

Each <Payload><OperationSet><Operation> element shall contain <Verb>, <Noun> and <OperationID> subelements (7.7.3).

7.7.3.6.2 Operation subelements

Table 13 describes the various subelements that may be present in a <Payload><OperationSet><Operation> element and their associated semantics.

Only one of the <ObjectID>, <Transaction>, <ObjectUpdate> or <AbstractPayloadType> subelements may be present. The choice of subelement depends on the <Payload><OperationSet><Operation><Verb> element (7.7.3.6.6).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table 13 – Subelements within the <Payload><OperationSet><Operation> element

Element	Range of Cardinality		XSD Data Type	Meaning
<OperationID>	1	1	UUIDType	See 7.6.4.5.
<Verb>	1	1	VerbType	The <Verb> and <Noun> elements shall together specify the semantics of the message and the interpretation of the request data supplied in the <Payload><OperationSet><Operation>AbstractPayloadType sublement (7.5.3.4, 7.5.4.4, 7.5.5.4).
<Noun>	1	1	xs:string	
<Comment>	0	1	xs:string	This element may contain any text. It has no normative meaning.
<Property>	0	many	MessagePropertyType	This element shall consist of a set of name/value pairs that may be used to customize the system behaviour in ways going beyond those defined by this International Standard. This may be used to provide some implementation-specific functionality. The values assigned to these elements are non-normative and have meaning only within the local implementation. In the interests of interoperability, usage of the <Property> element should be avoided wherever possible.
<Name>	1	1	xs:string	Name of the property
<Value>	0	1	xs:string	Associated value of the property
<ObjectID>	1	many	IdentifierType	Used to supply object IDs for cancel operations or object IDs for get or delete operations in cases where they are not otherwise specified using a type-specific payload (8.6.5).
<Transaction>	1	many	TransactionType	This element shall be present only in request messages (5.2) or event messages (5.4). This element shall be used to supply transaction IDs for cancel operations (8.5.3).
<ObjectUpdate>	1	many	UpdatePayloadType	This element shall be present only in request messages (5.2) or event messages (5.4). See 7.7.4
< ... >	1	many	AbstractPayloadType	The <Operation> element may contain application-specific data. These data depend on the precise semantics of the request and are left for the relevant working group or other application developer to define (7.7.5).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.7.3.6.3 Structure for request messages

For request messages (5.2), the contents of the <Payload><OperationSet> <Operation> sublement are depicted in Figure 40.

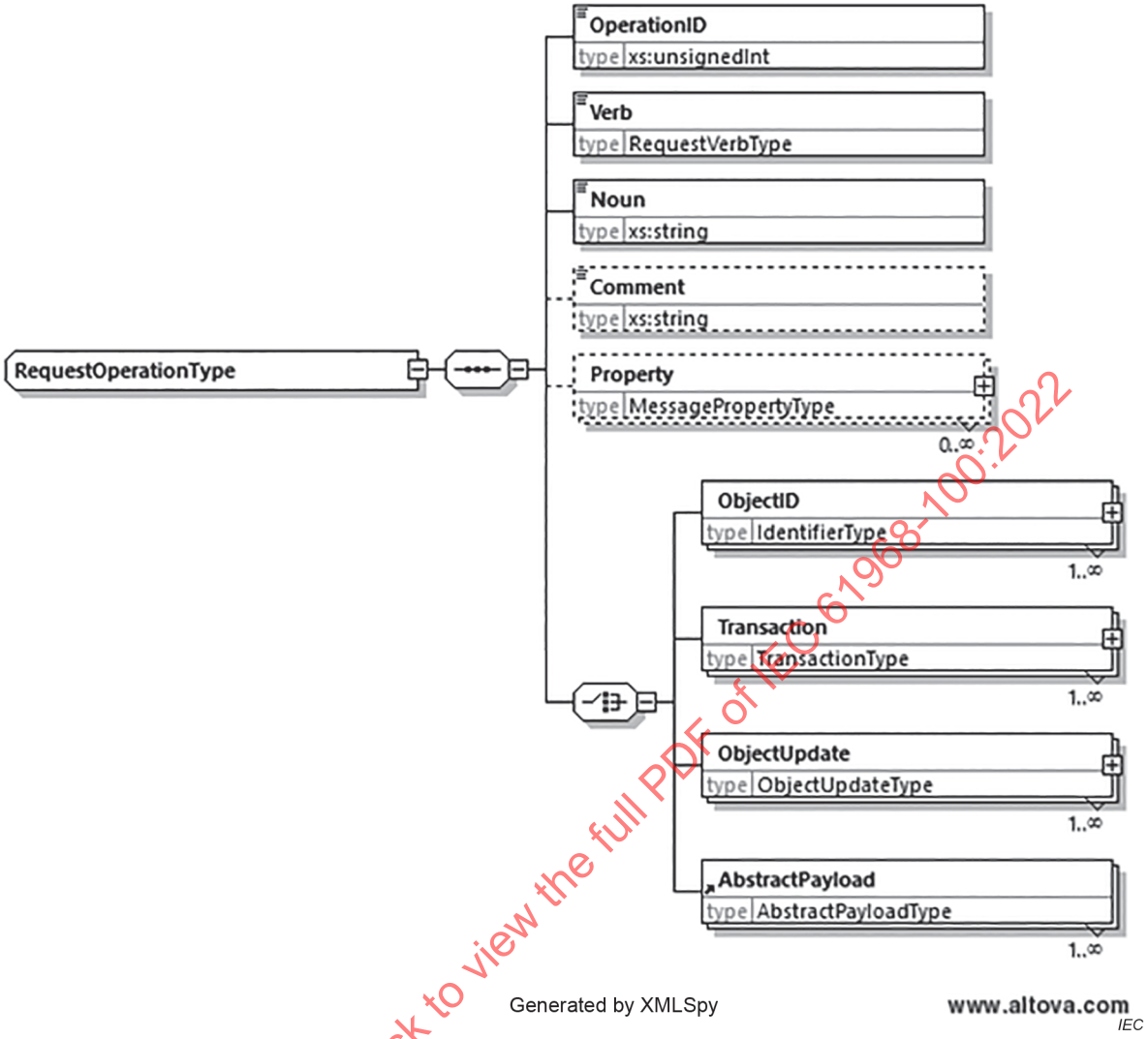


Figure 40 – Structure of the `<Payload><OperationSet><Operation>` element for request messages

7.7.3.6.4 Structure for response messages

For response messages (5.3), the contents of the `<Payload><OperationSet><Operation>` subelement are depicted in Figure 41.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

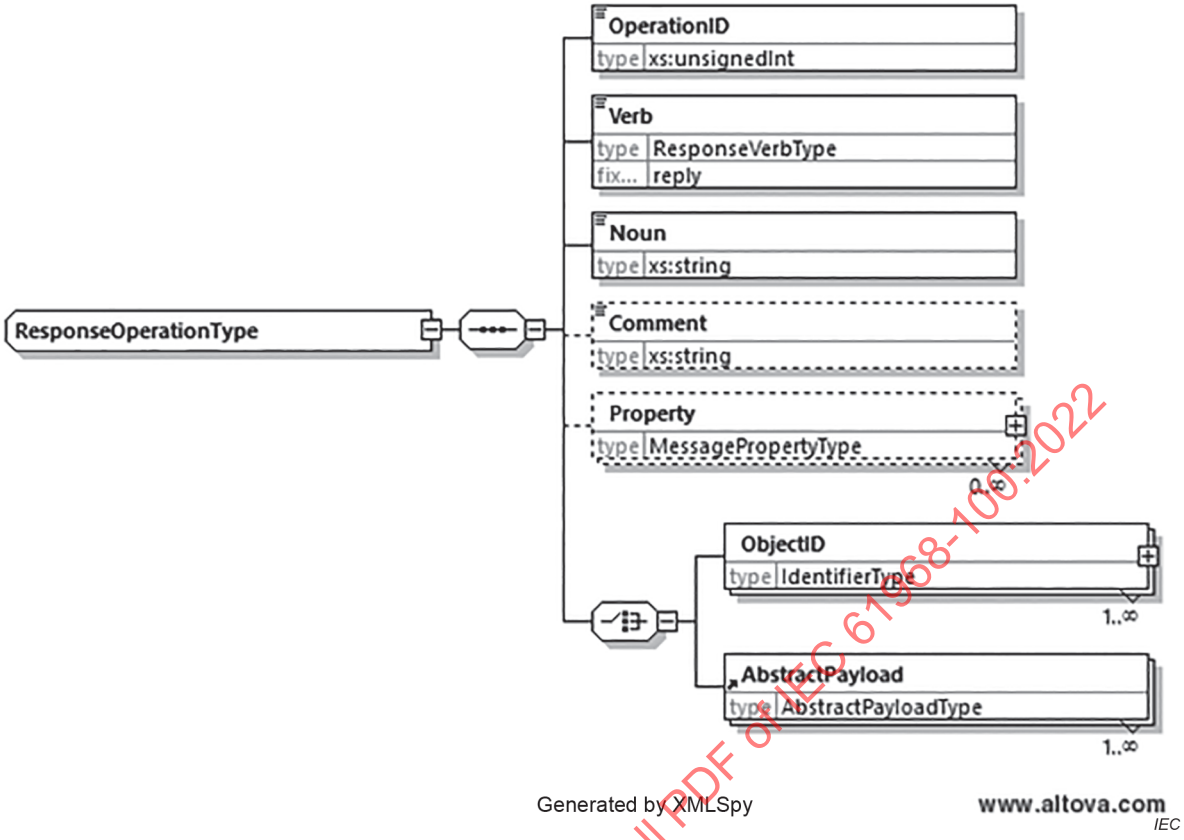


Figure 41 – Structure of the `<Payload><OperationSet><Operation>` element for response messages

7.7.3.6.5 Structure for event messages

For event messages (5.4), the contents of the `<Payload><OperationSet><Operation>` subelement are depicted in Figure 42.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

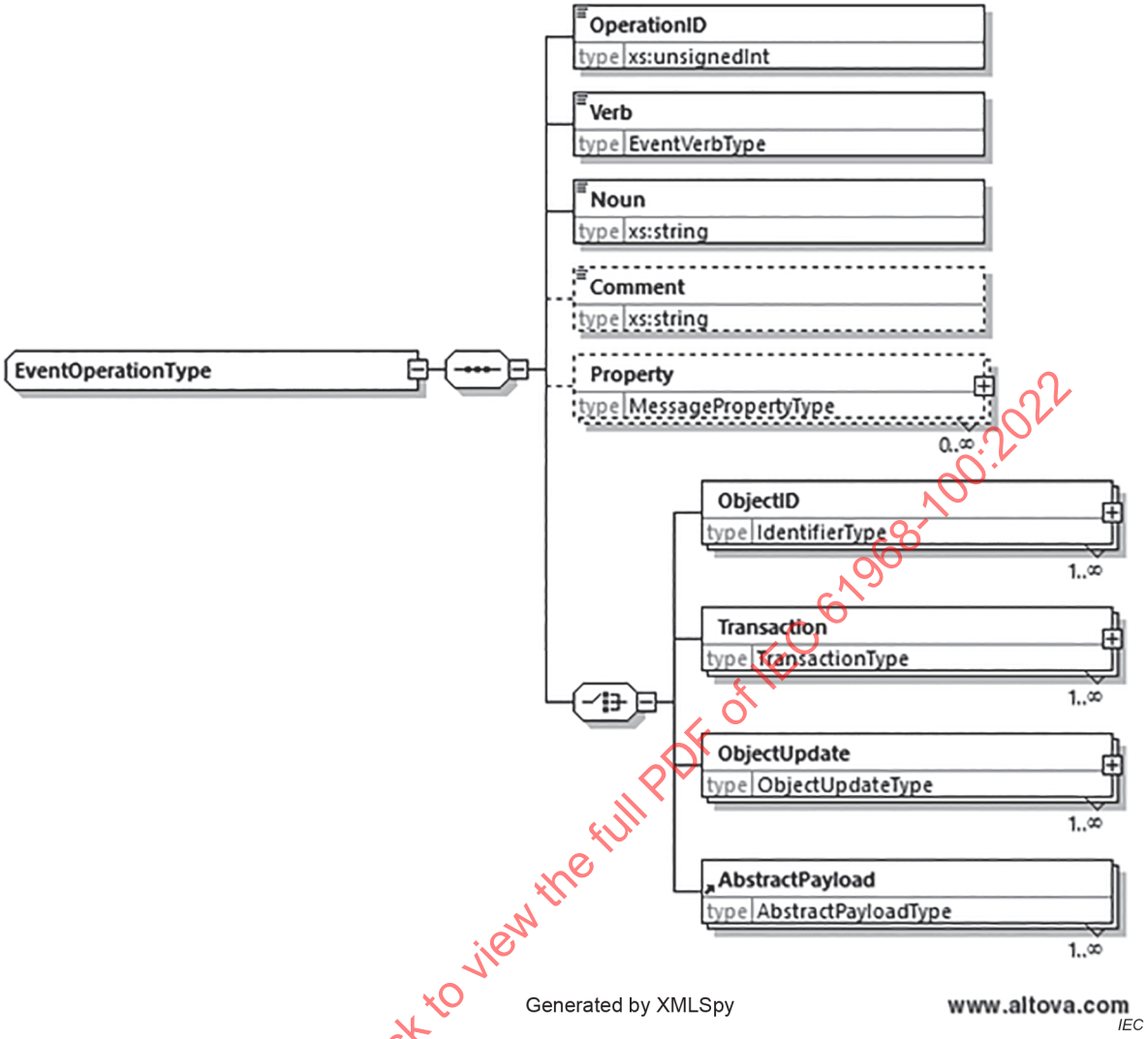


Figure 42 – Structure of the <Payload><OperationSet><Operation> element for event messages

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.7.3.6.6 The <Payload><OperationSet><Operation><Verb> element

For a request message (5.2) in which the <Header><Verb> is execute, the verbs in the <Payload><OperationSet><Operation><Verb> subelements shall be taken from Table 5.

For a response message (5.3) in which the <Header><Verb> is execute, the verbs in the <Payload><OperationSet><Operation><Verb> subelements shall be reply.

For an event message (5.4) in which the <Header><Verb> is execute, the verbs in the <Payload><OperationSet><Operation><Verb> subelements shall be taken from Table 6.

7.7.4 The <Payload><ObjectUpdate> element

7.7.4.1 General

The <Payload><ObjectUpdate> element shall be used for making partial changes (additions, modifications or replacements) either in part or in whole to an existing object (8.3.5).

Each <Payload><ObjectUpdate> element shall contain <UpdateID>, <UpdateAction> and <AbstractPayload> subelements

EXAMPLE Annex E contains example messages illustrating the usage of the <Payload><ObjectUpdate> sublement.

7.7.4.2 Structure

The contents of the <Payload><ObjectUpdate> sublement are depicted in Figure 43.

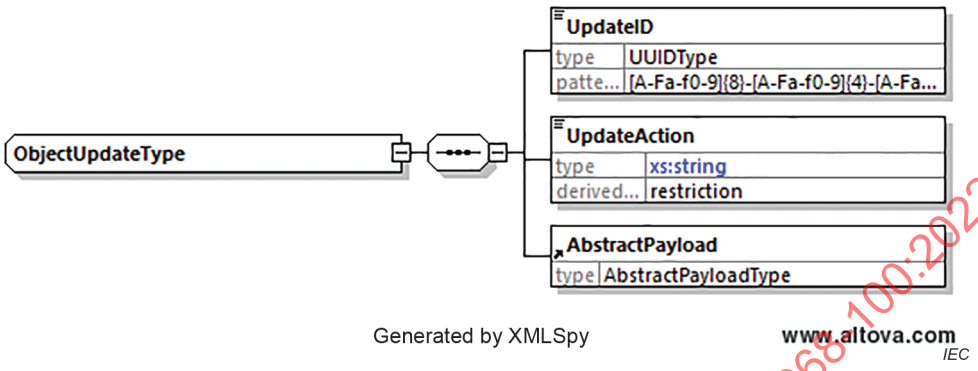


Figure 43 – Structure of the <Payload><ObjectUpdate> element

7.7.4.3 UpdatePayloadType subelements

Table 14 describes the various subelements that shall be present in a <Payload><ObjectUpdate> element and their associated semantics.

Table 14 – Subelements within the <Payload><ObjectUpdate> element

Element	Range of Cardinality		XSD Data Type	Meaning
<UpdateID>	1	1	xs:unsignedInt	See 7.6.4.5
<UpdateAction>	1	1	UpdateActionType	See 7.7.4.4
< ... >	1	1	AbstractPayloadType	This element may contain application-specific data. These data depend on the precise semantics of the request and are left for the relevant working group or other application developer to define. That is, these data have no normative meaning in the context of this International Standard. However, they may have normative semantics as far as other standards are concerned. See 7.7.5

7.7.4.4 UpdateAction

The value of the <Payload><ObjectUpdate><UpdateAction> element shall be one of the values listed in Table 15.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table 15 – Subelements within the <Payload><ObjectUpdate><UpdateAction> element

UpdateAction	Meaning	Clause/ Subclause
add	Specifies a request to add one or more elements to an existing object.	8.3.5.2
modify	Specifies a request to replace one or more elements within an existing object.	8.3.5.3
remove	Specifies a request to delete one or more elements from an existing object.	8.3.5.4

7.7.5 AbstractPayloadType data

An IEC 61968-100 message may convey application-specific data (8.2).

If present, such application-specific data shall be provided in the AbstractPayloadType sections of the <Payload> element (7.7), the <Payload><OperationSet><Operation> element (7.7.3.6) or the <Payload><OperationUpdate> (7.7.4) element. That is, the AbstractPayLoadType subelement is replaced by the application-specific data (8.2.2).

This International Standard does not prescribe these application-specific data which are generally expected to be normatively defined in other IEC 61968 Part standards. However, this does not preclude the use of IEC 61968-100 for other, non-IEC 61968 messaging purposes.

Clause 8 defines the normative conventions to which these application-specific data shall conform.

7.7.6 Multiple abstract payloads

A single message may contain multiple <AbstractPayload> elements (7.7.5). In such a case, the application-specific data contained in the <AbstractPayload> elements shall all have the same element name at the topmost level. However, this cannot be enforced by XSD validation alone.

EXAMPLE Clause G.5 shows an example a message containing multiple AbstractPayloadType elements.

7.8 The <CompressedPayload> element

7.8.1 General

The <CompressedPayload> element allows a payload to be compressed before being conveyed inside an IEC 61968-100 message.

The <Payload> and <CompressedPayload> elements are mutually exclusive.

The <CompressedPayload> element, if present, shall contain the same information as the equivalent <Payload> element. These data shall be compressed and then encoded into base64 representation according to RFC 4648.

7.8.2 Structure

The structure of the <CompressedPayload> element is shown in Figure 44.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

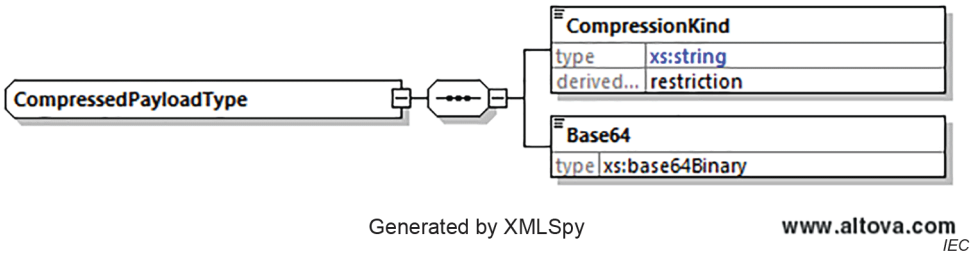


Figure 44 – Structure of the <CompressedPayload> element

7.8.3 CompressedPayload elements

Table 16 specifies the various subelements that shall be present in a <CompressedPayload> element.

Table 16 – Subelements within the <CompressedPayload> element

Element	Range of Cardinality		XSD Data Type	Meaning
<CompressionType>	1	1	enumeration	See 7.8.4
<Base64>	1	1	xs:base64Binary	See 7.8.5

7.8.4 The <CompressedPayload><CompressionType> subelement

The <CompressedPayload><CompressionType> subelement shall specify the method of compression that has been applied to the payload data. This element shall be assigned one of the values listed in Table 17.

Table 17 – Values of the <CompressedPayload><CompressionType> element

Value of the <CompressionType> Element	Meaning
zip	ZIP compression as defined by IEC 21320-1:2015
gzip	GZIP compression as defined by RFC 1952.
other	Some other compression mechanism

7.8.5 The <CompressedPayload><Base64> subelement

The <CompressedPayload><Base64> subelement shall contain the base64 representation of the compressed data.

7.9 The Transaction Type

The transaction type is used within the context of cancelling transactions (8.5).

An instance of a transaction type may occur as part of a <CancelRequestMessage><Payload><Transaction> element (7.4.2.2), a <Payload><OperationSet><Operation><Transaction> (7.7.3.6.2) or a <Reply><Status> element (7.6.4.12).

The transaction type has the structure depicted in Figure 45.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

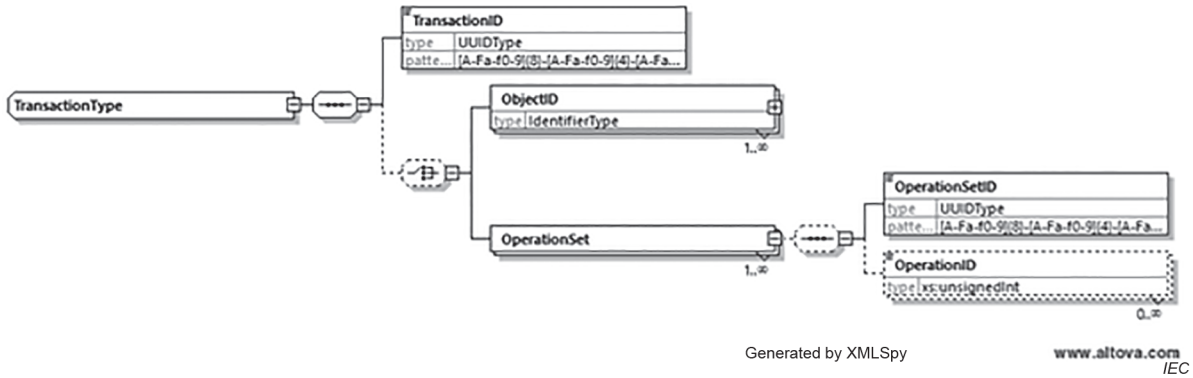


Figure 45 – Structure of the TransactionType

Table 18 describes the elements of a TransactionType.

Table 18 – Elements of a transaction type

Element	Range of Cardinality		XSD Data Type	Meaning
<TransactionID>	1	1	UUIDType	The transaction ID
<ObjectID>	0	many	IdentifierType	This element shall be used to identify the objects to which the transaction refers (8.6.5).
<OperationSet>	0	many		This element shall be used to identify the OperationSets to which the transaction refers (7.7.3).
<OperationSetID>	1	1	UUIDType	See 7.6.4.5
<OperationID>	0	many	xs:unsignedInt	This element shall be used to identify the subset of Operations within an OperationSet to which the transaction refers (7.7.3).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8 Constraints and conventions

8.1 General

This clause details constraints and conventions that are intended to help promote consistency and interoperability across all Parts and applications that make use of this document.

8.2 Application-specific data

8.2.1 General

This document does not prescribe the kind of application-specific data that are transmitted in a message (7.7.5).

The contents of the AbstractPayloadType sections are generally defined according to an application-specific XSD profile that has been specified by the relevant working group or local organization according to the application at hand (4.3).

The details of such application-specific profiles lie outside the scope of this document. They should, however, conform to the requirements laid out in the memorandum of understanding between the IEC and UN/CEFACT [1][1].

Such a XSD profile shall be assigned to a XML namespace that has been assigned for that purpose. The choice of XML namespace is non-normative and lies outside the scope of this International Standard.

8.2.2 The AbstractPayloadType

The semantics of any application-specific data are non-normative as far as this document is concerned. However, such data may have normative semantics as far as other application or domain standards are concerned.

The application-specific profile shall inherit from the XSD abstract data type `AbstractPayloadType` which is defined in the `AbstractPayloadType.xsd` file (Clause A.3). This `AbstractPayloadType` is defined as a XSD `<xs:substitutionGroup>` directive. This implies that the `AbstractPayloadType` sections should be replaced by the corresponding application-specific data.

An application-specific profile may be defined by including the `AbstractPayloadType.xsd` file using a `<xs:import>` directive. XSD inheritance shall be accomplished by means of the `<xs:extension>` and `<xs:substitutionGroup>` directives.

EXAMPLE Examples of XSD profiles that illustrate the usage of the AbstractPayloadType are shown in Annex C. Example messages exemplifying corresponding XML messages may be found in Annex D and Annex E.

8.2.3 Application-specific customisation

For the purposes of providing local customisation, an organisation may define its own application-specific XSD profiles. Such customisations are specific to the local organisation and have no normative meaning.

If such an organisation-specific profile is required, it shall be assigned an organisation-specific XML namespace in order to emphasize that a non-standard extension is being used.

An organisation may also use the `<Header><Revision>`, `<Header><Source>`, `<Header><User>` and `<Header><Property>` elements in order to provide further customisations.

8.2.4 RDF data

The normative XSD file listed in Clause A.5 may be used in the context of an AbstractPayloadType to carry an RDF payload as prescribed by IEC 61970-552:2016.

8.3 Object management requests and notifications

8.3.1 General

An object management request shall take place within the context of a request/response message exchange pattern (6.2).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

[1] Numbers in square brackets refer to the bibliography.

A request message with a verb of `create`, `change`, `delete` or `update` sent from a requesting to a responding system shall be interpreted as a request for the responding system to instantiate, replace, remove or modify the corresponding application-specific object respectively (8.3.2, 8.3.3, 8.3.4, 8.3.5).

This clause describes the success cases for the various object management requests. For failure cases, refer to 7.6.4.3.

This clause does not apply to creating, changing, deleting or updating the identifiers (`<mRID>` or `<Names>` elements) of an application-specific object (8.7).

8.3.2 Create request message

8.3.2.1 Making a create request message

A create request message shall, in the absence of errors, cause one or more application-specific objects to be instantiated on the responding system.

A create request message shall be denoted by a `<Header><Verb>` of `create` (7.5.3.3).

The associated information necessary for creating the new object or objects shall be contained in the `<Payload>` element of the create request message (7.7).

EXAMPLE D.2.3 provides an example of a create request message.

8.3.2.2 Naming an object

The newly instantiated object shall be given one or more identifiers by which it can be referenced in subsequent requests.

The newly instantiated object shall be identified by specifying the corresponding `<mRID>` or `<Name>` fields in the provided application-specific `<Payload>` element (7.7, 8.6).

An object may be known by more than one name.

8.3.2.3 Successful instantiation of an object

If the request to create a particular object is successful and the newly instantiated object contains no information other than that contained in the request, the responding system may return an identical copy of the object in the `<Payload>` element of a response message (7.7).

If the request to create a particular object is successful on the responding system, the newly instantiated object may be augmented with information beyond that contained in the request. For example, the new object may contain supplementary object identifiers, names or additional information. In such a case, the responding system may return a copy of the enhanced object inside the `<Payload>` element of a response message (7.7).

The responding system system may, at its discretion, elect not to return a copy of the newly instantiated object. For example, the responding system may not have at hand all information belonging to the object and to assemble a complete copy of the object by interrogating other systems would be prohibitively expensive in computational terms.

8.3.3 Change request message

8.3.3.1 Making a change request message

A change request shall be used to replace one or more existing application-specific objects on the responding system. The specified objects are replaced in their entirety with the new data.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

A change request message shall be denoted by a <Header><Verb> of change (7.5.3.3).

The objects for which the change request is being made shall be identified by specifying the corresponding <mRID> or <Name> fields in the provided application-specific <Payload> element (clauses 7.7, 8.6).

The application-specific data that are intended to replace the existing object or objects shall be provided in the <Payload> element of the change request message (7.7).

8.3.3.2 Successful replacement of an object

If the request to replace a particular object is successful, the replaced object may be associated with information beyond that contained in the request. For example, the replaced object may contain supplementary object identifiers, names or other additional information. In such a case, the responding system may return a copy of the replaced object inside the <Payload> element of a response message (7.7).

The responding system system may, at its discretion, elect not to return a copy of the replaced object. For instance, it may be prohibitively expensive in computational terms for the responding system to generate a complete copy of the replaced object.

8.3.4 Delete request message

8.3.4.1 General

A delete request shall be used to remove one or more application-specific objects in their entirety from the responding system. A delete request message shall be denoted by a <Header><Verb> of delete (7.5.3.3).

8.3.4.2 Successful deletion of an object

The <Payload> element of the delete request message should contain no values other than any <mRID> or <Name> fields. The responding system shall ignore any other fields.

The objects that are to be deleted in their entirety shall be identified by one of the following methods:

- Either by means of the <Payload><ObjectID><mRID> or <Payload><ObjectID><Name> elements to specify the objects in question (8.6.5);
- Or by specifying the corresponding <mRID> or <Name> fields in the provided application-specific <Payload> (7.7, 8.6).

EXAMPLE D.14.3 provides an example of a delete request message.

8.3.5 Update request message

8.3.5.1 General

An update request message shall be used to make modifications to one or more existing, application-specific objects on the responding system without replacing that object in its entirety.

The request for making such an object update is contained within a <Payload><ObjectUpdate> element.

The objects for which the update request is being made shall be identified by specifying the corresponding <mRID> or <Name> fields in the provided application-specific <Payload><ObjectUpdate><AbstractPayload> element (7.7, 8.6).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

The request to be carried out is determined according to the value of the `<Payload><ObjectUpdate><UpdateAction>` element (7.7.4.4, 8.3.5.2, 8.3.5.3, 8.3.5.4).

8.3.5.2 Adding elements to an existing object

If the value of the `<Payload><ObjectUpdate><UpdateAction>` element (7.7.4.4) is `add`, then any application-specific elements (other than the `<mRID>` or `<Name>` fields used to identify that object) contained in the `<Payload><ObjectUpdate><AbstractPayload>` element shall be added to the specified object on the responding system.

EXAMPLE Examples of adding elements to an object may be found in Clauses E.5, E.10, E.11, E.17 and E.20.

8.3.5.3 Modifying elements in an existing object

If the value of the `<Payload><ObjectUpdate><UpdateAction>` element (7.7.4.4) is `modify`, then any application-specific elements (other than the `<mRID>` or `<Name>` fields used to identify that object) contained in the `<Payload><ObjectUpdate><AbstractPayload>` element shall be used to replace those belonging to the specified object on the responding system.

EXAMPLE Examples of modifying elements in an existing object may be found in Clauses E.3, E.6, E.7, E.12, E.15 and E.18.

8.3.5.4 Removing elements from an existing object

If the value of the `<Payload><ObjectUpdate><UpdateAction>` element (7.7.4.4) is `remove`, then any application-specific elements (other than the `<mRID>` or `<Name>` fields used to identify that object) contained in the `<Payload><ObjectUpdate><AbstractPayload>` element shall be removed from the specified object on the responding system.

In some cases it may be necessary to use an element other than `<mRID>` or `<Name>` in order to identify those elements to be removed. An example of such a case is provided in Clause E.13.

Any value provided for an element being removed shall be ignored.

NOTE It may be necessary to provide a dummy value for an element that is being removed in order that the payload validates against the XSD schema.

EXAMPLE Examples of removing elements from an existing object may be found in Clauses E.4. E.8, E.9, E.13, E.14, E.19 and E.21.

8.3.6 Object management notifications

An object management notification shall be formed as an event message (5.4). Such a notification shall take place within the context of the unsolicited event message exchange pattern (6.3).

An event message with a verb of `created`, `changed`, `deleted` or `updated` sent from a sending system to a receiving system shall be interpreted as a notification that the specified object on the sending system has been instantiated, replaced, removed or modified respectively.

EXAMPLE An example of an object management notification may be found in Clause D.3.

8.4 Get requests

8.4.1 Making a get request

A request message with a `<Header><Verb>` of `get` shall be interpreted as a query to retrieve one or more application-specific objects whose type is specified by the noun in the request.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

The application-specific query parameters (4.3) that are used to specify the corresponding objects to be returned shall be identified by one of the following methods:

- Either by means of the `<Payload><ObjectID><mRID>` or `<Payload><ObjectID><Names>` elements (clause 8.6.5);
- Or by specifying the appropriate selection fields in a provided application-specific query profile. The query parameters are then specified in the `<AbstractPayload>` element.

In general, a get request should be idempotent. That is, it should not alter the state of the objects being queried. However, a get request may result in the creation of new data in the responding system. For example, an IEC 61968-9:2013 on-demand get(MeterReadings) request may cause data to be read from the AMI network into one of more new MeterReading objects.

EXAMPLE D.4.3, D.7.3, D.8.3, D.9.3, D.11.2 and D.13.2 provide examples of get request messages.

8.4.2 Specifying multiple constraints in a request message

8.4.2.1 General

A given XSD file that defines an application-specific query profile may allow a request message to specify multiple selection parameter constraints (7.7). This clause defines how such parameter constraints may be defined in order to achieve the desired logical "AND" and "OR" operations described in 8.4.2.2 to 8.4.2.5.

The rules stated in Subclauses 8.4.2.2 through 8.4.2.4 are known together as the cross-product rule.

8.4.2.2 Logical "OR" over the specified criteria

If a request specifies several selection parameter constraints that belong to the same class of criteria (that is, they share the same XML element name at the topmost level) these shall be interpreted as a logical "OR" operation.

EXAMPLE Clause G.2 contains an example of specifying logical "OR" operations over the specified criteria.

8.4.2.3 Logical "AND" over the specified criteria

When several top-level selection parameter constraints are specified in a request and these all refer to different classes of criteria (that is, they do not share the same XML element name at the topmost level), then these constraints shall be interpreted as a logical "AND" operation.

EXAMPLE Clause G.3 contains an example of specifying logical "AND" operations over the specified criteria.

8.4.2.4 Multiple constraints

If a request specifies combinations of the various selection parameter criteria, this shall be interpreted as a request to return data for all combinations of the specified criteria.

EXAMPLE Clause G.4 contains an example of specifying a combination of logical "AND" and "OR" operations over the specified criteria.

8.4.2.5 Specifying multiple application-specific profiles within a single message

If a get request message contains several application-specific profiles (7.7.6), this provides an equivalent means of specifying multiple selection criteria that are logically "OR"ed together.

EXAMPLE Clause G.5 contains an example of specifying multiple AbstractPayloadType profiles in a single message.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8.4.3 Responding to a get request

After attempting to carry out the get request, the responding system shall reply with one or more response messages that contain the requested data along with any informational conditions or error indications (5.3.5, 6.2.2, 7.7).

8.5 Cancelling transactions

8.5.1 General

A cancellation request message sent from a requesting to a responding system shall be interpreted as a request for the responding system to terminate the specified transactions.

NOTE 1 A "transaction" in this context refers to a single, indivisible operation.

NOTE 2 Cancellations are inherently problematic in that it may be impossible for a receiving system to cancel a transaction. If the cancellation is for a request whose action is to take place in the future or which defines an idempotent operation then there is generally no problem. However, if the cancellation request is for a transaction that has already started or has some external effect – for example, an IEC 61968-9:2013 create(EndDeviceControls) message – it may not always be possible to cancel all requested transactions. Network timing issues may also complicate the picture because the original transaction may be already in progress when the cancellation request is received.

8.5.2 Message exchange pattern

A message exchange pattern in which a request is partially or completely cancelled is shown in Figure 46. Here it is assumed that the underlying request is part of a consequential event message exchange pattern (6.4).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

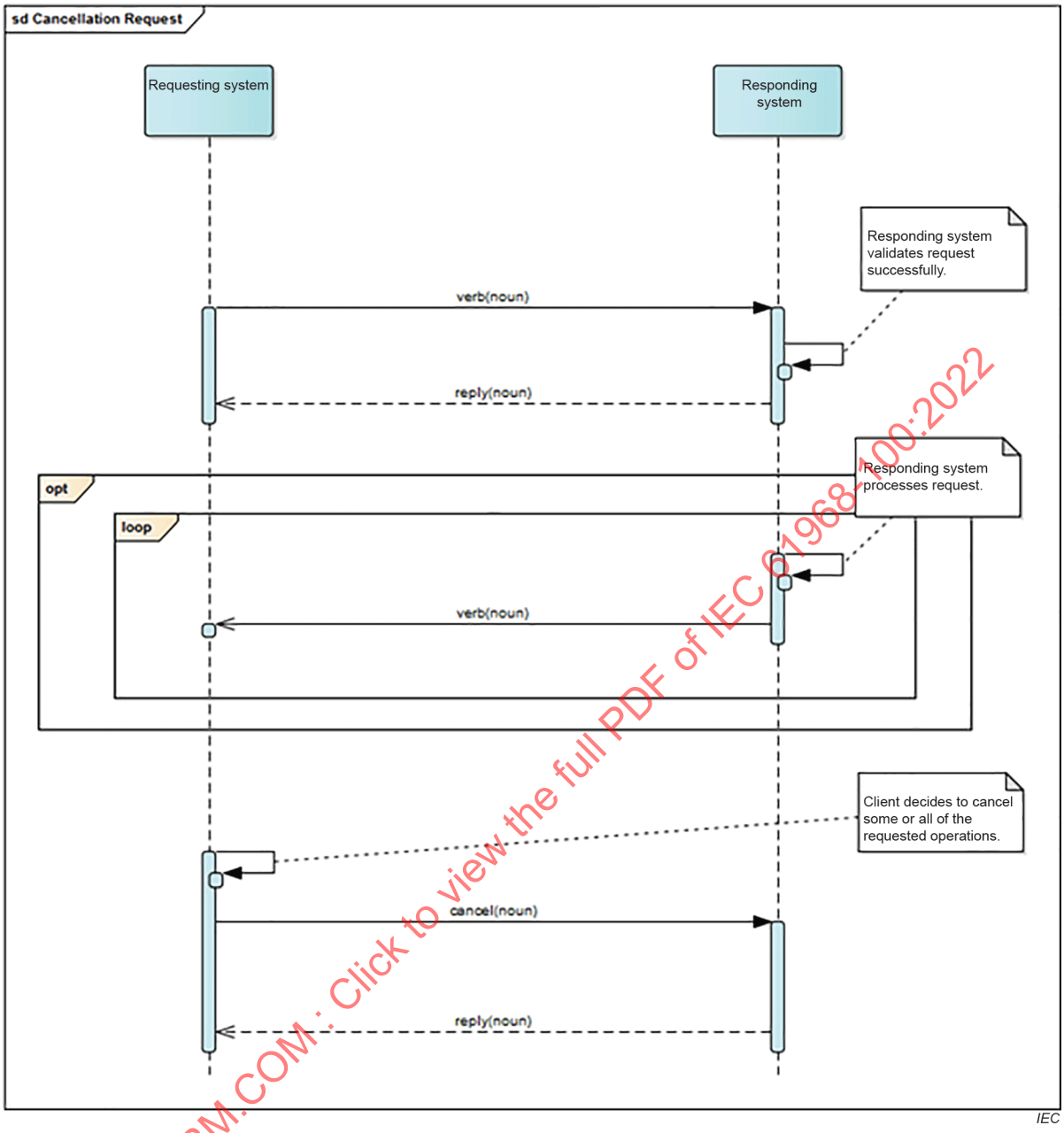


Figure 46 – Message Exchange Pattern for a Cancellation Request

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8.5.3 Transactions IDs

A transaction that is available for cancellation is identified by a transaction ID. A transaction ID shall be specified as a 128-bit universally unique identifier (UUID) generated according to RFC 4122.

When a responding system receives a request that it determines is capable of being cancelled, the responding system shall generate a transaction ID and return it to the requesting system. In order to request the subsequent cancellation of the original transaction, the requesting system shall supply this transaction ID in a subsequent cancellation request message.

A responding system may, at its discretion, return several transaction IDs for a given request. This allows a finer granularity for cancelling parts of the request.

EXAMPLE Clause D.16 shows an example of the usage of several transaction IDs.

Not all operations may be cancelled. The responding system has discretion to decide for which requests and at which level of granularity it will provide transaction IDs.

8.5.4 Receiving a transaction ID

A response message may contain zero or more transaction IDs. Such a transaction ID shall be carried within a `<Reply><Transaction><TransactionID>` element (8.6.5).

Each transaction ID shall be specified together with a `<Reply><Transaction><ObjectID><mRID>` or one or more `<Reply><Transaction><ObjectID><Names>` elements (8.6). These allow the requesting system to identify the object to which the transaction ID applies.

EXAMPLE Clause Clause D.15.2 and D.16.1 show examples of response messages that contain transaction IDs.

8.5.5 Sending a cancellation request message

A cancellation request message shall have a header verb of `cancel`. The header noun shall be the same as in the original request to which the cancellation message pertains.

The transactions to be cancelled shall be specified by the respective transaction IDs. These shall be carried in separate `<Payload><Transaction><TransactionID>` elements.

EXAMPLE D.15.3 and D.16.2 show examples of a cancellation request message.

8.5.6 Responding to a cancellation request message

The responding system shall respond to a cancellation request message with a response message (5.3).

This response message shall specify the status of the cancellation request in the `<Reply><Status><Code>` element (7.6.4.7). This may take on any one of three possible values.

`0.0` – This value shall indicate that the cancellation request was carried out successfully.

`0.4` – This value shall denote that the cancellation request was received and accepted successfully. The actual cancellation operation shall be queued for execution. After the responding system has carried out the cancellation operation, it shall inform the requesting system of the success or otherwise of the cancellation operation by sending a subsequent response message (5.3) containing an appropriate `<Reply><Status><Code>` element.

`2.16` – This value shall indicate that the cancellation request could not be carried out because an invalid transaction ID was specified.

Any other value of the `<Reply><Status><Code>` element shall indicate that some other error in the cancellation request was detected.

8.6 Object identification

8.6.1 General

Although this document does not express an opinion concerning the application-specific data that are transmitted within Part 100 messages, it specifies a consistent means for referencing any CIM or application-specific objects from within the `<Reply>` or `<Payload>` elements of a message.

The precise manner by which a given object may be referenced is termed its identification.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8.6.2 Rules for object identification

The classes and attributes for the naming of CIM objects are specified in IEC 61970-301.

An object shall be uniquely identified by means of a single <mRID> element (8.6.3) or by one or more <Name> elements (8.6.4.2). These means of identification are not mutually exclusive. The <mRID> method of naming should be used whenever possible.

The set of identifiers referring to an object may be extended or modified (8.7).

An object shall have at least one unique identifier that is assigned when it is instantiated (8.3.2). This identifier may subsequently be used to reference the object – for instance, when the object is to be queried, updated or deleted (8.6.5).

8.6.3 Identifying objects using an <mRID> element

A master resource identifer (mRID) should be used to identify an object.

An mRID shall be specified as a 128-bit universally unique identifier (UUID) generated according to RFC 4122.

A given object shall have no more than one mRID.

8.6.4 Identifying objects using a <Name> element

8.6.4.1 General

A <Name><name> element may be used to identify an object.

The content of the <Name><name> element is non-normative.

If the <Name><name> element does not uniquely identify the object, the name may be additionally qualified by specifying a <Name><NameType><name> element. The content of this <Name><NameType><name> element is non-normative.

If this still results in a potential ambiguity, the name may be further qualified by specifying a <Name><NameType><NameTypeAuthority><name> element. The content of this element is non-normative.

8.6.4.2 Uniqueness of <Name> elements

Not all <Name> identifiers must be unique. However, if an object is not associated with an <mRID> identifier then at least one of its <Name> identifiers must be unique.

A <Name> element may be assigned for purposes other than object identification. Such a <Name> element need not be unique.

8.6.4.3 Structure

The structure of the <Name> element is depicted in Figure 47.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

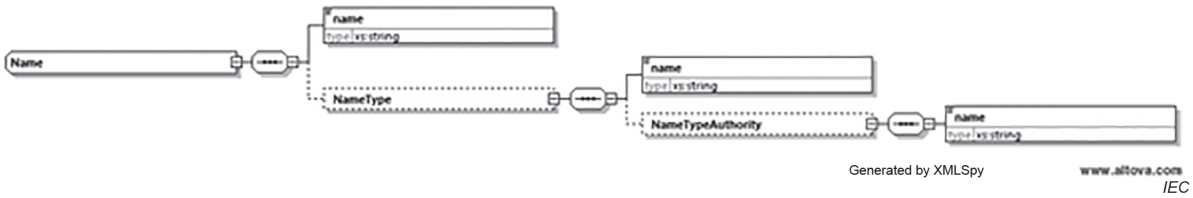


Figure 47 – General Naming Scheme

8.6.5 Referencing objects

8.6.5.1 General

It may be necessary to make direct reference to an object from within a message. This can be achieved by using the `<ObjectID>` subelement of the `<Reply><Status>` and `<Payload>` elements. Such examples include:

- using the `<Reply><Status><ObjectID>` element to reference an object from within a response message for the purpose of associating error or status notifications (7.6.4.6).
- using the `<Payload><ObjectID>` element to reference an object from within a "create", "change", "delete" or "update" request (8.3.2, 8.3.3, 8.3.4, 8.3.5)
- using the `<Payload><ObjectID>` element to reference an object from within a cancellation request message (8.5).

8.6.5.2 Structure

The `<ObjectID>` elements share a similar structure as shown in Figure 48.

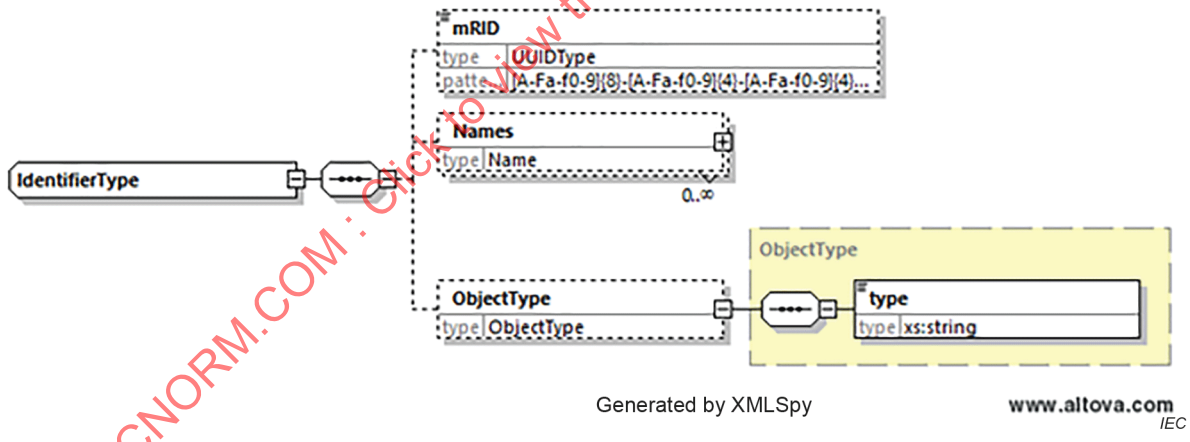


Figure 48 – Structure of an <ObjectID> element

Table 19 specifies the various subelements that may be present in an `<ObjectID>` element and their associated semantics.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table 19 – Subelements within an <ObjectID> element

Element	Range of Cardinality		XSD Data Type	Meaning
<mRID>	0	1	UUIDType	The mRID of the object being referenced (8.6.3).
<Names>	0	many	Name	The name of the object being referenced (8.6.4).
<ObjectType>	0	1	ObjectType	
<type>	1	1	xs:string	This element shall be used to identify the kind of object to which the name refers. This provides a further means for disambiguating named objects. The value of this element shall be the name of the actual class (and not that of a parent class) of the object being referenced.

8.7 The ObjectIdentificationRevisionSets profile

8.7.1 General

The ObjectIdentificationRevisionSets profile allows the identifiers (8.6) of an existing object to be added (8.7.4), removed (8.7.5) or modified.

Such a request shall take place within the context of a request/response message exchange pattern (6.2).

The XSD for the ObjectIdentificationRevisionSets profile is defined in the file `LegacyObjectIdentificationRevisionSets.xsd`. The contents of this file are listed in Clause A.4.

The ObjectIdentificationRevisionSets profile provides the only method for the identifiers of an existing object to be added, removed or modified.

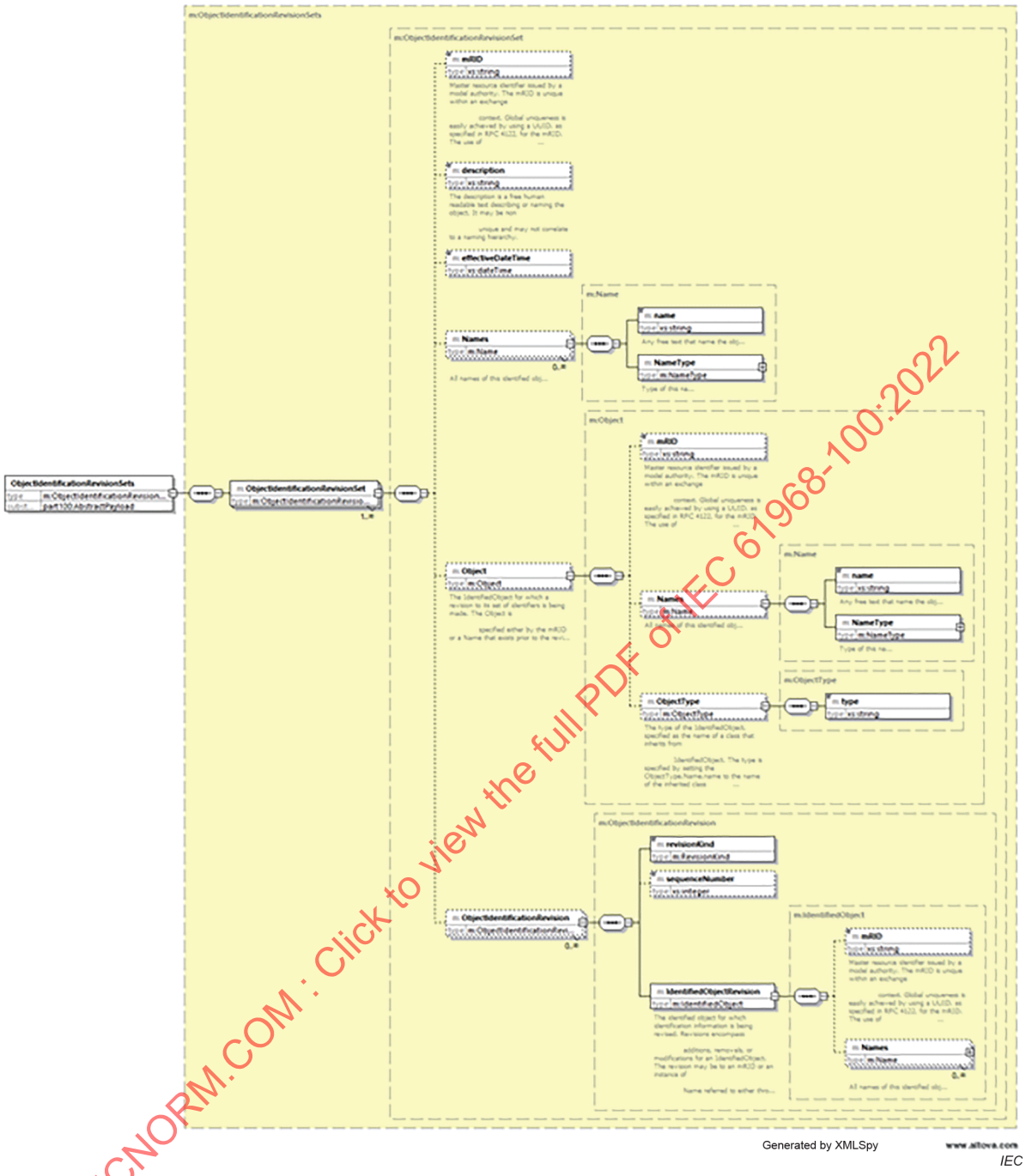
The modification of an identifier shall be accomplished by the combination of removing the existing identifier and adding a new one.

EXAMPLE Examples illustrating the use of ObjectIdentificationRevisionSets messages to add or remove identifiers may be found in Annex F.

8.7.2 Structure

The ObjectIdentificationRevisionSets profile is depicted in Figure 49.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 49 – ObjectIdentificationRevisionSets profile

Table 20 describes the various subelements that may be present in an ObjectIdentificationRevisionSets message and their associated semantics.

Table 20 – Elements of an ObjectIdentificationRevisionSets profile

Element	Range of Cardinality		XSD Data Type	Meaning
<mRID>	0	1	xs:string	Allows the ObjectIdentificationRevisionSets to be given an identity. .
<description>	0	1	xs:string	
<Names>	0	many	Name	
<effectiveDateTime>	0	1	xs:dateTime	Specifies the time at which the identifier addition or deletion should take place. If omitted, the default is for the operation to take place immediately.
<Object>	0	1	Object	Specifies the existing object whose identifier(s) are to be added or removed.
<mRID>	0	1	xs:string	The existing mRID of the object being referenced (8.6.3, 8.7.4, 8.7.5).
<Names>	0	1	Name	An existing Name of the object being referenced (8.6.4, 8.7.4, 8.7.5).
<ObjectType>	0	1	ObjectType	
<type>	1	1	xs:string	This element shall be used to identify the kind of object to which the name refers. This provides a further means for disambiguating named objects. The value of this element shall be the name of the actual class (and not that of a parent class) of the object being referenced.
<ObjectIdentificationRevision>	0	many	ObjectIdentificationRevision	Specifies the identifiers being added or removed
<revisionKind>	1	1	RevisionKind	Either add or remove
<sequenceNumber>	0	1	xs:integer	Specifies the order in which multiple addition or removal requests should be carried out.
<IdentifiedObjectRevision>			ObjectIdentifiedRevision	
<mRID>	0	1	UUIDType	The added or removed mRID of the object being referenced (8.6.3, 8.7.4, 8.7.5).
<Names>	0	many	Name	The added or removed Name(s) of the object being referenced (8.6.4, 8.7.4, 8.7.5).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8.7.3 Using the ObjectIdentificationRevisionSets profile

A request message with a <Header><Noun> of ObjectIdentificationRevisionSets shall be interpreted as a request to add, remove or modify one or more identifiers associated with a given object (clauses 8.7.4, 8.7.5).

An OperationSet (clause 7.7.3) may also be used in order to combine several of these operations within a single message.

8.7.4 Adding one or more new identifiers

The ObjectIdentificationRevisionSets profile may be used to assign new identifiers to an existing object.

The <Header><Verb> shall be create.

The <Payload><ObjectIdentificationRevisionSets><ObjectIdentificationRevisionSet><ObjectIdentificationRevision><revisionKind> element shall be add.

One of the <Payload><ObjectIdentificationRevisionSets><ObjectIdentificationRevisionSet><Object><mRID> or <Name> elements shall be specified. This defines the existing object whose set of identifiers is to be extended.

The <Payload><ObjectIdentificationRevisionSets><ObjectIdentificationRevisionSet><ObjectIdentificationRevision><IdentifiedObjectRevision><mRID> or <Name> elements shall also be specified. These shall specify the additional identifiers by which the object is to be known.

Because an object may have no more than one mRID (8.6.3), an <mRID> may only be added to an object that has no prior existing mRID.

8.7.5 Removing one or more identifiers

The ObjectIdentificationRevisionSets profile may be used to remove one or more identifiers of an existing object.

The existing mRID (8.6.3) of an object may be removed although such practice is discouraged.

The <Header><Verb> shall be create.

The <Payload><ObjectIdentificationRevisionSets>

<ObjectIdentificationRevisionSet><ObjectIdentificationRevision><revisionKind> element shall be remove.

One of the <Payload><ObjectIdentificationRevisionSets><ObjectIdentificationRevisionSet><Object><mRID> or <Name> elements shall be specified. This defines the existing object whose identifiers are to be removed.

The <Payload><ObjectIdentificationRevisionSets><ObjectIdentificationRevisionSet><ObjectIdentificationRevision><Identifier><mRID> or <Name> elements shall also be specified. These shall specify the identifiers that are to be removed.

8.7.6 Possible Errors Pertaining To ObjectIdentificationRevisionSets Requests

The <Reply><Status><Code> codes listed in Table 10 shall be used to communicate a notification either of success or that of an error encountered while executing an ObjectIdentificationRevisionSets request message.

9 Transporting messages

9.1 General

This clause details considerations pertaining to all IEC 61968-100 applications that exchange messages over SOAP Web Services (9.2) and JMS (9.3). Other transport services may also be used but their usage is non-normative and lies outside the scope of this International Standard.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

9.2 Transporting over SOAP web services

9.2.1 General

This clause describes the use of Simple Object Access Protocol (SOAP) 1.1 web services over HTTP for the transport of IEC 61968-100 messages.

9.2.2 Encoding of IEC 61968-100 messages inside SOAP messages

An IEC 61968-100 message shall be conveyed within the `<soap:Envelope><soap:Body>` element of the SOAP message.

The `SOAPAction` and other HTTP headers have no normative values as far as this International Standard is concerned. Their values are left for the system integrator to define.

9.2.3 Example SOAP message

Figure 50 illustrates how the example IEC 61968-100 message of Figure D.28 may be conveyed inside a SOAP message.

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header/>
  <soap:Body>
    <GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Header>
        <Verb>get</Verb>
        <Noun>MeterReadings</Noun>
        <Timestamp>2018-10-02T14:54:31-06:00</Timestamp>
        <MessageID>1f92de80-b835-4074-b42e-233164063a39</MessageID>
        <CorrelationID>6d6fe897-b588-40a2-ba30-55e4d838b6f4</CorrelationID>
        <Comment>Request for historical meter read data</Comment>
      </Header>
      <Payload>
        <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <EndDevice>
            <Names>
              <name>M001</name>
            </Names>
          </EndDevice>
          <ReadingType>
            <Names>
      <!--bulkQuantity forward electricitySecondaryMetered energy (kWh) -->
              <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0</name>
            </Names>
          </ReadingType>
          <TimeSchedule>
            <scheduleInterval>
              <end>2018-07-01T09:50:00-06:00</end>
              <start>2018-07-01T09:00:00-06:00</start>
            </scheduleInterval>
          </TimeSchedule>
        </GetMeterReadings>
      </Payload>
    </GetRequestMessage>
  </soap:Body>
</soap:Envelope>
```

Figure 50 – Example SOAP message

9.2.4 Mapping of IEC 61968-100 messages to SOAP messages

The usage of SOAP over HTTP requires that all messages at the transport level be exchanged in a synchronous request/response message exchange pattern.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

The message exchange patterns defined in 6.2, 6.3 and 6.4 allow for asynchronous response messages to be sent from a receiving to a sending system. In order to allow the underlying SOAP transport to send a response message independently of the application-level message exchange pattern, a simple acknowledgement message (5.3.6) shall be sent in the synchronous HTTP response (6.5).

EXAMPLE Examples of simple acknowledgement messages may be found in D.3.4, D.6.4, D.8.6, D.8.8 and D.12.6.

9.2.5 WSDL files

9.2.5.1 General

This document normatively defines two kinds of WSDL files for use when transporting IEC 61968-100 messages over SOAP. These are termed the generic WSDL and strongly-typed WSDL files. Both kinds of WSDL file shall be generated using the template files that form part of this International Standard (Annex B).

A strongly-typed WSDL file has the advantage over the generic WSDL file in that it offers stricter type checking. In particular, the AbstractPayloadType element (7.7.5) is replaced by a specific XSD profile.

A conformant system shall implement a web services interface defined by either the generic or strongly-typed WSDL template files.

An implementation may define its web services according to the integration task at hand. It may be necessary to merge or otherwise edit the normatively generated WSDL files to achieve this.

9.2.5.2 Generic WSDL files

For a SOAP client system, a generic WSDL specifies the HTTP endpoint to which all SOAP messages should be sent.

For a SOAP server system, a generic WSDL allows a SOAP endpoint to accept IEC 61968-100 messages and to dispatch these internally according to their respective root elements.

A generic WSDL file shall be created by means of the `GenericMessageWSDLTemplate.txt` template file whose contents are listed in Clause B.2.

In order to generate a generic WSDL file from this template file, it is necessary first to copy the template file to a new location and then to edit its contents according to Table 21. Each listed variable should be replaced globally throughout the file.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table 21 – Generic WSDL template variable names

Variable Name	Meaning	Example
`{PATHPREFIX}`	Folder in which the Message.xsd file resides	../XSDs
`{SOAPADDRESS}`	HTTP endpoint to which SOAP messages should be directed	http://example.com

9.2.5.3 Strongly-typed WSDL files

For a SOAP client system, a strongly-typed WSDL file specifies the HTTP endpoint to which all SOAP messages should be sent.

For a SOAP server system, a strongly-typed WSDL file allows a SOAP endpoint to accept and to dispatch received IEC 61968-100 messages internally according to their respective root elements and profile names.

Creation of a strongly-typed WSDL file is a two-stage process. First, a strongly-typed message XSD profile file must be created by copying and editing the `StrongMessageXSDTemplate.txt` template file whose contents are listed in Clause B.3. This is to be edited according to Table 22. Each listed variable should be replaced globally throughout the file.

Table 22 – Strongly-typed message XSD template variable names

Variable Name	Meaning	Example
{GETPROFILENAMESPACE}	Namespace for "get" requests (8.4)	http://iec.ch/TC57/2011/GetMeterReadings#
{GETPROFILENAME}	Profile name for "get" requests (8.4)	GetMeterReadings
{PROFILENAMESPACE}	Namespace for all other requests and responses	http://iec.ch/TC57/2011/MeterReadings#
{PROFILENAME}	Profile name for all other requests and responses	MeterReadings
{PROFILEPATHPREFIX}	Folder in which the part-specific XSD files resides	../Part9XSDs
{PART100PATHPREFIX}	Folder in which the Message.xsd file resides	../XSDs

Second, the strongly-typed WSDL file must be created by copying and editing the `StrongMessageWSDLTemplate.txt` template file whose contents are listed in annex B.4. This must be edited according to Table 23. Each listed variable should be replaced globally throughout the file.

Table 23 – Strongly-typed WSDL template variable names

Variable Name	Meaning	Example
{GETPROFILENAME}	Profile name for "get" requests (8.4)	GetMeterReadings
{PROFILENAME}	Profile name for all other requests and responses	MeterReadings
{PATHPREFIX}	Folder in which the Message.xsd file resides	../XSDs
{SOAPADDRESS}	HTTP endpoint to which SOAP messages should be directed	http://example.com

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

9.3 Transporting over JMS

9.3.1 General

This clause describes the use of the Java Message Service (JMS) for the transport of IEC 61968-100 messages.

According to the JMS specification, "JMS provides a common way for Java programs to create, send, receive and read an enterprise messaging system's messages". It describes a set of interfaces rather than an underlying implementation. It is generally left up to the implementor to choose a conforming implementation.

This clause describes normative and informative considerations that an application compliant with IEC 61968-100 shall take into account when using JMS to transport messages.

9.3.2 Message exchange patterns

In general, JMS queues should be used for sending and receiving messages in the context of the request/response message exchange pattern (6.2) or the consequential message exchange pattern (6.4).

JMS topics should generally be used for sending and receiving messages in the context of the unsolicited event message exchange pattern (6.3). JMS topics may also be suitable for the consequential event message pattern (6.4) where multiple parties are interested in the actual event.

Apart from these considerations, it is left up to the implementor to decide how the messaging infrastructure should be arranged in terms of JMS queues and topics.

9.3.3 Application-level acknowledgement messages

JMS allows a transport-level acknowledgement to be automatically transmitted by a responding system on the receipt of a message. JMS also allows for the persistence of messages. Taken together, automatic acknowledgement of messages and message persistence can go a long way at the transport level to provide assurance that a transmitted message is received by a responding system.

However, if it is necessary for an application to ensure that a transmitted message has been received by its peer application, the only guaranteed way of providing such assurance is for the peer application to return an acknowledgement at the application level. This may especially be the case if there are several hops between one system and its peer.

Therefore, an application should in general set the `<AckRequired>` flag in the IEC 61968-100 message header to the value `true` (7.5) in order to force the peer application system to return a simple acknowledgement message (6.5). The receipt of such a message ensures that the original message has been received by its peer.

NOTE Such a simple acknowledgement message is necessary only for the cases of sending an event message (5.4), or a request message (5.3) that has a <Header><Verb> of reply (7.5.4.3). For all other cases, the responding system returns a response message (6.2, 6.4).

9.3.4 JMS headers

9.3.4.1 General

JMS provides several transport-level header fields which may be set at the convenience of an application. These include JMSMessageID, JMSCorrelationID, JMSReplyTo and JMSTimestamp.

9.3.4.2 The JMSMessageID header field

The JMS specification states that the "JMSMessageID header field contains a value that uniquely identifies each message sent by a provider". JMS imposes the restriction that the `JMSMessageID` header must start with the prefix "`ID:`".

The `JMSMessageID` header field hence serves a similar purpose to the `<MessageID>` subelement of the IEC 61968-100 `<Header>` element. Therefore, for applications compliant with IEC 61968-100, the `JMSMessageID` header field should be assigned the value of the `<MessageID>` subelement (7.5) prefixed by "`ID:`".

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

9.3.4.3 The JMSCorrelationID header field

The JMS specification states that "A client can use the JMSCorrelationID header field to link one message with another. A typical use is to link a response message with its request message". JMS imposes the restriction that the `JMSCorrelationID` header field must start with the prefix "`ID:`".

The `JMSCorrelationID` field hence serves a similar purpose to the `<CorrelationID>` subelement of the IEC 61968-100 `<Header>` element. Therefore, for applications compliant with IEC 61968-100, the JMSCorrelationID header field should be assigned the value of the `<CorrelationID>` subelement (7.5) prefixed by "`ID:`".

9.3.4.4 The JMSReplyTo header field

The JMS specification states that "The JMSReplyTo header field contains a Destination supplied by a client when a message is sent. It is the destination where a reply to the message should be sent".

The `JMSReplyTo` header field thus serves a similar purpose to the `<ReplyAddress>` subelement of the IEC 61968-100 `<Header>` element. Therefore, for applications compliant with IEC 61968-100, the `JMSReplyTo` header field should be assigned the same value as the `<ReplyAddress>` subelement (7.5).

9.3.4.5 The JMSTimestamp header field

The JMS specification states that "The JMSTimestamp header field contains the time a message was handed off to a provider to be sent".

The `JMSTimestamp` header field thus serves a similar purpose to the `<Timestamp>` subelement of the IEC 61968-100 `<Header>` element. However, this JMSTimestamp header field is set automatically by the JMS message producer when the send operation is invoked and is not under direct control of the application. Therefore a receiving system may notice differences between the values of the JMSTimestamp header field and the `<Header><Timestamp>` element.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annex A
(normative)

XSD files

A.1 General

The XSD files listed in this annex comprise part of this document.

A.2 The Message.xsd file

The contents of the `Message.xsd` file (7.3) are listed in Figure A.1.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://iec.ch/TC57/2021/schema/message"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://iec.ch/TC57/2021/schema/message"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="1.0.0">
  <xs:annotation>
    <xs:documentation>
        This is the Message.xsd file which forms an integral part of the
International Standard
        IEC 61968-100:2021, "IEC IMPLEMENTATION PROFILES FOR APPLICATION
INTEGRATION".
        IEC 61968-100:2021 is available from the International
Electrotechnical Commission at http://www.iec.ch
        and contains a full description of the contents of this file.
    </xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="AbstractPayloadType.xsd"/>
  <xs:element name="CanceledEventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="canceled"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CanceledEventResponseMessage" type="SimpleAckReplyType"/>
  <xs:element name="CancelRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
              <xs:element name="Verb" type="RequestVerbType" fixed="cancel"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CancelResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ChangedEventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="changed"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ChangedEventResponseMessage" type="SimpleAckReplyType"/>
  <xs:element name="ChangeRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="change"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
    </xs:choice>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="ChangeResponseMessage">
 <xs:complexType>
  <xs:sequence>
    <xs:element name="Header" type="ResponseHeaderType"/>
    <xs:element name="Reply" type="ReplyType"/>
    <xs:choice minOccurs="0">
      <xs:element name="Payload">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
    </xs:choice>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="CreatedEventMessage">
 <xs:complexType>
  <xs:sequence>
    <xs:element name="Header">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Verb" type="EventVerbType" fixed="created"/>
          <xs:element name="Noun" type="xs:string"/>
          <xs:group ref="EventHeaderGroup"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:choice>
      <xs:element name="Payload">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
    </xs:choice>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="CreatedEventResponseMessage" type="SimpleAckReplyType"/>
<xs:element name="CreateRequestMessage">
 <xs:complexType>
  <xs:sequence>
    <xs:element name="Header">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Verb" type="RequestVerbType" fixed="create"/>
          <xs:element name="Noun" type="xs:string"/>
          <xs:group ref="RequestHeaderGroup"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:choice>
      <xs:element name="Payload">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CreateResponseMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Header" type="ResponseHeaderType"/>
      <xs:element name="Reply" type="ReplyType"/>
      <xs:choice minOccurs="0">
        <xs:element name="Payload">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
              <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="DeletedEventMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Header">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Verb" type="EventVerbType" fixed="deleted"/>
            <xs:element name="Noun" type="xs:string"/>
            <xs:group ref="EventHeaderGroup"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:choice>
        <xs:element name="Payload">
          <xs:complexType>
            <xs:choice>
              <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
              <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
        <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="DeletedEventResponseMessage" type="SimpleAckReplyType"/>
<xs:element name="DeleteRequestMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Header">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Verb" type="RequestVerbType" fixed="delete"/>
            <xs:element name="Noun" type="xs:string"/>
            <xs:group ref="RequestHeaderGroup"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:choice>
        <xs:element name="Payload">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DeleteResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecutedEventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="executed"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet" type="EventOperationSetType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecutedEventResponseMessage" type="SimpleAckReplyType"/>
  <xs:element name="ExecuteRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="execute"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet" type="RequestOperationSetType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecuteResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="ResponseHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Reply" type="ReplyType"/>
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet"
type="ResponseOperationSetType" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="FaultMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="get"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="UpdatedEventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="updated"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="UpdatedEventResponseMessage" type="SimpleAckReplyType"/>
  <xs:element name="UpdateRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="update"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="UpdateResponseMessage">
    <xs:complexType>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="CompressedPayloadType">
    <xs:sequence>
      <xs:element name="CompressionKind">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="zip"/>
            <xs:enumeration value="gzip"/>
            <xs:enumeration value="other"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Base64" type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="EventVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:group ref="EventHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="EnforceMsgSequence" type="xs:boolean" minOccurs="0"/>
      <xs:element name="EnforceTransactionalIntegrity" type="xs:boolean"
minOccurs="0"/>
      <xs:element name="Operation" type="EventOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="EventVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
        <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
        <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="IdentifierType">
    <xs:sequence>
      <xs:element name="mRID" type="UUIDType" minOccurs="0"/>
      <xs:element name="Names" type="Name" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="ObjectType" type="ObjectType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ObjectType">
    <xs:sequence>
      <xs:element name="type" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MessagePropertyType">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
  <xs:sequence>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Value" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="MessageSequenceType">
  <xs:sequence>
    <xs:element name="SequenceNumber" type="xs:unsignedInt" minOccurs="0"/>
    <xs:element name="LastMessage" type="xs:boolean" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Name">
  <xs:sequence>
    <xs:element name="mRID" type="xs:string" minOccurs="0"/>
    <xs:element name="name" type="xs:string" minOccurs="0"/>
    <xs:element name="NameAuthority" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="mRID" type="xs:string" minOccurs="0"/>
          <xs:element name="name" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="NameType" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="mRID" type="xs:string" minOccurs="0"/>
          <xs:element name="name" type="xs:string" minOccurs="0"/>
          <xs:element name="NameTypeAuthority" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="mRID" type="xs:string" minOccurs="0"/>
                <xs:element name="name" type="xs:string" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ObjectUpdateType">
  <xs:sequence>
    <xs:element name="UpdateID" type="UUIDType"/>
    <xs:element name="UpdateAction">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="add"/>
          <xs:enumeration value="modify"/>
          <xs:enumeration value="remove"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element ref="AbstractPayload"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ReplyType">
  <xs:sequence>
    <xs:element name="Status" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="StatusType"/>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Transaction" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="TransactionType"/>
        </xs:complexContent>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="RequestVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:group ref="RequestHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="EnforceMsgSequence" type="xs:boolean" minOccurs="0"/>
      <xs:element name="EnforceTransactionalIntegrity" type="xs:boolean"
minOccurs="0"/>
      <xs:element name="Operation" type="RequestOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="RequestVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
        <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
        <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:group ref="ResponseHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="Operation" type="ResponseOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="SimpleAckReplyType">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <xs:sequence>
      <xs:element name="Header" type="ResponseHeaderType"/>
      <xs:element name="Reply">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Status">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Level" type="xs:string" fixed="OK"/>
                  <xs:element name="Code" type="xs:string" fixed="0.3"/>
                  <xs:element name="Reason" type="xs:string" minOccurs="0"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="StatusType">
    <xs:sequence>
      <xs:element name="Level">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="OK"/>
            <xs:enumeration value="WARNING"/>
            <xs:enumeration value="ERROR"/>
            <xs:enumeration value="UNKNOWN"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Code" type="xs:string"/>
      <xs:element name="Namespace" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="http://.+"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Reason" type="xs:string" minOccurs="0"/>
      <xs:element name="ObjectID" type="IdentifierType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="OperationSet" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="OperationSetID" type="UUIDType"/>
            <xs:element name="Operation" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="OperationID" type="xs:unsignedInt"/>
                  <xs:element name="ObjectID" type="IdentifierType"
minOccurs="0" maxOccurs="unbounded"/>
                  <xs:element name="UpdateID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element name="TransactionID" type="UUIDType"
minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="UpdateID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="TransactionID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="TransactionType">
    <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="TransactionID" type="UUIDType"/>
      <xs:choice minOccurs="0">
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="OperationSet" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence minOccurs="0">
              <xs:element name="OperationSetID" type="UUIDType"/>
              <xs:element name="OperationID" type="xs:unsignedInt"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UserType">
    <xs:sequence>
      <xs:element name="UserID" type="xs:string"/>
      <xs:element name="Organization" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="EventVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="canceled"/>
      <xs:enumeration value="changed"/>
      <xs:enumeration value="created"/>
      <xs:enumeration value="deleted"/>
      <xs:enumeration value="executed"/>
      <xs:enumeration value="updated"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="RequestVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="cancel"/>
      <xs:enumeration value="change"/>
      <xs:enumeration value="create"/>
      <xs:enumeration value="delete"/>
      <xs:enumeration value="get"/>
      <xs:enumeration value="execute"/>
      <xs:enumeration value="update"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ResponseVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="reply"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="UUIDType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Fa-f0-9]{8}-[A-Fa-f0-9]{4}-[A-Fa-f0-9]{4}-[A-Fa-
f0-9]{4}-[A-Fa-f0-9]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:group name="EventHeaderGroup">
    <xs:sequence>
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="MessageSequence" type="MessageSequenceType"
minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="RequestHeaderGroup">
    <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="ReplyAddress" type="xs:string" minOccurs="0"/>
      <xs:element name="AckRequired" type="xs:boolean" minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="ResponseHeaderGroup">
    <xs:sequence>
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="MessageSequence" type="MessageSequenceType"
minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
</xs:schema>
```

Figure A.1 – Contents of Message.xsd file

A.3 The `AbstractPayloadType.xsd` file

The contents of the `AbstractPayloadType.xsd` file are listed in Figure A.2. This file may be included by application-specific XSD profiles (8.2.2, Annex C).

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://iec.ch/TC57/2021/schema/message"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified" version="1.0.0">
  <xs:annotation>
    <xs:documentation>
        This is the AbstractPayloadType.xsd file which forms an integral
        part of the International Standard IEC 61968-100:2021,
        "IEC IMPLEMENTATION PROFILES FOR APPLICATION INTEGRATION".
        IEC 61968-100:2021 is available from the International
        Electrotechnical Commission at http://www.iec.ch
        and contains a full description of the contents of this file.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType name="AbstractPayloadType"/>
  <xs:element name="AbstractPayload" type="AbstractPayloadType"
    abstract="true"/>
</xs:schema>
```

Figure A.2 – Contents of AbstractPayloadType.xsd file

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

A.4 The LegacyObjectIdentificationRevisionSets.xsd File

The contents of the `LegacyObjectIdentificationRevisionSets.xsd` file (8.7) are listed in Figure A.3.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:part100="http://iec.ch/TC57/2021/schema/message"
  xmlns:a="http://langdale.com.au/2005/Message#"
xmlns:sawsdl="http://www.w3.org/ns/sawsdl"
targetNamespace="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#"
elementFormDefault="qualified"
  attributeFormDefault="unqualified"
xmlns="http://langdale.com.au/2005/Message#"
xmlns:m="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#">
  <xs:annotation>
    <xs:documentation>
        This is the LegacyObjectIdentificationRevisionSets.xsd file which
forms an integral
        part of the International Standard IEC 61968-100:2021,
        "IEC IMPLEMENTATION PROFILES FOR APPLICATION INTEGRATION".
        IEC 61968-100:2021 is available from the International
        Electrotechnical Commission at http://www.iec.ch
        and contains a full description of the contents of this file.
    </xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
schemaLocation="AbstractPayloadType.xsd" xmlns="" />
  <xs:element name="ObjectIdentificationRevisionSets"
type="m:ObjectIdentificationRevisionSets"
substitutionGroup="part100:AbstractPayload"
    xmlns="" />
  <xs:complexType name="ObjectIdentificationRevisionSets" xmlns="">
    <xs:complexContent>
      <xs:extension base="part100:AbstractPayloadType">
        <xs:sequence>
          <xs:element name="ObjectIdentificationRevisionSet"
type="m:ObjectIdentificationRevisionSet" minOccurs="1"
            maxOccurs="unbounded" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType xmlns="" name="Identifier"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject">
    <xs:annotation>
      <xs:documentation>This is a root class to provide common identification
for all classes needing identification and
        naming attributes.</xs:documentation>
      <xs:documentation />
    </xs:annotation>
    <xs:sequence>
      <xs:element name="mRID" minOccurs="0" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.mRID">
        <xs:annotation>
          <xs:documentation>Master resource identifier issued by a model
authority. The mRID is unique within an exchange
            context. Global uniqueness is easily achieved by using a UUID, as
specified in RFC 4122, for the mRID. The use of
            UUID is strongly recommended.</xs:documentation>
          <xs:documentation>For CIMXML data files in RDF syntax conforming to
IEC 61970-552, the mRID is mapped to rdf:ID or
            rdf:about attributes that identify CIM object
elements.</xs:documentation>
        </xs:annotation>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      </xs:element>
      <xs:element name="Names" minOccurs="0" maxOccurs="unbounded"
type="m:Name"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.Names">
        <xs:annotation>
          <xs:documentation>All names of this identified
object.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="Name"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#Name">
    <xs:annotation>
      <xs:documentation>The Name class provides the means to define any number
of human readable names for an object. A
        name is &lt;b&gt;not&lt;/b&gt; to be used for defining inter-object
relationships. For inter-object relationships
        instead use the object identification 'mRID'.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="name" minOccurs="1" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#Name.name">
        <xs:annotation>
          <xs:documentation>Any free text that name the
object.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="NameType" minOccurs="1" maxOccurs="1"
type="m:NameType"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#Name.NameTyp
e">
        <xs:annotation>
          <xs:documentation>Type of this name.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="NameType"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameType">
    <xs:annotation>
      <xs:documentation>Type of name. Possible values for attribute 'name' are
implementation dependent but standard
        profiles may specify types. An enterprise may have multiple IT systems
each having its own local name for the same
        object, e.g. a planning system may have different names from an EMS.
An object may also have different names within
        the same IT system, e.g. localName as defined in CIM version 14. The
definition from CIM14 is:</xs:documentation>
      <xs:documentation>The localName is a human readable name of the object.
It is a free text name local to a node in a
        naming hierarchy similar to a file directory structure. A power system
related naming hierarchy may be: Substation,
        VoltageLevel, Equipment etc. Children of the same parent in such a
hierarchy have names that typically are unique
        among them.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="description" minOccurs="0" maxOccurs="1"
type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameType.des
cription">
        <xs:annotation>
          <xs:documentation>Description of the name type.</xs:documentation>
        </xs:annotation>
      </xs:element>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="name" minOccurs="1" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameType.nam
e">
        <xs:annotation>
          <xs:documentation>Name of the name type.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="NameTypeAuthority" minOccurs="0" maxOccurs="1"
type="m:NameTypeAuthority"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameType.Nam
eTypeAuthority">
        <xs:annotation>
          <xs:documentation>Authority responsible for managing names of this
type.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="NameTypeAuthority"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameTypeAuth
ority">
    <xs:annotation>
      <xs:documentation>Authority responsible for creation and management of
names of a given name type; typically an
        organization or an enterprise system.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="description" minOccurs="0" maxOccurs="1"
type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameTypeAuth
ority.description">
        <xs:annotation>
          <xs:documentation>Description of the name type
authority.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="name" minOccurs="1" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameTypeAuth
ority.name">
        <xs:annotation />
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="Object"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject">
    <xs:annotation>
      <xs:documentation>This is a root class to provide common identification
for all classes needing identification and
        naming attributes.</xs:documentation>
      <xs:documentation />
    </xs:annotation>
    <xs:sequence>
      <xs:element name="mRID" minOccurs="0" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.mRID">
        <xs:annotation>
          <xs:documentation>Master resource identifier issued by a model
authority. The mRID is unique within an exchange
            context. Global uniqueness is easily achieved by using a UUID, as
specified in RFC 4122, for the mRID. The use of
            UUID is strongly recommended.</xs:documentation>
          <xs:documentation>For CIMXML data files in RDF syntax conforming to
IEC 61970-552, the mRID is mapped to rdf:ID or
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            rdf:about attributes that identify CIM object
elements.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Names" minOccurs="0" maxOccurs="1" type="m:Name"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.Names">
        <xs:annotation>
          <xs:documentation>All names of this identified
object.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ObjectType" minOccurs="0" maxOccurs="1"
type="m:ObjectType"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.ObjectType">
        <xs:annotation>
          <xs:documentation>The type of the IdentifiedObject, specified as the
name of a class that inherits from
            IdentifiedObject. The type is specified by setting the
ObjectType.Name.name to the name of the inherited class
            (e.g., Customer, Switch, Meter, etc.).</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="ObjectIdentificationRevision"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevision">
    <xs:annotation>
      <xs:documentation>Each instance of an ObjectIdentificationRevision
identifies:</xs:documentation>
      <xs:documentation>1. The IdentifiedObject for which one or more
identifiers are being added, removed, or modified.
        The IdentifiedObject is specified by the association to
IdentifiedObject with the target role name of 'Object'.
      </xs:documentation>
      <xs:documentation>2. One or more identifiers that are being added,
removed or modified. These identifiers may be an
        mRID or an instance of Name. The identifiers are specified by the
association to IdentifiedObject with the target
        role name of 'Identifier'.</xs:documentation>
      <xs:documentation>3. The type of revision (add, remove, or modify). The
type of revision is specified by the
        ObjectIdentificationRevision.ObjectIdentificationRevisionKind, with
enumerated choices of 'add', 'remove', or 'modify'.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="revisionKind" minOccurs="1" maxOccurs="1"
type="m:RevisionKind"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevision.revisionKind">
        <xs:annotation />
      </xs:element>
      <xs:element name="sequenceNumber" minOccurs="0" maxOccurs="1"
type="xs:integer"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevision.sequenceNumber">
        <xs:annotation />
      </xs:element>
      <xs:element name="Identifier" minOccurs="1" maxOccurs="1"
type="m:Identifier"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevision.Identifier">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        <xs:annotation>
          <xs:documentation>The identifier that is being added to, removed
from, or modified for an IdentifiedObject. The
            identifier may be an mRID or an instance of
Name.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="ObjectIdentificationRevisionSet"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevisionSet">
    <xs:annotation />
    <xs:sequence>
      <xs:element name="mRID" minOccurs="0" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.mRID">
        <xs:annotation>
          <xs:documentation>Master resource identifier issued by a model
authority. The mRID is unique within an exchange
            context. Global uniqueness is easily achieved by using a UUID, as
specified in RFC 4122, for the mRID. The use of
            UUID is strongly recommended.</xs:documentation>
          <xs:documentation>For CIMXML data files in RDF syntax conforming to
IEC 61970-552, the mRID is mapped to rdf:ID or
            rdf:about attributes that identify CIM object
elements.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="description" minOccurs="0" maxOccurs="1"
type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.description">
        <xs:annotation>
          <xs:documentation>The description is a free human readable text
describing or naming the object. It may be non
            unique and may not correlate to a naming
hierarchy.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="effectiveDateTime" minOccurs="0" maxOccurs="1"
type="xs:dateTime"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevisionSet.effectiveDateTime">
        <xs:annotation />
      </xs:element>
      <xs:element name="Names" minOccurs="0" maxOccurs="unbounded"
type="m:Name"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.Names">
        <xs:annotation>
          <xs:documentation>All names of this identified
object.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Object" minOccurs="0" maxOccurs="1" type="m:Object"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevisionSet.Object">
        <xs:annotation>
          <xs:documentation>The IdentifiedObject for which a revision to its
set of identifiers is being made. The Object is
            specified either by the mRID or a Name that exists prior to the
revision.</xs:documentation>
        </xs:annotation>
      </xs:element>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="ObjectIdentificationRevision" minOccurs="0"
maxOccurs="unbounded" type="m:ObjectIdentificationRevision"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevisionSet.ObjectIdentificationRevision">
        <xs:annotation />
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="ObjectType"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectType">
    <xs:annotation>
      <xs:documentation>Identifies the specific type of the object that is
generically identified as an IdentifiedObject.
        This type will always be the name of a CIM class that inherits from
IdentifiedObject and is specified by setting the
        ObjectType.type to the name of the class (e.g., Customer, Switch,
Meter, etc.).</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="type" minOccurs="1" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectType.t
ype">
        <xs:annotation />
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType xmlns="" name="RevisionKind"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#RevisionKind
">
    <xs:annotation />
    <xs:restriction base="xs:string">
      <xs:enumeration value="add">
        <xs:annotation />
      </xs:enumeration>
      <xs:enumeration value="remove">
        <xs:annotation />
      </xs:enumeration>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Figure A.3 – Contents of LegacyObjectIdentificationRevisionSets.xsd file

A.5 The `RDF.xsd` file

The contents of the `RDF.xsd` file are listed in Figure A.4. This file is suitable for the transmission of RDF data (8.2.4). It may be used also for the transmission of XML data that does not correspond to any XML schema definition.

The data corresponding to this profile may therefore appear in the `AbstractPayload` section of a `<Payload>` element (7.7.5).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:a="http://langdale.com.au/2005/Message#"
xmlns:sawsdl="http://www.w3.org/ns/sawsdl"
xmlns="http://langdale.com.au/2005/Message#"
xmlns:m="http://iec.ch/TC57/2011/RDFData#"
xmlns:part100="http://iec.ch/TC57/2021/schema/message"
xmlns:ns1="http://iec.ch/TC57/2021/schema/message"
targetNamespace="http://iec.ch/TC57/2011/RDFData#"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation>
        This is the RDF.xsd file which forms an integral part of the
International Standard
        IEC 61968-100:2021,
        "IEC IMPLEMENTATION PROFILES FOR APPLICATION INTEGRATION".
        IEC 61968-100:2021 is available from the International
Electrotechnical Commission at http://www.iec.ch
        and contains a full description of the contents of this file.
    </xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
schemaLocation="../XSDs/AbstractPayloadType.xsd"/>
  <xs:element name="RDFData" type="m:RDFData"
substitutionGroup="part100:AbstractPayload"/>
  <xs:complexType name="RDFData">
    <xs:complexContent>
      <xs:extension base="part100:AbstractPayloadType">
        <xs:sequence>
          <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

Figure A.4 – Contents of the RDF.xsd file

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annex B
(normative)

Template files

B.1 General

The template files listed in this annex comprise part of this document. These files must be manually edited in order to generate the corresponding WSDL files (9.2.5).

B.2 The GenericMessageWsdlTemplate.txt file

The contents of the `GenericMessageWsdlTemplate.txt` file (9.2.5.2) are listed in Figure B.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  This is the GenericMessageXsdTemplate.txt file which forms an integral part
  of the International Standard  IEC 61968-100:2021,
 "IEC IMPLEMENTATION PROFILES FOR APPLICATION INTEGRATION".
  IEC 61968-100:2021 is available from the International Electrotechnical
  Commission at http://www.iec.ch
  and contains a full description of the contents of this file.

  This template file may be used to generate a generic WSDL file by globally
  substitutingthe following variable names:

  Variable Name           Example Value
  ====================================
  {PATHPREFIX}            ../XSDs
  {SOAPADDRESS}           http://example.com
-->
<wsdl:definitions
    xmlns:tns="http://iec.ch/TC57/2021/wsdl/IEC61968-100_Ed2_Generic"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:msg="http://iec.ch/TC57/2021/schema/message"
    name="IEC61968-100_Ed2_Generic"
    targetNamespace="http://iec.ch/TC57/2021/wsdl/IEC61968-100_Ed2_Generic">
  <wsdl:types>
    <xs:schema targetNamespace="http://iec.ch/TC57/2021/wsdl/IEC61968-
100_Ed2_Generic" elementFormDefault="qualified">
      <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
schemaLocation="{PATHPREFIX}/Message.xsd"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="CancelRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:CancelRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="CancelResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:CancelResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ChangeRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:ChangeRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="ChangeResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:ChangeResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="CreateRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:CreateRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="CreateResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:CreateResponseMessage"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
  </wsdl:message>
  <wsdl:message name="DeleteRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:DeleteRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="DeleteResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:DeleteResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecuteRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:ExecuteRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecuteResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:ExecuteResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="GetRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:GetRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="GetResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:GetResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="UpdateRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:UpdateRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="UpdateResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:UpdateResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="CanceledEventMessage">
    <wsdl:part name="EventMessage" element="msg:CanceledEventMessage"/>
  </wsdl:message>
  <wsdl:message name="CanceledEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:CanceledEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ChangedEventMessage">
    <wsdl:part name="EventMessage" element="msg:ChangedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="ChangedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:ChangedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="CreatedEventMessage">
    <wsdl:part name="EventMessage" element="msg:CreatedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="CreatedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:CreatedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="DeletedEventMessage">
    <wsdl:part name="EventMessage" element="msg:DeletedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="DeletedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:DeletedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecutedEventMessage">
    <wsdl:part name="EventMessage" element="msg:ExecutedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecutedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:ExecutedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="FaultMessage">
    <wsdl:part name="FaultMessage" element="msg:FaultMessage"/>
  </wsdl:message>
  <wsdl:message name="UpdatedEventMessage">
    <wsdl:part name="EventMessage" element="msg:UpdatedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="UpdatedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:UpdatedEventResponseMessage"/>
  </wsdl:message>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
  <wsdl:portType name="IEC61968-100_Ed2_Generic_Port">
    <wsdl:operation name="Cancel">
      <wsdl:input name="CancelInput" message="tns:CancelRequestMessage"/>
      <wsdl:output name="CancelOutput" message="tns:CancelResponseMessage"/>
      <wsdl:fault name="CancelFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Change">
      <wsdl:input name="ChangeInput" message="tns:ChangeRequestMessage"/>
      <wsdl:output name="ChangeOutput" message="tns:ChangeResponseMessage"/>
      <wsdl:fault name="ChangeFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Create">
      <wsdl:input name="CreateInput" message="tns:CreateRequestMessage"/>
      <wsdl:output name="CreateOutput" message="tns:CreateResponseMessage"/>
      <wsdl:fault name="CreateFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Delete">
      <wsdl:input name="DeleteInput" message="tns:DeleteRequestMessage"/>
      <wsdl:output name="DeleteOutput" message="tns:DeleteResponseMessage"/>
      <wsdl:fault name="DeleteFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Execute">
      <wsdl:input name="ExecuteInput" message="tns:ExecuteRequestMessage"/>
      <wsdl:output name="ExecuteOutput" message="tns:ExecuteResponseMessage"/>
      <wsdl:fault name="ExecuteFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Get">
      <wsdl:input name="GetInput" message="tns:GetRequestMessage"/>
      <wsdl:output name="GetOutput" message="tns:GetResponseMessage"/>
      <wsdl:fault name="GetFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Update">
      <wsdl:input name="UpdateInput" message="tns:UpdateRequestMessage"/>
      <wsdl:output name="UpdateOutput" message="tns:UpdateResponseMessage"/>
      <wsdl:fault name="UpdateFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Canceled">
      <wsdl:input name="CanceledInput" message="tns:CanceledEventMessage"/>
      <wsdl:output name="CanceledOutput"
message="tns:CanceledEventResponseMessage"/>
      <wsdl:fault name="CanceledFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Changed">
      <wsdl:input name="ChangedInput" message="tns:ChangedEventMessage"/>
      <wsdl:output name="ChangedOutput"
message="tns:ChangedEventResponseMessage"/>
      <wsdl:fault name="ChangedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Created">
      <wsdl:input name="CreatedInput" message="tns:CreatedEventMessage"/>
      <wsdl:output name="CreatedOutput"
message="tns:CreatedEventResponseMessage"/>
      <wsdl:fault name="CreatedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Deleted">
      <wsdl:input name="DeletedInput" message="tns:DeletedEventMessage"/>
      <wsdl:output name="DeletedOutput"
message="tns:DeletedEventResponseMessage"/>
      <wsdl:fault name="DeletedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Executed">
      <wsdl:input name="ExecutedInput" message="tns:ExecutedEventMessage"/>
      <wsdl:output name="ExecutedOutput"
message="tns:ExecutedEventResponseMessage"/>
      <wsdl:fault name="ExecutedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Updated">
      <wsdl:input name="UpdatedInput" message="tns:UpdatedEventMessage"/>
      <wsdl:output name="UpdatedOutput"
message="tns:UpdatedEventResponseMessage"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <wsdl:fault name="UpdatedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="IEC61968-100_Ed2_Generic_Binding" type="tns:IEC61968-
100_Ed2_Generic_Port">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Cancel">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Cancel" style="document"/>
      <wsdl:input name="CancelInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CancelOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CancelFault">
        <soap:fault name="CancelFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Change">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Change" style="document"/>
      <wsdl:input name="ChangeInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ChangeOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ChangeFault">
        <soap:fault name="ChangeFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Create">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Create" style="document"/>
      <wsdl:input name="CreateInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CreateOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CreateFault">
        <soap:fault name="CreateFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Delete">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Delete" style="document"/>
      <wsdl:input name="DeleteInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="DeleteOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="DeleteFault">
        <soap:fault name="DeleteFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Execute">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Execute" style="document"/>
      <wsdl:input name="ExecuteInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ExecuteOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ExecuteFault">
        <soap:fault name="ExecuteFault" use="literal"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Get">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Get" style="document"/>
      <wsdl:input name="GetInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="GetOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="GetFault">
        <soap:fault name="GetFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Update">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Update" style="document"/>
      <wsdl:input name="UpdateInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="UpdateOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="UpdateFault">
        <soap:fault name="UpdateFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Canceled">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Canceled" style="document"/>
      <wsdl:input name="CanceledInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CanceledOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CanceledFault">
        <soap:fault name="CanceledFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Changed">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Changed" style="document"/>
      <wsdl:input name="ChangedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ChangedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ChangedFault">
        <soap:fault name="ChangedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Created">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Created" style="document"/>
      <wsdl:input name="CreatedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CreatedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CreatedFault">
        <soap:fault name="CreatedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Deleted">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Deleted" style="document"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <wsdl:input name="DeletedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="DeletedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="DeletedFault">
        <soap:fault name="DeletedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Executed">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Executed" style="document"/>
      <wsdl:input name="ExecutedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ExecutedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ExecutedFault">
        <soap:fault name="ExecutedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Updated">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Updated" style="document"/>
      <wsdl:input name="UpdatedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="UpdatedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="UpdatedFault">
        <soap:fault name="UpdatedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="IEC61968-100_Ed2_Generic">
    <wsdl:port name="IEC61968-100_Ed2_Generic_Port" binding="tns:IEC61968-
100_Ed2_Generic_Binding">
      <!-- The SOAP address below must be changed for a particular
implementation. -->
      <soap:address location="{SOAPADDRESS}"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Figure B.1 – Contents of GenericMessageXsdTemplate.txt file

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

B.3 The `StrongMessageXsdTemplate.txt` file

The contents of the `StrongMessageXsdTemplate.txt` file (9.2.5.3) are listed in Figure B.2.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  This is the StrongMessageXsdTemplate.txt file which forms an integral part
of the International Standard
  IEC 61968-100:2021, "IEC IMPLEMENTATION PROFILES FOR APPLICATION
INTEGRATION".
  IEC 61968-100:2021 is available from the International Electrotechnical
Commission at http://www.iec.ch
  and contains a full description of the contents of this file.

  This template file may be used to generate a strongly-typed message XSD file
by globally substituting
  the following variable names:

  Variable Name          Example Value
  ====================================
  {GETPROFILENAMESPACE}  http://iec.ch/TC57/2011/GetMeterReadings#
  {GETPROFILENAME}       GetMeterReadings
  {PROFILENAMESPACE}     http://iec.ch/TC57/2011/MeterReadings#
  {PROFILENAME}          MeterReadings
  {PROFILEPATHPREFIX}    ../Part9XSDs
  {PART100PATHPREFIX}    ../XSDs
-->
<xs:schema xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:obj1="{PROFILENAMESPACE}"
    xmlns:obj2="{GETPROFILENAMESPACE}"
    targetNamespace="http://iec.ch/TC57/2021/schema/message"
    elementFormDefault="qualified" attributeFormDefault="unqualified"
version="1.0.0">
  <xs:import namespace="{PROFILENAMESPACE}"
schemaLocation="{PROFILEPATHPREFIX}/{PROFILENAME}.xsd"/>
  <xs:import namespace="{GETPROFILENAMESPACE}"
schemaLocation="{PROFILEPATHPREFIX}/{GETPROFILENAME}.xsd"/>
  <xs:include schemaLocation="{PART100PATHPREFIX}/AbstractPayloadType.xsd"/>
  <xs:element name="Canceled{PROFILENAME}EventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="canceled"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Canceled{PROFILENAME}EventResponseMessage"
type="SimpleAckReplyType"/>
  <xs:element name="Cancel{PROFILENAME}RequestMessage">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="cancel"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Cancel{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Changed{PROFILENAME}EventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="changed"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Changed{PROFILENAME}EventResponseMessage"
type="SimpleAckReplyType"/>
  <xs:element name="Change{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="change"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Change{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Created{PROFILENAME}EventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="created"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Created{PROFILENAME}EventResponseMessage"
type="SimpleAckReplyType"/>
  <xs:element name="Create{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="create"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Create{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Deleted{PROFILENAME}EventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="deleted"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Deleted{PROFILENAME}EventResponseMessage"
type="SimpleAckReplyType"/>
  <xs:element name="Delete{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
              <xs:element name="Verb" type="RequestVerbType" fixed="delete"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Delete{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecutedEventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="executed"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet" type="EventOperationSetType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecutedEventResponseMessage" type="SimpleAckReplyType"/>
  <xs:element name="ExecuteRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="execute"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet" type="RequestOperationSetType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecuteResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="ResponseHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Reply" type="ReplyType"/>
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet"
type="ResponseOperationSetType" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="FaultMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Get{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="get"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="obj2:{GETPROFILENAME}"
maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Get{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Updated{PROFILENAME}EventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="updated"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Updated{PROFILENAME}EventResponseMessage"
type="SimpleAckReplyType"/>
  <xs:element name="Update{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="update"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Update{PROFILENAME}ResponseMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Header" type="ResponseHeaderType"/>
      <xs:element name="Reply" type="ReplyType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="CompressedPayloadType">
  <xs:sequence>
    <xs:element name="CompressionKind">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="zip"/>
          <xs:enumeration value="gzip"/>
          <xs:enumeration value="other"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Base64" type="xs:base64Binary"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EventHeaderType">
  <xs:sequence>
    <xs:element name="Verb" type="EventVerbType"/>
    <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
    <xs:group ref="EventHeaderGroup"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EventOperationSetType">
  <xs:sequence>
    <xs:element name="OperationSetID" type="UUIDType"/>
    <xs:element name="EnforceMsgSequence" type="xs:boolean" minOccurs="0"/>
    <xs:element name="EnforceTransactionalIntegrity" type="xs:boolean"
minOccurs="0"/>
    <xs:element name="Operation" type="EventOperationType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EventOperationType">
  <xs:sequence>
    <xs:element name="OperationID" type="xs:unsignedInt"/>
    <xs:element name="Verb" type="EventVerbType"/>
    <xs:element name="Noun" type="xs:string"/>
    <xs:element name="Comment" type="xs:string" minOccurs="0"/>
    <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:choice>
      <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
      <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
      <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
      <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="IdentifierType">
  <xs:sequence>
    <xs:element name="mRID" type="UUIDType" minOccurs="0"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="Names" type="Name" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="ObjectType" type="ObjectType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ObjectType">
    <xs:sequence>
      <xs:element name="type" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MessagePropertyType">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Value" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MessageSequenceType">
    <xs:sequence>
      <xs:element name="SequenceNumber" type="xs:unsignedInt" minOccurs="0"/>
      <xs:element name="LastMessage" type="xs:boolean" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Name">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="NameType" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="NameTypeAuthority" minOccurs="0">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="name" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ObjectUpdateType">
    <xs:sequence>
      <xs:element name="UpdateID" type="UUIDType"/>
      <xs:element name="UpdateAction">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="add"/>
            <xs:enumeration value="modify"/>
            <xs:enumeration value="remove"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element ref="AbstractPayload"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ReplyType">
    <xs:sequence>
      <xs:element name="Status" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="StatusType"/>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="Transaction" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="TransactionType"/>
          </xs:complexContent>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="RequestVerbType"/>
      <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
      <xs:group ref="RequestHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="EnforceMsgSequence" type="xs:boolean" minOccurs="0"/>
      <xs:element name="EnforceTransactionalIntegrity" type="xs:boolean"
minOccurs="0"/>
      <xs:element name="Operation" type="RequestOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="RequestVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
        <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
        <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
      <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
      <xs:group ref="ResponseHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="Operation" type="ResponseOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="SimpleAckReplyType">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <xs:sequence>
      <xs:element name="Header" type="ResponseHeaderType"/>
      <xs:element name="Reply">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Status">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Level" type="xs:string" fixed="OK"/>
                  <xs:element name="Code" type="xs:string" fixed="0.3"/>
                  <xs:element name="Reason" type="xs:string" minOccurs="0"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="StatusType">
    <xs:sequence>
      <xs:element name="Level">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="OK"/>
            <xs:enumeration value="WARNING"/>
            <xs:enumeration value="ERROR"/>
            <xs:enumeration value="UNKNOWN"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Code" type="xs:string"/>
      <xs:element name="Namespace" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="http://.+"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Reason" type="xs:string" minOccurs="0"/>
      <xs:element name="ObjectID" type="IdentifierType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="OperationSet" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="OperationSetID" type="UUIDType"/>
            <xs:element name="Operation" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="OperationID" type="xs:unsignedInt"/>
                  <xs:element name="ObjectID" type="IdentifierType"
minOccurs="0" maxOccurs="unbounded"/>
                  <xs:element name="UpdateID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element name="TransactionID" type="UUIDType"
minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="UpdateID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="TransactionID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="TransactionType">
    <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="TransactionID" type="UUIDType"/>
      <xs:choice minOccurs="0">
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="OperationSet" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence minOccurs="0">
              <xs:element name="OperationSetID" type="UUIDType"/>
              <xs:element name="OperationID" type="xs:unsignedInt"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UserType">
    <xs:sequence>
      <xs:element name="UserID" type="xs:string"/>
      <xs:element name="Organization" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="EventVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="canceled"/>
      <xs:enumeration value="changed"/>
      <xs:enumeration value="created"/>
      <xs:enumeration value="deleted"/>
      <xs:enumeration value="executed"/>
      <xs:enumeration value="updated"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="RequestVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="cancel"/>
      <xs:enumeration value="change"/>
      <xs:enumeration value="create"/>
      <xs:enumeration value="delete"/>
      <xs:enumeration value="get"/>
      <xs:enumeration value="execute"/>
      <xs:enumeration value="update"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ResponseVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="reply"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="UUIDType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Fa-f0-9]{8}-[A-Fa-f0-9]{4}-[A-Fa-f0-9]{4}-[A-Fa-
f0-9]{4}-[A-Fa-f0-9]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:group name="EventHeaderGroup">
    <xs:sequence>
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="MessageSequence" type="MessageSequenceType"
minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="RequestHeaderGroup">
    <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="ReplyAddress" type="xs:string" minOccurs="0"/>
      <xs:element name="AckRequired" type="xs:boolean" minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="ResponseHeaderGroup">
    <xs:sequence>
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="MessageSequence" type="MessageSequenceType"
minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
</xs:schema>
```

Figure B.2 – Contents of StrongMessageXsdTemplate.txt file

B.4 The `StrongMessageWsdlTemplate.txt` file

The contents of the `StrongMessageWsdlTemplate.txt` file (9.2.5.3) are listed in Figure B.3.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  This is the StrongMessageWsdlTemplate.txt file which forms an integral part
of the International Standard IEC 61968-100:2021,
"IEC IMPLEMENTATION PROFILES FOR APPLICATION INTEGRATION".
  IEC 61968-100:2021 is available from the International Electrotechnical
Commission at http://www.iec.ch
  and contains a full description of the contents of this file.

  This template file may be used to generate a strongly-typed WSDL file by
globally substituting
  the following variable names:

  Variable Name          Example Value
  ===================================
  {GETPROFILENAME}       GetMeterReadings
  {PROFILENAME}          MeterReadings
  {PATHPREFIX}           ../Templates
  {SOAPADDRESS}          http://example.com
-->
<wsdl:definitions xmlns:tns="http://iec.ch/TC57/2021/wsdl/IEC61968-
100_Ed2_Strong/{PROFILENAME}"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:msg="http://iec.ch/TC57/2021/schema/message"
    name="IEC61968-100_Ed2_Strong"
    targetNamespace="http://iec.ch/TC57/2021/wsdl/IEC61968-
100_Ed2_Strong/{PROFILENAME}">
  <wsdl:types>
    <xs:schema targetNamespace="http://iec.ch/TC57/2021/wsdl/{PROFILENAME}"
elementFormDefault="qualified">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
schemaLocation="{PATHPREFIX}/{PROFILENAME}Message.xsd"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="Cancel{PROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:Cancel{PROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="Cancel{PROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Cancel{PROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Change{PROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:Change{PROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="Change{PROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Change{PROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Create{PROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:Create{PROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="Create{PROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Create{PROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Delete{PROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:Delete{PROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="Delete{PROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Delete{PROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecuteRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:ExecuteRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecuteResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:ExecuteResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="{GETPROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:{GETPROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="{GETPROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:{GETPROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Update{PROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:Update{PROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="Update{PROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Update{PROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Canceled{PROFILENAME}EventMessage">
    <wsdl:part name="{PROFILENAME}EventMessage"
element="msg:Canceled{PROFILENAME}EventMessage"/>
  </wsdl:message>
  <wsdl:message name="Canceled{PROFILENAME}EventResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Canceled{PROFILENAME}EventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Changed{PROFILENAME}EventMessage">
    <wsdl:part name="{PROFILENAME}EventMessage"
element="msg:Changed{PROFILENAME}EventMessage"/>
  </wsdl:message>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
  <wsdl:message name="Changed{PROFILENAME}EventResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Changed{PROFILENAME}EventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Created{PROFILENAME}EventMessage">
    <wsdl:part name="{PROFILENAME}EventMessage"
element="msg:Created{PROFILENAME}EventMessage"/>
  </wsdl:message>
  <wsdl:message name="Created{PROFILENAME}EventResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Created{PROFILENAME}EventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Deleted{PROFILENAME}EventMessage">
    <wsdl:part name="{PROFILENAME}EventMessage"
element="msg:Deleted{PROFILENAME}EventMessage"/>
  </wsdl:message>
  <wsdl:message name="Deleted{PROFILENAME}EventResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Deleted{PROFILENAME}EventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecutedEventMessage">
    <wsdl:part name="EventMessage" element="msg:ExecutedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecutedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:ExecutedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Updated{PROFILENAME}EventMessage">
    <wsdl:part name="{PROFILENAME}EventMessage"
element="msg:Updated{PROFILENAME}EventMessage"/>
  </wsdl:message>
  <wsdl:message name="Updated{PROFILENAME}EventResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Updated{PROFILENAME}EventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="FaultMessage">
    <wsdl:part name="FaultMessage" element="msg:FaultMessage"/>
  </wsdl:message>
  <wsdl:portType name="IEC61968-100_Ed2_Strong_Port">
    <wsdl:operation name="Cancel">
      <wsdl:input name="CancelInput"
message="tns:Cancel{PROFILENAME}RequestMessage"/>
      <wsdl:output name="CancelOutput"
message="tns:Cancel{PROFILENAME}ResponseMessage"/>
      <wsdl:fault name="CancelFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Change">
      <wsdl:input name="ChangeInput"
message="tns:Change{PROFILENAME}RequestMessage"/>
      <wsdl:output name="ChangeOutput"
message="tns:Change{PROFILENAME}ResponseMessage"/>
      <wsdl:fault name="ChangeFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Create">
      <wsdl:input name="CreateInput"
message="tns:Create{PROFILENAME}RequestMessage"/>
      <wsdl:output name="CreateOutput"
message="tns:Create{PROFILENAME}ResponseMessage"/>
      <wsdl:fault name="CreateFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Delete">
      <wsdl:input name="DeleteInput"
message="tns:Delete{PROFILENAME}RequestMessage"/>
      <wsdl:output name="DeleteOutput"
message="tns:Delete{PROFILENAME}ResponseMessage"/>
      <wsdl:fault name="DeleteFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Execute">
      <wsdl:input name="ExecuteInput" message="tns:ExecuteRequestMessage"/>
      <wsdl:output name="ExecuteOutput" message="tns:ExecuteResponseMessage"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <wsdl:fault name="ExecuteFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Get">
      <wsdl:input name="GetInput"
message="tns:{GETPROFILENAME}RequestMessage"/>
      <wsdl:output name="GetOutput"
message="tns:{GETPROFILENAME}ResponseMessage"/>
      <wsdl:fault name="GetFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Update">
      <wsdl:input name="UpdateInput"
message="tns:Update{PROFILENAME}RequestMessage"/>
      <wsdl:output name="UpdateOutput"
message="tns:Update{PROFILENAME}ResponseMessage"/>
      <wsdl:fault name="UpdateFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Created">
      <wsdl:input name="CreatedInput"
message="tns:Created{PROFILENAME}EventMessage"/>
      <wsdl:output name="CreatedOutput"
message="tns:Created{PROFILENAME}EventResponseMessage"/>
      <wsdl:fault name="CreatedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Changed">
      <wsdl:input name="ChangedInput"
message="tns:Changed{PROFILENAME}EventMessage"/>
      <wsdl:output name="ChangedOutput"
message="tns:Changed{PROFILENAME}EventResponseMessage"/>
      <wsdl:fault name="ChangedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Canceled">
      <wsdl:input name="CanceledInput"
message="tns:Canceled{PROFILENAME}EventMessage"/>
      <wsdl:output name="CanceledOutput"
message="tns:Canceled{PROFILENAME}EventResponseMessage"/>
      <wsdl:fault name="CanceledFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Deleted">
      <wsdl:input name="DeletedInput"
message="tns:Deleted{PROFILENAME}EventMessage"/>
      <wsdl:output name="DeletedOutput"
message="tns:Deleted{PROFILENAME}EventResponseMessage"/>
      <wsdl:fault name="DeletedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Executed">
      <wsdl:input name="ExecutedInput" message="tns:ExecutedEventMessage"/>
      <wsdl:output name="ExecutedOutput"
message="tns:ExecutedEventResponseMessage"/>
      <wsdl:fault name="ExecutedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Updated">
      <wsdl:input name="UpdatedInput"
message="tns:Updated{PROFILENAME}EventMessage"/>
      <wsdl:output name="UpdatedOutput"
message="tns:Updated{PROFILENAME}EventResponseMessage"/>
      <wsdl:fault name="UpdatedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="IEC61968-100_Ed2_Strong_Binding" type="tns:IEC61968-
100_Ed2_Strong_Port">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Cancel">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Cancel" style="document"/>
      <wsdl:input name="CancelInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CancelOutput">
        <soap:body use="literal"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      </wsdl:output>
      <wsdl:fault name="CancelFault">
        <soap:fault name="CancelFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Change">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Change" style="document"/>
      <wsdl:input name="ChangeInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ChangeOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ChangeFault">
        <soap:fault name="ChangeFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Create">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Create" style="document"/>
      <wsdl:input name="CreateInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CreateOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CreateFault">
        <soap:fault name="CreateFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Delete">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Delete" style="document"/>
      <wsdl:input name="DeleteInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="DeleteOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="DeleteFault">
        <soap:faults name="DeleteFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Execute">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Execute" style="document"/>
      <wsdl:input name="ExecuteInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ExecuteOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ExecuteFault">
        <soap:fault name="ExecuteFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Get">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Get" style="document"/>
      <wsdl:input name="GetInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="GetOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="GetFault">
        <soap:fault name="GetFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <wsdl:operation name="Update">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Update" style="document"/>
      <wsdl:input name="UpdateInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="UpdateOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="UpdateFault">
        <soap:fault name="UpdateFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Canceled">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Canceled" style="document"/>
      <wsdl:input name="CanceledInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CanceledOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CanceledFault">
        <soap:fault name="CanceledFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Changed">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Changed" style="document"/>
      <wsdl:input name="ChangedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ChangedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ChangedFault">
        <soap:fault name="ChangedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Created">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Created" style="document"/>
      <wsdl:input name="CreatedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CreatedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CreatedFault">
        <soap:fault name="CreatedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Deleted">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Deleted" style="document"/>
      <wsdl:input name="DeletedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="DeletedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="DeletedFault">
        <soap:fault name="DeletedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Executed">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Executed" style="document"/>
      <wsdl:input name="ExecutedInput">
        <soap:body use="literal"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      </wsdl:input>
      <wsdl:output name="ExecutedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ExecutedFault">
        <soap:fault name="ExecutedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Updated">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Updated" style="document"/>
      <wsdl:input name="UpdatedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="UpdatedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="UpdatedFault">
        <soap:fault name="UpdatedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="IEC61968-100_Ed2_Strong">
    <wsdl:port name="IEC61968-100_Ed2_Strong_Port" binding="tns:IEC61968-
100_Ed2_Strong_Binding">
      <!-- The SOAP address below must be changed for a particular
implementation. -->
      <soap:address location="{SOAPADDRESS}"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Figure B.3 – Contents of StrongMessageWsdlTemplate.txt file

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annex C
(informative)

Example XSD profiles inheriting from AbstractPayloadType

C.1 GetMeterReadings profile

The profile shown in Figure C.1 is an excerpt from the GetMeterReadings.xsd profile defined in IEC 61969-9:2013. This has been modified in order to inherit from the `AbstractPayloadType` XSD type. The required modifications are highlighted in bold type.

The data corresponding to this profile may therefore appear in the `AbstractPayload` section of a `<Payload>` element (7.7.5).

NOTE The source code made available in https://github.com/TC57WG14/Part-100-Ed-2-XSD-Updater may be useful for effecting the automatic conversion of existing XSD profile files for this document.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:m="http://iec.ch/TC57/2011/GetMeterReadings#"
    xmlns:part100="http://iec.ch/TC57/2021/schema/message"
    xmlns:sawsdl="http://www.w3.org/ns/sawsdl"
    targetNamespace="http://iec.ch/TC57/2011/GetMeterReadings#"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
     schemaLocation="AbstractPayloadType.xsd"/>
  <xs:annotation>
      <xs:documentation/>
  </xs:annotation>
  <xs:element name="GetMeterReadings" type="m:GetMeterReadings"
       substitutionGroup="part100:AbstractPayload"/>
  <xs:complexType name="GetMeterReadings">
    <xs:complexContent>
      <xs:extension base="part100:AbstractPayloadType">
      <xs:sequence>
        <xs:element name="EndDevice" type="m:EndDevice" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="EndDeviceGroup" type="m:EndDeviceGroup"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="MeterReadings" type="m:MeterReadings" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="Reading" type="m:Reading" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="ReadingQuality" type="m:ReadingQuality"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ReadingType" type="m:ReadingType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="TimeSchedule" type="m:TimeSchedule" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="TransformerTank" type="m:TransformerTank"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="UsagePoint" type="m:UsagePoint" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="UsagePointGroup" type="m:UsagePointGroup"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType . . .>
  . . . . .
  </xs:complexType>
</xs:schema>
```

Figure C.1 – Example XSD profile illustrating inheritance from `AbstractPayloadType`

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

C.2 BinaryData profile

The XSD profile shown in Figure C.2 is suitable for the transmission of arbitrary binary data. The data themselves are held in base64 format (RFC 4648).

The data corresponding to this profile may therefore appear in the `AbstractPayload` section of a `<Payload>` element (7.7.5).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
     xmlns:m="http://iec.ch/TC57/2011/JsonData#"
     xmlns:part100="http://iec.ch/TC57/2021/schema/message"
     targetNamespace="http://iec.ch/TC57/2021/schema/binary"
     elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
     schemaLocation="AbstractPayloadType.xsd"/>
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:element name="Binary" type="m:Binary"
      substitutionGroup="part100:AbstractPayload"/>
  <xs:complexType name="Binary">
    <xs:complexContent>
      <xs:extension base="part100:AbstractPayloadType">
        <xs:sequence>
          <xs:element name="Base64" type="m:Base64" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="Base64">
    <xs:restriction base="xs:base64Binary"/>
  </xs:simpleType>
</xs:schema>
```

Figure C.2 – Example XSD profile suitable for the transmission of binary data

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annex D
(informative)

Example IEC 61968-100:2021 messages

D.1 General

The example messages in this annex illustrate the use of IEC 61968-100:2021 for sending and receiving application-specific data defined according to IEC 61968-9:2013. Similar message patterns and examples may also be applied to other IEC 61968 parts.

The HTTP transport using the generic WSDL is assumed to be used throughout this annex (9.2.5.2). Consequently this results in some messages having a simple acknowledgement reply message.

Most of the examples in this annex use mRIDs for object identification. Refer to Annex F for examples that illustrate the use of the `<Name><name>`, `<Name><NameType><name>` and `<Name><NameType><NameTypeAuthority><name>` elements for object identification.

D.2 Messages dealing with object instantiation

D.2.1 General

This example using the request/response message exchange pattern illustrates the instantiation of a single meter on a responding system.

The requesting system sends a "meter instantiation" request to the responding system. This causes the responding system to attempt to add the given meter along with all the supplied details to its internal database of known meters. Notification of success or failure of this attempt is then returned back to the requesting system.

D.2.2 Sequence diagram

The sequence diagram for this example is shown in Figure D.1.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

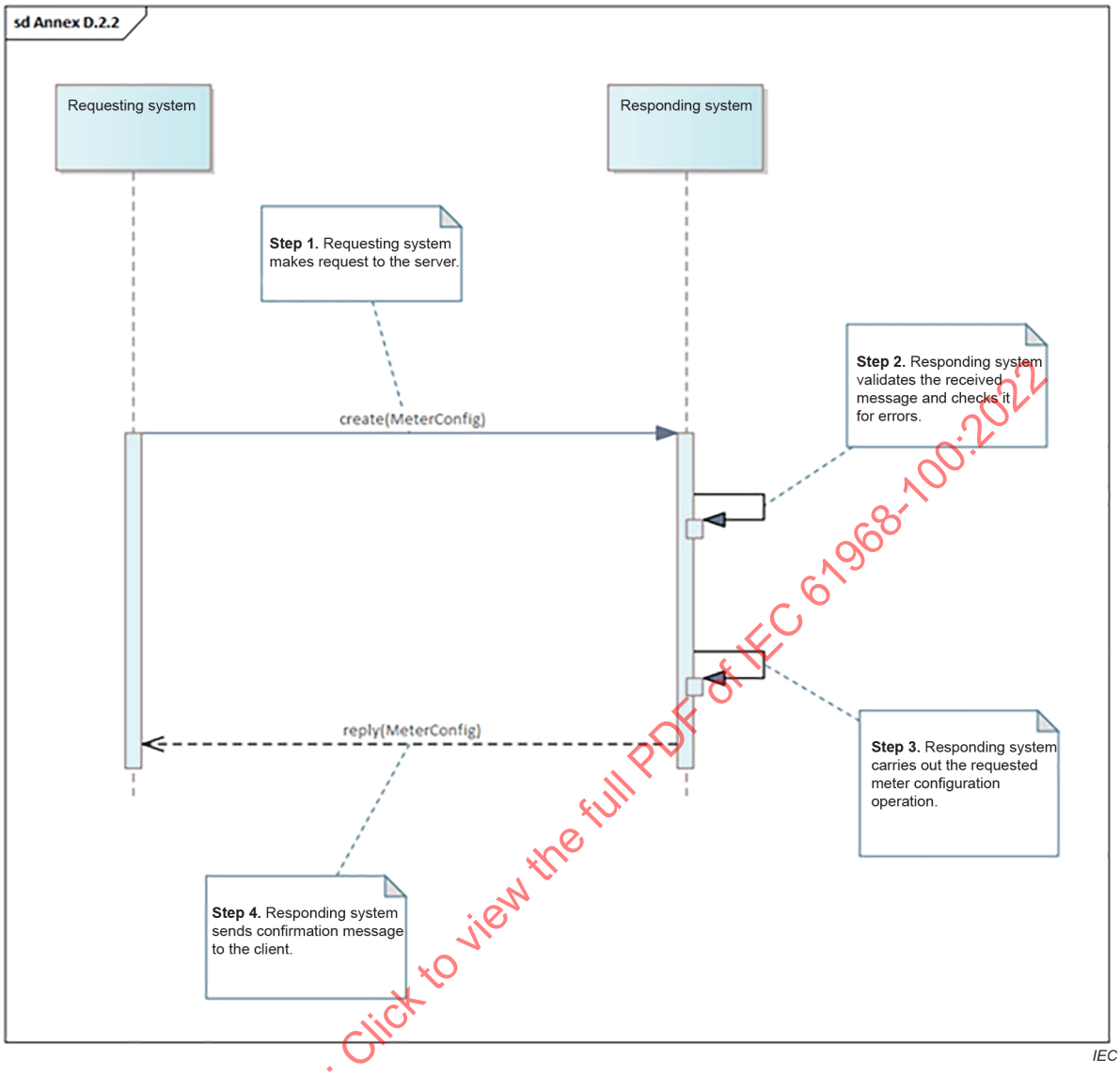


Figure D.1 – Example sequence diagram for a meter instantiation request

A request to instantiate a meter object is formulated as an IEC 61968-9:2013 create(MeterConfig) request message which is sent directly from the requesting system to the responding system (step 1). The responding system then checks the request for errors (step 2) and, assuming no problems are found with the request, carries out the requested meter instantiation (step 3). It then responds to the requesting system with a confirmation message in an IEC 61968-9:2013 reply(MeterConfig) message (step 4).

D.2.3 Meter instantiation request (request message)

The request message sent from the requesting system to the responding system (step 1) is a IEC 61968-9:2013 create(MeterConfig) message as shown in Figure D.2. In this example, the request specifies a single meter with a single register and two channels.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:10-06:00</Timestamp>
    <MessageID>fbb1c893-83c9-47b2-8cca-b8f8859315e6</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a169-ce38fba5d40f</CorrelationID>
    <Comment>Request to instantiate a meter</Comment>
  </Header>
  <Payload>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <ConnectDisconnectFunction>
        <enabled>true</enabled>
        <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
        <isRemoteAutoReconOp>true</isRemoteAutoReconOp>
        <Names>
          <name>CDF1</name>
        </Names>
      </ConnectDisconnectFunction>
      <Meter>
        <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
        <formNumber>2S</formNumber>
        <electronicAddress>
          <mac>98-5F-D3-42-05-39</mac>
        </electronicAddress>
        <ConnectDisconnectFunction ref="CDF1"/>
        <SimpleEndDeviceFunction ref="SEDF1"/>
        <EndDeviceInfo>
          <phaseCount>1</phaseCount>
          <ratedVoltage>240.0</ratedVoltage>
          <capability>
            <communication>true</communication>
            <connectDisconnect>true</connectDisconnect>
            <electricMetering>true</electricMetering>
            <onRequestRead>true</onRequestRead>
          </capability>
        </EndDeviceInfo>
        <Names>
          <name>M001</name>
          <NameType>
            <name>MeterID</name>
          </NameType>
        </Names>
      </Meter>
      <SimpleEndDeviceFunction>
        <programID>ER017A</programID>
        <Names>
          <name>SEDF1</name>
        </Names>
        <Registers>
          <Channels>
            <Names>
              <name>CH1</name>
            </Names>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
          <Channels>
            <Names>
              <name>CH2</name>
            </Names>
            <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          <Names>
            <name>REG1</name>
          </Names>
        </Registers>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </Payload>
</CreateRequestMessage>
```

Figure D.2 – Example meter instantiation request message

D.2.4 Meter instantiation response (step 4, success case)

Assuming that the meter instantiation request succeeds, the responding system may reply with a response message such as that shown in Figure D.3.

```
<CreateResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:12-06:00</Timestamp>
    <MessageID>ffb209a3-8f1e-4e5e-a8a1-a52a33eb81a4</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a169-ce38fba5d40f</CorrelationID>
    <Comment>Response to a meter instantiation request</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
</CreateResponseMessage>
```

Figure D.3 – Example meter instantiation response message (success case)

D.2.5 Meter instantiation response (step 4, failure case)

The request for meter instantiation may fail, for example, if the specified meter is already instantiated in the responding system. The responding system then replies with a message such as that shown in Figure D.4.

```
<CreateResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:12-06:00</Timestamp>
    <MessageID>da1eea46-b239-4685-9c2c-e0300084ddc7</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a169-ce38fba5d40f</CorrelationID>
    <Comment>Response to a repeated meter instantiation request</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>ERROR</Level>
      <Code>5.4</Code>
      <Reason>Referenced object already exists</Reason>
      <ObjectID>
        <Names>
          <name>M001</name>
        </Names>
        <ObjectType>Meter</ObjectType>
      </ObjectID>
    </Status>
  </Reply>
</CreateResponseMessage>
```

Figure D.4 – Example meter instantiation response message (failure case)

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

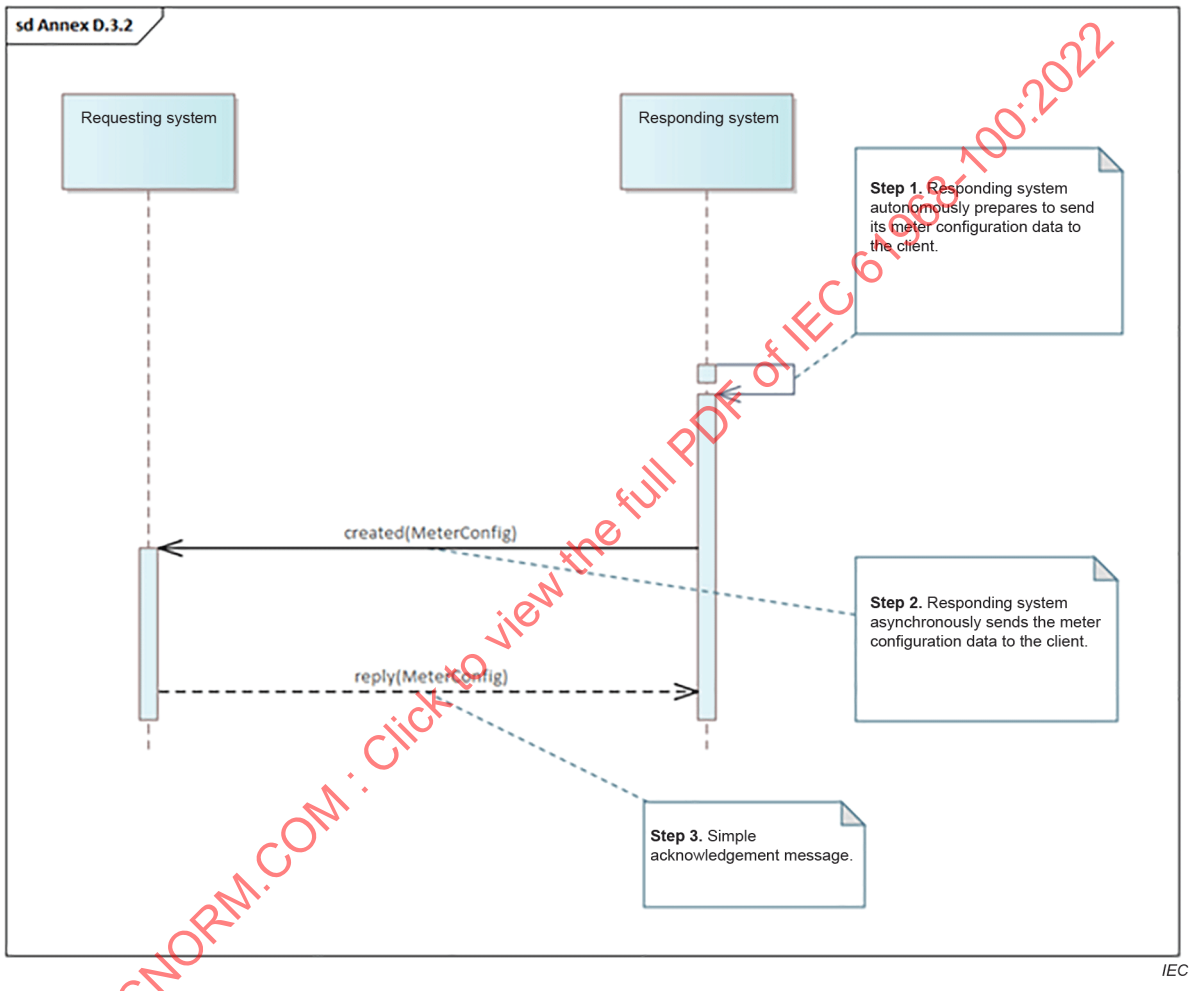
D.3 Messages dealing with the event notification of object instantiation

D.3.1 General

This example illustrates a responding system that autonomously reports a set of meter instantiations to an interested requesting system.

D.3.2 Sequence diagram

The sequence diagram for this example is shown in Figure D.5.



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure D.5 – Example sequence diagram for the publication of a meter instantiation

D.3.3 Publication of a meter instantiation (event message)

The event message sent from the responding system to the requesting system is an IEC 61968-9:2013 created(MeterConfig) message as shown in Figure D.6. This message may be sent from a responding system after being instantiated with the MeterConfig data shown for the example described in Clause D.2.

```
<CreatedEventMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>created</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-17T00:00:02-06:00</Timestamp>
    <MessageID>e0c9a3e6-38cd-429d-b4f8-1c0fb435102e</MessageID>
    <CorrelationID>7976ddcd-9244-4f88-b22d-65d0e1ffe81f</CorrelationID>
    <Comment>Reporting a meter instantiation</Comment>
  </Header>
  <Payload>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <ConnectDisconnectFunction>
        <enabled>true</enabled>
        <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
        <isRemoteAutoReconOp>true</isRemoteAutoReconOp>
        <Names>
          <name>CDF1</name>
        </Names>
      </ConnectDisconnectFunction>
      <ConnectDisconnectFunction>
        <enabled>false</enabled>
        <isRemoteAutoDisconOp>false</isRemoteAutoDisconOp>
        <isRemoteAutoReconOp>false</isRemoteAutoReconOp>
        <Names>
          <name>CDF2</name>
        </Names>
      </ConnectDisconnectFunction>
      <Meter>
        <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
        <formNumber>2S</formNumber>
        <electronicAddress>
          <mac>98-5F-D3-42-05-39</mac>
        </electronicAddress>
        <ConnectDisconnectFunction ref="CDF1"/>
        <SimpleEndDeviceFunction ref="SEDF1"/>
        <EndDeviceInfo>
          <phaseCount>1</phaseCount>
          <ratedVoltage>240.0</ratedVoltage>
          <capability>
            <communication>true</communication>
            <connectDisconnect>true</connectDisconnect>
            <electricMetering>true</electricMetering>
            <onRequestRead>true</onRequestRead>
          </capability>
        </EndDeviceInfo>
        <Names>
          <name>M001</name>
          <NameType>
            <name>MeterID</name>
          </NameType>
        </Names>
      </Meter>
      <Meter>
        <mRID>83009321-7883-4ea6-bc0e-71ddcd274f69</mRID>
        <formNumber>12S</formNumber>
        <electronicAddress>
          <mac>9A-5F-D3-42-04-39</mac>
        </electronicAddress>
        <ConnectDisconnectFunction ref="CDF1"/>
        <SimpleEndDeviceFunction ref="SEDF1"/>
        <EndDeviceInfo>
          <phaseCount>3</phaseCount>
          <ratedVoltage>240.0</ratedVoltage>
          <capability>
            <communication>true</communication>
            <connectDisconnect>true</connectDisconnect>
            <electricMetering>true</electricMetering>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <onRequestRead>true</onRequestRead>
    </capability>
  </EndDeviceInfo>
  <Names>
    <name>M002</name>
  </Names>
</Meter>
<Meter>
  <mRID>99dcbdde-3249-11e8-b467-0ed5f89f718b</mRID>
  <formNumber>2S</formNumber>
  <electronicAddress>
    <mac>9A-5F-D3-42-04-40</mac>
  </electronicAddress>
  <ConnectDisconnectFunction ref="CDF1"/>
  <SimpleEndDeviceFunction ref="SEDF1"/>
  <EndDeviceInfo>
    <phaseCount>1</phaseCount>
    <ratedVoltage>240.0</ratedVoltage>
    <capability>
      <communication>true</communication>
      <connectDisconnect>true</connectDisconnect>
      <electricMetering>true</electricMetering>
      <onRequestRead>true</onRequestRead>
    </capability>
  </EndDeviceInfo>
  <Names>
    <name>M003</name>
  </Names>
</Meter>
<Meter>
  <mRID>645707ea-37d6-11e8-b467-0ed5f89f718b</mRID>
  <formNumber>2S</formNumber>
  <electronicAddress>
    <mac>9A-5F-D3-42-04-4A</mac>
  </electronicAddress>
  <ConnectDisconnectFunction ref="CDF2"/>
  <SimpleEndDeviceFunction ref="SEDF2"/>
  <EndDeviceInfo>
    <phaseCount>1</phaseCount>
    <ratedVoltage>240.0</ratedVoltage>
    <capability>
      <communication>true</communication>
      <connectDisconnect>true</connectDisconnect>
      <electricMetering>true</electricMetering>
      <onRequestRead>true</onRequestRead>
    </capability>
  </EndDeviceInfo>
  <Names>
    <name>M004</name>
  </Names>
</Meter>
<Meter>
  <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
  <formNumber>12S</formNumber>
  <electronicAddress>
    <mac>9A-5F-D3-42-04-4B</mac>
  </electronicAddress>
  <ConnectDisconnectFunction ref="CDF2"/>
  <SimpleEndDeviceFunction ref="SEDF2"/>
  <EndDeviceInfo>
    <phaseCount>1</phaseCount>
    <ratedVoltage>240.0</ratedVoltage>
    <capability>
      <communication>true</communication>
      <connectDisconnect>true</connectDisconnect>
      <electricMetering>true</electricMetering>
      <onRequestRead>true</onRequestRead>
    </capability>
  </EndDeviceInfo>
  <Names>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <name>M005</name>
    </Names>
  </Meter>
  <SimpleEndDeviceFunction>
    <programID>ER017A</programID>
    <Names>
      <name>SEDF1</name>
    </Names>
    <Registers>
      <Channels>
        <Names>
          <name>CH1</name>
        </Names>
        <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
      </Channels>
      <Channels>
        <Names>
          <name>CH2</name>
        </Names>
        <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
      </Channels>
      <Names>
        <name>REG1</name>
      </Names>
    </Registers>
  </SimpleEndDeviceFunction>
  <SimpleEndDeviceFunction>
    <programID>ER017A</programID>
    <Names>
      <name>SEDF2</name>
    </Names>
    <Registers>
      <touTierName>overnight</touTierName>
      <Channels>
        <Names>
          <name>CH1</name>
        </Names>
        <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.1.0.0.0.3.72.0"/>
      </Channels>
      <Channels>
        <Names>
          <name>CH2</name>
        </Names>
        <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.1.0.0.0.3.72.0"/>
      </Channels>
      <Names>
        <name>REG1</name>
      </Names>
      <touTier>
        <end>06:00:00</end>
        <start>22:00:00</start>
      </touTier>
    </Registers>
    <Registers>
      <touTierName>daytime</touTierName>
      <Channels>
        <Names>
          <name>CH1</name>
        </Names>
        <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.2.0.0.0.3.72.0"/>
      </Channels>
      <Channels>
        <Names>
          <name>CH2</name>
        </Names>
        <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.2.0.0.0.3.72.0"/>
      </Channels>
      <Names>
        <name>REG2</name>
      </Names>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          <touTier>
            <end>22:00:00</end>
            <start>06:00:00</start>
          </touTier>
        </Registers>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </Payload>
</CreatedEventMessage>
```

Figure D.6 – Example publication of a meter instantiation (event message)

D.3.4 Publication of a meter instantiation (simple acknowledgement message)

After sending the created(MeterConfig) message (step 2), the responding system may wait until the requesting system confirms receipt by means of a simple acknowledgement message (step 3) as shown in Figure D.7.

```
<CreateResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-17T00:00:04-06:00</Timestamp>
    <MessageID>5be39bcc-5e55-4aaf-b10f-ab432bc91da7</MessageID>
    <CorrelationID>7976ddcd-9244-4f88-b22d-65d0e1ffe81f</CorrelationID>
    <Comment>Reporting a meter instantiation (simple ack)</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.3</Code>
      <Reason>simple ack</Reason>
    </Status>
  </Reply>
</CreateResponseMessage>
```

Figure D.7 – Example publication of a meter instantiaion (simple acknowledgement message)

D.4 Messages dealing with requests for information concerning an object instantiation

D.4.1 General

This example illustrates a requesting system that requests an object from a responding system.

On receipt of such a request, the responding system retrieves the requested data and returns these to the requesting system.

D.4.2 Sequence diagram

The sequence diagram for this example is shown in Figure D.8.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

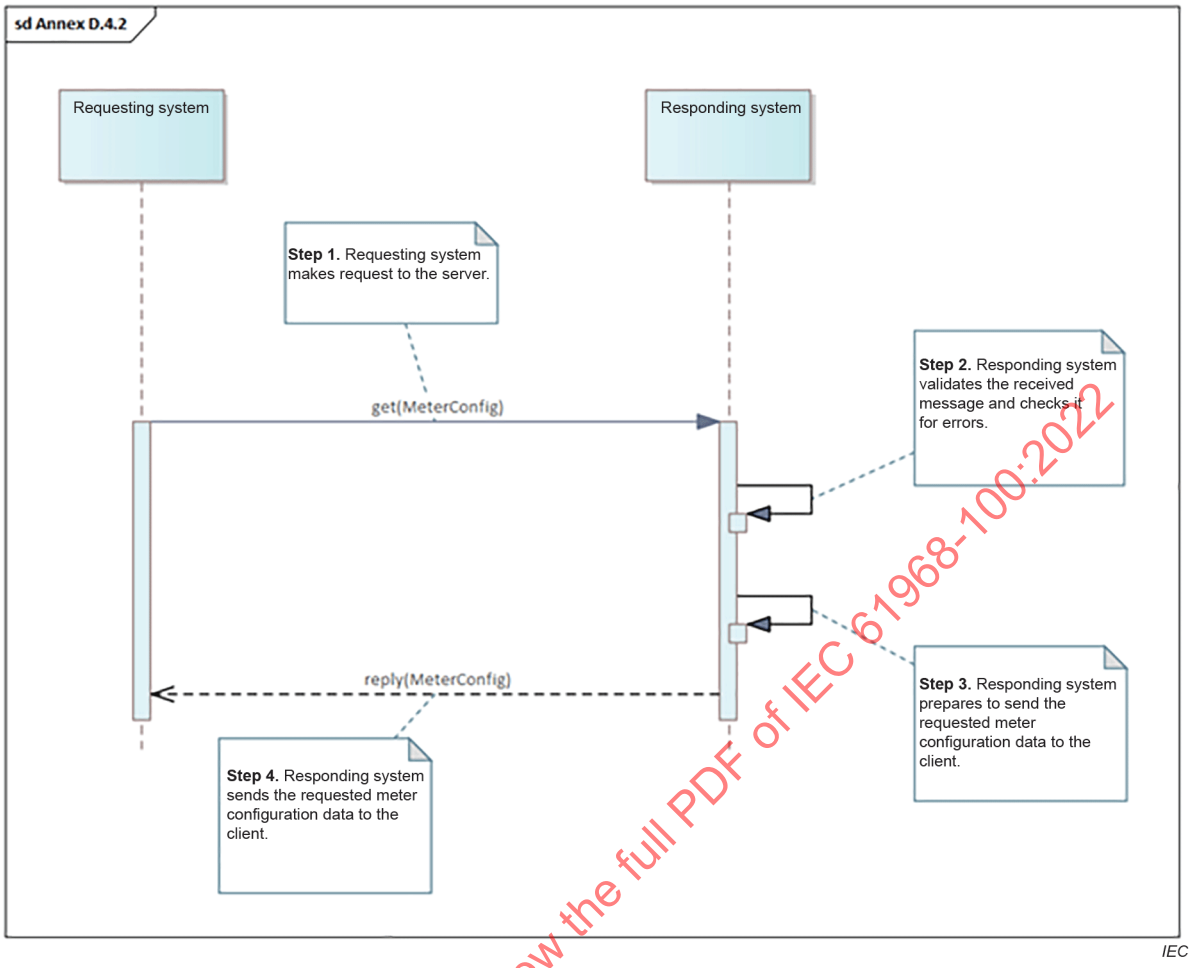


Figure D.8 – Example sequence diagram for requests for information concerning an object instantiation

A request for meter configuration data is formulated as an IEC 61968-9:2013 get(MeterConfig) request message which is sent directly from the requesting system to the responding system (step 1). The responding system then checks the request for errors (step 2) and, assuming no problems are found with the request, retrieves the requested meter configuration data (step 3). It then responds to the requesting system with the requested data in an IEC 61968-9:2013 reply(MeterConfig) message (step 4).

D.4.3 Request for information concerning a meter instantiation (request message)

The request message (step 1) sent from the requesting system to the responding system is an IEC 61968-9:2013 get(MeterConfig) message as shown in Figure D.9. In this example, the request is for all data pertaining to meter M001.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-03-01T08:04:56-06:00</Timestamp>
    <MessageID>e94d334d-a1ee-4a30-b780-9e4c775ccc5f</MessageID>
    <CorrelationID>705a9743-f3bd-4f5a-b1d9-a7f98bb6bbcd</CorrelationID>
    <Comment>Request for information concerning a meter
instantiation</Comment>
  </Header>
  <Payload>
    <GetMeterConfig xmlns="http://iec.ch/TC57/2011/GetMeterConfig#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Meter>
        <Names>
          <name>M001</name>
        </Names>
      </Meter>
    </GetMeterConfig>
  </Payload>
</GetRequestMessage>
```

Figure D.9 – Example request for information concerning a meter instantiation (request message)

D.4.4 Request for information concerning a meter instantiation (response message)

The response message (step 4) sent from the responding system to the requesting system is a IEC 61968-9:2013 reply(MeterConfig) message as shown in Figure D.10.

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-03-01T08:04:58-06:00</Timestamp>
    <MessageID>2eacb133-59ec-48e0-a07c-51206789d087</MessageID>
    <CorrelationID>705a9743-f3bd-4f5a-b1d9-a7f98bb6bbcd</CorrelationID>
    <Comment>Response to a request for information concerning a meter
instantiation</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
  <Payload>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <ConnectDisconnectFunction>
        <enabled>true</enabled>
        <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
        <isRemoteAutoReconOp>true</isRemoteAutoReconOp>
        <Names>
          <name>CDF1</name>
        </Names>
      </ConnectDisconnectFunction>
      <Meter>
        <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
        <formNumber>2S</formNumber>
        <electronicAddress>
          <mac>98-5F-D3-42-05-39</mac>
        </electronicAddress>
        <ConnectDisconnectFunction ref="CDF1"/>
        <SimpleEndDeviceFunction ref="SEDF1"/>
        <EndDeviceInfo>
          <phaseCount>1</phaseCount>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          <ratedVoltage>240.0</ratedVoltage>
          <capability>
            <communication>true</communication>
            <connectDisconnect>true</connectDisconnect>
            <electricMetering>true</electricMetering>
            <onRequestRead>true</onRequestRead>
          </capability>
        </EndDeviceInfo>
        <Names>
          <name>M001</name>
          <NameType>
            <name>MeterID</name>
          </NameType>
        </Names>
      </Meter>
      <SimpleEndDeviceFunction>
        <programID>ER017A</programID>
        <Names>
          <name>SEDF1</name>
        </Names>
        <Registers>
          <Channels>
            <Names>
              <name>CH1</name>
            </Names>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
          <Channels>
            <Names>
              <name>CH2</name>
            </Names>
            <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
          <Names>
            <name>REG1</name>
          </Names>
        </Registers>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </Payload>
</GetResponseMessage>
```

Figure D.10 – Example request for information concerning a meter instantiation (response message)

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.5 Messages utilizing an operationset to enforce transactional integrity

D.5.1 General

This example utilizes an OperationSet to enforce transactional integrity (7.7.3).

D.5.2 Sequence diagram

The sequence diagram for this example is shown in Figure D.11.

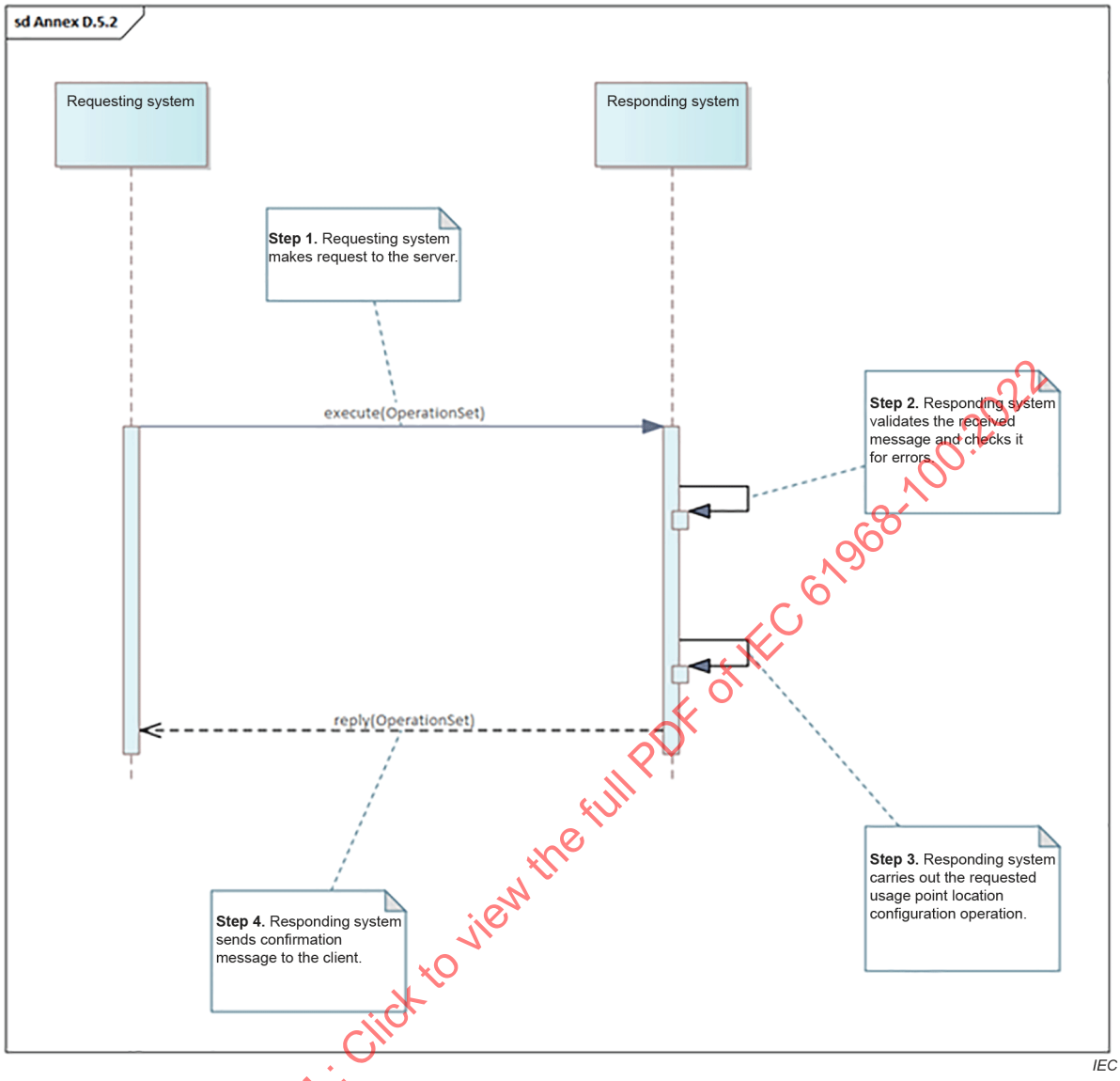


Figure D.11 – Example sequence diagram for an OperationSet request

This example illustrates a request to instantiate a meter config and a usage point location and associate them using a master data linkage. This is requested to be done in a pre-defined sequence and as a single transaction.

This is formulated as an IEC 61968-9:2013 execute(OperationSet) request message which is sent directly from the requesting system to the responding system (step 1). The responding system then checks the request for errors (step 2) and, assuming no problems are found with the request, carries out the requested operations (step 3). It then responds to the requesting system with a confirmation message in an IEC 61968-9:2013 reply(OperationSet) message (step 4).

D.5.3 Using an operationset to enforce transactional integrity (request message)

The request message sent from the requesting system to the responding system (step 1) is an IEC 61968-9:2013 execute(OperationSet) message as shown in Figure D.12. In this example it contains four operations.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<ExecuteRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>execute</Verb>
    <Noun>OperationSet</Noun>
    <Timestamp>2018-04-13T13:23:10-06:00</Timestamp>
    <MessageID>aa872373-6490-4ef6-833d-46c7f2eff5a9</MessageID>
    <CorrelationID>8cf2a42a-5c4f-40fc-822f-a0b223e6268d</CorrelationID>
    <Comment>Using an OperationSet to enforce transactional
integrity</Comment>
  </Header>
  <Payload>
    <OperationSet>
      <OperationSetID>80e293dc-cbcb-4e92-9b9a-ca6396c90d8b</OperationSetID>
      <EnforceMsgSequence>true</EnforceMsgSequence>
      <EnforceTransactionalIntegrity>true</EnforceTransactionalIntegrity>
      <Operation>
        <OperationID>1</OperationID>
        <Verb>create</Verb>
        <Noun>MeterConfig</Noun>
        <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <ConnectDisconnectFunction>
            <mRID>36917bfa-7909-4a9f-b997-29724be5f397</mRID>
            <enabled>true</enabled>
            <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
            <isRemoteAutoReconOp>true</isRemoteAutoReconOp>
          </ConnectDisconnectFunction>
          <Meter>
            <mRID>3f85f1ef-727b-458f-b45e-5fb40793579a</mRID>
            <formNumber>2S</formNumber>
            <electronicAddress>
              <mac>98-5F-D3-42-05-39</mac>
            </electronicAddress>
            <ConnectDisconnectFunction
ref="36917bfa-7909-4a9f-b997-29724be5f397"/>
            <SimpleEndDeviceFunction
ref="4485ce93-a285-47e3-a3ef-368c7fe7eb98"/>
            <EndDeviceInfo>
              <phaseCount>1</phaseCount>
              <ratedVoltage>240.0</ratedVoltage>
              <capability>
                <communication>true</communication>
                <connectDisconnect>true</connectDisconnect>
                <electricMetering>true</electricMetering>
                <onRequestRead>true</onRequestRead>
              </capability>
            </EndDeviceInfo>
            <Names>
              <name>M006</name>
            </Names>
          </Meter>
          <SimpleEndDeviceFunction>
            <mRID>4485ce93-a285-47e3-a3ef-368c7fe7eb98</mRID>
            <programID>ER017A</programID>
            <Registers>
              <Channels>
                <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
              </Channels>
              <Channels>
                <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
              </Channels>
            </Registers>
          </SimpleEndDeviceFunction>
        </MeterConfig>
      </Operation>
      <Operation>
        <OperationID>2</OperationID>
        <Verb>create</Verb>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
  <Noun>UsagePointLocationConfig</Noun>
  <UsagePointLocationConfig
      xmlns="http://iec.ch/TC57/2011/UsagePointLocationConfig#"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <UsagePointLocation>
      <mRID>3343dcc9-0de2-4d8f-af6d-dc4d79f0368f</mRID>
      <mainAddress>
        <streetDetail>
          <name>Chestnut</name>
          <number>207</number>
          <prefix>N</prefix>
          <suffix>St</suffix>
        </streetDetail>
        <townDetail>
          <code>52402</code>
          <country>USA</country>
          <name>Cedar Rapids</name>
          <stateOrProvince>IA</stateOrProvince>
        </townDetail>
      </mainAddress>
      <Names>
        <name>UPL006</name>
      </Names>
    </UsagePointLocation>
  </UsagePointLocationConfig>
</Operation>
<Operation>
  <OperationID>3</OperationID>
  <Verb>create</Verb>
  <Noun>UsagePointConfig</Noun>
  <UsagePointConfig xmlns="http://iec.ch/TC57/2011/UsagePointConfig#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <UsagePoint>
      <mRID>354383a4-23a6-4188-aa37-35db994df5d6</mRID>
      <isSdp>true</isSdp>
      <Names>
        <name>UP006</name>
      </Names>
      <UsagePointLocation>
        <Names>
          <name>UPL006</name>
        </Names>
      </UsagePointLocation>
    </UsagePoint>
  </UsagePointConfig>
</Operation>
<Operation>
  <OperationID>4</OperationID>
  <Verb>create</Verb>
  <Noun>MasterDataLinkageConfig</Noun>
  <MasterDataLinkageConfig
        xmlns="http://iec.ch/TC57/2011/MasterDataLinkageConfig#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Meter>
      <Names>
        <name>M006</name>
      </Names>
    </Meter>
    <UsagePoint>
      <Names>
        <name>UP006</name>
      </Names>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          </UsagePoint>
        </MasterDataLinkageConfig>
      </Operation>
    </OperationSet>
  </Payload>
</ExecuteRequestMessage>
```

Figure D.12 – Example OperationSet message to enforce transactional integrity (request message)

D.5.4 Using an operationset to enforce transactional integrity (response message)

The response message (step 4) is an IEC 61968-9:2013 reply(OperationSet) message as shown in Figure D.13.

```
<ExecuteResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>OperationSet</Noun>
    <Timestamp>2018-04-13T13:23:18-06:00</Timestamp>
    <MessageID>c75008b9-ec94-432b-af75-4ffc3c353193</MessageID>
    <CorrelationID>8cf2a42a-5c4f-40fc-822f-a0b223e6268d</CorrelationID>
    <Comment>Response to a request for using an OperationSet to enforce
transactional integrity</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
</ExecuteResponseMessage>
```

Figure D.13 – Example OperationSet message to enforce transactional integrity (response message)

D.6 Messages dealing with the publication of data

D.6.1 General

This example illustrates a responding system that autonomously reports a set of meter readings to an interested requesting system.

It may be assumed that the responding system in question has already collected the necessary meter read data according to some pre-defined configuration. Similarly, the trigger for sending out these data is pre-defined so that the act of publishing the meter read data takes place autonomously.

D.6.2 Sequence diagram

The sequence diagram for this example is shown in Figure D.14.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

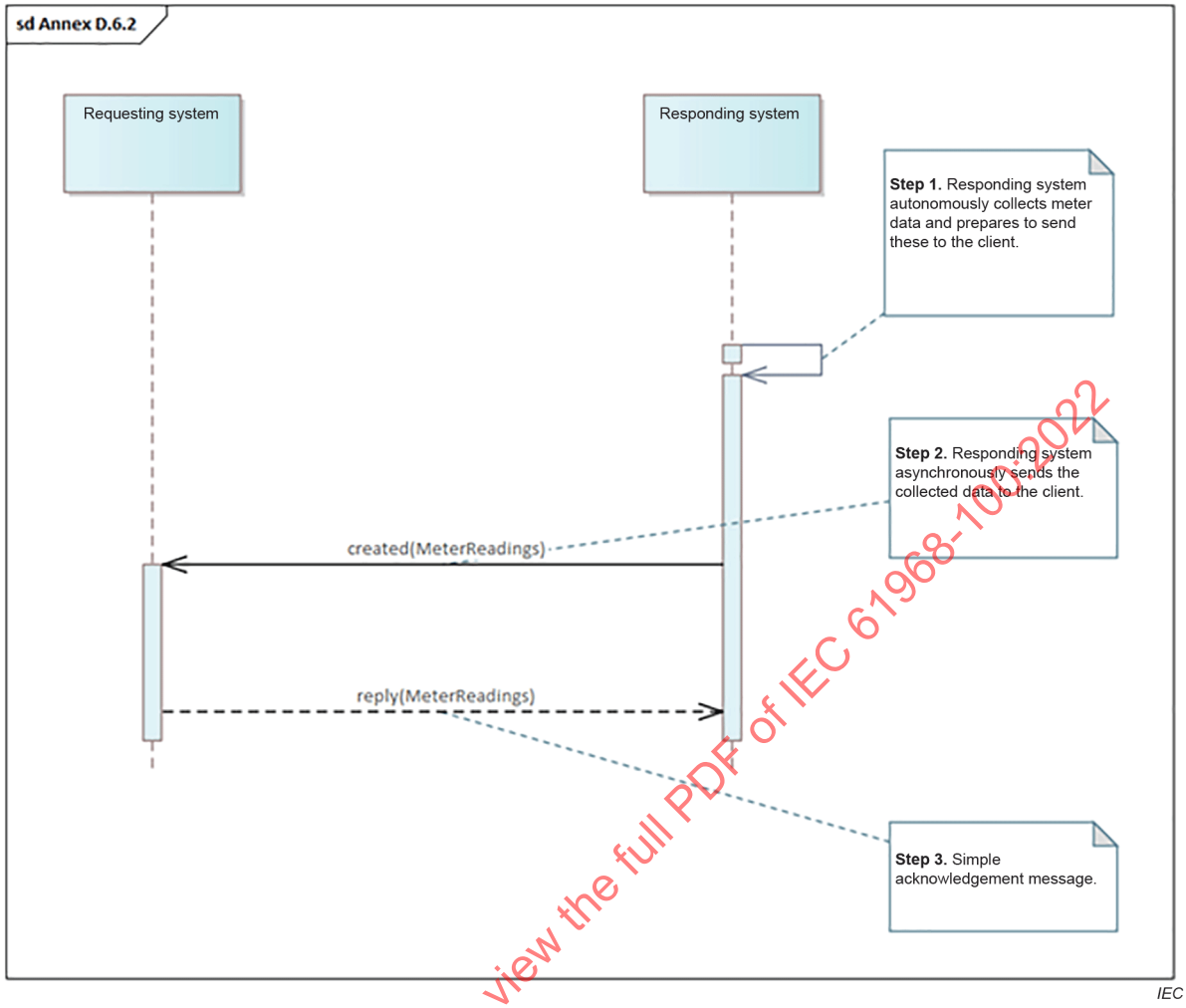


Figure D.14 – Example sequence diagram for the publication of meter readings

D.6.3 Publication of meter readings (event message)

The meter data are formulated as an IEC 61968-9:2013 unsolicited event message (step 2) as shown in Figure D.15.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreatedEventMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>created</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-03-18T02:00:05-06:00</Timestamp>
    <MessageID>686f942f-577d-4f81-bc0a-0d38edca5ff9</MessageID>
    <CorrelationID>6a99028f-4ba8-46f9-9d73-1237ca95c99b</CorrelationID>
    <Comment>Unsolicited publication of meter readings data </Comment>
  </Header>
  <Payload>
    <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <MeterReading>
<!-- Load profile/interval data values over the measurement period -->
        <IntervalBlocks>
          <IntervalReadings>
            <timeStamp>2018-03-18T01:15:00-06:00</timeStamp>
            <value>200.0</value>
          </IntervalReadings>
          <IntervalReadings>
            <timeStamp>2018-03-18T01:30:00-06:00</timeStamp>
            <value>300.0</value>
          </IntervalReadings>
          <IntervalReadings>
            <timeStamp>2018-03-18T01:45:00-06:00</timeStamp>
            <value>200.0</value>
          </IntervalReadings>
          <IntervalReadings>
            <timeStamp>2018-03-18T02:00:00-06:00</timeStamp>
            <value>100.0</value>
          </IntervalReadings>
<!-- 15Minute deltaData forward electricitySecondaryMetered energy (kWh)-->
          <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
        </IntervalBlocks>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
<!-- bulk quantity values at the start and end of the measurement period -->
        <Readings>
          <timeStamp>2018-03-18T01:00:00-06:00</timeStamp>
          <value>1000.0</value>
<!-- bulkQuantity forward electricitySecondaryMetered energy (kWh) -->
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
        <Readings>
          <timeStamp>2018-03-18T02:00:00-06:00</timeStamp>
          <value>1800.0</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
      </MeterReading>
    </MeterReadings>
  </Payload>
</CreatedEventMessage>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure D.15 – Example publication of routine meter readings (event message)

D.6.4 Publication of meter readings (simple acknowledgement message)

After sending the created(MeterReadings) message (step 2), the requesting system confirms receipt by means of a simple acknowledgement message (step 3) as shown in Figure D.16.

```
<CreatedEventResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-03-18T02:00:07-06:00</Timestamp>
    <MessageID>59cb031e-f456-4f32-98fd-4a7026c9c506</MessageID>
    <CorrelationID>6a99028f-4ba8-46f9-9d73-1237ca95c99b</CorrelationID>
    <Comment>Unsolicited publication of meter readings data (simple
ack)</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.3</Code>
      <Reason>simple ack</Reason>
    </Status>
  </Reply>
</CreatedEventResponseMessage>
```

Figure D.16 – Example publication of routine meter readings (simple acknowledgement message)

D.7 Messages dealing with an on-demand data request resulting in a single response

D.7.1 General

This example illustrates a requesting system that requests on-demand meter data from a responding system. On receipt of such a request, the responding system retrieves the requested data and returns these data back to the requesting system.

D.7.2 Sequence Diagram

The sequence diagram for this example is shown in Figure D.17.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

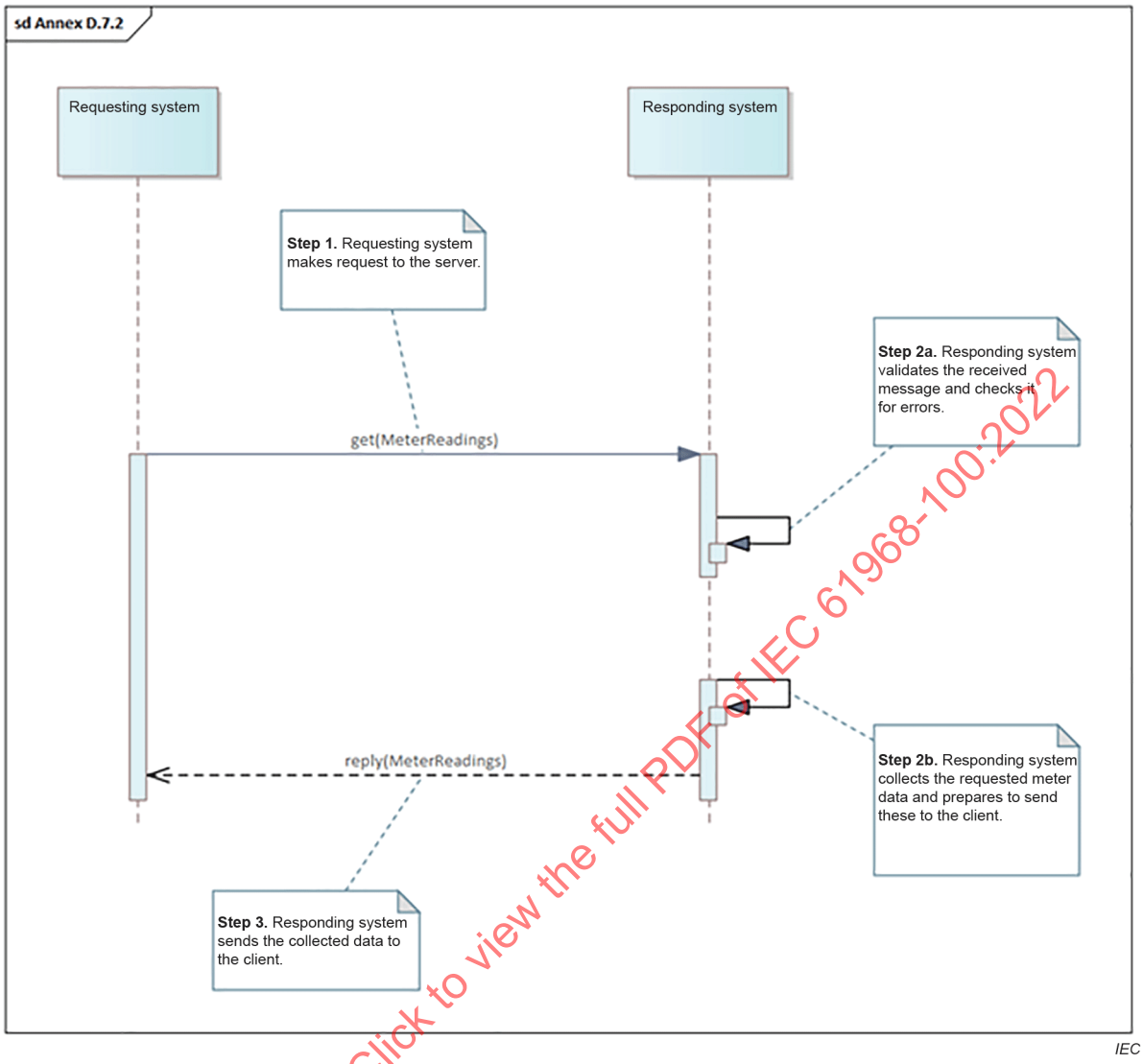


Figure D.17 – Example sequence diagram for the retrieval of on-demand meter readings

A request for meter read data is formulated as an IEC 61968-9:2013 get(MeterReadings) request message which is sent directly from the requesting system to the responding system (step 1). The responding system then checks the request for errors (step 2a) and, assuming no problems are found with the request, retrieves the required data (step 2b). It then responds to the requesting system with the requested data in an IEC 61968-9:2013 reply(MeterReadings) message (step 3).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.7.3 On-demand meter reading message (request message)

The request message sent from the requesting system to the responding system (step 1) is an IEC 61968-9:2013 get(MeterReadings) message as shown in Figure D.18.

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:16:00-06:00</Timestamp>
    <MessageID>c193e6a5-0ee9-449b-b354-1e99ab4b61e7</MessageID>
    <CorrelationID>346a7060-3aa8-49da-9384-bcfc38bf3d82</CorrelationID>
    <Comment>Request for an on-demand meter read data</Comment>
  </Header>
  <Payload>
    <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <EndDevice>
        <!-- name of an existing meter -->
        <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
      </EndDevice>
      <ReadingType>
        <Names>
<!--bulkQuantity forward electricitySecondaryMetered energy (kWh) -->
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0</name>
        </Names>
      </ReadingType>
    </GetMeterReadings>
  </Payload>
</GetRequestMessage>
```

Figure D.18 – Example on-demand meter readings request message

D.7.4 On-demand meter reading message (response message)

The response message from the responding system back to the requesting system (step 4) is an IEC 61968-9:2013 reply(MeterReadings) message as shown in Figure D.19.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:16:44-06:00</Timestamp>
    <MessageID>f37cfbbc-bfc3-46bd-9391-7d3328b193fb</MessageID>
    <CorrelationID>346a7060-3aa8-49da-9384-bcfc38bf3d82</CorrelationID>
    <Comment>Response to a request for on-demand meter read data</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
  <Payload>
    <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <MeterReading>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <Readings>
          <timeStamp>2018-10-02T14:16:12-06:00</timeStamp>
          <value>2012.1</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
      </MeterReading>
    </MeterReadings>
  </Payload>
</GetResponseMessage>
```

Figure D.19 – Example on-demand meter readings response message

D.8 Messages dealing with an on-demand data request resulting in multiple responses

D.8.1 General

This example illustrates a single request message targetting multiple end devices which results in multiple response messages being returned.

In this example, two response messages are returned from the responding system to the requesting system, each of which corresponds to the response for each individual end device. Because multiple response messages are generated for the single request message, the responding system utilizes the `<MessageSequence>` element in the header to indicate there is more than one message being returned (7.5.2).

D.8.2 Sequence diagram

The sequence diagram for this example is shown in Figure D.20.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

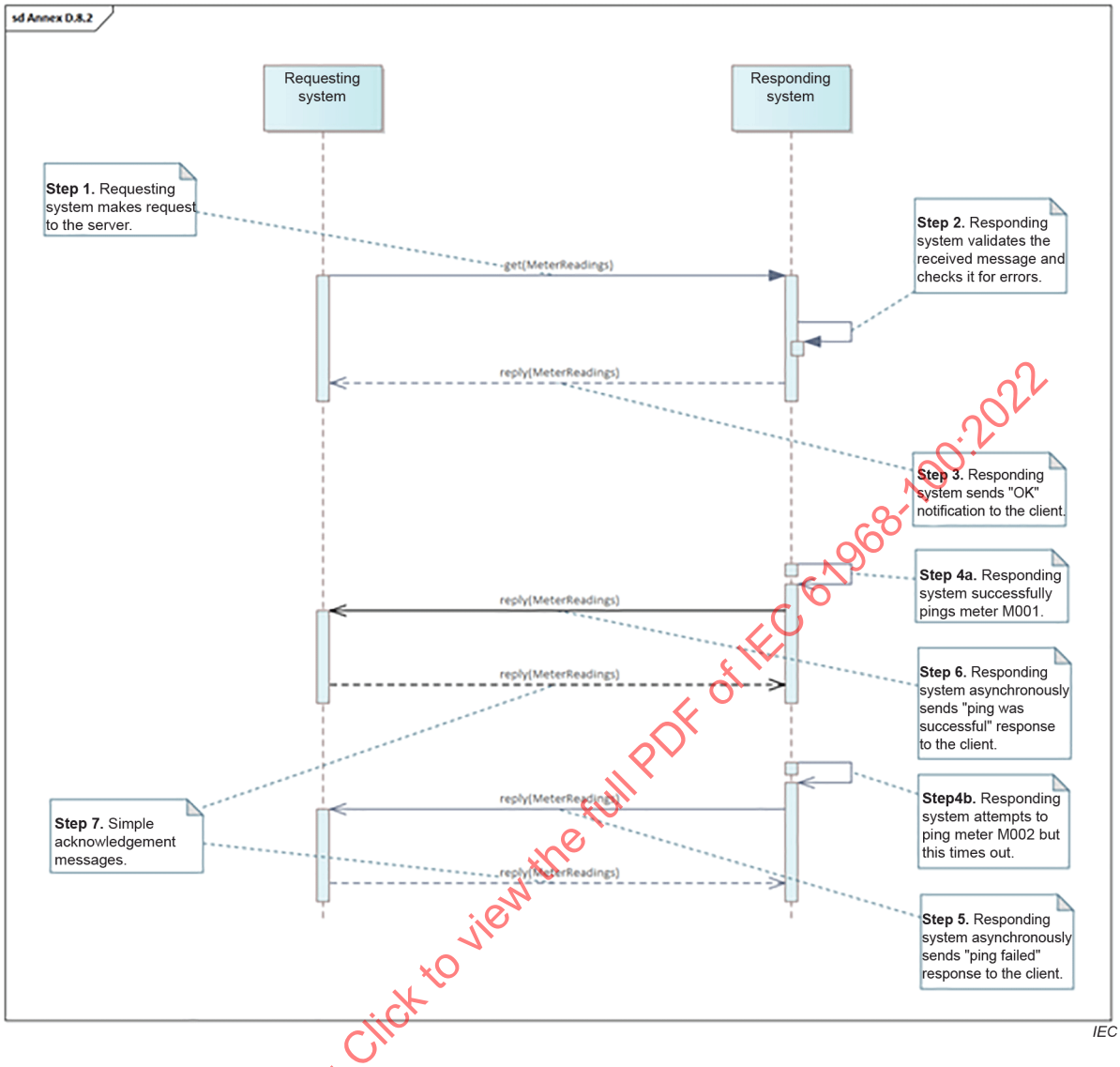


Figure D.20 – Example sequence diagram for an on-demand meter ping request

D.8.3 Request for an on-demand meter ping (request message)

The request message (step 1) sent from the requesting system to the responding system is an IEC 61968-9:2013 get(MeterReadings) message as shown in Figure D.21. It contains a request to ping two meters, one of which is in an energized state (meter M001) and a second which is in a deenergized state (M002). This example also utilizes the `<ReplyAddress>` element to provide a callback address for the responding system to use in sending the asynchronous replies back to the requesting system. This is illustrated in Figure D.21.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:26-06:00</Timestamp>
    <MessageID>41b7d6d4-2c4d-41fb-8c63-b6dd1fe400f5</MessageID>
    <CorrelationID>49882980-3254-11e8-b467-0ed5f89f718b</CorrelationID>
  <ReplyAddress>http://www.myclient.test/ReceiveMeterReadings</ReplyAddress>
    <Comment>Request for an on-demand meter ping</Comment>
  </Header>
  <Payload>
    <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <EndDevice>
        <Names>
          <name>M001</name>
          <NameType>
            <name>MeterID</name>
          </NameType>
        </Names>
      </EndDevice>
      <EndDevice>
        <Names>
          <name>M002</name>
          <NameType>
            <name>GasModuleID</name>
          </NameType>
        </Names>
      </EndDevice>
      <ReadingType>
        <Names>
          <!--electricitySecondaryMetered energization status -->
          <name>0.0.0.0.0.1.11.0.0.0.0.0.0.0.0.0.109.0</name>
        </Names>
      </ReadingType>
    </GetMeterReadings>
  </Payload>
</GetRequestMessage>
```

Figure D.21 – Example on-demand meter ping request (request message)

D.8.4 Request for an on-demand meter ping (response message 1)

On receipt of the request message, the responding system then checks it for errors (step 2) and, assuming no problems are found with the request, sends an IEC 61968-9:2013 reply(MeterReadings) message to the requesting system as shown in Figure D.22. This message contains an "OK" status. Because multiple response messages are generated for the single request message, the responding system utilizes the `<MessageSequence>` element with a `<SequenceNumber>` of 1 to indicate it is the first message in the sequence of messages being returned. Since it is not the last message in the sequence, the `<LastMessage>` element is set to false. This is illustrated in Figure D.22.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:27-06:00</Timestamp>
    <MessageID>f7fcb5ad-53c0-4636-a934-bd708c2c66c9</MessageID>
    <CorrelationID>49882980-3254-11e8-b467-0ed5f89f718b</CorrelationID>
    <MessageSequence>
      <SequenceNumber>1</SequenceNumber>
      <LastMessage>false</LastMessage>
    </MessageSequence>
    <Comment>Response 1 to a request for an on-demand meter ping</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
</GetResponseMessage>
```

Figure D.22 – Example on-demand meter ping request (response message)

D.8.5 Request for an on-demand meter ping (response message 2)

The responding system then attempts to ping the specified meters over the AMI network. The response to the ping for the first meter (M001) arrives over the AMI network and the responding system sends a corresponding IEC 61968-9:2013 reply(MeterReadings) message to the requesting system. This message contains a reading value of 1 to signify that the meter is in the energized state. Because multiple response messages are generated for the single request message, the responding system utilizes the `<MessageSequence>` element with a `<SequenceNumber>` of 2 to indicate it is the second message in the sequence of messages being returned. Since it is not the last message in the sequence, the `<LastMessage>` element is set to false. This is illustrated in Figure D.23.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:30-06:00</Timestamp>
    <MessageID>4071638e-3254-11e8-b467-0ed5f89f718b</MessageID>
    <CorrelationID>49882980-3254-11e8-b467-0ed5f89f718b</CorrelationID>
    <MessageSequence>
      <SequenceNumber>2</SequenceNumber>
      <LastMessage>false</LastMessage>
    </MessageSequence>
    <Comment>Response 2 to a request for an on-demand meter ping</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
  <Payload>
    <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <MeterReading>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <Readings>
          <timeStamp>2018-10-02T14:54:29-06:00</timeStamp>
          <!-- "1" means energized -->
          <value>1</value>
          <ReadingType ref="0.0.0.0.0.1.11.0.0.0.0.0.0.0.0.0.109.0"/>
        </Readings>
      </MeterReading>
    </MeterReadings>
  </Payload>
</GetResponseMessage>
```

Figure D.23 – Example on-demand meter ping request (response message 2)

D.8.6 Request for an on-demand meter ping (simple acknowledgement of message 2)

The requesting system acknowledges receipt of this reply(MeterReadings) message by means of a simple acknowledgement message (step 8) as shown in Figure D.24.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:30-06:00</Timestamp>
    <MessageID>cf22db89-98a7-43f2-98af-ee9eee0dbe6a</MessageID>
    <CorrelationID>49882980-3254-11e8-b467-0ed5f89f718b</CorrelationID>
    <Comment>Response 2 to a request for an on-demand meter ping (simple
ack)</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.3</Code>
      <Reason>simple ack</Reason>
    </Status>
  </Reply>
</GetResponseMessage>
```

Figure D.24 – Example on-demand meter ping request (simple acknowledgement message 2)

D.8.7 Request for an on-demand meter ping (response message 3)

When the responding system realizes that the second meter is not responding, it sends an IEC 61968-9:2013 reply(MeterReadings) message (step 6) as shown in Figure D.25. It contains a time-out notification for the meter M002.

Because multiple response messages are generated for the single request message, the responding system utilizes the `<MessageSequence>` element with a `<SequenceNumber>` of 3 to indicate it is the third message in the sequence of messages being returned. Since it is the last message in the sequence, the `<LastMessage>` element is set to true.

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:41-06:00</Timestamp>
    <MessageID>ed018ab0-b3f8-4373-983f-cf48a00e85e5</MessageID>
    <CorrelationID>49882980-3254-11e8-b467-0ed5f89f718b</CorrelationID>
    <MessageSequence>
      <SequenceNumber>3</SequenceNumber>
      <LastMessage>true</LastMessage>
    </MessageSequence>
    <Comment>Response 3 to a request for an on-demand meter ping</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>ERROR</Level>
      <Code>4.1</Code>
      <Reason>request timed out</Reason>
      <ObjectID>
        <Names>
          <name>M002</name>
        </Names>
        <ObjectType>
          <type>Meter</type>
        </ObjectType>
      </ObjectID>
    </Status>
  </Reply>
</GetResponseMessage>
```

Figure D.25 – Example on-demand meter ping request (response message 3)

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.8.8 Request for an on-demand meter ping (simple acknowledgement of message 3)

The requesting system acknowledges receipt of this reply(MeterReadings) message by means of a simple acknowledgement message as shown in Figure D.26.

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:43-06:00</Timestamp>
    <MessageID>e824635e-802d-49e6-aeec-1b17a173b270</MessageID>
    <CorrelationID>49882980-3254-11e8-b467-0ed5f89f718b</CorrelationID>
    <Comment>Response 3 to a request for an on-demand meter ping (simple
ack)</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.3</Code>
      <Reason>simple ack</Reason>
    </Status>
  </Reply>
</GetResponseMessage>
```

Figure D.26 – Example on-demand meter ping request (simple acknowledgement message 3)

D.9 Messages dealing with historical data request

D.9.1 General

This example illustrates a requesting system that requests historical meter data from a responding system. On receipt of such a request, the responding system retrieves the requested data and returns these to the requesting system.

D.9.2 Sequence diagram

The sequence diagram for this example is shown in Figure D.27.

A request for meter data is formulated as an IEC 61968-9:2013 get(MeterReadings) request message which is sent directly from the requesting system to the responding system (step 1). The responding system then checks the request for errors (step 2) and, assuming no problems are found with the request, retrieves the requested meter data (step 3). It then responds to the requesting system with the requested data in a IEC 61968-9:2013 reply(MeterReadings) message (step 4).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

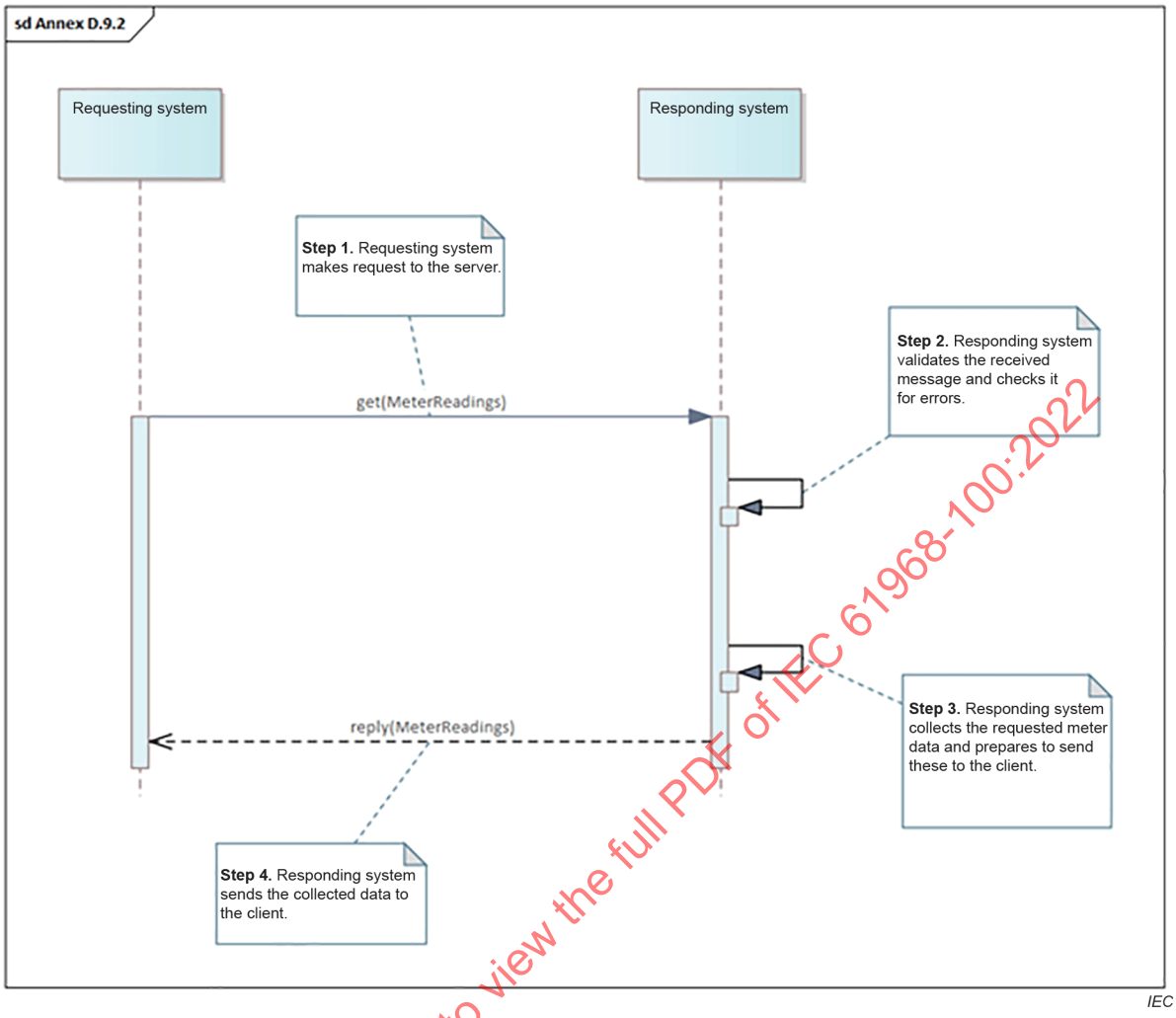


Figure D.27 – Example sequence diagram for the retrieval of historical meter readings

D.9.3 Historical meter read (request message)

The request message (step 1) sent from the requesting system to the responding system is an IEC 61968-9:2013 get(MeterReadings) message as shown in Figure D.28.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:31-06:00</Timestamp>
    <MessageID>1f92de80-b835-4074-b42e-233164063a39</MessageID>
    <CorrelationID>6d6fe897-b588-40a2-ba30-55e4d838b6f4</CorrelationID>
    <Comment>Request for historical meter read data</Comment>
  </Header>
  <Payload>
    <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <EndDevice>
        <Names>
          <name>M001</name>
          <NameType>
            <name>MeterID</name>
          </NameType>
        </Names>
      </EndDevice>
      <ReadingType>
        <Names>
          <!--bulkQuantity forward electricitySecondaryMetered energy (kWh) -
->
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0</name>
        </Names>
      </ReadingType>
      <TimeSchedule>
        <scheduleInterval>
          <end>2018-07-01T10:00:00-06:00</end>
          <start>2018-07-01T09:00:00-06:00</start>
        </scheduleInterval>
      </TimeSchedule>
    </GetMeterReadings>
  </Payload>
</GetRequestMessage>
```

Figure D.28 – Example historical meter read request (request message)

D.9.4 Historical meter read (response message)

The response message from the responding system back to the requesting system (step 4) is an IEC 61968-9:2013 reply(MeterReadings) message as shown in Figure D.29.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:36-06:00</Timestamp>
    <MessageID>5ce4e117-217b-4d43-989a-4af6de460195</MessageID>
    <CorrelationID>6d6fe897-b588-40a2-ba30-55e4d838b6f4</CorrelationID>
    <Comment>Response to a request for historical meter read data</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
  <Payload>
    <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <MeterReading>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <Readings>
          <timeStamp>2018-07-01T09:00:00-06:00</timeStamp>
          <value>1800.0</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
        <Readings>
          <timeStamp>2018-07-01T09:15:00-06:00</timeStamp>
          <value>1820.0</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
        <Readings>
          <timeStamp>2018-07-01T09:30:00-06:00</timeStamp>
          <value>1830.0</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
        <Readings>
          <timeStamp>2018-07-01T09:45:00-06:00</timeStamp>
          <value>1840.0</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
        <Readings>
          <timeStamp>2018-07-01T10:00:00-06:00</timeStamp>
          <value>1850.0</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
      </MeterReading>
    </MeterReadings>
  </Payload>
</GetResponseMessage>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure D.29 – Example historical meter read request (response message)

D.10 Messages dealing with historical data request (responses spread over several messages)

D.10.1 General

This example is similar to that described in Clause D.9. However, in this case, the response data are spread over several messages. This may be because the responding system wants to reduce the size of the response messages or because it does not have all the available information to hand.

Because several response message are transmitted, each one must contain a <Header><MessageSequence> element. This is so that the receiving system can detect when it has received all these messages even if they are delivered out of order (7.5.4.6, 7.5.5.6).

D.10.2 Historical meter read (response message 1 of 4)

The first response message from the responding system back to the requesting system (step 4) is an IEC 61968-9:2013 reply(MeterReadings) message as shown in Figure D.30.

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:36-06:00</Timestamp>
    <MessageID>5d09449a-4c9e-40f9-9618-e2600ca52625</MessageID>
    <CorrelationID>6d6fe897-b588-40a2-ba30-55e4d838b6f4</CorrelationID>
    <MessageSequence>
      <SequenceNumber>1</SequenceNumber>
      <LastMessage>false</LastMessage>
    </MessageSequence>
    <Comment>Response 1 of 4 to a request for historical meter read
data</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
  <Payload>
    <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <MeterReading>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <Readings>
          <timeStamp>2018-07-01T09:00:00-06:00</timeStamp>
          <value>1800.0</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
      </MeterReading>
    </MeterReadings>
  </Payload>
</GetResponseMessage>
```

Figure D.30 – Example historical meter read request (response message 1 of 4)

D.10.3 Historical meter read (response message 2 of 4)

The second response message from the responding system back to the requesting system (step 4) is an IEC 61968-9:2013 reply(MeterReadings) message as shown in Figure D.31.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:38-06:00</Timestamp>
    <MessageID>d301ec91-1ece-4e69-a006-df7c7efc2729</MessageID>
    <CorrelationID>6d6fe897-b588-40a2-ba30-55e4d838b6f4</CorrelationID>
    <MessageSequence>
      <SequenceNumber>2</SequenceNumber>
      <LastMessage>false</LastMessage>
    </MessageSequence>
<Comment>Response 2 of 4 to a request for historical meter read data</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
  <Payload>
    <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#">
      <MeterReading>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <Readings>
          <timeStamp>2018-07-01T09:15:00-06:00</timeStamp>
          <value>1820.0</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
      </MeterReading>
    </MeterReadings>
  </Payload>
</GetResponseMessage>
```

Figure D.31 – Example historical meter read request (response message 2 of 4)

D.10.4 Historical meter read (response message 3 of 4)

The third response message from the responding system back to the requesting system (step 4) is an IEC 61968-9:2013 reply(MeterReadings) message as shown in Figure D.32.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:40-06:00</Timestamp>
    <MessageID>e14abe98-4a04-4060-9119-a1272a1589f9</MessageID>
    <CorrelationID>6d6fe897-b588-40a2-ba30-55e4d838b6f4</CorrelationID>
    <MessageSequence>
      <SequenceNumber>3</SequenceNumber>
      <LastMessage>false</LastMessage>
    </MessageSequence>
<Comment>Response 3 of 4 to a request for historical meter read data</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
  <Payload>
    <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#">
      <MeterReading>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <Readings>
          <timeStamp>2018-07-01T09:30:00-06:00</timeStamp>
          <value>1830.0</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
      </MeterReading>
    </MeterReadings>
  </Payload>
</GetResponseMessage>
```

Figure D.32 – Example historical meter read request (response message 3 of 4)

D.10.5 Historical meter read (response message 4 of 4)

The fourth, and last response message from the responding system back to the requesting system is an IEC 61968-9:2013 reply(MeterReadings) message as shown in Figure D.33. This message indicates that it is the last message in the sequence by the `<Header><MessageSequence><LastMessage>` element being set to true.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T14:54:42-06:00</Timestamp>
    <MessageID>1d45540b-a1cc-4d78-a2b5-9e7d966e3e25</MessageID>
    <CorrelationID>6d6fe897-b588-40a2-ba30-55e4d838b6f4</CorrelationID>
    <MessageSequence>
      <SequenceNumber>4</SequenceNumber>
      <LastMessage>true</LastMessage>
    </MessageSequence>
<Comment>Response 4 of 4 to a request for historical meter read data</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
  <Payload>
    <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#">
      <MeterReading>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <Readings>
          <timeStamp>2018-07-01T09:45:00-06:00</timeStamp>
          <value>1840.0</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
      </MeterReading>
    </MeterReadings>
  </Payload>
</GetResponseMessage>
```

Figure D.33 – Example historical meter read request (response message 4 of 4)

D.11 Messages dealing with multiple statuses within a response

D.11.1 General

This example illustrates how to communicate a mixture of failures and successes within a single response message. Each indication of failure or success is correlated with the appropriate object in the original request message. This example also exercises the cross-product rule (8.4.2).

Upon receiving a request message, the responding system checks it for possible errors. The responding system then processes those parts of the request that are not in error. The responding system responds to the requesting system with a single response message containing both the requested data as well as the error notifications.

This example exercises the cross-product rule (8.4.2). In the sample XML for the request message in Figure D.35, various filtering parameters such as meter name, reading type and time schedule are defined. The cross-product rule defines how the responding system shall interpret such a request when multiple such parameters are specified. In the example, the request deals with three meter names (one of which is non-existent), three reading types (one of which is badly-formed) and one TimeSchedule interval. Because of the cross-product rule only four of these nine combinations are valid.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.11.2 Sequence diagram

The sequence diagram for this example is shown in Figure D.34.

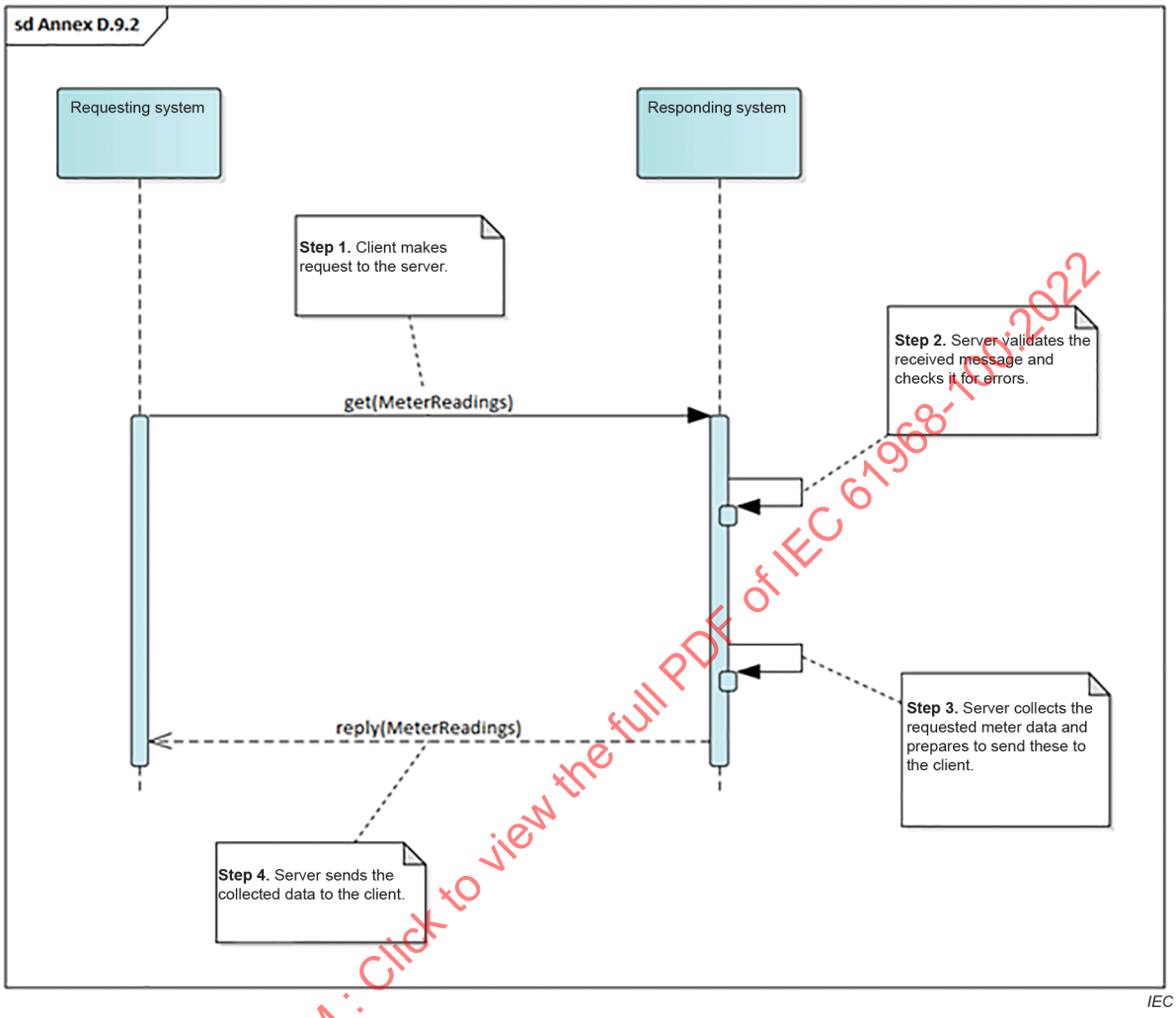


Figure D.34 – Example sequence diagram for the retrieval of historical meter readings (request message)

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

The following is a request for three meters and three reading types.

The request in Figure D.35 is to be interpreted as follows. Return meter readings, if any, for which:

- the meter name is one of M001, M002 or X001 (non-existent)
- and the reading type is one of 0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0 or 0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.73.0 or 0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72 (badly-formed)
- and the reading timestamp lies between 2018-04-01T00:00:00-06:00 and 2018-04-03T00:00:00-06:00.

The request message sent from the requesting system to the responding system (step 1) is an IEC 61968-9:2013 get(MeterReadings) message.

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-04-03T14:16:09-06:00</Timestamp>
    <MessageID>096aa913-98f2-4b92-8077-a486aee0c635</MessageID>
    <CorrelationID>25bbb4a8-9ca9-4314-9753-5d15d580baf0</CorrelationID>
    <Comment>Request for historical meter read data (mixed success and failure
cases)</Comment>
  </Header>
  <Payload>
    <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <EndDevice>
        <!-- valid meter name -->
        <Names>
          <name>M001</name>
        </Names>
      </EndDevice>
      <EndDevice>
        <!-- valid meter name -->
        <Names>
          <name>M002</name>
        </Names>
      </EndDevice>
      <EndDevice>
        <!-- non-existent meter name -->
        <Names>
          <name>X001</name>
        </Names>
      </EndDevice>
      <ReadingType>
        <Names>
<!--bulkQuantity forward electricitySecondaryMetered energy (kWh) -->
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0</name>
        </Names>
      </ReadingType>
      <ReadingType>
        <Names>
<!--bulkQuantity forward electricitySecondaryMetered energy (kVArh) -->
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.73.0</name>
        </Names>
      </ReadingType>
      <ReadingType>
        <Names>
          <!-- badly formed ReadingType -->
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72</name>
        </Names>
      </ReadingType>
      <TimeSchedule>
        <scheduleInterval>
          <end>2018-04-03T00:00:00-06:00</end>
          <start>2018-04-01T00:00:00-06:00</start>
        </scheduleInterval>
      </TimeSchedule>
    </GetMeterReadings>
  </Payload>
</GetRequestMessage>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure D.35 – Example historical meter read request, mixed success and failure cases (request message)

D.11.3 Historical meter read (response message)

The single IEC 61968-9:2013 reply(MeterReadings) response message (step 4) contains the requested data for the valid portions of the request message and error indications for the parts of the request message that cannot be satisfactorily processed (Figure D.36).

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-04-03T14:16:16-06:00</Timestamp>
    <MessageID>485c02d7-5f20-4b84-a2de-0e7759d1b84b</MessageID>
    <CorrelationID>25bbb4a8-9ca9-4314-9753-5d15d580baf0</CorrelationID>
    <Comment>Response to a request for historical meter read data (mixed
success and failure cases)</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>ERROR</Level>
      <Code>2.0</Code>
      <Namespace>http://iec.ch/TC57/2013/61968-9#</Namespace>
      <Reason>no such meter</Reason>
      <ObjectID>
        <Names>
          <name>X001</name>
        </Names>
        <ObjectType>
          <type>MeterID</type>
        </ObjectType>
      </ObjectID>
    </Status>
    <Status>
      <Level>ERROR</Level>
      <Code>2.6</Code>
      <Namespace>http://iec.ch/TC57/2013/61968-9#</Namespace>
      <Reason>invalid ReadingType</Reason>
      <ObjectID>
        <Names>
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72</name>
        </Names>
        <ObjectType>
          <type>ReadingType</type>
        </ObjectType>
      </ObjectID>
    </Status>
  </Reply>
  <Payload>
    <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <MeterReading>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <Readings>
          <timeStamp>2018-04-02T14:16:12-06:00</timeStamp>
          <value>2012.1</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
        <Readings>
          <timeStamp>2018-04-02T14:16:13-06:00</timeStamp>
          <value>606.82</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.73.0"/>
        </Readings>
      </MeterReading>
      <MeterReading>
        <Meter>
          <mRID>83009321-7883-4ea6-bc0e-71ddcd274f69</mRID>
          <Names>
            <name>M002</name>
          </Names>
        </Meter>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        <Readings>
          <timeStamp>2018-04-02T14:16:14-06:00</timeStamp>
          <value>15948.55</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </Readings>
        <Readings>
          <timeStamp>2018-04-02T14:16:11-06:00</timeStamp>
          <value>1305.19</value>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.73.0"/>
        </Readings>
      </MeterReading>
    </MeterReadings>
  </Payload>
</GetResponseMessage>
```

Figure D.36 – Example historical meter read request, mixed success and failure cases (response message)

D.12 Messages dealing with a consequential event message

D.12.1 General

This example illustrates a requesting system that sends a "meter connect" command to a responding system such as a head-end system. The latter checks the meter connect command for validity and sends a response message to the requesting system containing the appropriate status notifications. Assuming the command appears to be valid, the responding system then attempts to carry out the command and sends an asynchronous notification to the requesting system once this has completed. The message exchange pattern for this case therefore consists of a combination of a request/response message exchange followed by a consequential event message (4.7.4),

D.12.2 Sequence diagram

The sequence diagram for this example is shown in Figure D.37.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

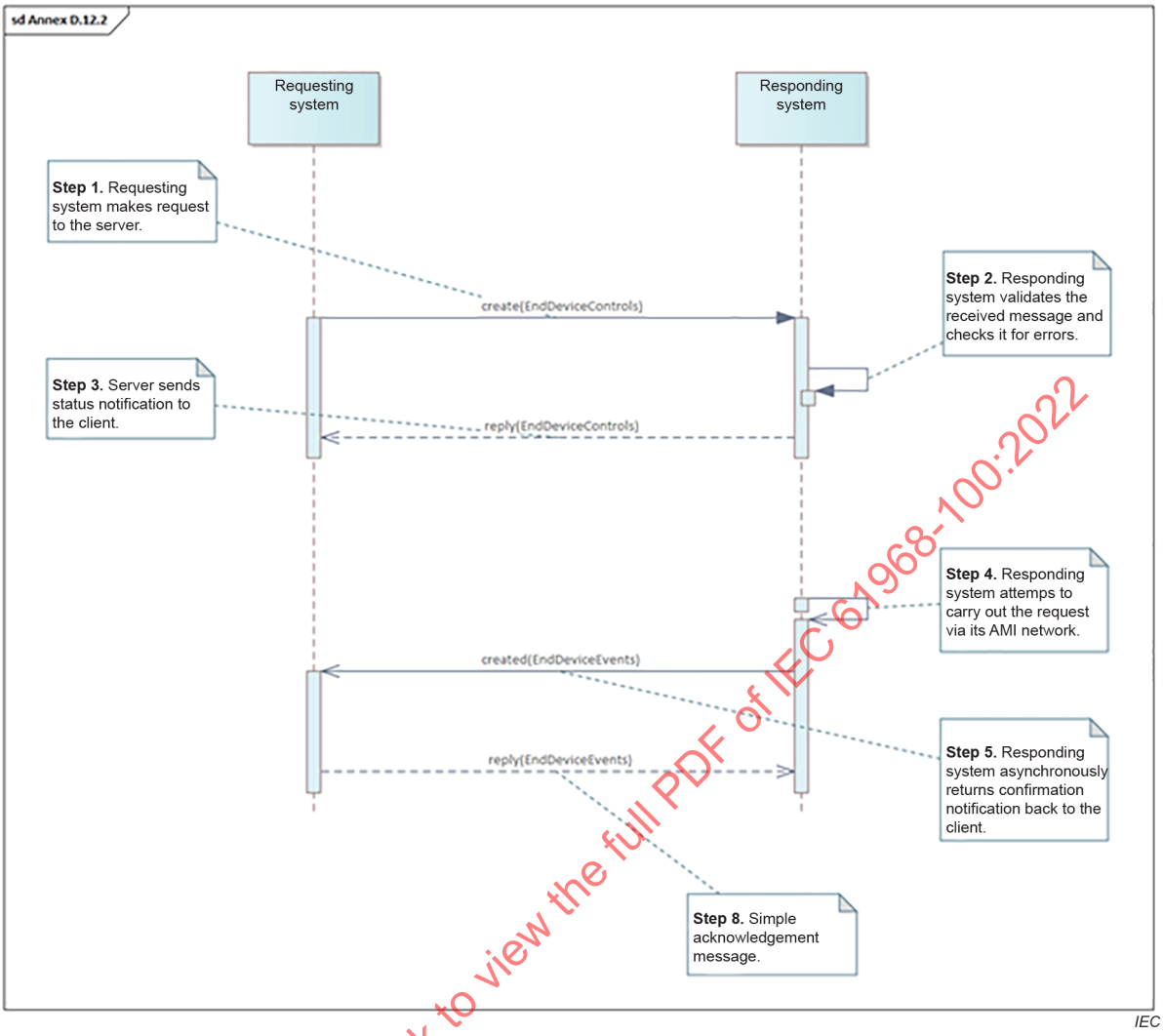


Figure D.37 – Example sequence diagram for the requesting a meter connect

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

In this example, it is to be assumed that a head-end system (HES) takes on the role of the responding system. A meter connect request is formulated as a create(EndDeviceControls) message which is sent directly from the requesting system to the HES (step 1). The HES checks this for errors (step 2) and responds with a reply(EndDeviceControls) message (step 3). In this example, we are sending EndDeviceControls for two devices, one of which does not exist in the HES. The HES sends a response message to identify the erroneous EndDevice and indicates that the command for the valid EndDevice was accepted. The HES then sends out the corresponding command over its AMI network to the valid EndDevice (step 4).

Once the meter has executed this AMI command and notified the HES appropriately, the HES then sends an asynchronous notification in the form of a created(EndDeviceEvents) message back to the original requestor, at the ReplyAddress specified in the request message header, (step 5) indicating that the connect operation succeeded.

D.12.3 Request for a meter connect (request message)

The request message sent from the requesting system to the responding system (step 1) is a IEC 61968-9:2013 create(EndDeviceControls) message as shown in Figure D.38. The EndDeviceControls type for a remote meter connect is `3.31.0.18`.

```
<CreateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>EndDeviceControls</Noun>
    <Timestamp>2018-08-01T09:26:32-06:00</Timestamp>
    <MessageID>87b4185b-fff5-45e3-8146-bbb7d80beff6</MessageID>
    <CorrelationID>ae2f7ab4-52cb-47b6-8bad-a64e296d417f</CorrelationID>
<ReplyAddress>http://www.myclient.test/ExecuteEndDeviceControls</ReplyAddress>
    <Comment>Request to connect  meters</Comment>
  </Header>
  <Payload>
    <EndDeviceControls xmlns="http://iec.ch/TC57/2011/EndDeviceControls#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <EndDeviceControl>
        <EndDeviceControlType ref="3.31.0.18"/>
        <EndDevices>
          <Names>
            <name>M001</name>
          </Names>
        </EndDevices>
        <EndDevices>
          <!-- non-existent meter name -->
          <Names>
            <name>X001</name>
          </Names>
        </EndDevices>
      </EndDeviceControl>
    </EndDeviceControls>
  </Payload>
</CreateRequestMessage>
```

Figure D.38 – Example meter connect (request message)

D.12.4 Request for a meter connect (response message)

The response message sent back from the responding system to the requesting system is a IEC 61968-9:2013 reply(EndDeviceControls) message as shown in Figure D.39.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>EndDeviceControls</Noun>
    <Timestamp>2018-08-01T09:26:34-06:00</Timestamp>
    <MessageID>f4c698bc-3241-11e8-b467-0ed5f89f718b</MessageID>
    <CorrelationID>ae2f7ab4-52cb-47b6-8bad-a64e296d417f</CorrelationID>
    <Comment>Response to a request to connect meters</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
      <ObjectID>
        <Names>
          <name>M001</name>
        </Names>
        <ObjectType>
          <type>Meter</type>
        </ObjectType>
      </ObjectID>
    </Status>
    <Status>
      <Level>ERROR</Level>
      <Code>5.3</Code>
      <Reason>Referenced object does not exist</Reason>
      <ObjectID>
        <Names>
          <name>X001</name>
        </Names>
        <ObjectType>
          <type>EndDevice</type>
        </ObjectType>
      </ObjectID>
    </Status>
  </Reply>
</CreateResponseMessage>
```

Figure D.39 – Example meter connect (response message)

D.12.5 Request for a meter connect (event message)

The event message sent back from the responding system to the requesting system (step 5) is a IEC 61968-9:2013 created(EndDeviceEvents) message as shown in Figure D.40. The EndDeviceEventType code for a successful remote meter connect is `3.31.0.42`.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreatedEventMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>created</Verb>
    <Noun>EndDeviceEvents</Noun>
    <Timestamp>2018-08-01T09:26:38-06:00</Timestamp>
    <MessageID>14127fe8-4546-48d6-95dd-894a5f57e74b</MessageID>
    <CorrelationID>ae2f7ab4-52cb-47b6-8bad-a64e296d417f</CorrelationID>
    <Comment>Publication of a meter event by means of an unsolicited event
message</Comment>
  </Header>
  <Payload>
    <EndDeviceEvents xmlns="http://iec.ch/TC57/2011/EndDeviceEvents#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
      <EndDeviceEvent>
        <createdDateTime>2018-08-01T09:26:36-06:00</createdDateTime>
        <Assets>
          <Names>
            <name>M001</name>
          </Names>
        </Assets>
        <!-- remote meter connect succeeded -->
        <EndDeviceEventType ref="3.31.0.42"/>
      </EndDeviceEvent>
    </EndDeviceEvents>
  </Payload>
</CreatedEventMessage>
```

Figure D.40 – Example meter connect (event message)

D.12.6 Request for a meter connect (simple acknowledgement message)

After sending the created(EndDeviceEvents) message (step 5), the responding system then waits until the requesting system confirms receipt by means of a simple acknowledgement message (step 8) as shown in Figure D.41.

```
<CreatedEventResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>EndDeviceEvents</Noun>
    <Timestamp>2018-08-01T09:26:39-06:00</Timestamp>
    <MessageID>e128980f-3267-4457-8d06-bb97e500e992</MessageID>
    <CorrelationID>ae2f7ab4-52cb-47b6-8bad-a64e296d417f</CorrelationID>
    <Comment>Publication of a meter event by means of an unsolicited event
message (simple ack)</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.3</Code>
      <Reason>simple ack</Reason>
    </Status>
  </Reply>
</CreatedEventResponseMessage>
```

Figure D.41 – Example meter connect (simple acknowledgement message)

D.13 Using two operationsets to sequence multiple operations with transactional integrity

D.13.1 General

This example illustrates how an OperationSet may be used to sequence a set of operations and to maintain transactional integrity across those operations.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.13.2 Using two OperationSets to read two meters before disconnecting them (request message)

Figure D.42 represents a combination of messages used to read meters before disconnecting them. Each meter read/disconnect command is placed into its own OperationSet. For each meter read/disconnect OperationSet, if the meter read operation fails for any reason, the corresponding disconnect operation will not be carried out. Each meter read/disconnect OperationSet, however, operates independently of the other OperationSet.

This example illustrates the OperationSet request and response messages. The disconnect operations will result in consequential example messages. However, these are not shown in this example. See the examples in Clause D.13.

```
<ExecuteRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>execute</Verb>
    <Noun>OperationSet</Noun>
    <Timestamp>2018-08-01T09:26:32-06:00</Timestamp>
    <MessageID>87b4185b-fff5-45e3-8146-bbb7d80beff6</MessageID>
    <CorrelationID>ae2f7ab4-52cb-47b6-8bad-a64e296d417f</CorrelationID>

<ReplyAddress>http://www.myclient.test/ExecuteEndDeviceControls</ReplyAddress>
    <Comment>Using an OperationSet to Read  and Disconnect Multiple
Meters</Comment>
  </Header>
  <Payload>
    <OperationSet>
      <OperationSetID>63c4cc52-38bc-4b45-9320-57d3863c2893</OperationSetID>
      <EnforceMsgSequence>true</EnforceMsgSequence>
      <EnforceTransactionalIntegrity>true</EnforceTransactionalIntegrity>
      <Operation>
        <OperationID>1</OperationID>
        <Verb>get</Verb>
        <Noun>MeterReadings</Noun>
        <Comment>Request for an on-demand meter read</Comment>
        <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <EndDevice>
            <!-- name of an existing meter -->
              <Names>
                <name>M001</name>
              </Names>
          </EndDevice>
          <ReadingType>
            <Names>
            <!--bulkQuantity forward electricitySecondaryMetered energy
(kWh) -->
              <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0</name>
            </Names>
          </ReadingType>
        </GetMeterReadings>
      </Operation>
      <Operation>
        <OperationID>2</OperationID>
        <Verb>create</Verb>
        <Noun>EndDeviceControls</Noun>
        <EndDeviceControls xmlns="http://iec.ch/TC57/2011/EndDeviceControls#"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <EndDeviceControl>
            <EndDeviceControlType ref="3.31.0.18"/>
            <EndDevices>
              <Names>
                <name>M001</name>
              </Names>
            </EndDevices>
          </EndDeviceControl>
        </EndDeviceControls>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      </Operation>
    </OperationSet>
    <OperationSet>
      <OperationSetID>4b9c6928-e974-4b0f-b715-db1b9283ddc8</OperationSetID>
      <EnforceMsgSequence>true</EnforceMsgSequence>
      <EnforceTransactionalIntegrity>true</EnforceTransactionalIntegrity>
      <Operation>
        <OperationID>1</OperationID>
        <Verb>get</Verb>
        <Noun>MeterReadings</Noun>
        <Comment>Request for an on-demand meter read</Comment>
        <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <EndDevice>
            <!-- name of an existing meter -->
              <Names>
                <name>M002</name>
              </Names>
          </EndDevice>
          <ReadingType>
            <Names>
              <!--bulkQuantity forward electricitySecondaryMetered energy
(kWh) -->
              <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0</name>
            </Names>
          </ReadingType>
        </GetMeterReadings>
      </Operation>
      <Operation>
        <OperationID>2</OperationID>
        <Verb>create</Verb>
        <Noun>EndDeviceControls</Noun>
        <Comment>Request for a meter disconnect operation</Comment>
        <EndDeviceControls xmlns="http://iec.ch/TC57/2011/EndDeviceControls#"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <EndDeviceControl>
            <EndDeviceControlType ref="3.31.0.23"/>
            <EndDevices>
              <Names>
                <name>M002</name>
              </Names>
            </EndDevices>
          </EndDeviceControl>
        </EndDeviceControls>
      </Operation>
    </OperationSet>
  </Payload>
</ExecuteRequestMessage>
```

Figure D.42 – Using two OperationSets to read two meters before disconnecting them (request message)

D.13.3 Using two OperationSets to read two meters before disconnecting them (response message)

Figure D.43 shows an example response to the request message of Figure D.42. It illustrates how an OperationSet in a response message may be used to convey the corresponding results of that request.

```
<ExecuteResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>OperationSet</Noun>
    <Timestamp>2018-10-02T14:16:44-06:00</Timestamp>
    <MessageID>d5f7b4d3-259c-44e9-82d6-081d57a2ea64</MessageID>
    <CorrelationID>ae2f7ab4-52cb-47b6-8bad-a64e296d417f</CorrelationID>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <Comment>Response to combined request for on-demand meter read data and
disconnect</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
  <Payload>
    <OperationSet>
      <OperationSetID>63c4cc52-38bc-4b45-9320-57d3863c2893</OperationSetID>
      <Operation>
        <OperationID>1</OperationID>
        <Verb>reply</Verb>
        <Noun>MeterReadings</Noun>
        <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <MeterReading>
          <Meter>
            <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
            <Names>
            <name>M001</name>
            </Names>
          </Meter>
          <Readings>
            <timeStamp>2018-10-02T14:16:12-06:00</timeStamp>
            <value>2012.1</value>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
          </Readings>
          </MeterReading>
        </MeterReadings>
      </Operation>
    </OperationSet>
    <OperationSet>
      <OperationSetID>4b9c6928-e974-4b0f-b715-db1b9283ddc8</OperationSetID>
      <Operation>
        <OperationID>1</OperationID>
        <Verb>reply</Verb>
        <Noun>MeterReadings</Noun>
        <MeterReadings xmlns="http://iec.ch/TC57/2011/MeterReadings#"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <MeterReading>
          <Meter>
            <mRID>83009321-7883-4ea6-bc0e-71ddcd274f69</mRID>
            <Names>
            <name>M002</name>
            </Names>
          </Meter>
          <Readings>
            <timeStamp>2018-10-02T14:16:13-06:00</timeStamp>
            <value>2013.1</value>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
          </Readings>
          </MeterReading>
        </MeterReadings>
      </Operation>
    </OperationSet>
  </Payload>
</ExecuteResponseMessage>
```

Figure D.43 – Using two OperationSets to Read two meters before disconnecting them (response message)

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

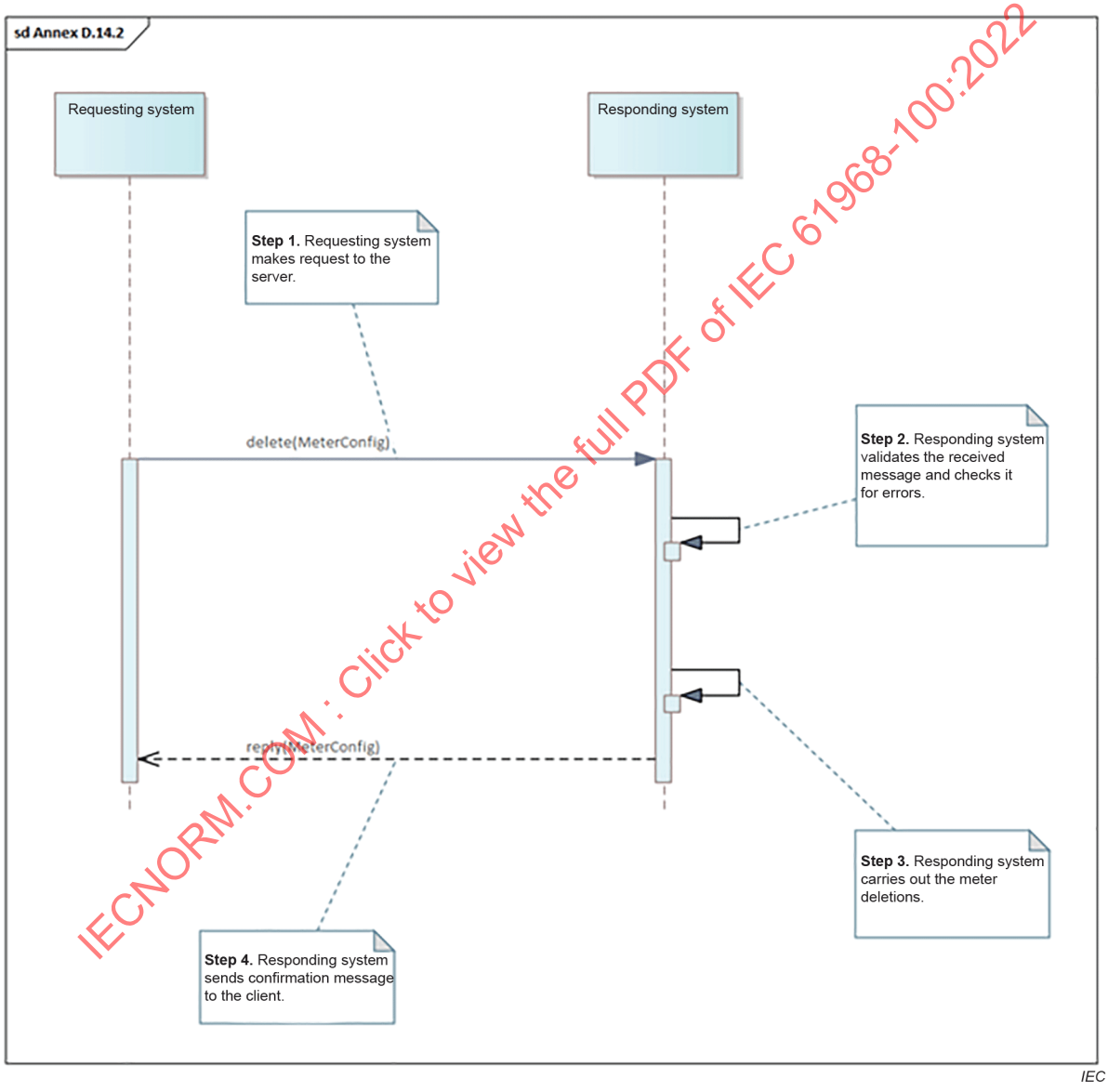
D.14 Messages dealing with object deletion

D.14.1 General

This example illustrates the deletion of a single meter from a responding system-side system. This is an action that would typically be taken when a meter is to be permanently removed from the responding system.

D.14.2 Sequence diagram

The sequence diagram for this example is shown in Figure D.44.



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure D.44 – Example sequence diagram for the deletion of a meter

A request to remove a meter is formulated as an IEC 61968-9:2013 delete(MeterConfig) request message which is sent directly from the requesting system to the responding system (step 1). The responding system then checks the request for errors (step 2) and, assuming no problems are found with the request, carries out the requested operation to remove the meter (step 3). It then responds to the requesting system with a confirmation message in an IEC 61968-9:2013 reply(MeterConfig) message (step 4).

D.14.3 Request to delete a meter (request message)

The request message from the requesting system to the responding system (step 1) is an IEC 61968-9:2013 delete(MeterConfig) message as shown in Figure D.45.

```
<DeleteRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>delete</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-11-18T10:30:32-06:00</Timestamp>
    <MessageID>525ef66f-9397-409a-840c-998f0a07b20b</MessageID>
    <CorrelationID>83159e6e-77df-40b0-90c5-58ad7e5dadfd</CorrelationID>
    <Comment>Request to delete a meter</Comment>
  </Header>
  <Payload>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Meter>
        <Names>
          <name>M001</name>
        </Names>
      </Meter>
    </MeterConfig>
  </Payload>
</DeleteRequestMessage>
```

Figure D.45 – Example meter deletion (request message)

D.14.4 Request to delete a meter (response message)

The response message from the responding system to the requesting system (step 4) is a IEC 61968-9:2013 reply(MeterConfig) message as shown in Figure D.46.

```
<DeleteResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-11-18T10:30:38-06:00</Timestamp>
    <MessageID>376b8f71-0e88-4f96-8f13-501bf385e5d7</MessageID>
    <CorrelationID>83159e6e-77df-40b0-90c5-58ad7e5dadfd</CorrelationID>
    <Comment>Response to a request to delete a meter</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
</DeleteResponseMessage>
```

Figure D.46 – Example meter deletion (response message)

D.15 Messages dealing with cancellation of a single transaction

D.15.1 Meter control request message

The example shown in Figure D.47 is similar to that shown in D.12.3. However in this case the meters (end devices) are identified by mRIDs instead of names. An mRID is also used to label the entire EndDeviceControl request.

The associated message exchange pattern is described in 8.5.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>EndDeviceControls</Noun>
    <Timestamp>2013-10-14T11:31:55Z</Timestamp>
    <MessageID>7c3bd408-c6df-4943-9ee7-38a326b83cca</MessageID>
    <CorrelationID>7c843d20-b47a-444a-90b3-3a23b6c52eae</CorrelationID>
    <ReplyAddress>http://localhost:8090/foobar</ReplyAddress>
  </Header>
  <Payload>
    <EndDeviceControls xmlns="http://iec.ch/TC57/2011/EndDeviceControls#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <EndDeviceControl>
        <mRID>4006A84F-F6B4-4F5C-8422-08D9539CE935</mRID>
        <EndDeviceControlType ref="3.31.0.23"/>
        <EndDevices>
          <mRID>73EF9D52-95B9-4840-B4FF-28D1C4BF675B</mRID>
        </EndDevices>
        <EndDevices>
          <mRID>9D175B1B-98A1-4011-BD75-DF270602E61A</mRID>
        </EndDevices>
        <EndDevices>
          <mRID>3375B9A1-BF9F-4883-A1D4-A7D1EBFB7666</mRID>
        </EndDevices>
        <EndDevices>
          <mRID>047400D3-C0A6-4BBF-9163-9ACE4D8698D2</mRID>
        </EndDevices>
        <primaryDeviceTiming>
          <interval>
            <end>2018-10-31T00:00:00Z</end>
            <start>2018-10-14T01:01:00Z</start>
          </interval>
        </primaryDeviceTiming>
      </EndDeviceControl>
    </EndDeviceControls>
  </Payload>
</CreateRequestMessage>
```

Figure D.47 – Example create(EndDeviceControls) request message

D.15.2 Response message containing a single transaction id

The example shown in Figure D.48 shows how a response message may return a transaction ID back to a requesting system for use in a subsequent cancellation message. The message depicted in Figure D.48 illustrates an example response to the request message shown in Figure D.47.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>EndDeviceControls</Noun>
    <Timestamp>2012-01-17T09:30:47Z</Timestamp>
    <MessageID>C8D4E105-205A-460A-ADE3-D3166746A0E9</MessageID>
    <CorrelationID>2CCE4A71-EF2D-4D33-A51E-124DB274861B</CorrelationID>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
    <Transaction>
      <TransactionID>8e792df5-afe0-46e3-a20f-3c28a063725a</TransactionID>
      <ObjectID>
        <mRID>4006A84F-F6B4-4F5C-8422-08D9539CE935</mRID>
        <ObjectType>EndDeviceControl</ObjectType>
      </ObjectID>
    </Transaction>
  </Reply>
</CreateResponseMessage>
```

Figure D.48 – Example response message containing a single transaction ID

D.15.3 Cancellation request (request message)

Figure D.49 illustrates how a cancellation request message may be made using the previously returned transaction ID. In this example, the cancel request message causes all pending requests associated with the original request message (Figure D.47) to be cancelled. The transaction ID is obtained from a previous response message such as that shown in Figure D.48.

```
<CancelRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>cancel</Verb>
    <Noun>EndDeviceControls</Noun>
    <Timestamp>2012-01-17T09:30:47Z</Timestamp>
    <MessageID>8B3EF3E8-C61C-4C91-BEF0-A1775570656A</MessageID>
    <CorrelationID>337887b5-f5d3-40d1-a0be-ee3d39bb64f8</CorrelationID>
  </Header>
  <Payload>
    <Transaction>
      <TransactionID>8e792df5-afe0-46e3-a20f-3c28a063725a</TransactionID>
    </Transaction>
  </Payload>
</CancelRequestMessage>
```

Figure D.49 – Example cancellation request message

D.15.4 Cancellation request (response message)

The response message from the responding system to the requesting system for the cancellation request is a reply(EndDeviceControls) message as shown in Figure D.50.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CancelResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>EndDeviceControls</Noun>
    <Timestamp>2012-01-17T09:31:04Z</Timestamp>
    <MessageID>24081e6c-ccbd-437b-8558-27811ef1cad1</MessageID>
    <CorrelationID>337887b5-f5d3-40d1-a0be-ee3d39bb64f8</CorrelationID>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
</CancelResponseMessage>
```

Figure D.50 – Example response to a cancellation request message

D.16 Messages dealing with cancellation of multiple transactions

D.16.1 Response message containing multiple transaction IDs

Instead of returning a response message containing a single transaction ID as shown in Figure D.48, the response message may return multiple transaction IDs back to a requesting system for use in a subsequent cancellation message. Such an example is shown in Figure D.51. Transaction IDs are returned for the entire end device control as well as for each end device referenced in the original request message (Figure D.47). The transaction IDs are distinguished according to which named objects they apply.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>EndDeviceControls</Noun>
    <Timestamp>2013-10-14T11:34:16Z</Timestamp>
    <MessageID>25b6e420-06b2-4318-a496-cb184ebc0091</MessageID>
    <CorrelationID>7c843d20-b47a-444a-90b3-3a23b6c52eae</CorrelationID>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
    <Transaction>
      <TransactionID>8e792df5-afe0-46e3-a20f-3c28a063725a</TransactionID>
      <ObjectID>
        <mRID>4006A84F-F6B4-4F5C-8422-08D9539CE935</mRID>
        <ObjectType>EndDeviceControl</ObjectType>
      </ObjectID>
    </Transaction>
    <Transaction>
      <TransactionID>18576e4a-1046-4952-9f13-9914cc6eb78f</TransactionID>
      <ObjectID>
        <mRID>73EF9D52-95B9-4840-B4FF-28D1C4BF675B</mRID>
        <ObjectType>EndDevice</ObjectType>
      </ObjectID>
    </Transaction>
    <Transaction>
      <TransactionID>d8935070-ed05-4c0e-b226-dc3454180d14</TransactionID>
      <ObjectID>
        <mRID>9D175B1B-98A1-4011-BD75-DF270602E61A</mRID>
        <ObjectType>EndDevice</ObjectType>
      </ObjectID>
    </Transaction>
    <Transaction>
      <TransactionID>986ea61a-0838-4fe0-9e7e-34850c8f1a4f</TransactionID>
      <ObjectID>
        <mRID>3375B9A1-BF9F-4883-A1D4-A7D1EBFB7666</mRID>
        <ObjectType>EndDevice</ObjectType>
      </ObjectID>
    </Transaction>
    <Transaction>
      <TransactionID>502537e2-2838-4daa-beca-4e522be57dbd</TransactionID>
      <ObjectID>
        <mRID>047400D3-C0A6-4BBF-9163-9ACE4D8698D2</mRID>
        <ObjectType>EndDevice</ObjectType>
      </ObjectID>
    </Transaction>
  </Reply>
</CreateResponseMessage>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure D.51 – Example response message containing several transaction IDs

D.16.2 Cancellation request (request message)

Figure D.52 illustrates how a cancellation request message may be made using previously returned transaction IDs. The transaction IDs are obtained from a previous response message such as that shown in Figure D.51. In this example only 3 of the initial end device controls are being cancelled such that the individual transaction IDs for the end devices are being used and not the transaction ID of the end device control.

```
<CancelRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>cancel</Verb>
    <Noun>EndDeviceControls</Noun>
    <Timestamp>2013-10-14T11:38:28Z</Timestamp>
    <MessageID>ac909114-d0da-449a-8d7a-21c127e1aafa</MessageID>
    <CorrelationID>fd881710-fb95-4ec9-a367-770478131138</CorrelationID>
  </Header>
  <Payload>
    <Transaction>
      <TransactionID>18576e4a-1046-4952-9f13-9914cc6eb78f</TransactionID>
    </Transaction>
    <Transaction>
      <TransactionID>d8935070-ed05-4c0e-b226-dc3454180d14</TransactionID>
    </Transaction>
    <Transaction>
      <TransactionID>986ea61a-0833-4fe0-9e7e-34850c8f1a4f</TransactionID>
    </Transaction>
  </Payload>
</CancelRequestMessage>
```

Figure D.52 – Example cancellation request message

D.16.3 Cancellation request (response message)

The response message from the responding system to the requesting system for the cancellation request is a reply(EndDeviceControls) message as shown in Figure D.53. Of the three cancellation requests, one was not able to be completed successfully as the command was already sent to the meter.

```
<CancelResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>EndDeviceControls</Noun>
    <Timestamp>2013-10-14T11:38:30Z</Timestamp>
    <MessageID>24081e6c-ccbd-437b-8558-27811ef1cad1</MessageID>
    <CorrelationID>fd881710-fb95-4ec9-a367-770478131138</CorrelationID>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
      <TransactionID>18576e4a-1046-4952-9f13-9914cc6eb78f</TransactionID>
      <TransactionID>d8935070-ed05-4c0e-b226-dc3454180d14</TransactionID>
    </Status>
    <Status>
      <Level>ERROR</Level>
      <Code>5.1</Code>
      <Reason>Unable to cancel transaction. Command already sent to
device.</Reason>
      <TransactionID>986ea61a-0833-4fe0-9e7e-34850c8f1a4f</TransactionID>
    </Status>
  </Reply>
</CancelResponseMessage>
```

Figure D.53 – Example response to a cancellation request message

D.17 Messages dealing with failed XSD validation

D.17.1 General

This example illustrates the scenario where an application receives a request that, while being a well-formed XML document, fails XSD validation.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

In such a case, the responding system should return as much useful information as possible to the requesting system. In particular, it should include a response message containing a `<Reply><Error><Code>` of `1.8`. If possible, it should also include more details in the `<Reply><Error><Reason>` element.

The responding system should also attempt to extract the `<Header><CorrelationID>` from the original request message and include this in the response message.

D.17.2 Message that fails XSD validation (request message)

The request message sent from the responding system to the requesting system (step 1) is an IEC 61968-9:2013 get(MeterReadings) message. It contains a deliberate error that prevents it from being validated. This is illustrated in Figure D.54.

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T18:16:00-06:00</Timestamp>
    <MessageID>b96122b3-91c7-429c-b1ea-fb430413dfb1</MessageID>
    <CorrelationID>eb56785e-95dd-4f78-89bb-a2988b977f5c</CorrelationID>
    <Comment>Request that fails XSD validation</Comment>
  </Header>
  <Payload>
<!-- Element <XGetMeterReadings> is not allowed under element <Payload> -->
    <XGetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <EndDevice>
        <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
      </EndDevice>
      <ReadingType>
        <Names>
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0</name>
        </Names>
      </ReadingType>
    </XGetMeterReadings>
  </Payload>
</GetRequestMessage>
```

Figure D.54 – Example message that fails XSD validation (request message)

D.17.3 Message that fails XSD validation (response message)

The response message from the responding system to the requesting system (step 3) is an IEC 61968-9:2013 reply(MeterReadings) message as shown in Figure D.55.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T18:16:02-06:00</Timestamp>
    <MessageID>b48e2a76-572e-4d44-b79c-684f78522a3b</MessageID>
    <CorrelationID>eb56785e-95dd-4f78-89bb-a2988b977f52</CorrelationID>
    <Comment>Response to a request that fails XSD validation</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>ERROR</Level>
      <Code>1.2</Code>
      <Reason>Format of request does not validate against schema: Line 14:
Element &lt;XMeterReadings&gt; is not allowed under element &lt;Payload
&gt;</Reason>
    </Status>
  </Reply>
</GetResponseMessage>
```

Figure D.55 – Example message that fails XSD validation (response message)

D.18 Messages dealing with malformed requests

D.18.1 General

This example illustrates the scenario where an application receives a request that is not a syntactically well-formed XML message. This prevents the responding system from parsing the message.

In such a case, the responding system should return a fault message to the requesting system. A fault message contains no `<Header>` but instead only a `<Reply>` element

D.18.2 Message that is malformed (request message)

The request message sent from the responding system to the requesting system (step 1) is an IEC 61968-9:2013 get(MeterReadings) message as shown in Figure D.56. It contains a deliberate error.

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2018-10-02T18:19:00-06:00</Timestamp>
    <MessageID>3f7ce710-eae8-42da-b51a-1edf32b41f40</MessageID>
    <CorrelationID>4c919335-7b34-48a0-a93d-3388ca2701de</CorrelationID>
    <Comment>Malformed request message</Comment>
  </Header>
  <Payload>
    <!-- Missing '>' on the next line -->
    <GetMeterReadings
      <EndDevice>
        <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
      </EndDevice>
      <ReadingType>
        <Names>
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0</name>
        </Names>
      </ReadingType>
    </GetMeterReadings>
  </Payload>
</GetRequestMessage>
```

Figure D.56 – Example malformed message (request message)

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.18.3 Message that is malformed (response message)

The response message from the responding system to the requesting system (step 3) is an IEC 61968-100 fault message as shown in Figure D.57.

```
<FaultMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Reply>
    <Status>
      <Level>ERROR</Level>
      <Code>1.4</Code>
      <Reason>request is malformed</Reason>
    </Status>
  </Reply>
</FaultMessage>
```

Figure D.57 – Example malformed message (response message)

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annex E
(informative)

Example IEC 61968-100:2021 messages for updating objects

E.1 General

The example messages in this annex illustrate the use of this document in the context of sending and receiving IEC 61968-9:2013 messages for the purposes of updating server-side objects. By "updating" is meant making incremental changes to these objects as opposed to instantiating, replacing or deleting them in their entirety (8.3.5).

Similar message patterns and examples may also be applied to other IEC 61868 Parts.

The examples in this annex start with a simple meter creation and then build from there. They include the addition, removal, and modification of parts of that meter object.

Some common aspects across these examples and which are not mentioned in each description are as follows:

- Identification of the object being modified is done by using the profile and specifying the mRID
- Except for example in Clause E.26, the use of OperationSet is not employed. The examples, for the most part, employ a single modification on a single object and the options for transactional integrity and enforcement of order of operations are not needed.

E.2 CreateMeterConfig.xml

The example shown in Figure E.1 illustrates the creation of a single meter. It forms the basis for the subsequent examples in this annex and which will make modifications to this meter. The meter, named M001, has a form factor of 2S, an end device function of SEDF1 with a program ID of ER017A, a CT Ratio of 200.5, a locked meter seal and a single register.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateRequestMessage>
  <Header>
    <Verb>create</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:10-06:00</Timestamp>
    <MessageID>fbb1c893-83c9-47b2-8cca-b8f8859315e6</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a169-ce38fba5d40f</CorrelationID>
  </Header>
  <Payload>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
      <Meter>
        <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
        <formNumber>2S</formNumber>
        <SimpleEndDeviceFunction ref="SEDF1"/>
        <MeterMultipliers>
          <kind>ctRatio</kind>
          <value>200.5</value>
        </MeterMultipliers>
        <Names>
          <name>M001</name>
        </Names>
        <Seals>
          <condition>locked</condition>
          <sealNumber>1</sealNumber>
        </Seals>
      </Meter>
      <SimpleEndDeviceFunction>
        <programID>ER017A</programID>
        <Names>
          <name>SEDF1</name>
        </Names>
        <Registers>
          <Channels>
            <Names>
              <name>CH1</name>
            </Names>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
          <Names>
            <name>REG1</name>
          </Names>
        </Registers>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </Payload>
</CreateRequestMessage>
```

Figure E.1 – CreateMeterConfig.xml

E.3 ModifyFormNumber.xml

The example shown in Figure E.2 illustrates the modification of the form number of the original meter from a 2S to a 12S. The remainder of the details of the meter remain the same. This example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `modify`. There is only one update action with ID of '1'.

The use case for this sample might be the correction of an error on the initial creation of the meter.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:40-06:00</Timestamp>
    <MessageID>fbb1c893-83c9-47b2-8cca-b8f8859315a7</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a169-ce38fba5d4a7</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>modify</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <formNumber>12S</formNumber>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.2 – ModifyFormNumber.xml

E.4 RemoveFormNumber.xml

The example shown in Figure E.3 illustrates the removal of the form number of '12S' from the meter. The remainder of the details of the meter remain the same. This example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `remove`. There is only one update action with ID of '1'.

This example demonstrates the removal of a simple leaf element as opposed to the complex element removal shown in Clause E.9.

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:25:30-06:00</Timestamp>
    <MessageID>fbb1c893-83c9-47b2-8cca-b8f8859315b8</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a169-ce38fba5d4b8</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>remove</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <formNumber>12S</formNumber>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.3 – RemoveFormNumber.xml

E.5 AddMeterMultiplier.xml

The example shown in Figure E.4 illustrates the addition of a meter multiplier to the meter. The multiplier being added is of kind 'kE' for pulse weight, with a value of 7. The other meter details remain the same. This example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `add`. There is only one update action with ID of '1'.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

The use case for this example incorporates the additional information being provided about the meter after the initial creation.

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>fbb1c893-83c9-47b2-8fef-b8f8859222b6</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a877-ce38fba5e12f</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>add</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <MeterMultipliers>
            <mRID>3c67f555-3119-1112-befc-0ed5f89f44cc</mRID>
            <kind>kE</kind>
            <value>7</value>
          </MeterMultipliers>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.4 – AddMeterMultiplier.xml

E.6 ModifyMeterMultiplierKind.xml

The example shown in Figure E.5 illustrates changing the ‘kind’ for the meter multiplier with the specified mRID. The original meter multiplier had a kind of ‘ctRatio’, and this example changes it to kH. The other meter details, including the value of the meter multiplier, remain the same. This example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `modify`. There is only one update action with ID of ‘1’. The mRID of the multiplier is required to be specified, which would be obtained by a get(MeterConfig) request as shown in the example in Clause D.4.

The use case for this example is the correction of erroneous information in the original meter creation.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>637b4fad-df81-482c-bb8b-4b013bc7fbe1</MessageID>
    <CorrelationID>60a6308c-4cca-4865-80b5-79b4ec560a36</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>modify</UpdateAction>
        <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <MeterMultipliers>
            <mRID>3c67f555-3119-1112-befc-0ed5f89f44cc</mRID>
            <kind>kH</kind>
          </MeterMultipliers>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.5 – ModifyMeterMultiplierKind.xml

E.7 ModifyMeterMultiplierValue.xml

The example shown in Figure E.6 illustrates the modification of the multiplier value for the meter multiplier with the specified mRID. The original meter had a multiplier value of 200.5 and this example changes it to 8. This example, together with F.6 essentially changes the meter from having a pulse weight (Ke) of 200.5 to having a meter constant (kH) of 8. This could also have been done in a single sample changing both attributes at the same time. The mRID of the multiplier is required to be specified, which would be obtained by a get of the meter config as in the example in Clause D.4. The other meter details remain the same. This example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `modify`. There is only one update action with ID of '1'.

The use case for this example is the correction of meter multiplier values after the initial creation of the meter.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>3df14c47-69c7-4431-af32-bf96ffd50935</MessageID>
    <CorrelationID>51e1c957-00e1-49fd-9a01-973e41b0d81f</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>modify</UpdateAction>
        <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <MeterMultipliers>
            <mRID>3c67f555-3119-1112-befc-0ed5f89f44cc</mRID>
            <value>8</value>
          </MeterMultipliers>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.6 – ModifyMeterMultiplierValue.xml

E.8 RemoveMeterMultiplierValue.xml

The example shown in Figure E.7 illustrates the removal of the value from the meter multiplier with the specified mRID from the meter. Building on the other examples, the user has initial set a Ke=200.5, then corrected it to be Kh=8 and then has removed the value altogether. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `remove`. There is only one update action with ID of '1'.

One use case for this example might be that the meter is being put into the meter shop and any subsequent re-deployment of the meter will require setting a proper multiplier value. The 'value' is specified so that the receiving system knows which property to remove, and the value of 0.0 is specified so that the payload will validate against the schema.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>c3fe25f8-9f36-4f14-9f5c-1d66566fe225</MessageID>
    <CorrelationID>01a00ffa-4add-45b9-94b2-487ee4a7626e</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>remove</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <MeterMultipliers>
            <mRID>3c67f555-3119-1112-befc-0ed5f89f44cc</mRID>
            <value>0.0</value>
          </MeterMultipliers>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.7 – RemoveMeterMultiplierValue.xml

E.9 RemoveMeterMultiplier.xml

The example shown in Figure E.8 illustrates the removal of the meter multiplier with the specified mRID from the meter. Building on the other examples, the user has initial set a Ke=200.5, then corrected it to be Kh=8, then has removed the value altogether. The user now removes the entire multiplier. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `remove`. There is only one update action with ID of '1'.

The use case for this example is the correction of erroneous information in the original meter creation.

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>20f20586-9ef2-4fa7-8330-4a4cfc602408</MessageID>
    <CorrelationID>bbc97629-bb7b-4001-9451-25eb1c731243</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>remove</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <MeterMultipliers>
            <mRID>3c67f555-3119-1112-befc-0ed5f89f44cc</mRID>
          </MeterMultipliers>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.8 – RemoveMeterMultiplier.xml

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

E.10 AddMeterSeal.xml

The example shown in Figure E.9 illustrates the addition of a new meter seal to the meter. The original meter had one seal with a sealNumber of 1 and a condition of locked. This example adds a second seal to the meter with a sealNumber of 2 and a condition of locked. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `add`. There is only one `<updateAction>` with an `<UpdateID>` of '1'.

The use case for this example is the incorporation of additional information being provided about the meter after the initial creation.

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>cf483cb7-7664-494a-8665-eb75ff68bf41</MessageID>
    <CorrelationID>fc43d242-aa54-412b-b146-a7f9849bf2b7</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>add</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Seals>
            <condition>locked</condition>
            <sealNumber>2</sealNumber>
          </Seals>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.9 – AddMeterSeal.xml

E.11 AddMeterSealKind.xml

The example shown in Figure E.10 illustrates the addition of a meter seal kind to the meter. In the example in Clause E.10, a second meter seal with seal number 2 and condition locked was added. In this example the kind of the meter seal is set to 'steel'. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `add`. There is only one `<updateAction>` with an `<UpdateID>` of '1'.

The use case for this example is the incorporation of additional information being provided about the meter after the initial creation.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>c8fc7028-cd23-4c70-bfd1-610caa748f50</MessageID>
    <CorrelationID>43de442d-865d-4a01-8cda-917bb6a9e816</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>add</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Seals>
            <kind>steel</kind>
            <sealNumber>2</sealNumber>
          </Seals>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.10 – AddMeterSealKind.xml

E.12 ModifyMeterSealCondition.xml

The example shown in Figure E.11 illustrates the modification of the condition of meter seal 2 on the meter. In the example in Clause E.11, the user added a second meter seal of condition locked, with a kind of steel. In this example the condition is updated to ‘broken’. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `modify`. There is only one `<updateAction>` with an `<UpdateID>` of ‘1’.

The use case for this example is the update of the meter seals condition after it has been found to be broken in the field.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>4e2ff938-368b-4d9b-8491-8fcd2a74dafc</MessageID>
    <CorrelationID>632fa9d4-ebc4-467a-b7a6-61b8c3b03bbe</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>modify</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Seals>
            <condition>open</condition>
            <sealNumber>2</sealNumber>
          </Seals>
          <Seals>
            <condition>broken</condition>
            <sealNumber>1</sealNumber>
          </Seals>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.11 – ModifyMeterSealCondition.xml

E.13 RemoveMeterSealCondition.xml

The example shown in Figure E.12 illustrates the removal of the meter seal condition for meter seal 1. The seal is identified by the number '1', and the condition of 'other' is provided so that the payload conforms to the schema and to indicate that the condition of the seal is not known. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `remove`. There is only one `<updateAction>` with an `<UpdateID>` of '1'.

The use case for this example would be to indicate that the state of the meter seal in the field is now unknown.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>c8fc7028-cd23-4c70-bfd1-610caa748f50</MessageID>
    <CorrelationID>43de442d-865d-4a01-8cda-917bb6a9e816</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>remove</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Seals>
            <condition>other</condition>
            <sealNumber>1</sealNumber>
          </Seals>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.12 – RemoveMeterDealCondition.xml

E.14 RemoveMeterSeal.xml

The example shown in Figure E.13 illustrates the removal of meter seal 1 from the meter. In the previous examples the meter had one initial meter seal with a second one added later. This example then removes the original meter seal and leaves on the second one. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `remove`. There is only one `<updateAction>` with an `<UpdateID>` of '1'.

The use case for this example is a meter seal being removed from the meter.

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>aa10fe03-52c7-4299-8d80-6f7b5d749d04</MessageID>
    <CorrelationID>d6edb684-7719-48ba-9626-e2d7c2183343</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>remove</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <Seals>
            <sealNumber>1</sealNumber>
          </Seals>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.13 – RemoveMeterSeal.xml

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

E.15 ModifyProgramId.xml

The example shown in Figure E.14 illustrates the modification of the program ID of the meter's simple end device function. The program ID of the meters simple end device function SEDF1 is being changed to 'ER017B' from an original program ID of ER017A. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `modify`. There is only one `<updateAction>` with an `<UpdateID>` of '1'.

The use case for this example is a reprogram of the meter.

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>4fe13aee-f5e8-4323-9852-800cb8d0c0fb</MessageID>
    <CorrelationID>7c24ca34-3b22-42a0-b7e3-b909a44f6eaa</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>modify</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <SimpleEndDeviceFunction ref="SEDF1"/>
        </Meter>
        <SimpleEndDeviceFunction>
          <programID>ER017B</programID>
          <Names>
            <name>SEDF1</name>
          </Names>
        </SimpleEndDeviceFunction>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.14 – ModifyProgramId.xml

E.16 RemoveProgramId.xml

The example shown in Figure E.15 illustrates the removal of program ID for the meter's simple end device function of SEDF1. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `remove`. There is only one `<updateAction>` with an `<UpdateID>` of '1'.

The use case for this example is the return of a meter back into inventory and an indication that the program ID will need to be set again once it is re-deployed. The value of 'xxxx' for the program ID indicates the property to be removed.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>48f1d454-c661-421e-8993-d8f47535d6c8</MessageID>
    <CorrelationID>fcc64ed0-ace0-4452-a53f-4a939ba7b5fe</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>remove</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <SimpleEndDeviceFunction ref="SEDF1"/>
        </Meter>
        <SimpleEndDeviceFunction>
          <programID>xxxx</programID>
          <Names>
            <name>SEDF1</name>
          </Names>
        </SimpleEndDeviceFunction>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.15 – RemoveProgramId.xml

E.17 AddChannel.xml

The example shown in Figure E.16 illustrates the addition of a second channel to the meter. The original meter had a single channel of reading type ‘0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0’. A second channel named CH2 is being added to the meters simple end device function SEDF1, with a reading type of ‘0.0.2.4.1.1.12.0.0.0.0.0.0.0.3.72.0'. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `add`. There is only one `<updateAction>` with an `<UpdateID>` of ‘1’.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>b406ae71-6ed2-437a-a2e8-c84064dd7ce3</MessageID>
    <CorrelationID>08b68bb5-601a-4292-8f3b-9336729ea18c</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>add</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <SimpleEndDeviceFunction ref="SEDF1"/>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <SimpleEndDeviceFunction>
          <Names>
            <name>SEDF1</name>
          </Names>
          <Registers>
            <Channels>
              <Names>
                <name>CH2</name>
              </Names>
              <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
            </Channels>
            <Names>
              <name>REG1</name>
            </Names>
          </Registers>
        </SimpleEndDeviceFunction>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.16 – AddChannelId.xml

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

E.18 ModifyChannelReadingType.xml

The example shown in Figure E.17 illustrates the modification of the reading type of the second channel that was added in the example in Clause E.17. The original value of reading type is being changed to '0.0.7.4.1.1.12.0.0.0.0.0.0.0.3.72.0'. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `modify`. There is only one `<updateAction>` with an `<UpdateID>` of '1'.

The use case for this example is the correction of erroneous information in the original addition of the second channel.

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>4ab1bcbe-316b-41e1-b908-781d4547e8ab</MessageID>
    <CorrelationID>eed57b7d-28d1-436b-bcfa-7d32c04dee0d</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>modify</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <SimpleEndDeviceFunction ref="SEDF1"/>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <SimpleEndDeviceFunction>
          <Names>
            <name>SEDF1</name>
          </Names>
          <Registers>
            <Channels>
              <Names>
                <name>CH2</name>
              </Names>
              <ReadingType ref="0.0.7.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
            </Channels>
            <Names>
              <name>REG1</name>
            </Names>
          </Registers>
        </SimpleEndDeviceFunction>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.17 – ModifyChannelReadingType.xml

E.19 RemoveChannel.xml

The example shown in Figure E.18 illustrates the removal of the second channel that was added to the meter in the example in Clause E.17. The remainder of the details of the meter remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `remove`. There is only one `<updateAction>` with an `<UpdateID>` of '1'.

The use case for this example is the correction of erroneous information in the original addition of the second channel.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>4ab1bcbe-316b-41e1-b908-781d4547e8ab</MessageID>
    <CorrelationID>eed57b7d-28d1-436b-bcfa-7d32c04dee0d</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>remove</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <SimpleEndDeviceFunction ref="SEDF1"/>
          <Names>
            <name>M001</name>
          </Names>
        </Meter>
        <SimpleEndDeviceFunction>
          <Names>
            <name>SEDF1</name>
          </Names>
          <Registers>
            <Channels>
              <Names>
                <name>CH2</name>
              </Names>
            </Channels>
            <Names>
              <name>REG1</name>
            </Names>
          </Registers>
        </SimpleEndDeviceFunction>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.18 – RemoveChannel.xml

E.20 AddConnectDisconnectFunction.xml

The example shown in Figure E.19 illustrates the addition of a connect/disconnect function name of CDF1 to the meter. The connect/disconnect function is enabled, is in a 'disconnected' state, and has the remote auto disconnect option set to true. The other meter details remain the same. The example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `add`. There is only one `<updateAction>` with an `<UpdateID>` of '1'.

The use case for this example is that the meter is being reprogrammed to enable connect/disconnection functionality. Alternatively, a 3rd party disconnect switch has been attached to the meter.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>cf7bb7a5-0206-4764-879b-2aaa20d2043f</MessageID>
    <CorrelationID>4caea4bd-3c4e-485a-ab82-0160c35ad20d</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>add</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <ConnectDisconnectFunction>
          <enabled>true</enabled>
          <isConnected>false</isConnected>
          <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
          <Names>
            <name>CDF1</name>
          </Names>
        </ConnectDisconnectFunction>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <ConnectDisconnectFunction ref="CDF1"/>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.19 – AddConnectDisconnectFunction.xml

E.21 RemoveConnectDisconnectFunction.xml

The example shown in Figure E.20 illustrates the remove of the connect/disconnect function that was added to the meter in the example in Clause E.20. The other meter details remain the same. This example uses a `<Header><Verb>` of `update` and an `<UpdateAction>` of `remove`. There is only one `<updateAction>` with an `<UpdateID>` of '1'.

The use case for this example is that the customer has decided to opt out of connect/disconnect functionality or that the 3rd party disconnect switch has been removed.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>72990914-ac7e-44ca-b33f-ec9ce117426f</MessageID>
    <CorrelationID>9c988e78-35ae-4b72-a8fe-875f7e05915b</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>remove</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <ConnectDisconnectFunction>
          <Names>
            <name>CDF1</name>
          </Names>
        </ConnectDisconnectFunction>
        <Meter>
          <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
          <ConnectDisconnectFunction ref="CDF1"/>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.20 – RemoveConnectDisconnectFunction.xml

E.22 DeleteMeterConfig.xml

The example shown in Figure E.21 illustrates the deletion of the meter config that was created in the example in Clause E.2. This example varies from the example in Clause E.23 in that this one uses the MeterConfig profile with an mRID to identify the meter, while the example in E.23 uses the ID in the payload. Both are equivalent approaches to the deletion of a meter config.

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>delete</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:10-06:00</Timestamp>
    <MessageID>72990914-ac7e-44ca-b33f-ec9ce117426f</MessageID>
    <CorrelationID>9c988e78-35ae-4b72-a8fe-875f7e05915b</CorrelationID>
  </Header>
  <Payload>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
      <Meter>
        <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
      </Meter>
    </MeterConfig>
  </Payload>
</UpdateRequestMessage>
```

Figure E.21 – DeleteMeterConfig.xml

E.23 DeleteMeterConfig.xml

The example shown in Figure E.22 illustrates the deletion of the meter config that was created in the example in Clause E.2. Note that example varies from the example in Clause E.22 in that this one uses ID in the payload to identify the meter, while the example in Clause E.22 uses the MeterConfig profile with an mRID in the payload. Both are equivalent approaches to the deletion of a meter config.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>delete</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:10-06:00</Timestamp>
    <MessageID>72990914-ac7e-44ca-b33f-ec9ce117426f</MessageID>
    <CorrelationID>9c988e78-35ae-4b72-a8fe-875f7e05915b</CorrelationID>
  </Header>
  <Payload>
    <ObjectID>
      <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
      <ObjectType>
        <type>Meter</type>
      </ObjectType>
    </ObjectID>
  </Payload>
</UpdateRequestMessage>
```

Figure E.22 – DeleteMeterConfig.xml

E.24 CreateMeterConfigs.xml

The example shown in Figure E.23 is similar to that of E.2 except it creates multiple meter config in one message. In this scenario multiple meter configs are provided outside of an operation set. As such, there is no order imposed and no transactional integrity. Each meter creation is independent of the others and they can be created in any order. If any one of them fails, the others will still be created.

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>create</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:10-06:00</Timestamp>
    <MessageID>fbb1c893-83c9-47b2-8eca-b8f8859315e6</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a169-ce38fba5d40f</CorrelationID>
  </Header>
  <Payload>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
      <Meter>
        <formNumber>2S</formNumber>
        <SimpleEndDeviceFunction ref="SEDF1"/>
        <MeterMultipliers>
          <kind>ctRatio</kind>
          <value>200.5</value>
        </MeterMultipliers>
        <Names>
          <name>M001</name>
        </Names>
        <Seals>
          <condition>locked</condition>
          <sealNumber>1</sealNumber>
        </Seals>
      </Meter>
      <SimpleEndDeviceFunction>
        <programID>ER017A</programID>
        <Names>
          <name>SEDF1</name>
        </Names>
        <Registers>
          <Channels>
            <Names>
              <name>CH1</name>
            </Names>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
          <Names>
            <name>REG1</name>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        </Names>
      </Registers>
    </SimpleEndDeviceFunction>
  </MeterConfig>
  <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
    <Meter>
      <formNumber>2S</formNumber>
      <SimpleEndDeviceFunction ref="SEDF2"/>
      <MeterMultipliers>
        <kind>ctRatio</kind>
        <value>207.0</value>
      </MeterMultipliers>
      <Names>
        <name>M002</name>
      </Names>
      <Seals>
        <condition>open</condition>
        <sealNumber>1</sealNumber>
      </Seals>
    </Meter>
    <SimpleEndDeviceFunction>
      <programID>ER017B</programID>
      <Names>
        <name>SEDF2</name>
      </Names>
      <Registers>
        <Channels>
          <Names>
            <name>CH1</name>
          </Names>
          <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
        </Channels>
        <Names>
          <name>REG1</name>
        </Names>
      </Registers>
    </SimpleEndDeviceFunction>
  </MeterConfig>
  <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
    <Meter>
      <formNumber>2S</formNumber>
      <SimpleEndDeviceFunction ref="SEDF03"/>
      <MeterMultipliers>
        <kind>ctRatio</kind>
        <value>190.1</value>
      </MeterMultipliers>
      <Names>
        <name>M003</name>
      </Names>
    </Meter>
    <SimpleEndDeviceFunction>
      <programID>ER017C</programID>
      <Names>
        <name>SEDF03</name>
      </Names>
      <Registers>
        <Channels>
          <Names>
            <name>CH1</name>
          </Names>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
          <Names>
            <name>REG1</name>
          </Names>
        </Registers>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </Payload>
</UpdateRequestMessage>
```

Figure E.23 – CreateMeterConfigs.xml

E.25 MultipleObjectChanges.xml

The example shown in Figure E.24 illustrates multiple object changes in one message. Each update is encapsulated in its own ObjectUpdate element with appropriate UpdateID and UpdateAction. In this scenario multiple meter updates are provided outside of an operation set. As such, there is no order imposed and no transactional integrity. Each meter update is independent of the others and they can happen in any order. If any one or more of them fails, the others will still be attempted. Because some use cases may necessitate imposing sequence order and transactional integrity, the example in Clause E.26 illustrates the same example with the use of OperationSets.

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>update</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>cf7bb7a5-0206-4764-879b-2aaa20d2043f</MessageID>
    <CorrelationID>4caea4bd-3c4e-485a-ab82-0160c35ad20d</CorrelationID>
  </Header>
  <Payload>
    <ObjectUpdate>
      <UpdateID>1</UpdateID>
      <UpdateAction>add</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <ConnectDisconnectFunction>
          <enabled>true</enabled>
          <isConnected>false</isConnected>
          <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
          <Names>
            <name>CD1</name>
          </Names>
        </ConnectDisconnectFunction>
        <Meter>
          <ConnectDisconnectFunction ref="CD1"/>
          <Names>
            <name>M001</name>
          </Names>
          <Seals>
            <condition>open</condition>
            <kind>steel</kind>
            <sealNumber>2</sealNumber>
          </Seals>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
    <ObjectUpdate>
      <UpdateID>2</UpdateID>
      <UpdateAction>modify</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <Meter>
          <SimpleEndDeviceFunction ref="SEDF1"/>
          <Names>
            <name>M001</name>
          </Names>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      </Meter>
      <SimpleEndDeviceFunction>
        <programID>ER018A</programID>
        <Names>
          <name>SEDF1</name>
        </Names>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </ObjectUpdate>
  <ObjectUpdate>
    <UpdateID>3</UpdateID>
    <UpdateAction>modify</UpdateAction>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
      <Meter>
        <formNumber>9S</formNumber>
        <SimpleEndDeviceFunction ref="SEDF2"/>
        <Names>
          <name>M002</name>
        </Names>
      </Meter>
      <SimpleEndDeviceFunction>
        <programID>ER018Z</programID>
        <Names>
          <name>SEDF2</name>
        </Names>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </ObjectUpdate>
  <ObjectUpdate>
    <UpdateID>4</UpdateID>
    <UpdateAction>add</UpdateAction>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
      <Meter>
        <SimpleEndDeviceFunction ref="SEDF2"/>
        <Names>
          <name>M002</name>
        </Names>
      </Meter>
      <SimpleEndDeviceFunction>
        <Names>
          <name>SEDF2</name>
        </Names>
        <Registers>
          <Channels>
            <Names>
              <name>CH1</name>
            </Names>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.73.0"/>
          </Channels>
          <Names>
            <name>REG2</name>
          </Names>
        </Registers>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </ObjectUpdate>
  <ObjectUpdate>
    <UpdateID>5</UpdateID>
    <UpdateAction>remove</UpdateAction>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
      <Meter>
        <Names>
          <name>M002</name>
        </Names>
        <Seals>
          <sealNumber>1</sealNumber>
        </Seals>
      </Meter>
    </MeterConfig>
  </ObjectUpdate>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <ObjectUpdate>
      <UpdateID>6</UpdateID>
      <UpdateAction>add</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
        <ConnectDisconnectFunction>
          <enabled>true</enabled>
          <isConnected>true</isConnected>
          <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
          <Names>
            <name>CD2</name>
          </Names>
        </ConnectDisconnectFunction>
        <Meter>
          <ConnectDisconnectFunction ref="CD2"/>
          <Names>
            <name>M003</name>
          </Names>
          <Seals>
            <condition>open</condition>
            <kind>steel</kind>
            <sealNumber>1</sealNumber>
          </Seals>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
    <ObjectUpdate>
      <UpdateID>7</UpdateID>
      <UpdateAction>modify</UpdateAction>
      <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
      <Meter>
          <formNumber>12S</formNumber>
          <Names>
            <name>M003</name>
          </Names>
        </Meter>
      </MeterConfig>
    </ObjectUpdate>
  </Payload>
</UpdateRequestMessage>
```

Figure E.24 – MultipleObjectChanges.xml

E.26 MultipleObjectChangesWithOperationSet.xml

The example shown in Figure E.25 illustrates multiple object changes in one message. It performs the same actions as in the example in Clause E.25 except in this case each update is encapsulated in an Operation inside an OperationSet. The updates are specified to be in a specific order and transactional integrity is enforced.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<UpdateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>execute</Verb>
    <Noun>OperationSet</Noun>
    <Timestamp>2018-01-02T13:33:47-06:00</Timestamp>
    <MessageID>71d267ed-7957-419d-aef3-d91dccba399f</MessageID>
    <CorrelationID>dbc73dbd-e71c-4c3e-b9fa-955449ac75d6</CorrelationID>
  </Header>
  <Payload>
    <OperationSet>
      <OperationSetID>1</OperationSetID>
      <EnforceMsgSequence>true</EnforceMsgSequence>
      <EnforceTransactionalIntegrity>true</EnforceTransactionalIntegrity>
      <Operation>
        <OperationID>1</OperationID>
        <Verb>update</Verb>
        <Noun>MeterConfig</Noun>
        <ObjectUpdate>
          <UpdateID>1</UpdateID>
          <UpdateAction>add</UpdateAction>
          <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
            <ConnectDisconnectFunction>
              <enabled>true</enabled>
              <isConnected>false</isConnected>
              <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
              <Names>
                <name>CD1</name>
              </Names>
            </ConnectDisconnectFunction>
            <Meter>
              <ConnectDisconnectFunction ref="CD1"/>
              <Names>
                <name>M001</name>
              </Names>
              <Seals>
                <condition>open</condition>
                <kind>steel</kind>
                <sealNumber>2</sealNumber>
              </Seals>
            </Meter>
          </MeterConfig>
        </ObjectUpdate>
      </Operation>
      <Operation>
        <OperationID>2</OperationID>
        <Verb>update</Verb>
        <Noun>MeterConfig</Noun>
        <ObjectUpdate>
          <UpdateID>1</UpdateID>
          <UpdateAction>modify</UpdateAction>
          <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
            <Meter>
              <SimpleEndDeviceFunction ref="SEDF1"/>
              <Names>
                <name>M001</name>
              </Names>
            </Meter>
            <SimpleEndDeviceFunction>
              <programID>ER018A</programID>
              <Names>
                <name>SEDF1</name>
              </Names>
            </SimpleEndDeviceFunction>
          </MeterConfig>
        </ObjectUpdate>
      </Operation>
      <Operation>
        <OperationID>3</OperationID>
        <Verb>update</Verb>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
  <Noun>MeterConfig</Noun>
  <ObjectUpdate>
    <UpdateID>1</UpdateID>
    <UpdateAction>modify</UpdateAction>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
      <Meter>
        <formNumber>9S</formNumber>
        <SimpleEndDeviceFunction ref="SEDF2"/>
        <Names>
          <name>M002</name>
        </Names>
      </Meter>
      <SimpleEndDeviceFunction>
        <programID>ER018Z</programID>
        <Names>
          <name>SEDF2</name>
        </Names>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </ObjectUpdate>
</Operation>
<Operation>
  <OperationID>4</OperationID>
  <Verb>update</Verb>
  <Noun>MeterConfig</Noun>
  <ObjectUpdate>
    <UpdateID>1</UpdateID>
    <UpdateAction>add</UpdateAction>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
      <Meter>
        <SimpleEndDeviceFunction ref="SEDF2"/>
        <Names>
          <name>M002</name>
        </Names>
      </Meter>
      <SimpleEndDeviceFunction>
        <Names>
          <name>SEDF2</name>
        </Names>
        <Registers>
          <Channels>
            <Names>
              <name>CH1</name>
            </Names>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.73.0"/>
          </Channels>
          <Names>
            <name>REG2</name>
          </Names>
        </Registers>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </ObjectUpdate>
</Operation>
<Operation>
  <OperationID>5</OperationID>
  <Verb>update</Verb>
  <Noun>MeterConfig</Noun>
  <ObjectUpdate>
    <UpdateID>1</UpdateID>
    <UpdateAction>remove</UpdateAction>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
      <Meter>
        <Names>
          <name>M002</name>
        </Names>
        <Seals>
          <sealNumber>1</sealNumber>
        </Seals>
      </Meter>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          </MeterConfig>
        </ObjectUpdate>
      </Operation>
      <Operation>
        <OperationID>6</OperationID>
        <Verb>update</Verb>
        <Noun>MeterConfig</Noun>
        <ObjectUpdate>
          <UpdateID>1</UpdateID>
          <UpdateAction>add</UpdateAction>
          <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
            <ConnectDisconnectFunction>
              <enabled>true</enabled>
              <isConnected>true</isConnected>
              <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
              <Names>
                <name>CD2</name>
              </Names>
            </ConnectDisconnectFunction>
            <Meter>
              <ConnectDisconnectFunction ref="CD2"/>
              <Names>
                <name>M003</name>
              </Names>
              <Seals>
                <condition>open</condition>
                <kind>steel</kind>
                <sealNumber>1</sealNumber>
              </Seals>
            </Meter>
          </MeterConfig>
        </ObjectUpdate>
      </Operation>
      <Operation>
        <OperationID>7</OperationID>
        <Verb>update</Verb>
        <Noun>MeterConfig</Noun>
        <ObjectUpdate>
          <UpdateID>1</UpdateID>
          <UpdateAction>modify</UpdateAction>
          <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#">
            <Meter>
              <formNumber>12S</formNumber>
              <Names>
                <name>M003</name>
              </Names>
            </Meter>
          </MeterConfig>
        </ObjectUpdate>
      </Operation>
    </OperationSet>
  </Payload>
</UpdateRequestMessage>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure E.25 – MultipleObjectChangesWithOperationSet.xml

Annex F
(informative)

Examples of ObjectIdentificationRevisionSets Messages

F.1 General

This annex provides examples showing the operation of the ObjectIdentificationRevisionSets request messages (8.7).

These examples are based around a meter object and a customer object. Their initial naming states are shown in Table F.1 and Table F.2 respectively.

Table F.1 – Initial naming states of the example meter object

	Object Type	Meter
	mRID	
Name One	Name.name	1467A35
	NameType.name	SerialNumber
	NameTypeAuthority.name	Itron
Name Two	Name.name	MTR46889
	NameType.name	BadgeNumber
	NameTypeAuthority.name	UtilityXYZ
Name Three	Name.name	
	NameType.name	
	NameTypeAuthority.name	

Table F.2 – Initial naming states of the example customer object

	Object Type	Customer
	mRID	dbe9fba9-a67a-4be4-92b1-62c1c96abc18
Name One	Name.name	
	NameType.name	
	NameTypeAuthority.name	
Name Two	Name.name	
	NameType.name	
	NameTypeAuthority.name	
Name Three	Name.name	
	NameType.name	
	NameTypeAuthority.name	

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

F.2 Add an mRID for the meter object

The example shown in Figure F.1 is a create(ObjectIdentificationRevisionSets) request message that causes the responding system to add the mRID of the meter object identified by the serial number 1467A35. The resulting state of the object's identification is shown in Table F.3.

Table F.3 – Add an mRID for the meter object

		Before Step	After Step
	Object Type	Meter	Meter
	mRID		c7c4f1c5-92ce-475d-b6d4-d831605ff56e
Name One	Name.name	1467A35	1467A35
	NameType.name	SerialNumber	SerialNumber
	NameTypeAuthority.name	Itron	Itron
Name Two	Name.name	MTR46889	MTR46889
	NameType.name	BadgeNumber	BadgeNumber
	NameTypeAuthority.name	UtilityXYZ	UtilityXYZ
Name Three	Name.name		
	NameType.name		
	NameTypeAuthority.name		

```
<CreateObjectIdentificationRevisionSetsRequestMessage
xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>create</Verb>
    <Noun>ObjectIdentificationRevisionSets</Noun>
    <Timestamp>2021-12-17T09:30:40Z</Timestamp>
    <MessageID>89136dde-079e-4dfa-94b0-3762498850cb</MessageID>
    <CorrelationID>b229f8fe-e59d-4f12-ba1c-31511b27a8e0</CorrelationID>
    <Comment>ObjectIdentificationRevisionSets Step 1</Comment>
  </Header>
  <Payload>
    <ObjectIdentificationRevisionSets
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#">
      <ObjectIdentificationRevisionSet>
        <mRID>62483e9b-fbb1-46ba-9cb3-5f3fb91ac396</mRID>
        <effectiveDateTime>2021-12-17T09:30:47Z</effectiveDateTime>
        <Object>
          <Names>
            <name>1467A35</name>
            <NameType>
              <name>SerialNumber</name>
              <NameTypeAuthority>
                <name>Itron</name>
              </NameTypeAuthority>
            </NameType>
          </Names>
          <ObjectType>
            <type>Meter</type>
          </ObjectType>
        </Object>
        <ObjectIdentificationRevision>
          <revisionKind>add</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
            <mRID>c7c4f1c5-92ce-475d-b6d4-d831605ff56e</mRID>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
      </ObjectIdentificationRevisionSet>
    </ObjectIdentificationRevisionSets>
  </Payload>
</CreateObjectIdentificationRevisionSetsRequestMessage>
```

Figure F.1 – Add an mRID for the Meter Object

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

F.3 Modify name two of the meter object

The example shown in Figure F.2 is a create(ObjectIdentificationRevisionSets) request message that causes the responding system to modify name two of the meter object identified by the mRID c7c4f1c5-92ce-475d-b6d4-d831605ff56e. The resulting state of the object's identification is shown in Table F.4.

Table F.4 – Modify name two of the meter object

		Before Step	After Step
	Object Type	Meter	Meter
	mRID	c7c4f1c5-92ce-475d-b6d4-d831605ff56e	c7c4f1c5-92ce-475d-b6d4-d831605ff56e
Name One	Name.name	1467A35	1467A35
	NameType.name	SerialNumber	SerialNumber
	NameTypeAuthority.name	Itron	Itron
Name Two	Name.name	MTR46889	**47CC907448**
	NameType.name	BadgeNumber	**RFID Tag**
	NameTypeAuthority.name	UtilityXYZ	UtilityXYZ
Name Three	Name.name		
	NameType.name		
	NameTypeAuthority.name		

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateObjectIdentificationRevisionSetsRequestMessage
xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>ObjectIdentificationRevisionSets</Noun>
    <Timestamp>2021-12-17T09:30:40Z</Timestamp>
    <MessageID>3d2fe703-bc3b-4f52-b380-c8c7daadd06a</MessageID>
    <CorrelationID>6f5ee9b4-a7f4-4321-a5a8-a386dc988979</CorrelationID>
    <Comment>ObjectIdentificationRevisionSets Step 2</Comment>
  </Header>
  <Payload>
    <ObjectIdentificationRevisionSets
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#">
      <ObjectIdentificationRevisionSet>
        <mRID>66a4149e-f4c7-4ff5-990a-6e5527be9ac2</mRID>
        <effectiveDateTime>2021-12-17T09:30:47Z</effectiveDateTime>
        <Object>
          <mRID>c7c4f1c5-92ce-475d-b6d4-d831605ff56e</mRID>
          <ObjectType>
            <type>Meter</type>
          </ObjectType>
        </Object>
        <ObjectIdentificationRevision>
          <revisionKind>remove</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
            <Names>
              <name>MTR46889</name>
              <NameType>
                <name>BadgeNumber</name>
                <NameTypeAuthority>
                  <name>UtilityXYZ</name>
                </NameTypeAuthority>
              </NameType>
            </Names>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
        <ObjectIdentificationRevision>
          <revisionKind>add</revisionKind>
          <sequenceNumber>2</sequenceNumber>
          <IdentifiedObjectRevision>
            <Names>
              <name>47CC907448</name>
              <NameType>
                <name>RFID Tag</name>
                <NameTypeAuthority>
                  <name>UtilityXYZ</name>
                </NameTypeAuthority>
              </NameType>
            </Names>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
      </ObjectIdentificationRevisionSet>
    </ObjectIdentificationRevisionSets>
  </Payload>
</CreateObjectIdentificationRevisionSetsRequestMessage>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure F.2 – Modify name two of the meter object

F.4 Add name three of the meter object

The example shown in Figure F.3 is a create(ObjectIdentificationRevisionSets) request message that causes the responding system to add a new name three of asset number 698877690 to the meter object identified by the mRID c7c4f1c5-92ce-475d-b6d4-d831605ff56e. The resulting state of the meter object's identification is shown in Table F.5.

Table F.5 – Add name three of the meter object

		Before Step	After Step
	Object Type	Meter	Meter
	mRID	c7c4f1c5-92ce-475d-b6d4-d831605ff56e	c7c4f1c5-92ce-475d-b6d4-d831605ff56e
Name One	Name.name	1467A35	1467A35
	NameType.name	SerialNumber	SerialNumber
	NameTypeAuthority.name	Itron	Itron
Name Two	Name.name	47CC907448	47CC907448
	NameType.name	RFID Tag	RFID Tag
	NameTypeAuthority.name	UtilityXYZ	UtilityXYZ
Name Three	Name.name		**698877690**
	NameType.name		**AssetNumber**
	NameTypeAuthority.name		

```
<CreateObjectIdentificationRevisionSetsRequestMessage
xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>ObjectIdentificationRevisionSets</Noun>
    <Timestamp>2021-12-17T09:30:40Z</Timestamp>
    <MessageID>e59ad6d9-aae3-41be-8c45-87d50ad042cb</MessageID>
    <CorrelationID>2ecf57de-fc3e-47a2-b4a1-899056e27f82</CorrelationID>
    <Comment>ObjectIdentificationRevisionSets Step 3</Comment>
  </Header>
  <Payload>
    <ObjectIdentificationRevisionSets
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#">
      <ObjectIdentificationRevisionSet>
        <mRID>a9bb4967-8be1-485d-a814-d20ea42bc22f</mRID>
        <effectiveDateTime>2021-12-17T09:30:47Z</effectiveDateTime>
        <Object>
          <mRID>c7c4f1c5-92ce-475d-b6d4-d831605ff56e</mRID>
          <ObjectType>
            <type>Meter</type>
          </ObjectType>
        </Object>
        <ObjectIdentificationRevision>
          <revisionKind>add</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
            <Names>
              <name>698877690</name>
              <NameType>
                <name>AssetNumber</name>
              </NameType>
            </Names>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
      </ObjectIdentificationRevisionSet>
    </ObjectIdentificationRevisionSets>
  </Payload>
</CreateObjectIdentificationRevisionSetsRequestMessage>
```

Figure F.3 – Add name three of the meter object

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

F.5 Remove Name One of the Meter Object

The example shown in Figure F.4 is a create(ObjectIdentificationRevisionSets) request message that causes the responding system to remove name one (serial number 1467A35) from the meter object identified by the mRID c7c4f1c5-92ce-475d-b6d4-d831605ff56e. The resulting state of the meter object's identification is shown in Table F.6.

Table F.6 – Remove name one of the meter object

		Before Step	After Step
	Object Type	Meter	Meter
	mRID	c7c4f1c5-92ce-475d-b6d4-d831605ff56e	c7c4f1c5-92ce-475d-b6d4-d831605ff56e
Name One	Name.name	1467A35	(deleted)
	NameType.name	SerialNumber	(deleted)
	NameTypeAuthority.name	Itron	(deleted)
Name Two	Name.name	47CC907448	47CC907448
	NameType.name	RFID Tag	RFID Tag
	NameTypeAuthority.name	UtilityXYZ	UtilityXYZ
Name Three	Name.name	698877690	698877690
	NameType.name	AssetNumber	AssetNumber
	NameTypeAuthority.name		

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateObjectIdentificationRevisionSetsRequestMessage
xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>ObjectIdentificationRevisionSets</Noun>
    <Timestamp>2021-12-17T09:30:40Z</Timestamp>
    <MessageID>d2374f01-9bde-42d1-9098-084f9ea31ce3</MessageID>
    <CorrelationID>64c952d5-beee-430c-87c8-2b1ba355a628</CorrelationID>
    <Comment>ObjectIdentificationRevisionSets Step 4</Comment>
  </Header>
  <Payload>
    <ObjectIdentificationRevisionSets
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#">
      <ObjectIdentificationRevisionSet>
        <mRID>3d2e7f37-b4e6-475e-b61d-75a2ecac34bf</mRID>
        <effectiveDateTime>2021-12-17T09:30:47Z</effectiveDateTime>
        <Object>
          <mRID>c7c4f1c5-92ce-475d-b6d4-d831605ff56e</mRID>
          <ObjectType>
            <type>Meter</type>
          </ObjectType>
        </Object>
        <ObjectIdentificationRevision>
          <revisionKind>remove</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
            <Names>
              <name>1467A35</name>
              <NameType>
                <name>SerialNumber</name>
                <NameTypeAuthority>
                  <name>Itron</name>
                </NameTypeAuthority>
              </NameType>
            </Names>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
      </ObjectIdentificationRevisionSet>
    </ObjectIdentificationRevisionSets>
  </Payload>
</CreateObjectIdentificationRevisionSetsRequestMessage>
```

Figure F.4 – Remove name one of the meter object

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

F.6 Modify the mRID of the meter object

The example shown in Figure F.5 is a create(ObjectIdentificationRevisionSets) request message that causes the responding system to replace the existing mRID of the meter object identified by the mRID c7c4f1c5-92ce-475d-b6d4-d831605ff56e with a41addcf-3542-4b69-b740-ca3abcf6384c. The resulting state of the meter object's identification is shown in Table F.7.

Table F.7 – Modify the mRID of the meter object

		Before Step	After Step
	Object Type	Meter	Meter
	mRID	c7c4f1c5-92ce-475d-b6d4-d831605ff56e	**a41addcf-3542-4b69-b740-ca3abcf6384c**
Name One	Name.name		
	NameType.name		
	NameTypeAuthority.name		
Name Two	Name.name	47CC907448	47CC907448
	NameType.name	RFID Tag	RFID Tag
	NameTypeAuthority.name	UtilityXYZ	UtilityXYZ
Name Three	Name.name	698877690	698877690
	NameType.name	AssetNumber	AssetNumber
	NameTypeAuthority.name		

```
<CreateObjectIdentificationRevisionSetsRequestMessage
xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>ObjectIdentificationRevisionSets</Noun>
    <Timestamp>2021-12-17T09:30:40Z</Timestamp>
    <MessageID>e4902d19-0b42-43c5-9e18-7afeddb5770c</MessageID>
    <CorrelationID>0d19c42b-2804-4782-9bdf-972c0bacac73</CorrelationID>
    <Comment>ObjectIdentificationRevisionSets Step 5</Comment>
  </Header>
  <Payload>
    <ObjectIdentificationRevisionSets
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <ObjectIdentificationRevisionSet>
        <mRID>306e126e-4ab5-4863-8e65-816744fc1556</mRID>
        <effectiveDateTime>2021-12-17T09:30:47Z</effectiveDateTime>
        <Object>
          <mRID>c7c4f1c5-92ce-475d-b6d4-d831605ff56e</mRID>
          <ObjectType>
            <type>Meter</type>
          </ObjectType>
        </Object>
        <ObjectIdentificationRevision>
          <revisionKind>remove</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
            <mRID>c7c4f1c5-92ce-475d-b6d4-d831605ff56e</mRID>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
        <ObjectIdentificationRevision>
          <revisionKind>add</revisionKind>
          <sequenceNumber>2</sequenceNumber>
          <IdentifiedObjectRevision>
            <mRID>a41addcf-3542-4b69-b740-ca3abcf6384c</mRID>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
      </ObjectIdentificationRevisionSet>
    </ObjectIdentificationRevisionSets>
  </Payload>
</CreateObjectIdentificationRevisionSetsRequestMessage>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure F.5 – Modify the mRID of the meter object

F.7 Remove the mRID of the meter object

The example shown in Figure F.6 is a create(ObjectIdentificationRevisionSets) request message that causes the responding system to remove the mRID of the meter object identified by the mRID a41addcf-3542-4b69-b740-ca3abcf6384c. The resulting state of the meter object's identification is shown in Table F.8.

Table F.8 – Remove the mRID of the meter object

		Before Step	After Step
	Object Type	Meter	Meter
	mRID	a41addcf-3542-4b69-b740-ca3abcf6384c	(deleted)
Name One	Name.name		
	NameType.name		
	NameTypeAuthority.name		
Name Two	Name.name	47CC907448	47CC907448
	NameType.name	RFID Tag	RFID Tag
	NameTypeAuthority.name	UtilityXYZ	UtilityXYZ
Name Three	Name.name	698877690	698877690
	NameType.name	AssetNumber	AssetNumber
	NameTypeAuthority.name		

```
<CreateObjectIdentificationRevisionSetsRequestMessage
xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>ObjectIdentificationRevisionSets</Noun>
    <Timestamp>2021-12-17T09:30:40Z</Timestamp>
    <MessageID>ebcec5a3-a1d5-4b37-88af-11bf5e3eac00</MessageID>
    <CorrelationID>6dc91a4a-e58f-4cd1-8a12-0121220c8f4f</CorrelationID>
    <Comment>ObjectIdentificationRevisionSets Step 6</Comment>
  </Header>
  <Payload>
    <ObjectIdentificationRevisionSets
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#">
      <ObjectIdentificationRevisionSet>
        <mRID>c7e18116-de86-4fe1-8906-8034fedc973a</mRID>
        <effectiveDateTime>2021-12-17T09:30:47Z</effectiveDateTime>
        <Object>
          <mRID>a41addcf-3542-4b69-b740-ca3abcf6384c</mRID>
          <ObjectType>
            <type>Meter</type>
          </ObjectType>
        </Object>
        <ObjectIdentificationRevision>
          <revisionKind>remove</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
            <mRID>a41addcf-3542-4b69-b740-ca3abcf6384c</mRID>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
      </ObjectIdentificationRevisionSet>
    </ObjectIdentificationRevisionSets>
  </Payload>
</CreateObjectIdentificationRevisionSetsRequestMessage>
```

Figure F.6 – Remove the mRID of the meter object

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

F.8 Add an mRID and modify name two of the meter object

The example shown in Figure F.7 is a create(ObjectIdentificationRevisionSets) request message that causes the responding system to add a new mRID bfd736ae-19c0-4757-80e7-c922f91bd3eb to the meter object identified by the RFID tag 47CC907448. It also modifies name two of the meter object so that this contains the MAC address 98-5F-D3-42-05-39. The resulting state of the meter object's identification is shown in Table F.9.

Table F.9 – Add an mRID and modify name two of the meter object

		Before Step	After Step
	Object Type	Meter	Meter
	mRID		**bfd736ae-19c0-4757-80e7-c922f91bd3eb**
Name One	Name.name		
	NameType.name		
	NameTypeAuthority.name		
Name Two	Name.name	47CC907448	**98-5F-D3-42-05-39**
	NameType.name	RFID Tag	**MAC Address**
	NameTypeAuthority.name	UtilityXYZ	**Itron**
Name Three	Name.name	698877690	698877690
	NameType.name	AssetNumber	AssetNumber
	NameTypeAuthority.name		

```
<CreateObjectIdentificationRevisionSetsRequestMessage
xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>ObjectIdentificationRevisionSets</Noun>
    <Timestamp>2021-12-17T09:30:40Z</Timestamp>
    <MessageID>26fe0aef-2031-4bb6-8b27-6f50b102b67c</MessageID>
    <CorrelationID>8422842c-25c4-4944-af56-2d45dd764c55</CorrelationID>
    <Comment>ObjectIdentificationRevisionSets Step 7</Comment>
  </Header>
  <Payload>
    <ObjectIdentificationRevisionSets
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#">
      <ObjectIdentificationRevisionSet>
        <mRID>98814df5-0159-42ac-bae4-0d337c13d52a</mRID>
        <effectiveDateTime>2021-12-17T09:30:47Z</effectiveDateTime>
        <Object>
          <Names>
            <name>47CC907448</name>
            <NameType>
              <name>RFID Tag</name>
              <NameTypeAuthority>
                <name>UtilityXYZ</name>
              </NameTypeAuthority>
            </NameType>
          </Names>
          <ObjectType>
            <type>Meter</type>
          </ObjectType>
        </Object>
        <ObjectIdentificationRevision>
          <revisionKind>add</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            <mRID>bfd736ae-19c0-4757-80e7-c922f91bd3eb</mRID>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
        <ObjectIdentificationRevision>
          <revisionKind>remove</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
            <Names>
              <name>47CC907448</name>
              <NameType>
                <name>RFID Tag</name>
                <NameTypeAuthority>
                  <name>UtilityXYZ</name>
                </NameTypeAuthority>
              </NameType>
            </Names>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
        <ObjectIdentificationRevision>
          <revisionKind>add</revisionKind>
          <sequenceNumber>2</sequenceNumber>
          <IdentifiedObjectRevision>
            <Names>
              <name>98-5F-D3-42-05-39</name>
              <NameType>
                <name>MAC Address</name>
                <NameTypeAuthority>
                  <name>Itron</name>
                </NameTypeAuthority>
              </NameType>
            </Names>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
      </ObjectIdentificationRevisionSet>
    </ObjectIdentificationRevisionSets>
  </Payload>
</CreateObjectIdentificationRevisionSetsRequestMessage>
```

Figure F.7 – Add an mRID and modify name two of the meter object

F.9 Modify name two of the meter object and add name one of the customer object

The example shown in Figure F.8 is a create(ObjectIdentificationRevisionSets) request message that causes the responding system to modify name two of the meter object identified by the asset number 698877690 to be 2.16.840.1.114416.0.334983423, an ESN. It also adds a new name one "ACME Tool Company" to the customer object identified by the mRID dbe9fba9-a67a-4be4-92b1-62c1c96abc18. The resulting states of the objects' identifications are shown in Table F.10 and Table F.11.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table F.10 – Modify name two of the meter object

		Before Step	After Step
	Object Type	Meter	Meter
	mRID	bfd736ae-19c0-4757-80e7-c922f91bd3eb	bfd736ae-19c0-4757-80e7-c922f91bd3eb
Name One	Name.name		
	NameType.name		
	NameTypeAuthority.name		
Name Two	Name.name	98-5F-D3-42-05-39	**2.16.840.1.114416.0.334983423**
	NameType.name	MAC Address	**ESN**
	NameTypeAuthority.name	Itron	**Itron**
Name Three	Name.name	698877690	698877690
	NameType.name	AssetNumber	AssetNumber
	NameTypeAuthority.name		

Table F.11 – Add name one of the customer object

		Before Step	After Step
	Object Type	Customer	Customer
	mRID	dbe9fba9-a67a-4be4-92b1-62c1c96abc18	dbe9fba9-a67a-4be4-92b1-62c1c96abc18
Name One	Name.name		**ACME Tool Company**
	NameType.name		**CompanyName**
	NameTypeAuthority.name		**ACME Tool Company**
Name Two	Name.name		
	NameType.name		
	NameTypeAuthority.name		
Name Three	Name.name		
	NameType.name		
	NameTypeAuthority.name		

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateObjectIdentificationRevisionSetsRequestMessage
xmlns="http://iec.ch/TC57/2021/schema/message"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>ObjectIdentificationRevisionSets</Noun>
    <Timestamp>2021-12-17T09:30:40Z</Timestamp>
    <MessageID>f53b1c93-a09a-41f9-bf4d-63b692af4128</MessageID>
    <CorrelationID>5e5eb400-7529-425f-984c-568c3af2c650</CorrelationID>
    <Comment>ObjectIdentificationRevisionSets Step 8</Comment>
  </Header>
  <Payload>
    <ObjectIdentificationRevisionSets
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#">
      <ObjectIdentificationRevisionSet>
        <mRID>98814df5-0159-42ac-bae4-0d337c13d52a</mRID>
        <effectiveDateTime>2021-12-17T09:30:47Z</effectiveDateTime>
        <Object>
          <mRID>dbe9fba9-a67a-4be4-92b1-62c1c96abc18</mRID>
          <ObjectType>
```

```
            <type>Customer</type>
          </ObjectType>
        </Object>
        <ObjectIdentificationRevision>
          <revisionKind>add</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
            <Names>
              <name>ACME Tool Company</name>
              <NameType>
                <name>CompanyName</name>
                <NameTypeAuthority>
                  <name>ACME Tool Company</name>
                </NameTypeAuthority>
              </NameType>
            </Names>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
      </ObjectIdentificationRevisionSet>
      <ObjectIdentificationRevisionSet>
        <mRID>b6548ed8-fc39-4234-b303-b457f29afb74</mRID>
        <effectiveDateTime>2021-12-17T09:30:47Z</effectiveDateTime>
        <Object>
          <Names>
            <name>698877690</name>
            <NameType>
              <name>AssetNumber</name>
            </NameType>
          </Names>
          <ObjectType>
            <type>Meter</type>
          </ObjectType>
        </Object>
        <ObjectIdentificationRevision>
          <revisionKind>add</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
            <Names>
              <name>2.16.840.1.114416.0.334983423</name>
              <NameType>
                <name>ESN</name>
                <NameTypeAuthority>
                  <name>Itron</name>
                </NameTypeAuthority>
              </NameType>
            </Names>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
        <ObjectIdentificationRevision>
          <revisionKind>remove</revisionKind>
          <sequenceNumber>2</sequenceNumber>
          <IdentifiedObjectRevision>
            <Names>
              <name>98-5F-D3-42-05-39</name>
              <NameType>
                <name>MAC Address</name>
                <NameTypeAuthority>
                  <name>Itron</name>
                </NameTypeAuthority>
              </NameType>
            </Names>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
      </ObjectIdentificationRevisionSet>
    </ObjectIdentificationRevisionSets>
  </Payload>
</CreateObjectIdentificationRevisionSetsRequestMessage>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure F.8 – Add name one of the customer object and modify name two of the meter object

F.10 Add name two of the customer object to carry a new IdentifiedObject.name

The example shown in Figure F.9 is a create(ObjectIdentificationRevisionSets) request message that causes the responding system to modify the customer object identified by the mRID dbe9fba9-a67a-4be4-92b1-62c1c96abc18 by adding an IdentifiedObject.name of "ACME" as the NameType.description of name two. The resulting state of the customer object's identification is shown in Table F.12.

Table F.12 – Add name two of the customer object to carry a new IdentifiedObject.name

		Before Step	After Step
	Object Type	Customer	Customer
	mRID	dbe9fba9-a67a-4be4-92b1-62c1c96abc18	dbe9fba9-a67a-4be4-92b1-62c1c96abc18
Name One	Name.name	ACME Tool Company	ACME Tool Company
	NameType.name	CompanyName	CompanyName
	NameTypeAuthority.name	ACME Tool Company	ACME Tool Company
Name Two	Name.name		**ACME**
	NameType.name		**IdentifiedObject.name**
	NameType.description		**Reserved by IEC TC57 WG13**
	NameTypeAuthority.name		**IEC TC57 WG13**
Name Three	Name.name		
	NameType.name		
	NameTypeAuthority.name		

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateObjectIdentificationRevisionSetsRequestMessage
xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>ObjectIdentificationRevisionSets</Noun>
    <Timestamp>2021-12-17T09:30:40Z</Timestamp>
    <MessageID>f53b1c93-a09a-41f9-bf4d-63b692af4128</MessageID>
    <CorrelationID>5e5eb400-7529-425f-984c-568c3af2c650</CorrelationID>
    <Comment>ObjectIdentificationRevisionSets Step 9</Comment>
  </Header>
  <Payload>
    <ObjectIdentificationRevisionSets
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#">
      <ObjectIdentificationRevisionSet>
        <mRID>98814df5-0159-42ac-bae4-0d337c13d52a</mRID>
        <effectiveDateTime>2021-12-17T09:30:47Z</effectiveDateTime>
        <Object>
          <mRID>dbe9fba9-a67a-4be4-92b1-62c1c96abc18</mRID>
          <ObjectType>
            <type>Customer</type>
          </ObjectType>
        </Object>
        <ObjectIdentificationRevision>
          <revisionKind>add</revisionKind>
          <sequenceNumber>1</sequenceNumber>
          <IdentifiedObjectRevision>
            <Names>
              <name>ACME</name>
              <NameType>
                <description>Reserved by IEC TC57 WG13</description>
                <name>IdentifiedObject.name</name>
                <NameTypeAuthority>
                  <name>IEC TC57 WG13</name>
                </NameTypeAuthority>
              </NameType>
            </Names>
          </IdentifiedObjectRevision>
        </ObjectIdentificationRevision>
      </ObjectIdentificationRevisionSet>
    </ObjectIdentificationRevisionSets>
  </Payload>
</CreateObjectIdentificationRevisionSetsRequestMessage>
```

Figure F.9 – Add name two of the customer object to carry a new IdentifiedObject.name

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annex G
(informative)

Example IEC 61968-100:2021 messages illustrating the cross-product rule

G.1 General

This clause provides examples showing the operation of the cross-product rule (8.4.2.5).

G.2 Specifying "OR" operations over the selection criteria

The example shown in Figure G.1 is a get(MeterConfig) request that specifies two meter names: M001 and M002. Because these selection parameter constraints belong to the same class of criteria – in this example, the selection is over the specified <Meter><Names> – these are interpreted as a logical "OR" operation (8.4.2.2). That is, the responding system is requested to return all meter configuration data that match either meter M001 or meter M002.

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-03-01T08:04:56-06:00</Timestamp>
    <MessageID>e94d334d-a1ee-4a30-b780-9e4c775cdc5f</MessageID>
    <CorrelationID>705a9743-f3bd-4f5a-b1d9-a7198bb6bbcd</CorrelationID>
  </Header>
  <Payload>
    <GetMeterConfig xmlns="http://iec.ch/TC57/2011/GetMeterConfig#">
      <Meter>
        <Names>
          <name>M001</name>
        </Names>
      </Meter>
      <Meter>
        <Names>
          <name>M002</name>
        </Names>
      </Meter>
    </GetMeterConfig>
  </Payload>
</GetRequestMessage>
```

Figure G.1 – Example get(MeterConfig) request illustrating a logical "OR" operation

G.3 Specifying "AND" operations over the selection criteria

The request message shown in Figure G.2 specifies a single reading type, a single a time-schedule and a single meter. Because these selection parameter constraints belong to different classes of criteria, these are interpreted as a logical "AND" operation (8.4.2.3). The responding system returns all meter readings that match the specified reading type, the specified time-schedule and the specified meter.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2012-10-02T14:16:09Z</Timestamp>
    <MessageID>d1d5c311-1e9f-401b-b4b9-7e73d25de404</MessageID>
    <CorrelationID>d8b6c828-e5f6-443e-b872-805ba4e3b2d8</CorrelationID>
  </Header>
  <Payload>
    <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#">
      <EndDevice>
        <Names>
          <name>meter1</name>
        </Names>
      </EndDevice>
      <ReadingType>
        <Names>
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0</name>
        </Names>
      </ReadingType>
      <TimeSchedule>
       <scheduleInterval>
        <end>2013-07-25T09:40:00Z</end>
        <start>2013-07-25T09:40:00Z</start>
       </scheduleInterval>
      </TimeSchedule>
    </GetMeterReadings>
  </Payload>
</GetRequestMessage>
```

Figure G.2 – Example get(MeterReadings) request illustrating a logical "AND" operation

G.4 Specifying a combination of logical "AND" and "OR" operations

A request such as the one shown in Figure G.3 depicts how logical "AND" and "OR" operations may be combined in a single request (8.4.2.4). In this example, the request specifies those meter readings for which:

- the meter name is either meter1 or meter2
- and a ReadingType of either 0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0 or 0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.73.0
- and the specified time range.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2012-10-02T14:16:09Z</Timestamp>
    <MessageID>d1d5c311-1e9f-401b-b4b9-7e73d25de404</MessageID>
    <CorrelationID>d8b6c828-e5f6-443e-b872-805ba4e3b2d8</CorrelationID>
  </Header>
  <Payload>
    <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#">
      <EndDevice>
        <Names>
          <name>meter1</name>
        </Names>
      </EndDevice>
      <EndDevice>
        <Names>
          <name>meter2</name>
        </Names>
      </EndDevice>
      <ReadingType>
        <Names>
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0</name>
        </Names>
      </ReadingType>
      <ReadingType>
        <Names>
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.73.0</name>
        </Names>
      </ReadingType>
      <TimeSchedule>
       <scheduleInterval>
        <end>2013-07-25T09:40:00Z</end>
        <start>2013-07-25T09:40:00Z</start>
       </scheduleInterval>
      </TimeSchedule>
    </GetMeterReadings>
  </Payload>
</GetRequestMessage>
```

Figure G.3 – Example request message illustrating combined logical "AND" and "OR" operations

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

G.5 Example of multiple AbstractPayloadType profiles in a single message

The example in Figure G.4 shows a request message that contains two `<GetMeterReadings>` profiles. The responding system is expected to process this as if they were two separate request messages that share the same header information. That is, the responding system should return those meter read data that match the selection criteria specified in either one or the other selection criteria (8.4.2.5).

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterReadings</Noun>
    <Timestamp>2012-10-02T14:16:09Z</Timestamp>
    <MessageID>c6d2cb99-0b0c-4532-ad16-a28c7cabdcbe</MessageID>
    <CorrelationID>cca4968f-9163-4c8e-8fb6-e43a79a74d06</CorrelationID>
  </Header>
  <Payload>
    <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#">
      <EndDevice>
        <Names>
          <name>meter1</name>
        </Names>
      </EndDevice>
      <ReadingType>
        <Names>
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0</name>
        </Names>
      </ReadingType>
      <TimeSchedule>
        <scheduleInterval>
          <end>2013-07-25T09:40:00Z</end>
          <start>2013-07-25T09:40:00Z</start>
        </scheduleInterval>
      </TimeSchedule>
    </GetMeterReadings>
    <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#">
      <EndDevice>
        <Names>
          <name>meter2</name>
        </Names>
      </EndDevice>
      <ReadingType>
        <Names>
          <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.73.0</name>
        </Names>
      </ReadingType>
      <TimeSchedule>
        <scheduleInterval>
          <end>2013-07-25T09:40:00Z</end>
          <start>2013-07-25T09:40:00Z</start>
        </scheduleInterval>
      </TimeSchedule>
    </GetMeterReadings>
  </Payload>
</GetRequestMessage>
```

Figure G.4 – Example of multiple AbstractPayloadType profiles in a single message

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annex H
(informative)

Validation of received messages

H.1 General

This clause provides examples illustrating how a received IEC 61968-100 message may be validated against the `Message.xsd` (Clause A.2) and other, application-specific XSD files.

It general, it should be possible to validate an entire message in a single step even though the message may reference several, application-specific namespaces.

H.2 Example message validation using Altova XMLSpy

When using a static tool such as Altova XMLSpy[2], the `Message.xsd` file, the application-specific `XSD` profile file and the message to be validated are typically located in the same folder. For this to work, the file containing the message to be validated must be edited by adding appropriate `xsi:schemaLocation` statements. These specify the locations of the `Message.xsd` and `XSD` profile files acording to their respective namespaces.

An example message file is shown in Figure H.1. The XSD `xsi:schemaLocation` statements are shown highlighted.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

2 http://www.altova.com – XMLSpy is the trade name of a product supplied by Altova GmbH. This information is given for the convenience of users of this document and does not constitute an endorsement by the IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

```
<CreatedEventMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://iec.ch/TC57/2021/schema/message Message.xsd">
  <Header>
    <Verb>created</Verb>
    <Noun>EndDeviceEvents</Noun>
    <Timestamp>2013-10-14T11:31:57Z</Timestamp>
    <MessageID>6de7556b-b370-4040-9be5-1ce75838c6a4</MessageID>
    <CorrelationID>806454a3-8ecb-46b5-a296-e0e2e3c9d8ea</CorrelationID>
  </Header>
  <Payload>
    <EndDeviceEvents xmlns="http://iec.ch/TC57/2011/EndDeviceEvents#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://iec.ch/TC57/2011/EndDeviceEvents#
EndDeviceEvents.xsd">
      <EndDeviceEvent>
        <createdDateTime>2013-10-14T11:31:57Z</createdDateTime>
        <EndDeviceEventType ref="3.8.0.215"/>
        <UsagePoint>
          <Names>
            <name>M1001</name>
          </Names>
        </UsagePoint>
      </EndDeviceEvent>
      <EndDeviceEvent>
        <createdDateTime>2013-10-14T11:31:57Z</createdDateTime>
        <EndDeviceEventType ref="3.8.0.215"/>
        <UsagePoint>
          <Names>
            <name>M1002</name>
          </Names>
        </UsagePoint>
      </EndDeviceEvent>
    </EndDeviceEvents>
  </Payload>
</CreatedEventMessage>
```

Figure H.1 – Example message suitably modified to specify the locations of the relevant XSD files

H.3 Example message validation using a Java validator

When using a programmatic validator such as one of the many available Java-based XSD validators, it is generally not convenient to edit the message to be validated. Instead, it may be more convenient to "pre-load" the `Message.xsd` file by adding appropriate `xs:import` statements near the top of this file. An example is shown in Figure H.2.

```
<xs:schema xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://iec.ch/TC57/2021/schema/message"
    elementFormDefault="qualified" attributeFormDefault="unqualified"
    version="1.0.0">
      <xs:include schemaLocation="AbstractPayloadType.xsd"/>
     <xs:import namespace="http://iec.ch/TC57/2011/GetMeterReadings#"
         schemaLocation="GetMeterReadings.xsd"/>
     <xs:import namespace="http://iec.ch/TC57/2011/EndDeviceControls#"
         schemaLocation="EndDeviceControls.xsd"/>
     <xs:import namespace="http://iec.ch/TC57/2011/EndDeviceEvents#"
         schemaLocation="EndDeviceEvents.xsd"/>
     <xs:element name="RequestMessage" type="RequestMessageType">
     . . . .
```

Figure H.2 – Example of local modifications made to the Message.xsd file

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annex I
(informative)

Significant changes between IEC 61968-100:2013 and IEC 61968-100:2021

Table I.1 lists the significant changes introduced in this current edition of IEC 61968-100:2021 since the publication of the previous edition of this document, IEC 61968-100:2013.

Table I.2 and Table I.3 list the detailed technical changes introduced in IEC 61968-100:2021, and provide appropriate suggestions for upgrading from IEC 61968-100:2013.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table I.1 – Significant changes introduced in IEC 61968-100:2021 since IEC 61968-100:2013

Change	Reason for change
IEC 61968-100:2013 provided no normative definitions for the various message archetypes, message exchange patterns, nor for the elements used in the message structures. These omissions have been rectified in IEC 61968-100:2021.	
The `Message.xsd` file has generally been "tightened up" and made more precise.	These changes allow for stronger type checking.
The `<RequestMessage>` root element has been replaced by those listed in Table 5.	
The `<ResponseMessage>` root element has been replaced by those listed in Table 5.	
The `<EventMessage>` root element has been replaced by those listed in Table 6.	
The elements of some data structures have been reordered and the cardinalities of some elements have also been changed.	
The set of error codes defined in 7.6.4.7 has been made normative.	
It is now possible to extend the set of defined error codes (7.6.4.7) by defining an appropriate, application-dependent namespace (7.6.4.8).	This allows other working groups or organisations to define their own set of error codes independently of one another.
There are now normatively defined means for adding to, replacing or removing parts of an existing object (8.3.5).	IEC 61968-100:2013 provided no normative definition for adding to or replacing part of an object.
There is now a normatively defined means for cancelling a request in progress (8.5). The concept of a transaction id has been introduced (7.9).	IEC 61968-100:2013 provided no normative definition for cancelling an operation in progress.
There is now a normatively defined means for modifying the set of identifiers (`<mRID>` and `<Names>` elements) associated with an object (8.6).	The `LegacyObjectIdentificationRevisionSets.xsd` profile (8.7) is new to IEC 61968-100:2021. This profile allows a requesting system to query and update the set of identifiers associated with a server-side object.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Table I.2 – Detailed changes introduced from IEC 61968-100:2013 to IEC 61968-100:2021

Change Made Since IEC 61968-100:2013	Reason for change	Recommended Upgrade Path
The application-specific data sections of the `<Request>` and `<Payload>` elements are now defined as an `<AbstractDataType>` rather than `xs:any`.	This change has been made in order to ensure compliance with a memorandum of understanding between the IEC and UN/CEFACT [1].	Existing application-specific profiles must be updated so that they inherit from `<AbstractDataType>` (7.7.5, A.3). Sample source code to facilitate this conversion may be found here: https://github.com/TC57WG14/Part-100-Ed-2-XSD-Updater
The `<Header><Verb>` element was defined as a xs:string in IEC 61968-100:2013. This is now defined as an enumeration type.	This change allows for stronger type checking.	No changes to application code should be required.
The `<Header><MessageSequence>` element is newly defined.	This element was not present in IEC 61968-100:2013.	7.5.2 contains a definition for this element.
The `<Header><MessageID>` element was defined as a xs:string in IEC 61968-100:2013. This is now defined as a pattern.	This change allows for stronger type checking.	This element should be supplied as a UUID as defined be RFC 4122.
The `<Header><CorrelationID>` element was defined as a xs:string in IEC 61968-100:2013. This is now defined as a pattern. This element is now mandatory in all messages.	This change allows for stronger type checking.	This element should be supplied as a UUID as defined by RFC 4122.
The following element had no normative definition in IEC 61968-100:2013: `<Header><AsynchReplyFlag>`.	IEC 61968-100:2013 provided no normative definition for this element.	This element now has a normative definition.
The `<Reply><Error>` element is now renamed as `<Reply><Status>`.	This element may contain informational conditions and not just errors.	
The following element has been updated: `<Reply><Status><Level>`.	IEC 61968-100:2013 provided no normative definition for this element.	New enumerated values for this element replace the old ones (7.6.4.6).
The IDKind, IDKindType and IDAtts definitions in the IEC 61968-100:2013 version of the Message.xsd file have been replaced.	IEC 61968-100:2013 provided no normative definition for these elements.	These are replaced by the IdentifierType (8.6).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Change Made Since IEC 61968-100:2013	Reason for change	Recommended Upgrade Path
The <Payload><OperationSet><Operation><Verb> and <Payload><OperationSet><Operation><Noun> elements have a cardinality of 1.	These elements were previously of cardinality 0 to 1. There is no reason for these to be optional.	
The <Payload><OperationSet><OperationSetId> element has been introduced (7.7.3.2).	This allows the <Reply><Status> element of a response message to be tied to the original request message (7.6.4.5).	

Table I.3 – Elements removed from IEC 61968-100:2013

Change Made Since IEC 61968-100:2013	Reason for change	Recommended Upgrade Path
The following verbs have been removed: close and closed.	IEC 61968-100:2013 provided no normative definitions for these verbs.	Possibly use the delete and deleted verbs instead. Possibly use the change, changed, update or updated verbs to change the status of an application-specific object.
The following element has been removed: <Header><Revision>.	IEC 61968-100:2013 provided no normative definition for this element.	
The following element has been removed: <Header><Context>.	IEC 61968-100:2013 provided no normative definitions for this element.	Use the <Header><Source> and <Header><Property> elements instead.
The following element has been removed: <Header><ReplayDetection>.	Security concerns lie outside the scope of IEC 61968:2018.	See the IEC 62351 series for definitions and practices relating to the secure transmission of messages.
The application-specific data (xs:any) section of the message <Header> element is now removed from IEC 61968-100:2013.	IEC 61968-100:2013 provided no normative definition for this section.	Possibly use the <Header><Property> element to provide non-normative, local extensions of the header information.
The following elements have been removed: <Request><StartTime> and <Request><EndTime>	IEC 61968-100:2013 provided no normative definitions for these elements.	These elements should be added to the application-specific profiles where appropriate.
The <Request> element has been removed.	There is no need for a separate element for carrying application-specific data.	Information that was previously in the <Request> element should now be carried in the <Payload> element

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Change Made Since IEC 61968-100:2013	Reason for change	Recommended Upgrade Path
The following element has been removed: `<Payload><Compressed>`	IEC 61968-100:2013 provided no normative definition for this element.	A new element `<CompressedPayload>` has been introduced (7.8).
The following elements has been removed: `<Payload><Format>`	IEC 61968-100:2013 provided no normative definition for this element.	Use an application-specific profile that is defined in terms of `<AbstractDataType>` (8.5.6).
The following element has been removed: `<Payload><OperationSet><elementOperation>`	IEC 61968-100:2013 provided no normative definition for this element. Annex K of IEC 61968-9:2013 discusses how this element may be used to control the effect of the delete verb.	8.3.4 and 8.3.5.4 provide information on deleting an object either in its entirety or partially. Annex K of IEC 61968-9:2013 no longer applies.
The following elements have been removed: `<Reply><Error><Location>`; `<Reply><Error><details>`; `<Reply><Error><stackTrace>`; `<Reply><Error><xpath>`.	IEC 61968-100:2013 provided no normative definitions for these elements.	Non-normative information relating to the cause of an error may be placed in the `<Reply><Status><Reason>` element (7.6.4.5).
The following element has been removed: `<Reply><Error><object>`.		Use the `<Reply><Status><ObjectID>` element instead.
The following element has been removed: `<Reply><Error><Code>`.	IEC 61968-100:2013 provided no normative definition for this element.	New values for this element replace the old ones. Application-specific values may also be defined (7.6.4.5).
The following element has been removed: `<Reply><Error><relatedID>`.	This element is now subsumed by the `<Reply><Status><ObjectID>` element which now has a cardinality of 0 to many.	Use the `<Reply><Status><ObjectID>` element instead
The following element has been removed: `<Reply><Result>`.	This element contained no information that could not be otherwise gleaned from the other data in the `<Reply>` element.	This element is not needed.
The generic XSD MessageType has been removed.	No IEC 61968-100 messages corresponded to this type.	
The application-specific data (`xs:any`) section of the message `<Header>` and `<Reply>` elements has been removed	IEC 61968-100:2013 provided no normative definitions for these elements	

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Change Made Since IEC 61968-100:2013	Reason for change	Recommended Upgrade Path
The mechanism whereby an `<Payload><OperationSet>` may be used to request the deletion of part of an object is no longer applicable.	This is described in annex L of IEC 61968-9:2013 and was never an offical part of IEC 61968-100.	In order to request the partial deletion of an object, use the appropriate object management request (8.3.5.4).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Bibliography

[1] United Nations Centre for Trade Facilitation and Electronic Business – XML Naming and Design Rules Technical Specification Version 3.0 – 17 December 2009. Available from https://www.unece.org/fileadmin/DAM/cefact/xml/UNCEFACT%2BXML%2BNDR%2BV3p0.pdf

[2] RFC 4627, The application/json Media Type for JavaScript Object Notation (JSON). Available from https://tools.ietf.org/html/rfc4627

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

SOMMAIRE

AVANT-PROPOS 261
INTRODUCTION 263
1 Domaine d'application 264
1.1 Généralités 264
1.2 À propos du présent document 264
1.3 Éléments non couverts par le présent document 264
1.4 Considérations futures 264
1.4.1 Choix des mécanismes de codage 264
1.4.2 Choix des technologies de service Web 264
2 Références normatives 265
3 Termes, définitions et termes abrégés 265
3.1 Termes et définitions 265
3.2 Termes abrégés 266
4 Aperçu 266
4.1 Généralités 266
4.2 Format de message 266
4.3 Profils 267
4.4 Archétypes de message 267
4.5 Verbe et nom de l'en-tête 267
4.6 Exemple de messages IEC 61968-100 268
4.7 Modèles d'échange de messages 268
4.7.1 Généralités 268
4.7.2 Modèle d'échange de messages de demande/réponse 268
4.7.3 Modèle d'échange de messages d'événement non sollicité 269
4.7.4 Modèle d'échange de messages d'événement induit 269
4.8 Types de messages de base demande/réponse 269
5 Archétypes de message 270
5.1 Généralités 270
5.2 Message de demande 270
5.2.1 Généralités 270
5.2.2 Élément Root (racine) 271
5.2.3 Structure 271
5.2.4 Élément Header (en-tête) 271
5.2.5 Données spécifiques à l'application 271
5.2.6 Effets collatéraux d'un message de demande 271
5.3 Message de réponse 271
5.3.1 Généralités 271
5.3.2 Élément Root 272
5.3.3 Structure 272
5.3.4 Élément Header 272
5.3.5 Données spécifiques à l'application et notifications d'état 272
5.3.6 Message d'accusé de réception simple 273
5.4 Message d'événement 273
5.4.1 Généralités 273
5.4.2 Élément Root 273
5.4.3 Structure 273

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

5.4.4 Élément Header 274
5.4.5 Données spécifiques à l'application 274
5.5 Message de défaut 274
5.5.1 Généralités 274
5.5.2 Élément Root 274
5.5.3 Structure 274
5.5.4 Élément de réponse 274
6 Modèles d'échange de messages élémentaires 274
6.1 Généralités 274
6.2 Modèle d'échange de messages de demande/réponse 275
6.2.1 Généralités 275
6.2.2 Messages de réponse 275
6.3 Modèle d'échange de messages d'événement non sollicité 276
6.4 Modèle d'échange de messages d'événement induit 277
6.5 Modèles d'échange de messages d'accusé de réception simples 278
6.5.1 Généralités 278
6.5.2 Modèle d'échange de messages de demande/réponse avec messages d'accusé de réception simples 279
6.5.3 Modèle d'échange de messages d'événement non sollicité avec message d'accusé de réception simple 280
6.5.4 Modèle d'échange de messages d'événement induit avec messages d'accusé de réception simple 280
7 Informations détaillées du message 281
7.1 Généralités 281
7.2 Codage de message 282
7.3 Définition et validation de schéma 282
7.4 Structure de niveau supérieur 282
7.4.1 Généralités 282
7.4.2 Structure des messages de demande 282
7.4.3 Structure des messages de réponse 285
7.4.4 Structure des messages d'événement 287
7.5 Élément <Header> 290
7.5.1 Généralités 290
7.5.2 Sous-éléments de Message header (en-tête de message) 290
7.5.3 En-tête des messages de demande 292
7.5.4 En-tête des messages de réponse 294
7.5.5 En-tête des messages d'événement 296
7.6 Élément <Reply> (Réponse) 298
7.6.1 Généralités 298
7.6.2 Structure 299
7.6.3 Sous-éléments de Reply 299
7.6.4 Sous-élément <Reply><Status> 299
7.6.5 Combinaison des éléments <Reply><Status><ObjectID>, <OperarionSet>, <UpdateID> et <TransactionID> 307
7.6.6 Élément <Reply><Transaction> 307
7.7 Élément <Payload> 307
7.7.1 Généralités 307
7.7.2 Sous-éléments de Payload (charge utile) 307
7.7.3 Élément <Payload><OperationSet> 308

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.7.4 Élément <Payload><ObjectUpdate> 315
7.7.5 Données AbstractPayloadType 317
7.7.6 Charges utiles abstraites multiples 317
7.8 Élément <CompressedPayload> 317
7.8.1 Généralités 317
7.8.2 Structure 317
7.8.3 Éléments CompressedPayload 318
7.8.4 Sous-élément <CompressedPayload><CompressionType> 318
7.8.5 Sous-élément <CompressedPayload><Base64> 318
7.9 Type de transaction 318
8 Contraintes et conventions 319
8.1 Généralités 319
8.2 Données spécifiques à l'application 319
8.2.1 Généralités 319
8.2.2 AbstractPayloadType 320
8.2.3 Personnalisation spécifique à l'application 320
8.2.4 Données RDF 320
8.3 Demandes et notifications de gestion d'objets 320
8.3.1 Généralités 320
8.3.2 Message de demande de création 321
8.3.3 Message de demande de modification 322
8.3.4 Message de demande de suppression 322
8.3.5 Message de demande de mise à jour 323
8.3.6 Notifications de gestion d'objet 324
8.4 Demandes de récupération 324
8.4.1 Formuler une demande de récupération 324
8.4.2 Spécification de plusieurs contraintes dans un message de demande 324
8.4.3 Réponse à une demande de récupération 325
8.5 Annulation de transactions 325
8.5.1 Généralités 325
8.5.2 Modèle d'échange de message 325
8.5.3 ID transaction 326
8.5.4 Réception d'un ID transaction 327
8.5.5 Envoi d'un message de demande d'annulation 327
8.5.6 Réponse à un message de demande d'annulation 327
8.6 Identification d'objet 327
8.6.1 Généralités 327
8.6.2 Règles d'identification d'objet 328
8.6.3 Identification d'objets à l'aide d'un élément n <mRID> 328
8.6.4 Identification d'objets à l'aide d'un élément <Name> 328
8.6.5 Référencement d'objets 329
8.7 Profil ObjectIdentificationRevisionSets 330
8.7.1 Généralités 330
8.7.2 Structure 330
8.7.3 Utilisation du profil ObjectIdentificationRevisionSets 332
8.7.4 Ajout d'un ou de plusieurs nouveaux identifiants 332
8.7.5 Suppression d'un ou de plusieurs identifiants 333
8.7.6 Erreurs possibles concernant les demandes ObjectIdentificationRevisionSet 333

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

9 Transport des messages 333
9.1 Généralités 333
9.2 Transport sur les services Web SOAP 334
9.2.1 Généralités 334
9.2.2 Codage des messages IEC 61968-100 à l'intérieur des messages SOAP 334
9.2.3 Exemple de message SOAP 334
9.2.4 Mapping des messages IEC 61968-100 aux messages SOAP 334
9.2.5 Fichiers WSDL 335
9.3 Transport sur JMS 336
9.3.1 Généralités 336
9.3.2 Modèle d'échange de messages 337
9.3.3 Messages d'accusé de réception au niveau de l'application 337
9.3.4 En-têtes JMS 337
Annexe A (normative) Fichiers XSD 339
A.1 Généralités 339
A.2 Fichier Message.xsd 339
A.3 Fichier `AbstractPayloadType.xsd` 351
A.4 Fichier LegacyObjectIdentificationRevisionSets.xsd 352
A.5 Fichier `RDF.xsd` 357
Annexe B (normative) Modèles de fichier 359
B.1 Généralités 359
B.2 Fichier GenericMessageWsdlTemplate.txt 359
B.3 Fichier `StrongMessageXsdTemplate.txt` 365
B.4 Fichier `StrongMessageWsdlTemplate.txt` 377
Annexe C (informative) Exemple de profils XSD héritant d'AbstractPayloadType 384
C.1 Profil GetMeterReadings 384
C.2 Profil BinaryData 385
Annexe D (informative) Exemple de messages IEC 61968-100:2021 387
D.1 Généralités 387
D.2 Messages relatifs à l'instanciation d'objet 387
D.2.1 Généralités 387
D.2.2 Diagramme de séquence 387
D.2.3 Demande d'instanciation de compteur (message de demande) 388
D.2.4 Réponse d'instanciation de compteur (étape 4, cas de réussite) 390
D.2.5 Réponse d'instanciation de compteur (étape 4, cas d'échec) 390
D.3 Messages relatifs à la notification d'événement de l'instanciation d'objet 391
D.3.1 Généralités 391
D.3.2 Diagramme de séquence 391
D.3.3 Publication d'une instanciation de compteur (message d'événement) 392
D.3.4 Publication d'une instanciation de compteur (message d'accusé de réception simple) 395
D.4 Messages relatifs aux demandes d'informations concernant une instanciation d'objet 396
D.4.1 Généralités 396
D.4.2 Diagramme de séquence 396
D.4.3 Demande d'informations concernant une instanciation de compteur (message de demande) 397

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.4.4 Demande d'informations concernant une instanciation de compteur (message de réponse) 398
D.5 Messages utilisant un OperationSet pour assurer l'intégrité transactionnelle 399
D.5.1 Généralités 399
D.5.2 Diagramme de séquence 399
D.5.3 Utilisation d'un OperationSet pour assurer l'intégrité transactionnelle (message de demande) 400
D.5.4 Utilisation d'un OperationSet pour assurer l'intégrité transactionnelle (message de réponse) 403
D.6 Messages relatifs à la publication des données 403
D.6.1 Généralités 403
D.6.2 Diagramme de séquence 403
D.6.3 Publication des relevés de compteur (message d'événement) 404
D.6.4 Publication des relevés de compteur (message d'accusé de réception simple) 405
D.7 Messages relatifs à une requête de données sur demande donnant lieu à une seule réponse 406
D.7.1 Généralités 406
D.7.2 Diagramme de séquence 406
D.7.3 Message de relevé de compteur sur demande (message de demande) 407
D.7.4 Message de relevé de compteur sur demande (message de réponse) 408
D.8 Messages relatifs à une requête de données sur demande donnant lieu à plusieurs réponses 409
D.8.1 Généralités 409
D.8.2 Diagramme de séquence 409
D.8.3 Requête d'un ping de compteur sur demande (message de demande) 410
D.8.4 Requête d'un ping de compteur sur demande (message de réponse 1) 411
D.8.5 Requête d'un ping de compteur sur demande (message de réponse 2) 412
D.8.6 Requête d'un ping de compteur sur demande (message d'accusé de réception simple 2) 413
D.8.7 Requête d'un ping de compteur sur demande (message de réponse 3) 414
D.8.8 Requête d'un ping de compteur sur demande (message d'accusé de réception simple 3) 415
D.9 Messages relatifs à la demande de données historiques 416
D.9.1 Généralités 416
D.9.2 Diagramme de séquence 416
D.9.3 Relevé de compteur historique (message de demande) 417
D.9.4 Relevé de compteur historique (message de réponse) 417
D.10 Messages relatifs à la demande de données historiques (réponses diffusées sur plusieurs messages) 418
D.10.1 Généralités 418
D.10.2 Relevé de compteur historique (message de réponse 1 sur 4) 419
D.10.3 Relevé de compteur historique (message de réponse 2 sur 4) 419
D.10.4 Relevé de compteur historique (message de réponse 3 sur 4) 420
D.10.5 Relevé de compteur historique (message de réponse 4 sur 4) 421
D.11 Messages relatifs à plusieurs états dans une réponse 422
D.11.1 Généralités 422
D.11.2 Diagramme de séquence 423
D.11.3 Relevé de compteur historique (message de réponse) 424
D.12 Messages relatifs à un message d'événement induit 426
D.12.1 Généralités 426

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.12.2 Diagramme de séquence 426
D.12.3 Demande d'un branchement de compteur (message de demande) 427
D.12.4 Demande d'un branchement de compteur (message de réponse) 428
D.12.5 Demande d'un branchement de compteur (message d'événement) 429
D.12.6 Demande d'un branchement de compteur (message d'accusé de réception simple) 430
D.13 Utilisation de deux OperationSets pour séquencer plusieurs opérations avec intégrité transactionnelle 431
D.13.1 Généralités 431
D.13.2 Utilisation de deux OperationSets pour le relevé de deux compteurs avant de les débrancher (message de demande) 431
D.13.3 Utilisation de deux OperationSets pour le relevé de deux compteurs avant de les débrancher (message de réponse) 433
D.14 Messages relatifs à la suppression d'objet 434
D.14.1 Généralités 434
D.14.2 Diagramme de séquence 434
D.14.3 Demande de suppression d'un compteur (message de demande) 435
D.14.4 Demande de suppression d'un compteur (message de réponse) 436
D.15 Messages relatifs à l'annulation d'une seule transaction 436
D.15.1 Message de demande de commande de compteur 436
D.15.2 Message de réponse qui contient un seul ID transaction 437
D.15.3 Demande d'annulation (message de demande) 438
D.15.4 Demande d'annulation (message de réponse) 438
D.16 Messages relatifs à l'annulation de plusieurs transactions 439
D.16.1 Message de réponse qui contient plusieurs ID transaction 439
D.16.2 Demande d'annulation (message de demande) 440
D.16.3 Demande d'annulation (message de réponse) 441
D.17 Messages relatifs à l'échec de la validation XSD 441
D.17.1 Généralités 441
D.17.2 Message qui empêche la validation XSD (message de demande) 442
D.17.3 Message qui empêche la validation XSD (message de réponse) 442
D.18 Messages relatifs aux demandes malformées 443
D.18.1 Généralités 443
D.18.2 Message malformé (message de demande) 443
D.18.3 Message malformé (message de réponse) 444
Annexe E (informative) Exemple de messages IEC 61968-100:2021 pour la mise à jour d'objets 445
E.1 Généralités 445
E.2 CreateMeterConfig.xml 445
E.3 ModifyFormNumber.xml 446
E.4 RemoveFormNumber.xml 447
E.5 AddMeterMultiplier.xml 447
E.6 ModifyMeterMultiplierKind.xml 448
E.7 ModifyMeterMultiplierValue.xml 449
E.8 RemoveMeterMultiplierValue.xml 450
E.9 RemoveMeterMultiplier.xml 451
E.10 AddMeterSeal.xml 452
E.11 AddMeterSealKind.xml 452
E.12 ModifyMeterSealCondition.xml 453
E.13 RemoveMeterSealCondition.xml 454

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

E.14 RemoveMeterSeal.xml 455
E.15 ModifyProgramId.xml 456
E.16 RemoveProgramId.xml 456
E.17 AddChannel.xml 457
E.18 ModifyChannelReadingType.xml 458
E.19 RemoveChannel.xml 459
E.20 AddConnectDisconnectFunction.xml 460
E.21 RemoveConnectDisconnectFunction.xml 461
E.22 DeleteMeterConfig.xml 462
E.23 DeleteMeterConfig.xml 462
E.24 CreateMeterConfigs.xml 463
E.25 MultipleObjectChanges.xml 465
E.26 MultipleObjectChangesWithOperationSet.xml 467
Annexe F (informative) Exemples de messages ObjectIdentificationRevisionSets 471
F.1 Généralités 471
F.2 Ajout d'un mRID pour l'objet Compteur 471
F.3 Modification du Nom deux de l'objet Compteur 473
F.4 Ajout du Nom trois de l'objet Compteur 474
F.5 Suppression du Nom un de l'objet Compteur 476
F.6 Modification du mRID de l'objet Compteur 477
F.7 Suppression du mRID de l'objet Compteur 479
F.8 Ajout d'un mRID et modification du Nom deux de l'objet Compteur 480
F.9 Modification du Nom deux de l'objet Compteur et ajout du Nom un de l'objet Client 481
F.10 Ajout du Nom deux de l'objet Client pour porter un nouveau IdentifiedObject.name 484
Annexe G (informative) Exemple de messages IEC 61968-100:2021 de représentation de la règle du produit vectoriel 486
G.1 Généralités 486
G.2 Spécification des opérations "OR" sur les critères de sélection 486
G.3 Spécification des opérations "AND" sur les critères de sélection 486
G.4 Spécification d'une combinaison d'opérations "AND" et "OR" logiques 487
G.5 Exemple de plusieurs profils AbstractPayloadType Profiles dans un seul message 488
Annexe H (informative) Validation des messages reçus 490
H.1 Généralités 490
H.2 Exemple de validation de message à l'aide d'Altova XMLSpy 490
H.3 Exemple de validation de message à l'aide d'un programme de validation Java 491
Annexe I (informative) Modifications importantes entre l'IEC 61968-100:2013 et l'IEC 61968-100:2021 492
Bibliographie 498

Figure 1 – Structure d'un message de demande 271
Figure 2 – Structure d'un ResponseMessage 272
Figure 3 – Structure d'un EventMessage 273
Figure 4 – Structure d'un FaultMessage 274
Figure 5 – Modèle d'échange de messages de demande/réponse 275
Figure 6 – Modèle d'échange de messages d'événement non sollicité 277

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 7 – Modèle d'échange de messages d'événement induit 278
Figure 8 – Modèle d'échange de messages de demande/réponse avec messages d'accusé de réception simples 279
Figure 9 – Modèle d'échange de messages d'événement non sollicité avec messages d'accusé de réception simple 280
Figure 10 – Modèle d'échange de messages d'événement induit avec messages d'accusé de réception simple 281
Figure 11 – Structure de l'élément <CancelRequestMessage> 283
Figure 12 – Structure de l'élément <ChangeRequestMessage> 283
Figure 13 – Structure de l'élément <CreateRequestMessage> 283
Figure 14 – Structure de l'élément <DeleteRequestMessage> 284
Figure 15 – Structure de l'élément <ExecuteRequestMessage> 284
Figure 16 – Structure de l'élément <GetRequestMessage> 284
Figure 17 – Structure de l'élément <UpdateRequestMessage> 285
Figure 18 – Structure de l'élément <CancelResponseMessage> 285
Figure 19 – Structure de l'élément <ChangeResponseMessage> 286
Figure 20 – Structure de l'élément <CreateResponseMessage> 286
Figure 21 – Structure de l'élément <DeleteResponseMessage> 286
Figure 22 – Structure de l'élément <ExecuteResponseMessage> 287
Figure 23 – Structure de l'élément <GetResponseMessage> 287
Figure 24 – Structure de l'élément <UpdateResponseMessage> 287
Figure 25 – Structure de l'élément <CanceledEventMessage> 288
Figure 26 – Structure de l'élément <ChangedEventMessage> 288
Figure 27 – Structure de l'élément <CreatedEventMessage> 289
Figure 28 – Structure de l'élément <DeletedEventMessage> 289
Figure 29 – Structure de l'élément <ExecutedEventMessage> 289
Figure 30 – Structure de l'élément <UpdatedEventMessage> 290
Figure 31 – Structure de l'élément <Header> pour les messages de demande 293
Figure 32 – Structure de l'élément <Header> pour les messages de réponse 295
Figure 33 – Structure de l'élément <Header> pour les messages d'événement 297
Figure 34 – Structure de l'élément <Reply> 299
Figure 35 – Structure de l'élément <Reply><Status> 300
Figure 36 – Exemple d'indications d'erreur dans un élément <Reply> 306
Figure 37 – Structure de l'élément <Payload><OperationSet> pour les messages de demande 310
Figure 38 – Structure de l'élément <Payload><OperationSet> pour les messages de réponse 310
Figure 39 – Structure de l'élément <Payload><OperationSet> pour les messages d'événement 311
Figure 40 – Structure de l'élément <Payload><OperationSet><Operation> pour les messages de demande 313
Figure 41 – Structure de l'élément <Payload><OperationSet><Operation> pour les messages de réponse 314

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 42 – Structure de l'élément <Payload><OperationSet><Operation> pour les messages d'événement 315
Figure 43 – Structure de l'élément <Payload><ObjectUpdate> 316
Figure 44 – Structure de l'élément <CompressedPayload> 318
Figure 45 – Structure de TransactionType 319
Figure 46 – Modèle d'échange de message pour une demande d'annulation 326
Figure 47 – Schéma de dénomination général 329
Figure 48 – Structure d'un élément <ObjectID> 329
Figure 49 – Profil ObjectIdentificationRevisionSets 331
Figure 50 – Exemple de message SOAP 334
Figure A.1 – Contenu du fichier Message.xsd 351
Figure A.2 – Contenu du fichier AbstractPayloadType.xsd 351
Figure A.3 – Contenu du fichier LegacyObjectIdentificationRevisionSets.xsd 357
Figure A.4 – Contenu du fichier RDF.xsd 358
Figure B.1 – Contenu du fichier GenericMessageXsdTemplate.txt 364
Figure B.2 – Contenu du fichier StrongMessageXsdTemplate.txt 377
Figure B.3 – Contenu du fichier StrongMessageWsdlTemplate.txt 383
Figure C.1 – Exemple de profil XSD présentant l'héritage d'AbstractPayloadType 385
Figure C.2 – Exemple de profil XSD adapté à la transmission de données binaires 386
Figure D.1 – Exemple de diagramme de séquence pour une demande d'instanciation de compteur 388
Figure D.2 – Exemple de message de demande d'instanciation de compteur 390
Figure D.3 – Exemple de message de réponse d'instanciation de compteur (cas de réussite) 390
Figure D.4 – Exemple de message de réponse d'instanciation de compteur (cas d'échec) 391
Figure D.5 – Exemple de diagramme de séquence pour la publication d'une instanciation de compteur 392
Figure D.6 – Exemple de publication d'une instanciation de compteur (message d'événement) 395
Figure D.7 – Exemple de publication d'une instanciation de compteur (message d'accusé de réception simple) 396
Figure D.8 – Exemple de diagramme de séquence pour les demandes d'informations concernant une instanciation d'objet 397
Figure D.9 – Exemple de demande d'informations concernant une instanciation de compteur (message de demande) 398
Figure D.10 – Exemple de demande d'informations concernant une instanciation de compteur (message de réponse) 399
Figure D.11 – Exemple de diagramme de séquence pour une demande OperationSet 400
Figure D.12 – Exemple de message OperationSet pour assurer l'intégrité transactionnelle (message de demande) 403
Figure D.13 – Exemple de message OperationSet pour assurer l'intégrité transactionnelle (message de réponse) 403
Figure D.14 – Exemple de diagramme de séquence pour la publication des relevés de compteur 404
Figure D.15 – Exemple de publication de relevés de compteur de routine (message d'événement) 405

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure D.16 – Exemple de publication de relevés de compteur de routine (message d'accusé de réception simple) 406
Figure D.17 – Exemple de diagramme de séquence pour l'extraction de relevés de compteur sur demande 407
Figure D.18 – Exemple de message de requête de relevés de compteur sur demande 408
Figure D.19 – Exemple de message de réponse de relevés de compteur sur demande 409
Figure D.20 – Exemple de diagramme de séquence pour une requête de ping de compteur sur demande 410
Figure D.21 – Exemple de requête d'un ping de compteur sur demande (message de demande) 411
Figure D.22 – Exemple de requête d'un ping de compteur sur demande (message de réponse) 412
Figure D.23 – Exemple de requête d'un ping de compteur sur demande (message de réponse 2) 413
Figure D.24 – Exemple de requête d'un ping de compteur sur demande (message d'accusé de réception simple 2) 414
Figure D.25 – Exemple de requête d'un ping de compteur sur demande (message de réponse 3) 415
Figure D.26 – Exemple de requête d'un ping de compteur sur demande (message d'accusé de réception simple 3) 415
Figure D.27 – Exemple de diagramme de séquence pour l'extraction de relevés de compteur historiques 416
Figure D.28 – Exemple de demande de relevé de compteur historique (message de demande) 417
Figure D.29 – Exemple de demande de relevé de compteur historique (message de réponse) 418
Figure D.30 – Exemple de demande de relevé de compteur historique (message de réponse 1 sur 4) 419
Figure D.31 – Exemple de demande de relevé de compteur historique (message de réponse 2 sur 4) 420
Figure D.32 – Exemple de demande de relevé de compteur historique (message de réponse 3 sur 4) 421
Figure D.33 – Exemple de demande de relevé de compteur historique (message de réponse 4 sur 4) 422
Figure D.34 – Exemple de diagramme de séquence pour l'extraction de relevés de compteur historiques (message de demande) 423
Figure D.35 – Exemple de demande de relevé de compteur historique, cas de succès et de défaillances mélangés (message de demande) 424
Figure D.36 – Exemple de demande de relevé de compteur historique, cas de succès et de défaillances mélangés (message de réponse) 426
Figure D.37 – Exemple de diagramme de séquence pour la demande d'un branchement de compteur 427
Figure D.38 – Exemple de branchement de compteur (message de demande) 428
Figure D.39 – Exemple de branchement de compteur (message de réponse) 429
Figure D.40 – Exemple de branchement de compteur (message d'événement) 430
Figure D.41 – Exemple de branchement de compteur (message d'accusé de réception simple) 430
Figure D.42 – Utilisation de deux OperationSets pour le relevé de deux compteurs avant de les débrancher (message de demande) 432

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure D.43 – Utilisation de deux OperationSets pour le relevé de deux compteurs avant de les débrancher (message de réponse) 434
Figure D.44 – Exemple de diagramme de séquence pour la suppression d'un compteur 435
Figure D.45 – Exemple de suppression de compteur (message de demande) 436
Figure D.46 – Exemple de suppression de compteur (message de réponse) 436
Figure D.47 – Exemple de message de demande create(EndDeviceControls) 437
Figure D.48 – Exemple de message de réponse qui contient un seul ID transaction 438
Figure D.49 – Exemple de message de demande d'annulation 438
Figure D.50 – Exemple de réponse à un message de demande d'annulation 439
Figure D.51 – Exemple de message de réponse qui contient plusieurs ID transaction 440
Figure D.52 – Exemple de message de demande d'annulation 441
Figure D.53 – Exemple de réponse à un message de demande d'annulation 441
Figure D.54 – Exemple de message qui empêche la validation XSD (message de demande) 442
Figure D.55 – Exemple de message qui empêche la validation XSD (message de réponse) 443
Figure D.56 – Exemple de message malformé (message de demande) 444
Figure D.57 – Exemple de message malformé (message de réponse) 444
Figure E.1 – CreateMeterConfig.xml 446
Figure E.2 – ModifyFormNumber.xml 447
Figure E.3 – RemoveFormNumber.xml 447
Figure E.4 – AddMeterMultiplier.xml 448
Figure E.5 – ModifyMeterMultiplierKind.xml 449
Figure E.6 – ModifyMeterMultiplierValue.xml 450
Figure E.7 – RemoveMeterMultiplierValue.xml 451
Figure E.8 – RemoveMeterMultiplier.xml 451
Figure E.9 – AddMeterSeal.xml 452
Figure E.10 – AddMeterSealKind.xml 453
Figure E.11 – ModifyMeterSealCondition.xml 454
Figure E.12 – RemoveMeterDealCondition.xml 455
Figure E.13 – RemoveMeterSeal.xml 455
Figure E.14 – ModifyProgramId.xml 456
Figure E.15 – RemoveProgramId.xml 457
Figure E.16 – AddChannelId.xml 458
Figure E.17 – ModifyChannelReadingType.xml 459
Figure E.18 – RemoveChannel.xml 460
Figure E.19 – AddConnectDisconnectFunction.xml 461
Figure E.20 – RemoveConnectDisconnectFunction.xml 462
Figure E.21 – DeleteMeterConfig.xml 462
Figure E.22 – DeleteMeterConfig.xml 463
Figure E.23 – CreateMeterConfigs.xml 465
Figure E.24 – MultipleObjectChanges.xml 467
Figure E.25 – MultipleObjectChangesWithOperationSet.xml 470
Figure F.1 – Ajout d'un mRID pour l'objet Compteur 472

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure F.2 – Modification du Nom deux de l'objet Compteur 474
Figure F.3 – Ajout du Nom trois de l'objet Compteur 475
Figure F.4 – Suppression du Nom un de l'objet Compteur 477
Figure F.5 – Modification du mRID de l'objet Compteur 478
Figure F.6 – Suppression du mRID de l'objet Compteur 479
Figure F.7 – Ajout d'un mRID et modification du Nom deux de l'objet Compteur 481
Figure F.8 – Ajout du Nom un de l'objet Client et modification du Nom deux de l'objet Compteur 483
Figure F.9 – Ajout du Nom deux de l'objet Client pour porter un nouveau IdentifiedObject.name 485
Figure G.1 – Exemple de demande get(MeterConfig) qui présente une opération "OR" logique 486
Figure G.2 – Exemple de demande get(MeterReadings) qui présente une opération "AND" logique 487
Figure G.3 – Exemple de message de demande qui présente des opérations "AND" et "OR" logiques combinées 488
Figure G.4 – Exemple de plusieurs profils AbstractPayloadType dans un seul message 489
Figure H.1 – Exemple de message correctement modifié pour spécifier les emplacements des fichiers XSD pertinents 491
Figure H.2 – Exemple de modifications locales apportées au fichier Message.xsd 491

Tableau 1 – Liste des éléments racines de niveau supérieur dans les messages de demande 282
Tableau 2 – Liste des éléments racines de niveau supérieur dans les messages de réponse 285
Tableau 3 – Liste des éléments racines de niveau supérieur dans les messages d'événement 288
Tableau 4 – Éléments dans l'élément <Header> du message 290
Tableau 5 – Verbes utilisés dans les messages de demande 293
Tableau 6 – Verbes utilisés dans les messages d'événement 297
Tableau 7 – Sous-éléments de l'élément <Reply> 299
Tableau 8 – Sous-éléments de l'élément <Reply><Status> 301
Tableau 9 – Valeurs de l'élément <Reply><Status><Level> 302
Tableau 10 – Valeurs normatives de l'élément <Reply><Status><Code> 302
Tableau 11 – Sous-éléments de l'élément <Payload> 308
Tableau 12 – Sous-éléments de l'élément <Payload><OperationSet> 309
Tableau 13 – Sous-éléments de l'élément <Payload><OperationSet><Operation> 312
Tableau 14 – Sous-éléments de l'élément <Payload><ObjectUpdate> 316
Tableau 15 – Sous-éléments de l'élément <Payload><ObjectUpdate><UpdateAction> 317
Tableau 16 – Sous-éléments de l'élément <CompressedPayload> 318
Tableau 17 – Valeurs de l'élément <CompressedPayload><CompressionType> 318
Tableau 18 – Éléments d'un type de transaction 319
Tableau 19 – Sous-éléments d'un élément <ObjectID> 330
Tableau 20 – Éléments d'un profil ObjectIdentificationRevisionSets 332

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Tableau 21 – Noms de variable du modèle de fichier WSDL générique 335
Tableau 22 – Noms de variable du modèle XSD de message fortement typé 336
Tableau 23 – Noms de variables du modèle WSDL fortement typé 336
Tableau F.1 – États de dénomination initiaux de l'exemple d'objet Meter 471
Tableau F.2 – États de dénomination initiaux de l'exemple d'objet Client 471
Tableau F.3 – Ajout d'un mRID pour l'objet Compteur 472
Tableau F.4 – Modification du Nom deux de l'objet Compteur 473
Tableau F.5 – Ajout du Nom trois de l'objet Compteur 475
Tableau F.6 – Suppression du Nom un de l'objet Compteur 476
Tableau F.7 – Modification du mRID de l'objet Compteur 478
Tableau F.8 – Suppression du mRID de l'objet Compteur 479
Tableau F.9 – Ajout d'un mRID et modification du Nom deux de l'objet Compteur 480
Tableau F.10 – Modification du Nom deux de l'objet Compteur 482
Tableau F.11 – Ajout du Nom un de l'objet Client 482
Tableau F.12 – Ajout du Nom deux de l'objet Client pour porter un nouveau IdentifiedObject.name 484
Tableau I.1 – Modifications importantes présentées dans l'IEC 61968-100:2021 depuis l'IEC 61968-100:2013 493
Tableau I.2 – Modifications détaillées présentées de l'IEC 61968-100:2013 à l'IEC 61968-100:2021 494
Tableau I.3 – Éléments supprimés de l'IEC 61968-100:2013 495

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

INTÉGRATION D'APPLICATIONS POUR LES SERVICES ÉLECTRIQUES – INTERFACES SYSTÈME POUR LA GESTION DE LA DISTRIBUTION –

Partie 100: Profils de mise en œuvre IEC pour l'intégration d'application

AVANT-PROPOS

1) La Commission Électrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

L'IEC 61968-100 a été établie par le comité d'études 57 de l'IEC: Gestion des systèmes de puissance et échanges d'informations associés. Il s'agit d'une Norme internationale.

Les futures normes de cette série porteront le nouveau titre général cité ci-dessus. Le titre des normes qui existent déjà dans cette série sera mis à jour lors de leur prochaine édition.

La présente édition de l'IEC 61968-100:2022 annule et remplace l'IEC 61968-100:2013 parue en 2013. Cette édition constitue une révision technique.

a) Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente: L'IEC 61968-100:2022 a été affinée pour éliminer les ambiguïtés et les redondances. Les articles informatifs et normatifs sont marqués comme tels et, dans ce dernier cas, des formes verbales appropriées sont utilisées.

b) En règle générale, les messages définis par l'IEC 61968-100:2022 ne sont pas rétrocompatibles avec ceux de l'IEC 61968-100:2013. L'Annexe I répertorie les modifications techniques importantes apportées depuis la publication de l'IEC 61968-100:2013 et propose des suggestions appropriées pour passer à l'IEC 61968-100:2022.

Le texte de cette Norme internationale est issu des documents suivants:

Projet	Rapport de vote
57/2446/FDIS	57/2455/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à son approbation.

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Le présent document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/standardsdev/publications.

Une liste de toutes les parties de la série IEC 61968, publiées sous le titre général *Intégration d'applications pour les services électriques – Interfaces système pour la gestion de la distribution*, se trouve sur le site web de l'IEC.

NOTE

Les caractères d'imprimerie suivants sont utilisés:

- Texte littéral tel que des fragments de programme ou des éléments XML en `Courier New 10`.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous webstore.iec.ch dans les données relatives au document recherché. À cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de ce document indique qu'il contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer ce document en utilisant une imprimante couleur.

INTRODUCTION

Le présent document spécifie la manière dont l'IEC 61968 peut être rendue interopérable à l'aide de technologies communes d'intégration en entreprise. En particulier, il décrit la manière dont les charges utiles de message définies dans l'IEC 61968 (Partie 3 à Partie 9, et Partie 13), l'IEC 61970 et l'IEC 62325 doivent être échangées entre les systèmes à l'aide de mécanismes de transport tels que des services Web et Java Message Service.

Le présent document peut également se révéler utile au-delà des échanges d'informations définis pour ces Normes internationales, pour l'intégration des systèmes du marché ou l'intégration générale d'entreprise par exemple.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

INTÉGRATION D'APPLICATIONS POUR LES SERVICES ÉLECTRIQUES – INTERFACES SYSTÈME POUR LA GESTION DE LA DISTRIBUTION –

Partie 100: Profils de mise en œuvre IEC pour l'intégration d'application

1 Domaine d'application

1.1 Généralités

La présente partie de l'IEC 61968 définit la manière dont les messages peuvent être échangés entre des systèmes coopératifs afin de faciliter le transfert de données spécifiques à l'application. Ces données spécifiques à l'application incluent, entre autres, les charges utiles de message définies dans l'IEC 61968 (Partie 3 à Partie 9, et Partie 13), l'IEC 61970 et l'IEC 62325.

1.2 À propos du présent document

Le présent document fournit des définitions normatives:

- d'un ensemble d'archétypes de message (Article 5);
- d'un ensemble de modèles d'échange de message que les systèmes d'envoi et de réception sont réputés mettre en œuvre (Article 6);
- du format exact des messages qui doivent être transmis dans le cadre des différentes technologies d'intégration, y compris une description précise des informations que chaque message doit contenir (Article 7);
- d'un ensemble de contraintes et de conventions auxquelles les applications doivent adhérer afin de faciliter l'échange de messages à l'aide de l'IEC 61968-100 (Article 8);
- des informations détaillées de la manière dont il convient de mettre en œuvre les messages de l'IEC 61968-100 à l'aide de différents mécanismes de transport sous-jacents (Article 9).

1.3 Éléments non couverts par le présent document

Les considérations en matière de sécurité ne relèvent pas du domaine d'application de l'IEC 61968-100. Le présent document s'en remet à la série IEC 62351 quant aux définitions et pratiques relatives à la transmission sécurisée des messages.

1.4 Considérations futures

1.4.1 Choix des mécanismes de codage

Le présent document préconise d'utiliser le langage XML comme mécanisme de codage normatif de tous les messages définis par le présent document.

Les éditions futures de l'IEC 61968-100 peuvent spécifier d'autres méthodes normatives de codage, y compris la prise en charge de documents JSON (RFC 4627) dont la sémantique est définie par l'IEC CIM et la syntaxe par un schéma IETF JSON.

1.4.2 Choix des technologies de service Web

Le présent document donne des définitions normatives pour l'utilisation de services Web SOAP (9.2) et de Java Message Service (9.3) pour le transport des messages.

Les éditions futures de l'IEC 61968-100 peuvent spécifier d'autres technologies de service Web normatives (REST, par exemple).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 62325-301:2018 *Cadre pour les communications pour le marché de l'énergie - Partie 301: Extensions du modèle d'information commun (CIM) pour les marchés*

IEC TS 61968-2:2011, *Application integration at electric utilities – System interfaces for distribution management – Part 2: Glossary* (disponible en anglais seulement)

IEC 61968-9:2013 *Intégration d'applications pour les services électriques - Interfaces système pour la gestion de distribution - Partie 9*: *Interface pour le relevé et la commande des compteurs*

IEC 61968-11:2013 *Intégration d'applications pour les services électriques - Interfaces système pour la gestion de distribution - Partie 11*: *Extensions du modèle d'information commun (CIM) pour la distribution*

IEC 61970-301:2020 *Interface de programmation d'application pour système de gestion d'énergie (EMS-API) - Partie 301*: *Base de modèle d'information commun (CIM)*

IEC 61970-552:2016 *Interface de programmation d'application pour système de gestion d'énergie (EMS-API) - Partie 552*: *Format d'échange de modèle CIMXML*

ISO IEC 21320-1:2015, *Information technology – Document Container File - Part 1*: *Core* (disponible en anglais seulement)

ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times* (disponible en anglais seulement)

Java Message Service (version 1.0.2 9 novembre 1999). Disponible à l'adresse https://docs.oracle.com/cd/E19957-01/816-5904-10/816-5904-10.pdf.

RFC 1952 GZIP file format specification version 4.3. Disponible à l'adresse https://tools.ietf.org/html/rfc1952.

RFC 4122 A Universally Unique IDentifier (UUID) URN Namespace. Disponible à l'adresse https://tools.ietf.org/html/rfc4122.

RFC 4648 The Base16, Base32, and Base64 Data Encodings. Disponible à l'adresse https://tools.ietf.org/html/rfc4648.

SOAP (Simple Object Access Protocol). Disponible à l'adresse http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

3 Termes, définitions et termes abrégés

3.1 Termes et définitions

Pour les besoins de la présente spécification, les termes et définitions de l'IEC TS 61968- 2:2011 s'appliquent.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

3.2 Termes abrégés

Les termes abrégés suivants sont utilisés dans le présent document:

AMI	Advanced Metering Infrastructure (infrastructure de mesure avancée)
CIM	Common Information Model (modèle d'information commun)
CIS	Customer Information System (système d'information des clients)
HES	Head End System (système de tête de réseau)
HTTP	Hypertext Transfer Protocol (protocole de transfert hypertexte)
IEC	International Electrotechnical Commission (Commission électrotechnique internationale)
IETF	Internet Engineering Task Force
JMS	Java Message Service (service de messagerie Java)
JSON	JavaScript Object Notation (notation d'objet JavaScript)
MDMS	Meter Data Management System (système de gestion des données du compteur)
mRID	CIM master Resource Identifier (identifiant de la ressource principale CIM)
RDF	Resource Description Framework (cadre de description des ressources)
REST	REpresentational State Transfer (transfert d'état représentationel)
RFC	Request For Comments (demande de commentaires)
SOAP	Simple Object Access Protocol (protocole simple d'accès aux objets)
URL	Uniform Resource Locator (localisateur uniforme de ressource)
WS	Web Services (services Web)
WSDL	Web Services Definition Language (langage de définition de services Web)
XML	eXtensible Markup Language (langage de balisage extensible)
XSD	XML Schema Definition (définition de schéma XML)

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

4 Aperçu

4.1 Généralités

Le présent article donne un aperçu du présent document. Les informations détaillées des concepts décrits dans le présent article sont fournies dans les articles suivants du présent document.

4.2 Format de message

Un message IEC 61968-100 est un document XML. À ce titre, il est conforme au fichier XSD (XML Schema Definition) `Message.xsd`qui fait également partie intégrante du présent document (A.2) et peut être validé en fonction de celui-ci (Annexe H).

L'Annexe D, l'Annexe E, l'Annexe F et l'Annexe G donnent des exemples informatifs de messages IEC 61968-100. Ces exemples indiquent qu'un message IEC 61968-100 est composé d'informations d'en-tête et de données spécifiques à l'application.

Les informations d'en-tête sont contenues dans un élément `<Header>` à l'intérieur du message. Cet en-tête fournit des métadonnées qui décrivent la manière dont le reste du message doit être interprété et traité. Par exemple, l'en-tête indique si le message est une commande, une demande de données, une réponse à un message précédent ou une notification asynchrone.

Les données spécifiques à l'application se trouvent dans les éléments `<Reply>` ou `<Payload>`, en fonction de l'objet du message. L'élément `<Reply>` renvoie le statut au système demandeur (les indications de succès, de mise en garde ou d'erreur associées à une demande précédente, par exemple). L'élément `<Payload>` contient des données spécifiques à l'application conformes aux profils définis dans les autres normes IEC.

Le présent document n'a pas vocation à définir ces données spécifiques à l'application. Cette définition est laissée aux groupes de travail IEC ou à l'organisme local compétents. Toutefois, le présent document impose certaines contraintes et conventions que l'acheminement des messages, tels que les données, doit respecter (Article 8).

4.3 Profils

Les données spécifiques à l'application sont en général conformes à un profil. Un profil est une définition de schéma XML (XSD) qui a été spécifiée par le groupe de travail IEC ou l'organisme local correspondant. En règle générale, et bien que cela ne soit pas indispensable, un profil est déduit d'un modèle d'information commun (CIM) tel que défini dans des normes telles que l'IEC 62325-301, l'IEC 61968-11 et l'IEC 61970-301.

Un profil entre généralement dans l'une des catégories suivantes:

- il peut correspondre à une classe, avec éventuellement d'autres classes associées, qui appartiennent au CIM et qui sont liées les unes aux autres par héritage ou association;

  EXEMPLE 1 Un profil IEC 61968-9:2013 MeterReadings contient des informations relatives aux données lues dans un compteur intelligent. MeterReading est une classe définie dans le CIM.

  EXEMPLE 2 Un profil IEC 61968-9:2013 CustomerConfig contient des informations relatives à la gestion des clients. Customer est une classe CIM, à l'inverse de CustomerConfig.

- un profil peut décrire les liens entre les objets CIM plutôt que les objets eux-mêmes;

  EXEMPLE 3 Un profil IEC 61968-9:2013 MasterDataLinkageConfig spécifie la relation entre différents objets CIM tels que Customer, Meter, UsagePoint, etc. Toutefois, MasterDataLinkageConfig n'est pas lui-même une classe CIM.

- un profil peut décrire un ensemble de paramètres de requête par lequel une demande d'un ou de plusieurs objets CIM existants peut être formulée. Dans le présent document, il s'agit de paramètres de requête spécifiques à l'application;

  EXEMPLE 4 Un profil IEC 61968-9:2013 GetMeterReadings permet de définir les paramètres de requête par lesquels les données lues dans un ou plusieurs compteurs peuvent être extraites.

- un organisme local peut définir son propre ensemble de profils qui n'ont de sens qu'au sein dudit organisme.

4.4 Archétypes de message

Les messages sont classés selon qu'il s'agit de messages de demande, de réponse ou d'événement (notification asynchrone). Ces catégories sont appelées archétypes de message (Article 5).

4.5 Verbe et nom de l'en-tête

L'archétype de message peut être déduit du verbe de l'en-tête placé dans l'élément `<Header><Verb>`. Le verbe est un élément d'un ensemble fixe de verbes énumérés en 7.5.3.3, 7.5.4.3 et 7.5.5.3.

Un profil porte un nom qui l'identifie dans l'en-tête de message (MeterReadings, CustomerConfig ou MasterDataLinkageConfig, par exemple). Ce nom est appelé 'noun' de l'en-tête dans l'élément `<Header><Noun>` (les règles précises de spécification du nom sont plus compliquées que cette description abrégée. Les paragraphes 7.5.3.4, 7.5.4.4 et 7.5.5.4 donnent de plus amples informations).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Le verbe de l'en-tête et le nom de l'en-tête déterminent l'objet du message. Ils sont désignés par la terminologie `verb(noun)`.

4.6 Exemple de messages IEC 61968-100

Le paragraphe D.7.3 présente un message IEC 61968-100 simple qui indique les éléments `<Header>` et `<Payload>`. Dans cet exemple, les données spécifiques à l'application dans l'élément `<Payload>` sont définies conformément à l'IEC 61968-9:2013 et forment une demande d'extraction des données de relevé de compteur.

Le paragraphe D.7.4 décrit le message de réponse correspondant et présente un message IEC 61968-100 avec les éléments `<Header>`, `<Reply>` et `<Payload>`. L'élément `<Reply>` contient une notification d'état (une notification de succès, en l'occurrence). L'élément `<Payload>` contient les données de relevé de compteur renvoyées en réponse à la demande précédente. Ces données spécifiques à l'application sont également définies conformément à l'IEC 61968-9:2013.

4.7 Modèles d'échange de messages

4.7.1 Généralités

Le présent document définit plusieurs types de modèles d'échange de messages de base (Article 6). Un échange de messages IEC 61968-100 entre des systèmes coopératifs est composé d'un ou de plusieurs de ces modèles d'échange de messages.

4.7.2 Modèle d'échange de messages de demande/réponse

Le modèle d'échange de messages de demande/réponse (6.2) est utilisé lorsqu'il est nécessaire qu'un système demandeur requiert des données ou qu'une action soit réalisée par un système répondant. Le système répondant traite la demande et renvoie un ou plusieurs messages de réponse au système demandeur.

Le `<Header><Verb>` dans le message de demande doit être l'un parmi `cancel`, `create`, `delete`, `change`, `execute`, `get` ou `update`. Le verbe de l'en-tête dans le message de réponse est toujours `reply`.

EXEMPLE 1 Un message IEC 61968-9:2013 `create(CustomerConfig)` représente une demande entre un système et un autre pour instancier un objet CIM Customer à l'aide des données fournies dans la charge utile. Le nom de ce message est CustomerConfig et le profil CustomerConfig est utilisé dans la section de charge utile. Lors de la réception de ce type de message, le système de réception le traite, puis renvoie un message `reply(CustomerConfig)` à l'expéditeur de la demande. Ce message de réponse présente le statut de la demande "create".

Les messages qui donnent un exemple similaire peuvent être consultés à l'Article D.2.

À l'inverse du paradigme conventionnel de demande/réponse, le présent document permet d'associer plusieurs messages de réponse à une seule demande. Cela peut être le cas lorsqu'il est nécessaire de séparer des charges utiles de réponse plus volumineuses en plusieurs messages ou lorsque plusieurs opérations ont lieu en même temps et indépendamment les unes des autres.

EXEMPLE 2 Une demande IEC 61968-9:2013 get(MeterReadings) indique que le message donné est une demande de données de relevé de compteur spécifique. Le nom du profil dans le message de demande est GetMeterReadings. Le nom du profil dans la réponse correspondante est MeterReadings. La réception d'une demande get(MeterReadings) peut amener le système de tête de réseau (HES) à demander les données des compteurs dans son réseau AMI. Le retour vers le système de tête de réseau des données en provenance des compteurs peut prendre un certain temps. Le système de tête de réseau peut décider d'attendre de recevoir les données de tous les compteurs avant de les regrouper dans un seul message reply(MeterReadings) renvoyé à l'expéditeur de la demande. En variante, il peut choisir de renvoyer les données au demandeur au fur et à mesure que chaque compteur répond séparément sur le réseau AMI. Dans ce dernier cas, le seul message de demande peut donner lieu à plusieurs messages reply(MeterReadings) générés et envoyés au demandeur.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

4.7.3 Modèle d'échange de messages d'événement non sollicité

Le modèle d'échange de messages d'événement non sollicité (6.3), parfois appelé "fire and forget" (tire et oublie), est composé d'un seul message envoyé par un système à un autre et dont aucune réponse n'est attendue. Cela peut être le cas lorsque le message est envoyé après une action autonome à l'intérieur d'un système. Ce type de message est appelé message d'événement non sollicité.

L'élément `<Header><Verb>` dans un message d'événement non sollicité appartient à l'un des ensembles `canceled`, `created`, `changed`, `deleted`, `executed` ou `updated`.

EXEMPLE Un système de tête de réseau peut envoyer un message IEC 61968-9:2013 `created(EndDeviceEvents)` lorsqu'un compteur est confronté à une panne de courant ou pour envoyer une alarme de haute température.

Un exemple exhaustif peut être consulté à l'Article D.6.

4.7.4 Modèle d'échange de messages d'événement induit

Le modèle d'échange de messages d'événement induit (6.4) est composé de la demande/réponse suivie du modèle d'échange de messages d'événement non sollicité. Dans ce cas, le message d'événement associé est appelé message d'événement induit. La séquence complète des messages est composée de la demande d'origine, d'une seule réponse qui indique le statut de cette demande, puis d'aucun ou de plusieurs messages d'événements induits.

EXEMPLE Un système de gestion des données du compteur (MDMS) peut envoyer un message IEC 61968-9:2013 `create(EndDeviceControls)` à un système de tête de réseau pour demander de réaliser une opération sur les compteurs connectés à son réseau AMI. Le système de tête de réseau vérifie la demande et renvoie un message `reply(EndDeviceControls)` au MDMS. Ce message de réponse indique si la commande "create" peut aboutir ou si elle contient des erreurs (des références à des compteurs inexistants, par exemple) ou d'autres conditions informationnelles. Il envoie ensuite les commandes appropriées sur le réseau AMI aux compteurs. Lorsqu'ils réagissent à ces commandes, ils répondent au HES avec les résultats de la commande. En retour, le HES renvoie un ou plusieurs messages `created(EndDeviceEvents)` au MDMS pour l'informer que l'état des compteurs a été modifié.

Un exemple exhaustif peut être consulté à l'Article D.12.

4.8 Types de messages de base demande/réponse

Un message IEC 61968-100 qui apparaît dans un modèle d'échange de messages de demande/réponse entre dans l'un des ensembles de catégories en fonction de son `<Header><Verb>`.

- Un message contenant un verbe `create` est une commande d'instanciation d'un nouvel objet sur le système de réception. L'exécution d'une certaine action externe par le système de réception peut être un effet collatéral de l'instanciation d'un objet.

  EXEMPLE 1 Un système d'information des clients (CIS) peut envoyer des informations à un MDMS à l'aide d'un message IEC 61968-9:2013 create(CustomerConfig). Le MDMS utilise ensuite les informations fournies pour instancier le nouveau client dans sa base de données locale.

  EXEMPLE 2 Un MDMS peut envoyer un message IEC 61968-9:2013 `create(MeterReadSchedule)` à un HES pour lui demander de procéder à de futures opérations de relevé de compteur.

- Un message contenant le verbe `change` est une commande de remplacement de la totalité d'un objet existant sur le système de réception. Cela peut déclencher l'exécution de certaines actions externes.

  EXEMPLE 3 Un message IEC 61968-9:2013 `change(CustomerConfig)` peut être utilisé pour demander à un CIS de remplacer les informations dans sa base de données locale.

- Un message contenant le verbe `delete` est une commande de suppression complète d'un objet existant du système de réception. Cela peut déclencher l'exécution de certaines actions externes.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

EXEMPLE 4 Un message IEC 61968-9:2013 `delete(CustomerConfig)` peut être utilisé pour demander à un CIS de supprimer les entrées de client correspondantes de sa base de données locale.

EXEMPLE 5 Un MDMS peut envoyer un message IEC 61968-9:2013 `delete(MeterReadSchedule)` à un HES pour lui demander de supprimer un programme de relevé de compteur.

- Un message contenant le verbe `get` permet d'interroger un système de réception et donc de provoquer le renvoi d'un ou de plusieurs objets. La section `<Payload>` du message donne les paramètres de requête, et le système de réception renvoie les données demandées en conséquence.

  EXEMPLE 6 Un message IEC 61968-9:2013 `get(CustomerConfig)` peut amener un CIS à renvoyer des informations relatives aux clients demandés depuis sa base de données locale.

- Un message contenant un verbe `execute` permet de regrouper plusieurs commandes dans un message. L'intégrité transactionnelle peut être éventuellement renforcée de sorte que toutes les commandes aboutissent ou pas. Le verbe "execute" est utilisé uniquement conjointement avec le nom OperationSet (7.5.3, 7.7.3).
- Un message contenant le verbe `cancel` est une commande d'annulation d'une opération en cours.

  EXEMPLE 7 Un message IEC 61968-9:2013 `cancel(EndDeviceControls)` peut être utilisé pour annuler une opération en attente qui résulte d'un précédent message `create(EndDeviceControls)`.

- Un message contenant un verbe `update` est une commande de modification partielle d'un objet existant (8.3.5). Le verbe `update` est nouveau dans cette édition de l'IEC 61968-100.

  EXEMPLE 8 Un message IEC 61968-9:2013 update(CustomerConfig) peut être utilisé pour demander à un CIS de mettre à jour les informations sélectionnées (le numéro de téléphone d'un client, par exemple) dans sa base de données locale.

- Un message contenant un verbe `reply` est utilisé par un système pour informer un autre qu'il a reçu une commande ou une demande dans un message qui contient l'un des autres verbes. Le message de réponse peut contenir toutes les données demandées, ainsi que toutes les notifications d'état de l'application.

  EXEMPLE 9 Un message IEC 61968-9:2013 `reply(MeterReadings)` peut être envoyé d'un HES à un MDMS pour fournir les lectures de compteur en réponse à un précédent message IEC 61968-9:2013 `get(MeterReadings)`. Il peut également contenir les notifications d'erreur (si la demande d'origine spécifie des compteurs non valides, par exemple).

5 Archétypes de message

5.1 Généralités

Le présent document définit quatre grands types de messages: les messages de demande, les messages de réponse, les messages d'événement et les messages de défaut (5.2, 5.3, 5.4, 5.5). Ces types de messages sont appelés archétypes de message.

Chaque message IEC 61968-100 doit être conforme à l'un de ces archétypes.

5.2 Message de demande

5.2.1 Généralités

Un message de demande IEC 61968-100 doit être utilisé pour envoyer une demande d'un système demandeur à un système répondant.

Un message de demande doit uniquement être utilisé dans les modèles d'échange de messages de demande/réponse (6.2) ou d'événement induit (6.4).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

5.2.2 Élément Root (racine)

Le nom de l'élément racine d'un message de demande dépend de la valeur de l'élément `<Header><Verb>` dans le message (7.5.3.3) et doit être choisi parmi l'un des noms d'élément racine qui figurent dans le Tableau 1.

Cette règle ne doit pas s'appliquer dans le cas des services Web fortement typés (9.2.5).

5.2.3 Structure

La structure d'un message de demande est décrite à la Figure 1.

NOTE Le nom générique RequestMessage qui apparaît à la Figure 1 indique l'un des noms d'élément de demande qui figurent dans le Tableau 1.

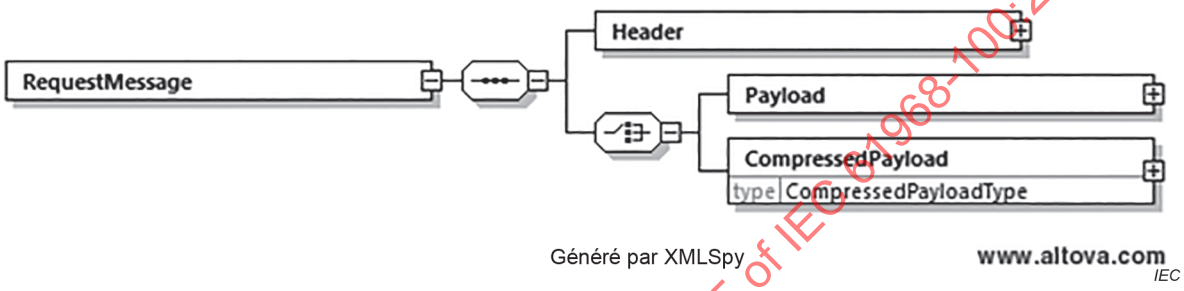


Figure 1 – Structure d'un message de demande

Un message de demande doit contenir un sous-élément `<Header>` (7.5) accompagné d'un sous-élément `<Payload>` (7.7) ou d'un sous-élément `<CompressedPayload>` (7.8).

5.2.4 Élément Header (en-tête)

Le paragraphe 7.5.2 spécifie l'élément `<Header>` en ce qui concerne les messages de demande.

5.2.5 Données spécifiques à l'application

Toutes les données spécifiques à l'application associées à la demande doivent être placées dans le sous-élément `<Payload>` ou `<CompressedPayload>` (7.7, 7.8).

5.2.6 Effets collatéraux d'un message de demande

La réception d'un message de demande par un système répondant peut amener ce dernier à exécuter certaines actions spécifiques à l'application. Ces actions ou leurs effets collatéraux ne sont pas normatifs et ne relèvent pas du domaine d'application du présent document. Leur définition doit être laissée aux groupes de travail et développeurs d'application compétents.

5.3 Message de réponse

5.3.1 Généralités

Un message de réponse IEC 61968-100 doit être utilisé pour envoyer la réponse d'un système répondant à un système demandeur qui a préalablement envoyé un message de demande.

Un message de réponse doit être utilisé dans les modèles d'échange de messages de demande/réponse (6.2) ou d'événement induit (6.4). Un message de réponse peut également être utilisé pour accuser réception d'un message si cela est exigé par le transport sous-jacent (6.5).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

5.3.2 Élément Root

Le nom de l'élément racine d'un message de réponse dépend de la valeur de l'élément `<Header><Verb>` dans le message de demande d'origine (7.5.3.3) et doit être choisi parmi l'un des noms d'élément racine qui figurent dans le Tableau 2.

Cette règle ne doit pas s'appliquer dans le cas des services Web fortement typés (9.2.5).

5.3.3 Structure

La structure d'un message de réponse est décrite à la Figure 2.

NOTE Le nom générique ResponseMessage qui apparaît à la Figure 2 indique l'un des noms d'élément de réponse qui figurent dans le Tableau 2.

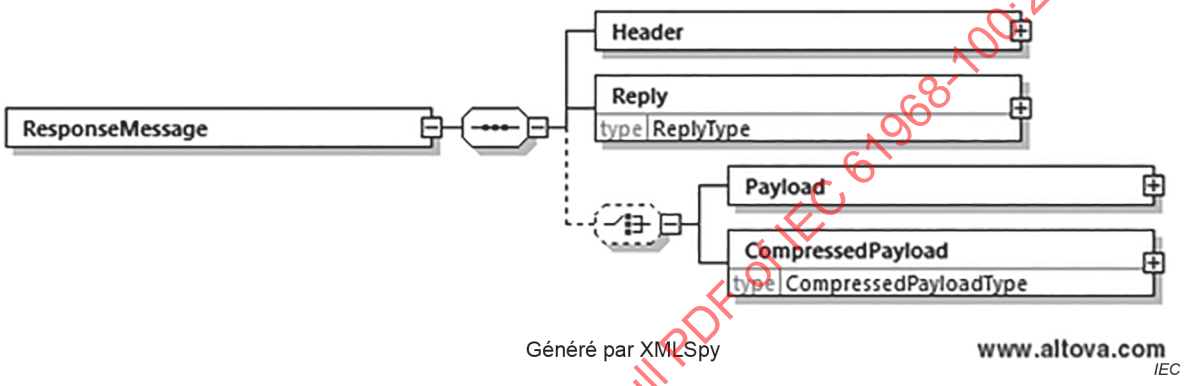


Figure 2 – Structure d'un ResponseMessage

Un message de réponse doit contenir les sous-éléments `<Header>` et `<Reply>` (7.5, 7.6). Il peut également contenir un sous-élément `<Payload>` (7.7) ou `<CompressedPayload>` (7.8).

5.3.4 Élément Header

Le paragraphe 7.5.2 spécifie l'élément `<Header>` en ce qui concerne les messages de réponse.

5.3.5 Données spécifiques à l'application et notifications d'état

Un message de réponse doit contenir des indications d'état. Il peut également contenir des données spécifiques à l'application.

Les données spécifiques à l'application, si elles sont présentes, doivent être placées dans le sous-élément `<Payload>` (7.7) ou `<CompressedPayload>` (7.8).

Toutes les indications de statut doivent être placées dans les éléments `<Reply><Status>` (7.6.4).

Si un message de réponse renvoie des notifications de succès sans notification d'informations, un élément `<Reply>` qui contient une indication explicite de succès doit être inclus dans le message (7.6.4.3).

EXEMPLE 1 Le paragraphe D.4.3 donne un exemple de message de réponse utilisé pour acheminer des données spécifiques à l'application.

EXEMPLE 2 Le paragraphe D.11.3 donne un exemple de message de réponse utilisé pour acheminer tant des données spécifiques à l'application que des notifications d'erreur.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

5.3.6 Message d'accusé de réception simple

Un message d'accusé de réception simple est un message de répons spécialisé utilisé uniquement pour acheminer la réception d'un message (6.5).

Les éléments suivants doivent être définis dans un message d'accusé de réception simple:

- `<Header><Verb>` doit être `reply`.
- `<Header><Noun>` doit avoir la même valeur que celle du message d'origine pour lequel l'accusé de réception simple est envoyé.
- Un seul élément `<Reply><Status>` doit être inclus dans le message d'accusé de réception simple (7.6.4). Il doit contenir les sous-éléments `<Reply><Status><Level>` et `<Reply><Status><Code>` avec les valeurs `OK` et `0.3` respectivement.
- Il peut également contenir un élément `<Reply><Status><Reason>` facultatif qui contient un texte arbitraire.

Aucune donnée spécifique à l'application ne doit être acheminée à l'intérieur d'un message d'accusé de réception simple.

5.4 Message d'événement

5.4.1 Généralités

Un message d'événement IEC 61968-100 doit être utilisé pour envoyer des données d'un système à l'autre lorsqu'aucune réponse n'est attendue ou admise. Il doit être utilisé pour la publication des données ou la notification d'une occurrence ou d'un incident.

Un message d'événement doit être utilisé dans le cadre des modèles d'échange de messages d'événement non sollicité (6.3) ou d'événement induit (6.4).

5.4.2 Élément Root

Le nom de l'élément racine d'un message d'événement dépend de la valeur de l'élément `<Header><Verb>` dans le message (7.5.5.3) et doit être choisi parmi l'une des valeurs énumérées qui figurent dans le Tableau 3.

Cette règle ne doit pas s'appliquer dans le cas des services Web fortement typés (9.2.5).

5.4.3 Structure

La structure d'un message d'événement est décrite à la Figure 3.

NOTE Le nom générique EventMessage qui apparaît à la Figure 3 indique l'un des noms d'élément qui figurent dans le Tableau 3.

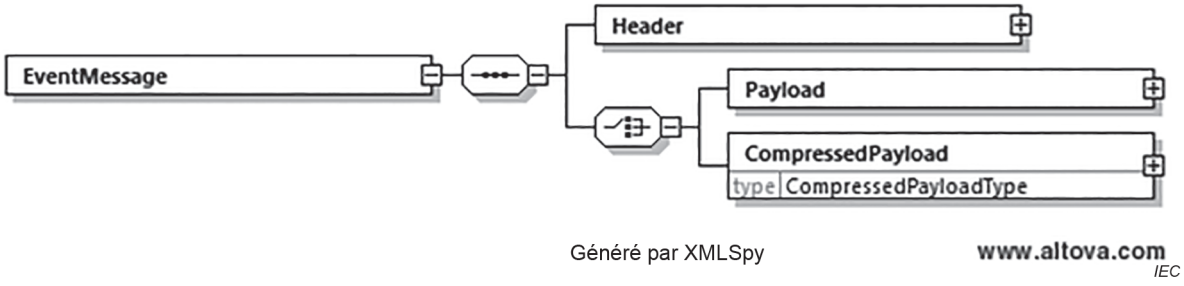


Figure 3 – Structure d'un EventMessage

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Un message d'événement doit contenir un sous-élément <Header> (7.5) accompagné d'un sous-élément `<Payload>` (7.7) ou `<CompressedPayload>` (7.8).

5.4.4 Élément Header

Le paragraphe 7.5.2 spécifie l'élément `<Header>` en ce qui concerne les messages d'événement.

5.4.5 Données spécifiques à l'application

Les données spécifiques à l'application doivent être placées dans l'élément `<Payload>` (7.7.3).

EXEMPLE Le paragraphe D.3.3 donne un exemple de message d'événement.

5.5 Message de défaut

5.5.1 Généralités

Un message de défaut IEC 61968-100 doit être utilisé uniquement lorsqu'un système destinataire a reçu un message de demande malformé, c'est-à-dire qui ne peut pas être analysé comme un document XML valide. Dans un tel cas, le système répondant doit renvoyer un message de défaut au système demandeur.

Un message de défaut ne doit pas être utilisé pour tout autre type d'erreur détectée par le système destinataire.

Les erreurs de validation de schéma XML doivent invoquer un message de réponse qui contient un élément `<Reply><Status><Code>` avec une valeur appropriée (7.6.4.7).

EXEMPLE Le paragraphe D.18.3 donne un exemple de message de défaut.

5.5.2 Élément Root

Un message de défaut doit contenir l'élément racine `<FaultMessage>`. Ce message est représenté à la Figure 4.

5.5.3 Structure

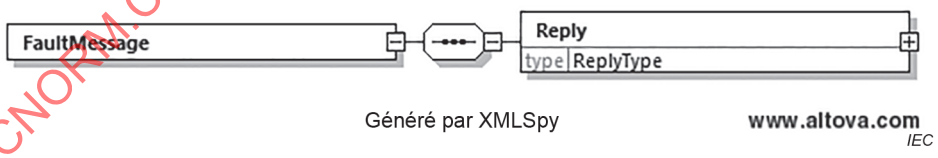


Figure 4 – Structure d'un FaultMessage

5.5.4 Élément de réponse

Le paragraphe 7.6.4.5 spécifie l'élément `<Reply>` en ce qui concerne les messages de défaut

6 Modèles d'échange de messages élémentaires

6.1 Généralités

Le présent article décrit l'ensemble de modèles d'échange de messages élémentaires IEC 61968-100. Un échange de messages IEC 61968-100 entre des systèmes coopératifs doit être composé d'un ou de plusieurs de ces modèles d'échange de messages élémentaires.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

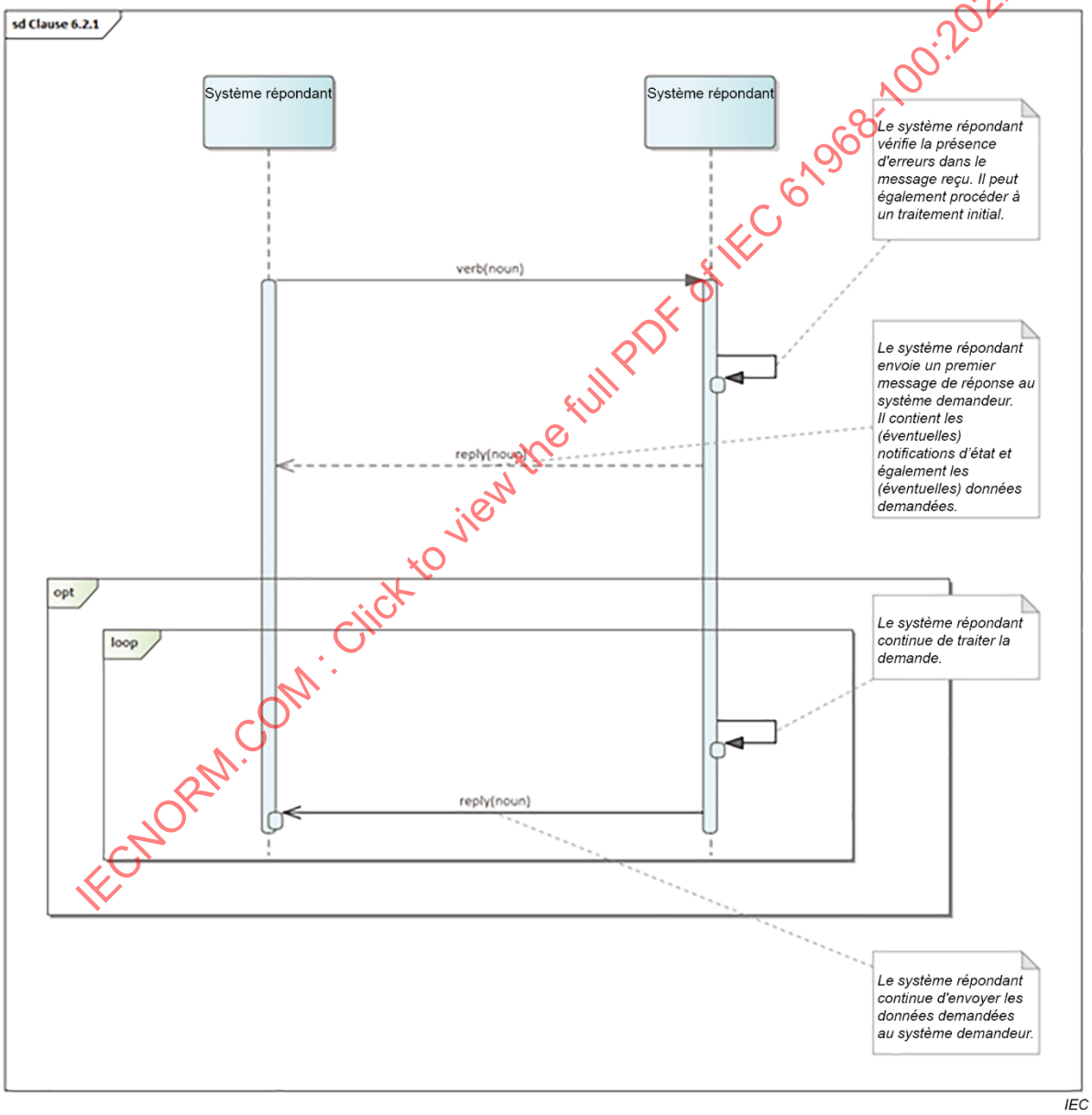
6.2 Modèle d'échange de messages de demande/réponse

6.2.1 Généralités

Un modèle d'échange de messages de demande/réponse doit être composé d'un seul message de demande (5.2) envoyé par un système demandeur à un système répondant, suivi d'un ou de plusieurs messages de réponse (5.3) envoyés par le système répondant au système demandeur.

EXEMPLE L'Article D.1 donne un exemple d'échange de messages fondé sur le modèle d'échange de messages de demande/réponse.

Le modèle d'échange de messages de demande/réponse est décrit à la Figure 5.



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 5 – Modèle d'échange de messages de demande/réponse

6.2.2 Messages de réponse

Le message de réponse (5.3) d'un modèle d'échange de messages de demande/réponse est utilisé pour acheminer les indications de statut (7.6.4) et les données de réponse spécifiques à l'application (7.7).

Un seul message de demande peut amener le système répondant à générer un seul ou plusieurs messages de réponse, à son appréciation. Les raisons possibles pour lesquelles un système répondant génère plusieurs messages de réponse à une seule demande sont les suivantes:

- Les données à renvoyer sont trop volumineuses pour tenir dans un seul message et, de ce fait, doivent être réparties sur plusieurs messages.
- Les données à renvoyer ne sont pas toutes immédiatement disponibles. En d'autres termes, des problèmes de synchronisation peuvent déterminer la manière dont les données demandées sont renvoyées au système demandeur.

EXEMPLE Un exemple de demande simple qui induit plusieurs messages de réponse à générer est présenté en D.10.3.

6.3 Modèle d'échange de messages d'événement non sollicité

Un modèle d'échange de messages d'événement non sollicité doit être utilisé pour envoyer une notification d'un système à un autre lorsqu'aucune réponse de niveau d'application n'est attendue ou exigée. Un modèle d'échange de messages d'événement non sollicité doit être composé d'un seul message d'événement (5.4).

Un message d'événement non sollicité doit être envoyé après une action autonome dans un système et pas en conséquence directe d'une précédente demande.

Le modèle d'échange de messages d'événement non sollicité est décrit à la Figure 6.

EXEMPLE L'Article D.6 donne un exemple de message d'événement non sollicité.

Si un système envoie un message à un système de réception destiné à déclencher une certaine opération et s'il est nécessaire que le système d'envoi sache si cette opération a abouti ou a échoué, il doit alors utiliser le modèle d'échange de messages de demande/réponse et pas le modèle d'échange de messages d'événement non sollicité.

Il convient d'envoyer un message d'événement non sollicité uniquement lorsque l'opération sous-jacente est terminée.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

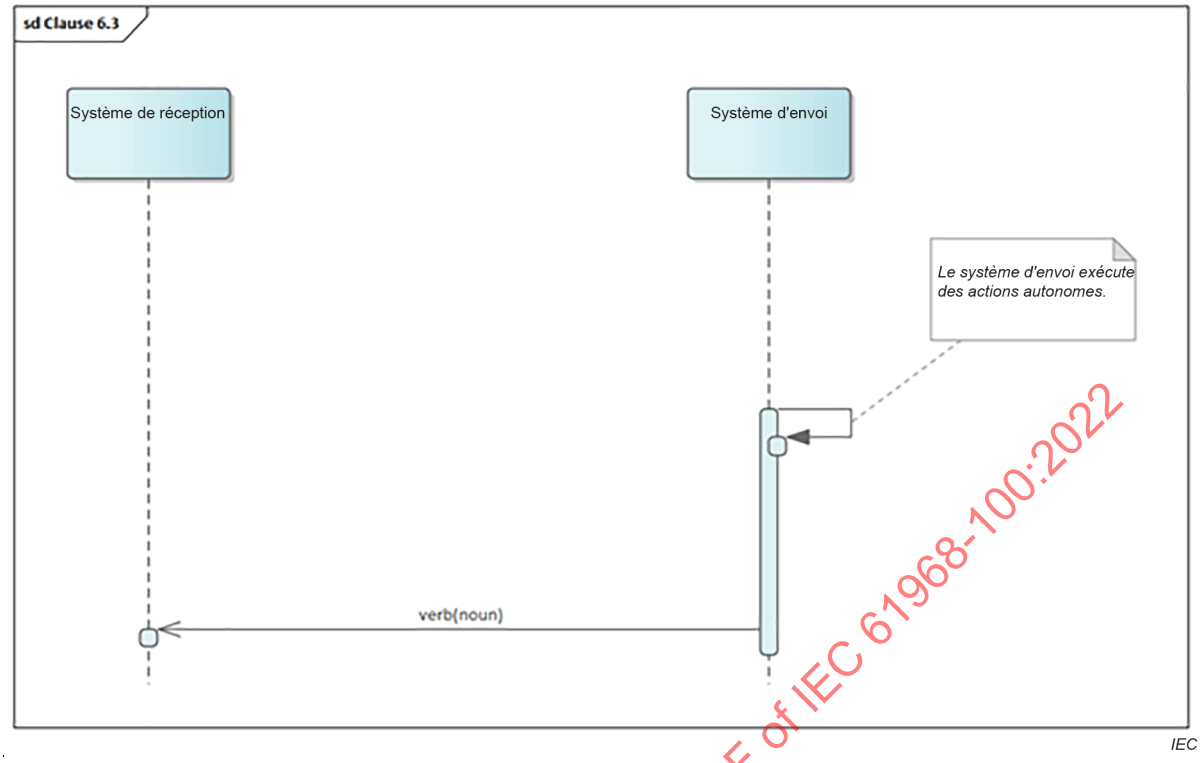


Figure 6 – Modèle d'échange de messages d'événement non sollicité

6.4 Modèle d'échange de messages d'événement induit

Le modèle d'échange de messages d'événement induit doit être composé d'un message de demande (5.2) suivi d'un seul message de réponse (5.3) suivi d'aucun ou de plusieurs messages d'événement (5.4).

Un modèle d'échange de messages d'événement induit est différent d'un modèle d'échange de messages d'événement non sollicité en ce sens que l'événement communiqué est le résultat d'un précédent message de demande.

Le modèle d'échange de messages d'événement induit est décrit à la Figure 7.

EXEMPLE L'Article D.12 donne un exemple d'échange de messages fondé sur le modèle d'échange de messages d'événement induit.

NOTE Les messages étant acheminés dans un modèle d'échange de messages de demande/réponse partagent le même élément `<Header><Noun>` (7.5.2). Toutefois, pour le modèle de messages d'événement induit, l'élément `<Header><Noun>` des messages d'événement est différent de ceux de la demande/réponse initiale. De plus, l'élément `<Header><Verb>` des messages d'événement n'est pas "reply".

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

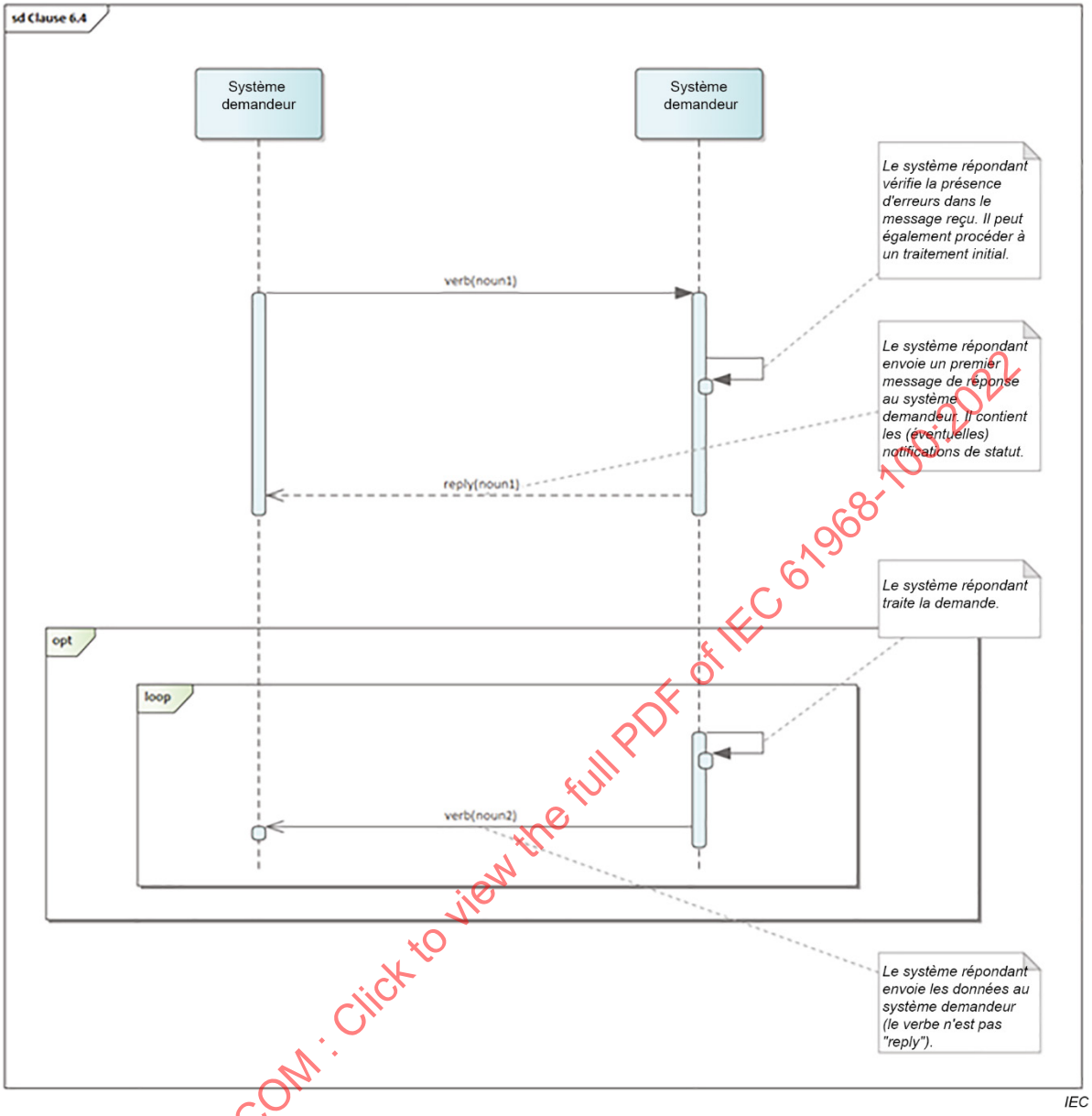


Figure 7 – Modèle d'échange de messages d'événement induit

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

6.5 Modèles d'échange de messages d'accusé de réception simples

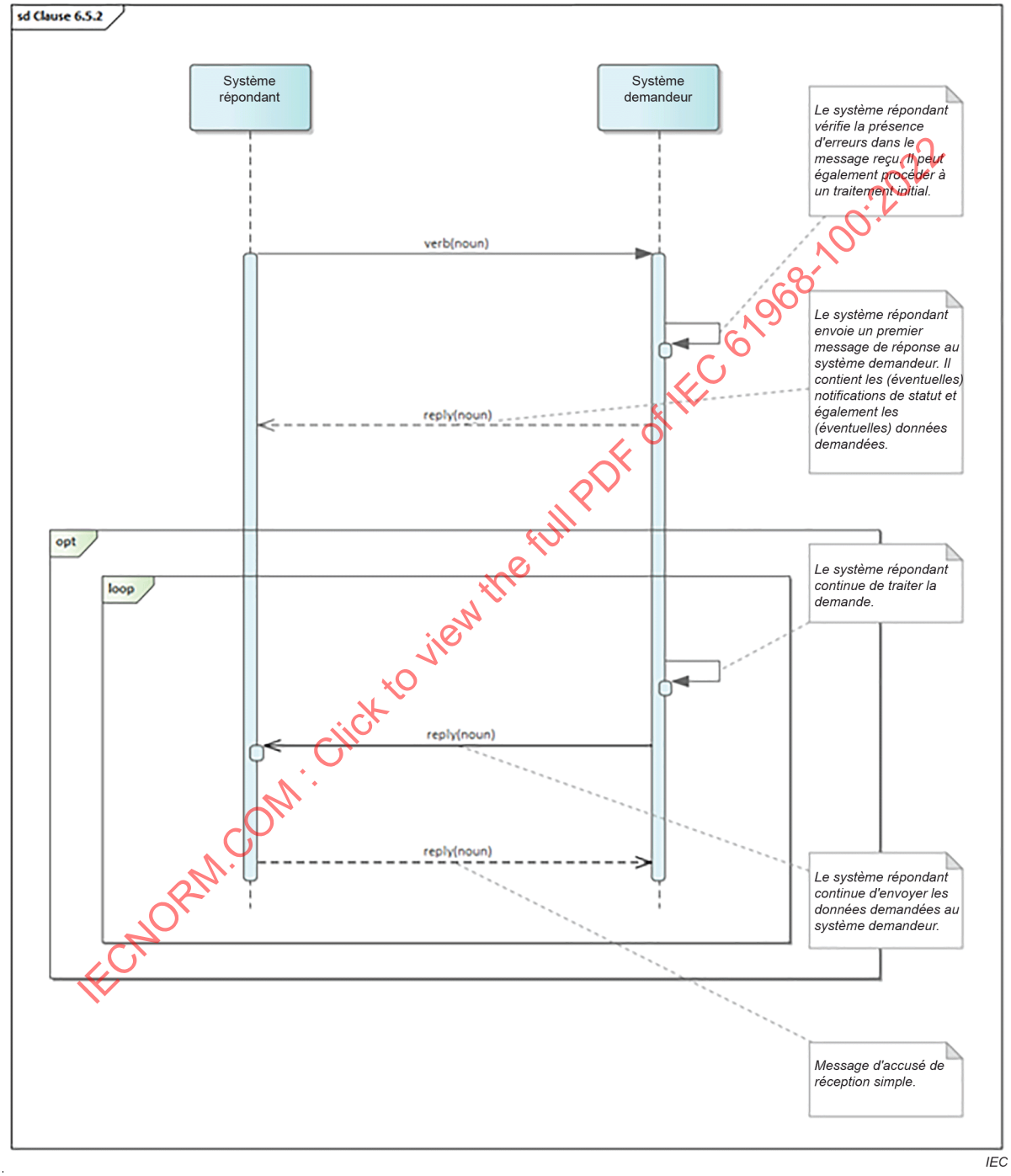
6.5.1 Généralités

Un système répondant doit envoyer un message d'accusé de réception simple dans l'un des cas suivants:

- Le système répondant a reçu un message de demande dans lequel l'élément `<Header><AckRequired>` est présent et dont la valeur est `true` (7.5.2).
- Le système répondant a reçu un message de réponse (5.2) ou un message d'événement (5.4) et le transport sous-jacent exige d'envoyer un accusé de réception au niveau du transport du message de réponse ou d'événement (9.2.4).

6.5.2 Modèle d'échange de messages de demande/réponse avec messages d'accusé de réception simples

Le modèle d'échange de messages de demande/réponse de la (Figure 5) est modifié comme cela est représenté à la Figure 8 lorsqu'un message d'accusé de réception simple doit être envoyé.



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 8 – Modèle d'échange de messages de demande/réponse avec messages d'accusé de réception simples

NOTE Si le système demandeur détecte une erreur dans le message de réponse reçu de la part du système répondant, il peut remplacer le message d'accusé de réception simple avec un code d'état de 0.3 par un message de réponse qui contient un code d'état correspondant à l'erreur rencontrée.

6.5.3 Modèle d'échange de messages d'événement non sollicité avec message d'accusé de réception simple

Le modèle d'échange de messages d'événement non sollicité de la Figure 6 est modifié comme cela est représenté à la Figure 9 lorsqu'un message d'accusé de réception simple doit être envoyé.

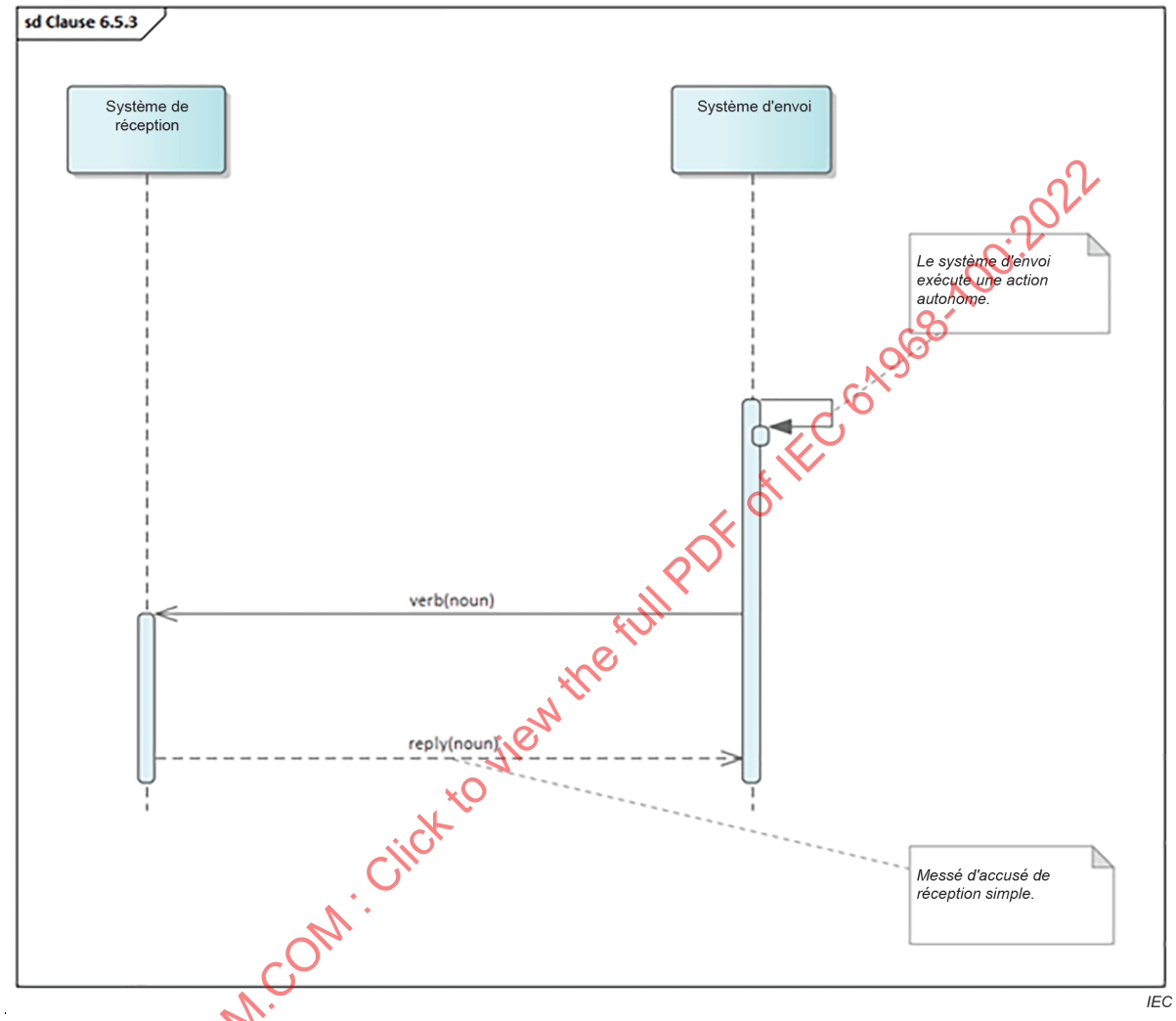


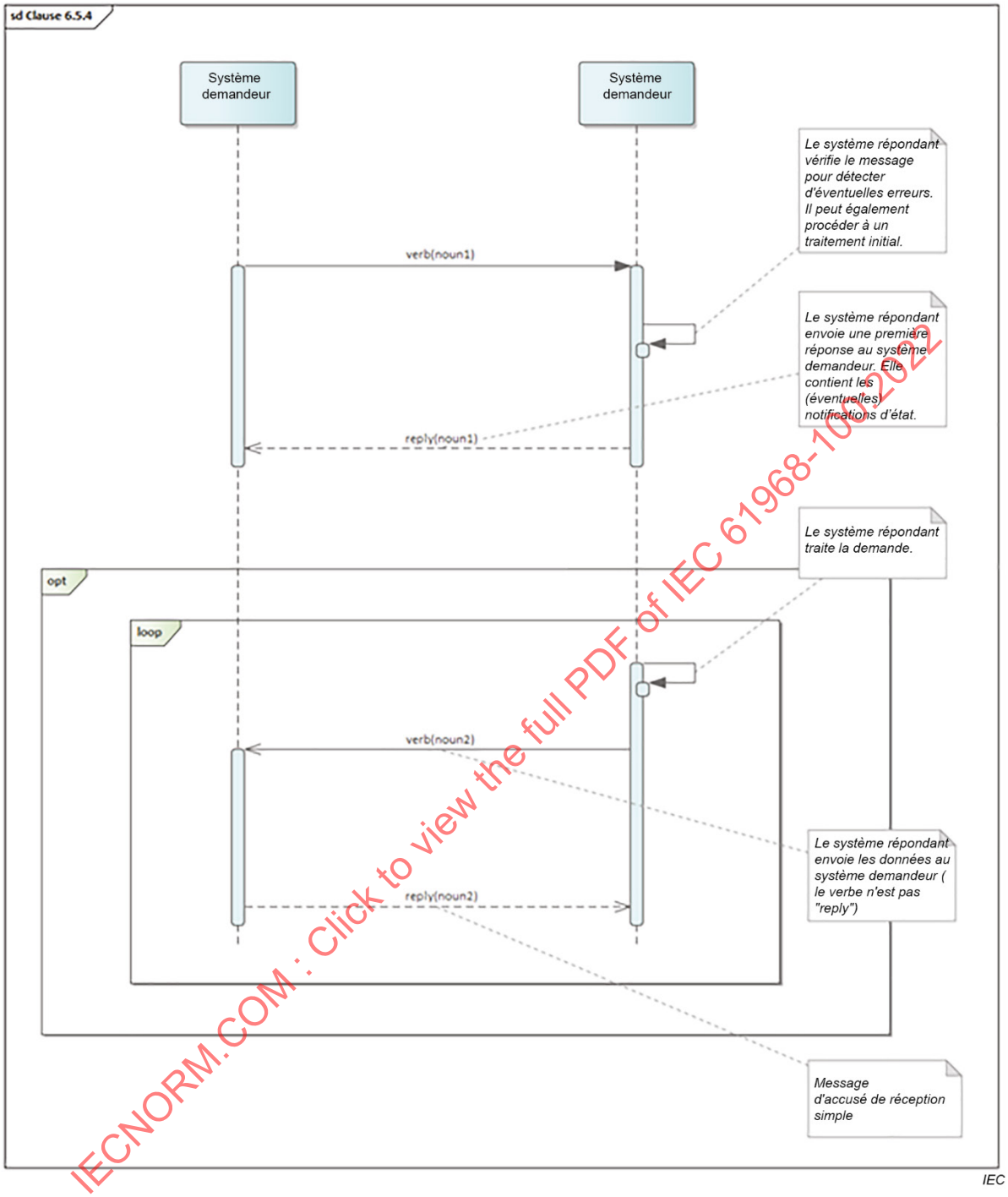
Figure 9 – Modèle d'échange de messages d'événement non sollicité avec messages d'accusé de réception simple

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

NOTE Si le système demandeur détecte une erreur dans le message d'événement non sollicité reçu de la part du système répondant, il peut remplacer le message d'accusé de réception simple avec un code d'état de 0.3 par un message de réponse qui contient un code d'état correspondant à l'erreur rencontrée.

6.5.4 Modèle d'échange de messages d'événement induit avec messages d'accusé de réception simple

Le modèle d'échange de messages d'événement induit de la Figure 7 est modifié comme cela est représenté à la Figure 10 lorsque des messages d'accusé de réception simple doivent être envoyés.



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 10 – Modèle d'échange de messages d'événement induit avec messages d'accusé de réception simple

NOTE Si le système demandeur détecte une erreur dans le message d'événement induit reçu de la part du système répondant, il peut remplacer le message d'accusé de réception simple avec un code d'état de 0.3 par un message de réponse qui contient un code d'état correspondant à l'erreur rencontrée.

7 Informations détaillées du message

7.1 Généralités

Le présent article décrit en détail le format de tous les messages IEC 61968-100.

7.2 Codage de message

Un message IEC 61968-100 doit être composé comme un document XML.

7.3 Définition et validation de schéma

Le fichier de définition de schéma XML `Message.xsd` énuméré à l'Article A.2 définit la structure syntaxique précise de tout message IEC 61968-100.

Un message IEC 61968-100 doit pouvoir faire l'objet d'une validation de schéma XML réussie par rapport à ce fichier XSD (Annexe H).

NOTE D'autres normes peuvent imposer des contraintes supplémentaires au-delà de celles de l'IEC 61968-100.

7.4 Structure de niveau supérieur

7.4.1 Généralités

Le présent article spécifie la structure de niveau supérieur des types de messages IEC 61968-100 en fonction de leurs éléments racines respectifs.

D'autres noms d'élément racine doivent être utilisés dans le cas des services Web fortement typés (9.2.5).

7.4.2 Structure des messages de demande

7.4.2.1 Généralités

Le présent article décrit la structure des messages de demande (5.2) en fonction de leurs éléments racines.

Le Tableau 1 répertorie les possibles éléments racines de message et leurs éléments `<Header><Verb>` respectifs (7.5.3.3).

Tableau 1 – Liste des éléments racines de niveau supérieur dans les messages de demande

Élément `<Header><Verb>`	Élément racine correspondant	Voir Article/paragraphe
`cancel`	`<CancelRequestMessage>`	7.4.2.2
`change`	`<ChangeRequestMessage>`	7.4.2.3
`create`	`<CreateRequestMessage>`	7.4.2.4
`delete`	`<DeleteRequestMessage>`	7.4.2.5
`execute`	`<ExecuteRequestMessage>`	7.4.2.6
`get`	`<GetRequestMessage>`	7.4.2.7
`update`	`<UpdateRequestMessage>`	7.4.2.8

7.4.2.2 Structure de l'élément `<CancelRequestMessage>`

La structure de niveau supérieur d'un message de demande (5.2) dont l'élément racine est `<CancelRequestMessage>` est représentée à la Figure 11.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

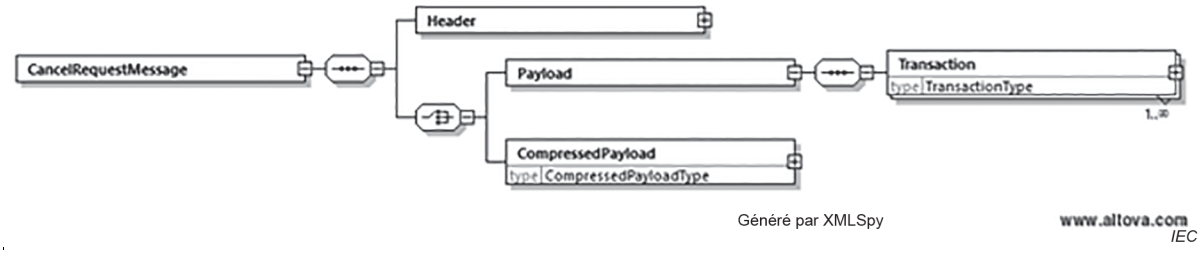


Figure 11 – Structure de l'élément <CancelRequestMessage>

7.4.2.3 Structure de l'élément <ChangeRequestMessage>

La structure de niveau supérieur d'un message de demande dont l'élément racine est <ChangeRequestMessage> est représentée à la Figure 12.

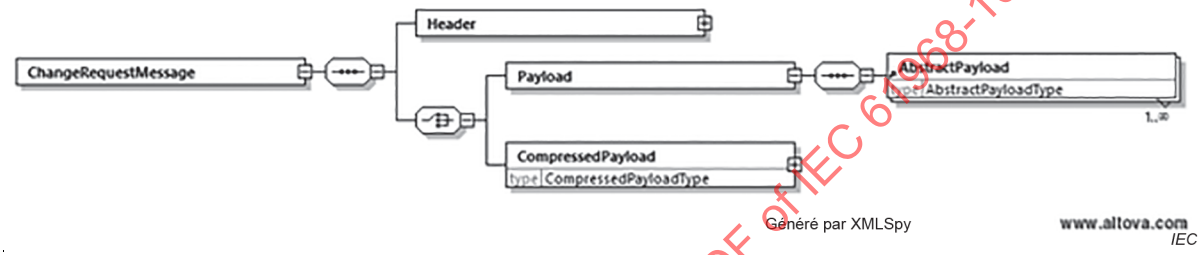


Figure 12 – Structure de l'élément <ChangeRequestMessage>

7.4.2.4 Structure de l'élément <CreateRequestMessage>

La structure de niveau supérieur d'un message de demande dont l'élément racine est <CreateRequestMessage> est représentée à la Figure 13.

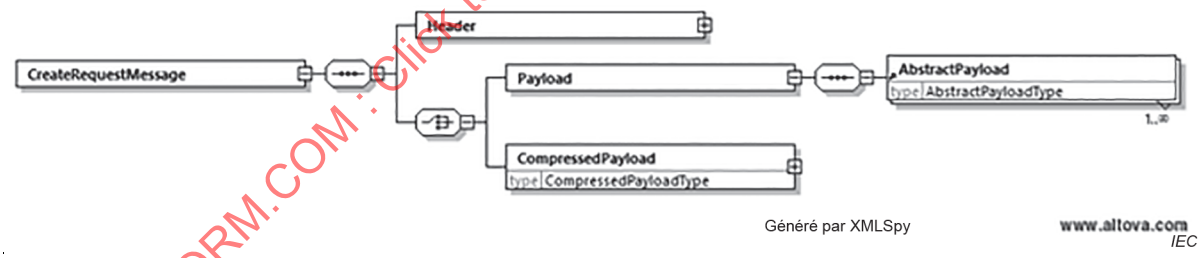


Figure 13 – Structure de l'élément <CreateRequestMessage>

7.4.2.5 Structure de l'élément <DeleteRequestMessage>

La structure de niveau supérieur du message de demande dont l'élément racine est <DeleteRequestMessage> est représentée à la Figure 14.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

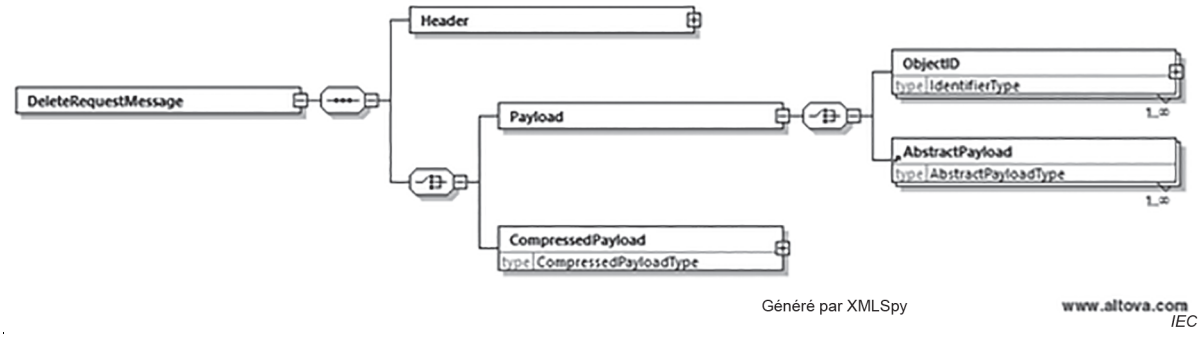


Figure 14 – Structure de l'élément <DeleteRequestMessage>

7.4.2.6 Structure de l'élément <ExecuteRequestMessage>

La structure de niveau supérieur d'un message de demande dont l'élément racine est <ExecuteRequestMessage> est représentée à la Figure 15.

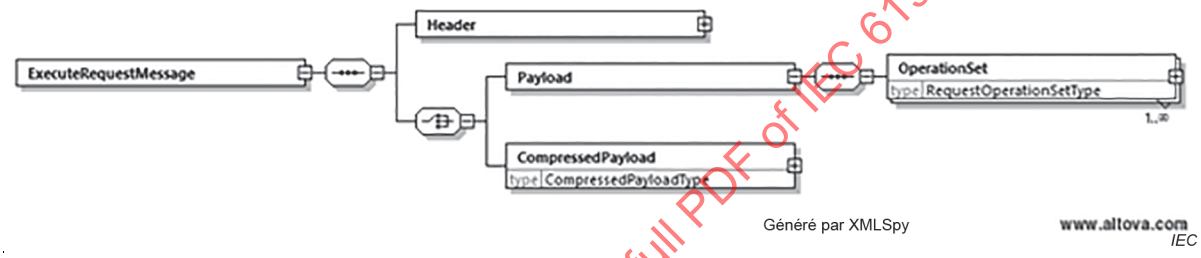


Figure 15 – Structure de l'élément <ExecuteRequestMessage>

7.4.2.7 Structure de l'élément <GetRequestMessage>

La structure de niveau supérieur d'un message de demande dont l'élément racine est <GetRequestMessage> est représentée à la Figure 16.

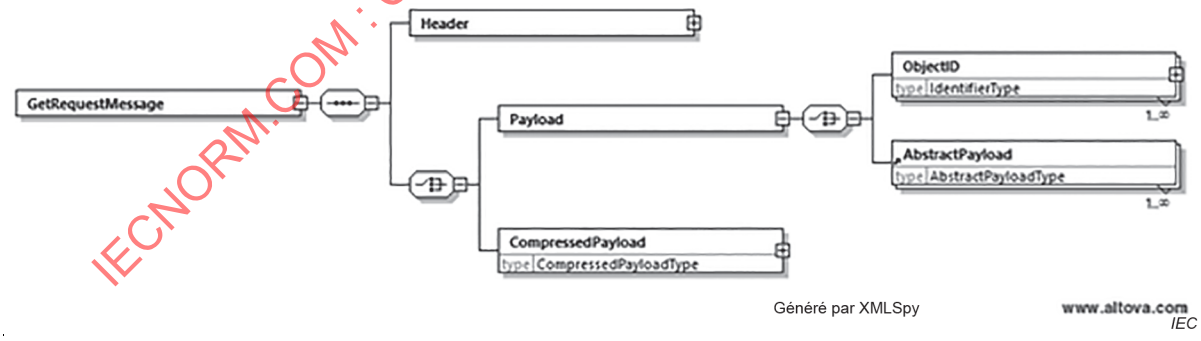


Figure 16 – Structure de l'élément <GetRequestMessage>

7.4.2.8 Structure de l'élément <UpdateRequestMessage>

La structure de niveau supérieur du message de demande dont l'élément racine est <UpdateRequestMessage> est représentée à la Figure 17.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

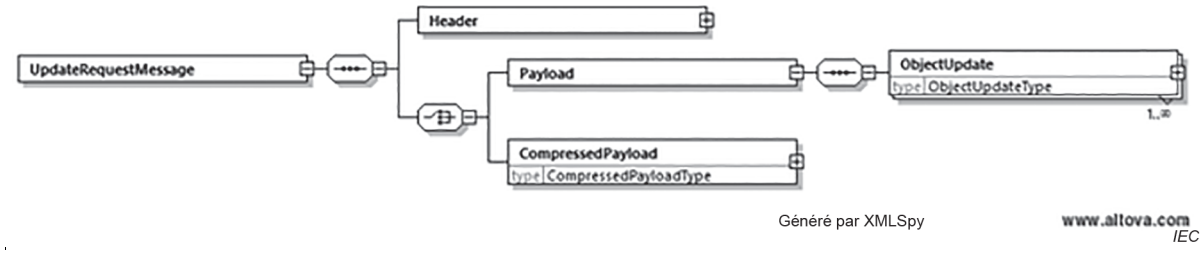


Figure 17 – Structure de l'élément <UpdateRequestMessage>

7.4.3 Structure des messages de réponse

7.4.3.1 Généralités

Le présent article décrit les structures de niveau supérieur des messages de réponse (5.3) en fonction de leurs éléments racines.

Le Tableau 2 répertorie les possibles éléments racines du message tels que définis par les éléments <Header><Verb> du message de demande d'origine (7.5.3.3).

Tableau 2 – Liste des éléments racines de niveau supérieur dans les messages de réponse

Élément <Header><Verb> dans le message de demande	Élément racine correspondant	Voir Article/paragraphe
cancel	<CancelResponseMessage>	7.4.3.2, 8.5
change	<ChangeResponseMessage>	7.4.3.3, 8.3.3
create	<CreateResponseMessage>	7.4.3.4, 8.3.2
delete	<DeleteResponseMessage>	7.4.3.5, 8.3.4
execute	<ExecuteResponseMessage>	7.4.3.6, 7.7.3
get	<GetResponseMessage>	7.4.3.7, 8.4
update	<UpdateResponseMessage>	7.4.3.8, 8.3.5

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.4.3.2 Structure de l'élément <CancelResponseMessage>

La structure du message de réponse dont l'élément racine est <CancelResponseMessage> est représentée à la Figure 18.

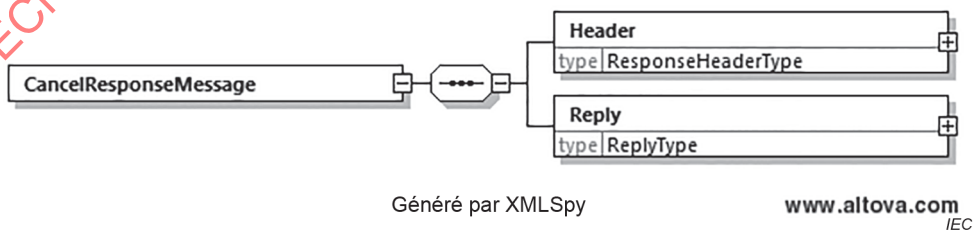


Figure 18 – Structure de l'élément <CancelResponseMessage>

7.4.3.3 Structure de l'élément <ChangeResponseMessage>

La structure du message de réponse dont l'élément racine est <ChangeResponseMessage> est représentée à la Figure 19.

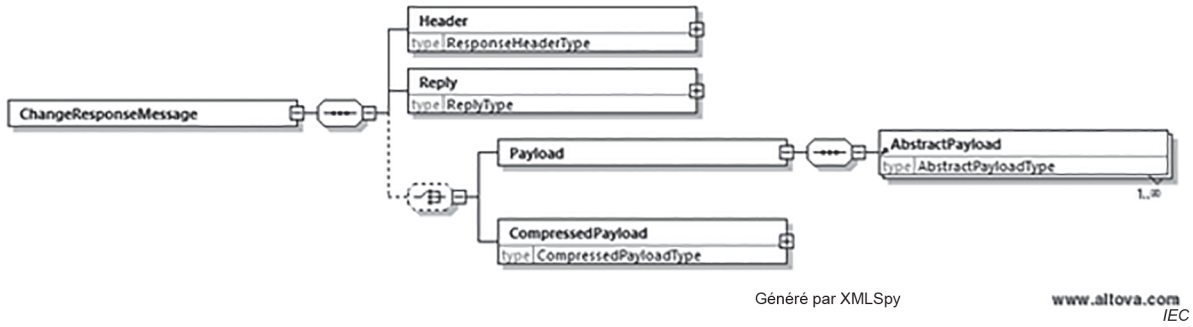


Figure 19 – Structure de l'élément <ChangeResponseMessage>

7.4.3.4 Structure de l'élément <CreateResponseMessage>

La structure du message de réponse dont l'élément racine est <ChangeResponseMessage> est représentée à la Figure 20.

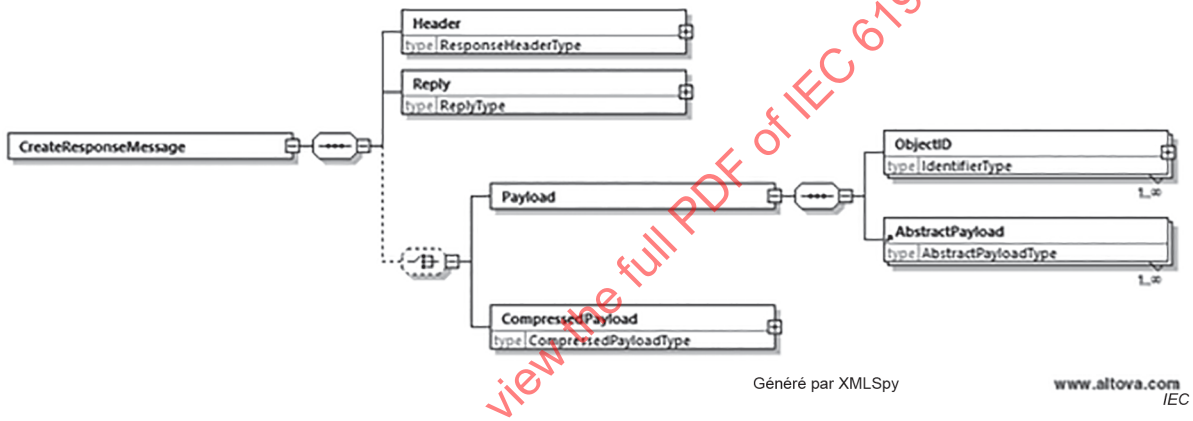


Figure 20 – Structure de l'élément <CreateResponseMessage>

7.4.3.5 Structure de l'élément <DeleteResponseMessage>

La structure du message de réponse dont l'élément racine est <DeleteResponseMessage> est représentée à la Figure 21.

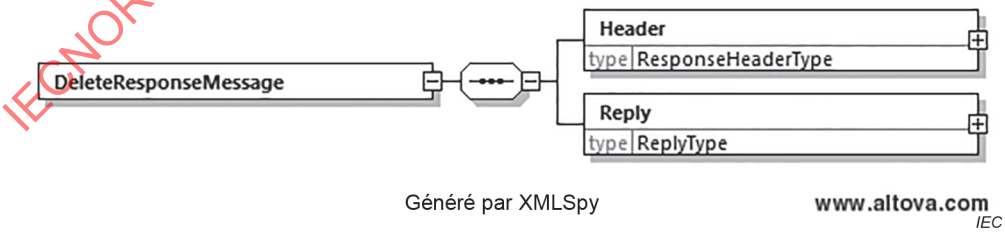


Figure 21 – Structure de l'élément <DeleteResponseMessage>

7.4.3.6 Structure de l'élément <ExecuteResponseMessage>

La structure du message de réponse dont l'élément racine est <ExecuteResponseMessage> est représentée à la Figure 22.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

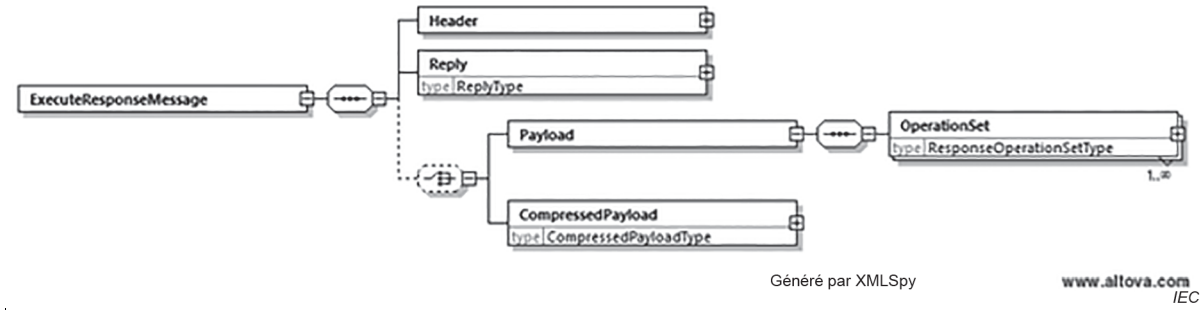


Figure 22 – Structure de l'élément <ExecuteResponseMessage>

7.4.3.7 Structure de l'élément <GetReponseMessage>

La structure du message de réponse dont l'élément racine est <GetReponseMessage> est représentée à la Figure 23.

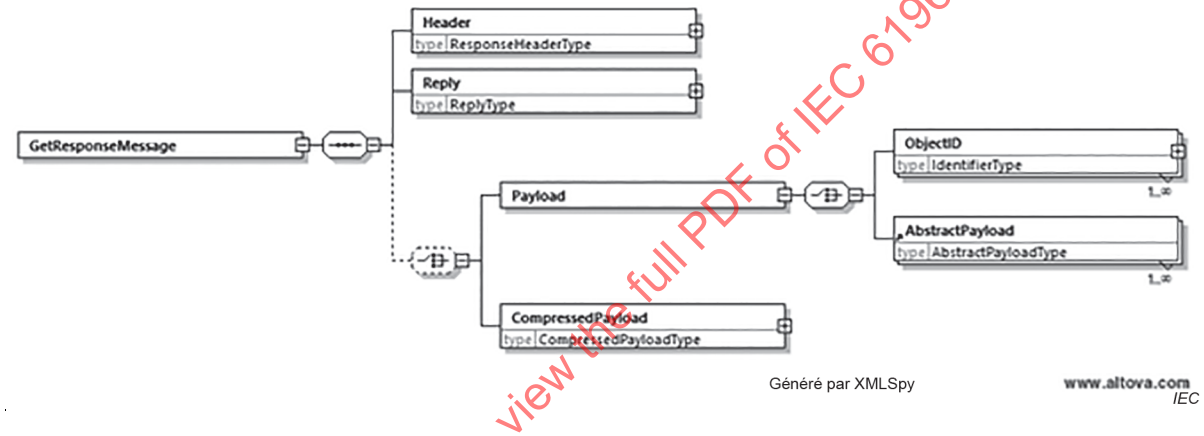


Figure 23 – Structure de l'élément <GetResponseMessage>

7.4.3.8 Structure de l'élément <UpdateResponseMessage>

La structure du message de réponse dont l'élément racine est <UpdateResponseMessage> est représentée à la Figure 24.

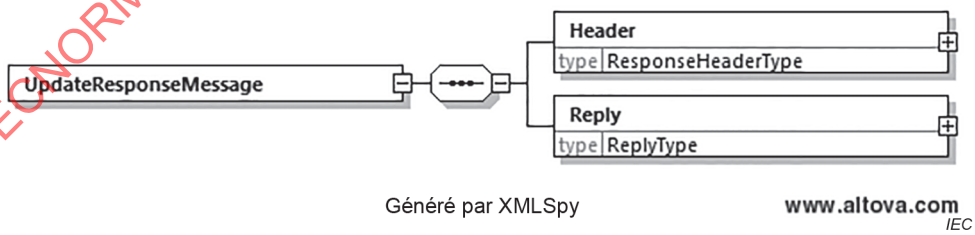


Figure 24 – Structure de l'élément <UpdateResponseMessage>

7.4.4 Structure des messages d'événement

7.4.4.1 Généralités

Le présent article décrit les structures de niveau supérieur des messages d'événement (5.4) en fonction de leurs éléments racines.

Le Tableau 3 répertorie les possibles éléments racines de message et leurs éléments <Header><Verb> respectifs (7.5.5.3)

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Tableau 3 – Liste des éléments racines de niveau supérieur dans les messages d'événement

Élément <Header><Verb>	Élément racine correspondant	Voir Article/paragraphe
canceled	<CanceledEventMessage>	7.4.4.2
changed	<ChangedEventMessage>	7.4.4.3
created	<CreatedEventMessage>	7.4.4.4
deleted	<DeletedEventMessage>	7.4.4.5
executed	<ExecutedEventMessage>	7.4.4.6
updated	<UpdatedEventMessage>	7.4.4.7

7.4.4.2 Structure de l'élément <CanceledEventMessage>

La structure du message d'événement dont l'élément racine est <CanceledEventMessage> est représentée à la Figure 25.

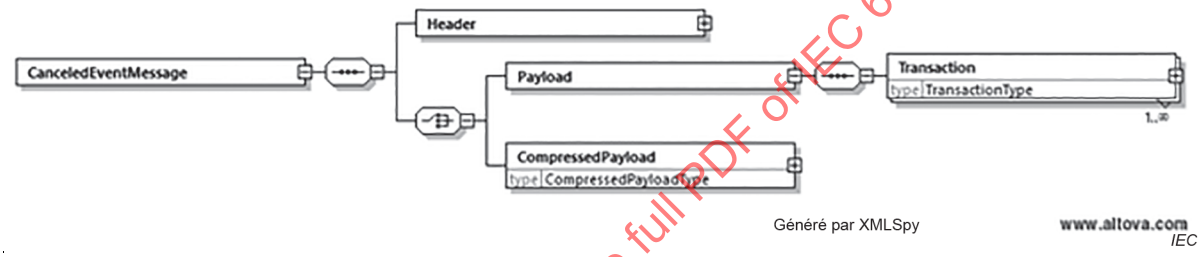


Figure 25 – Structure de l'élément <CanceledEventMessage>

7.4.4.3 Structure de l'élément <ChangedEventMessage>

La structure du message d'événement dont l'élément racine est <ChangedEventMessage> est représentée à la Figure 26.

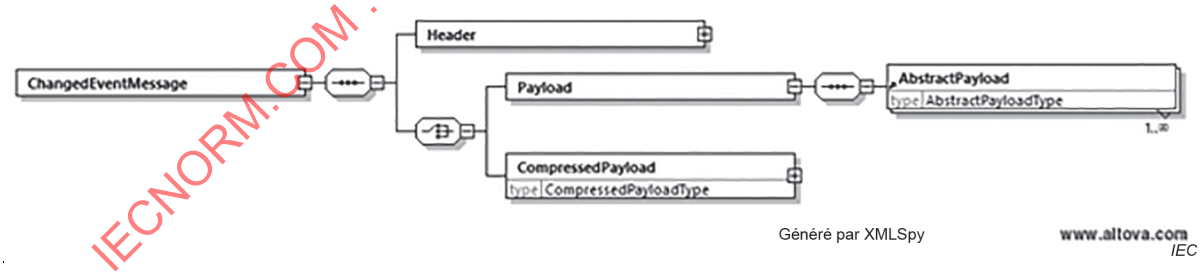


Figure 26 – Structure de l'élément <ChangedEventMessage>

7.4.4.4 Structure de l'élément <CreatedEventMessage>

La structure du message d'événement dont l'élément racine est <CreatedEventMessage> est représentée à la Figure 27.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

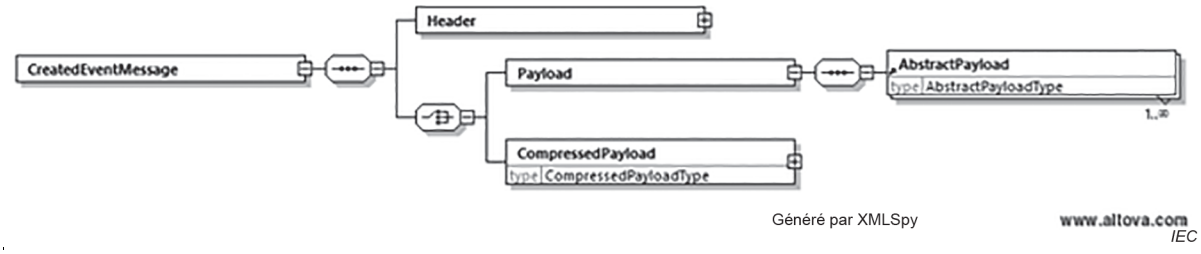


Figure 27 – Structure de l'élément <CreatedEventMessage>

7.4.4.5 Structure de l'élément <DeletedEventMessage>

La structure du message d'événement dont l'élément racine est <DeletedEventMessage> est représentée à la Figure 28.

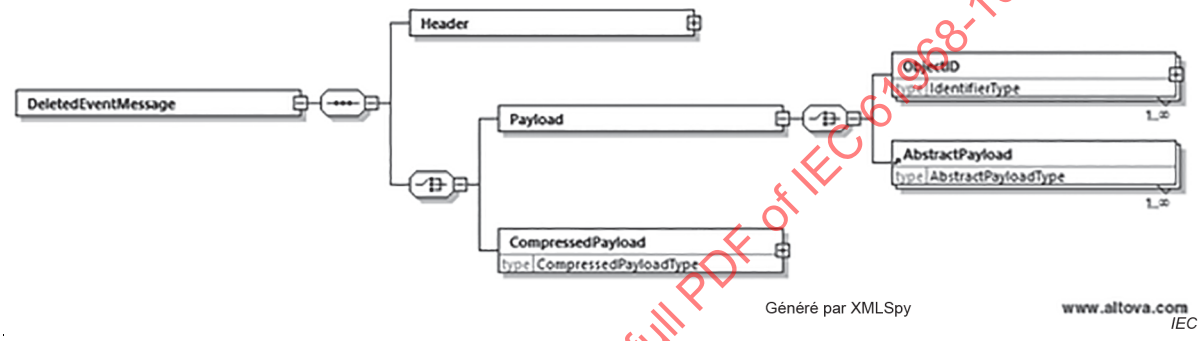


Figure 28 – Structure de l'élément <DeletedEventMessage>

7.4.4.6 Structure de l'élément <ExecutedEventMessage>

La structure du message d'événement dont l'élément racine est <ExecutedEventMessage> est représentée à la Figure 29.

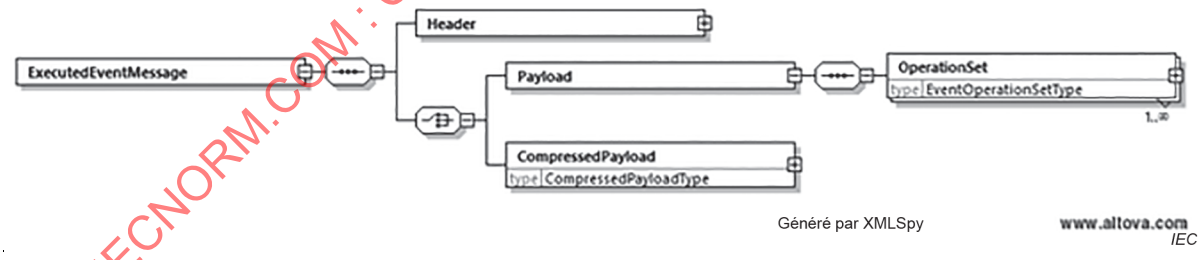


Figure 29 – Structure de l'élément <ExecutedEventMessage>

7.4.4.7 Structure de l'élément <UpdatedEventMessage>

La structure du message d'événement dont l'élément racine est <UpdatedEventMessage> est représentée à la Figure 30.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

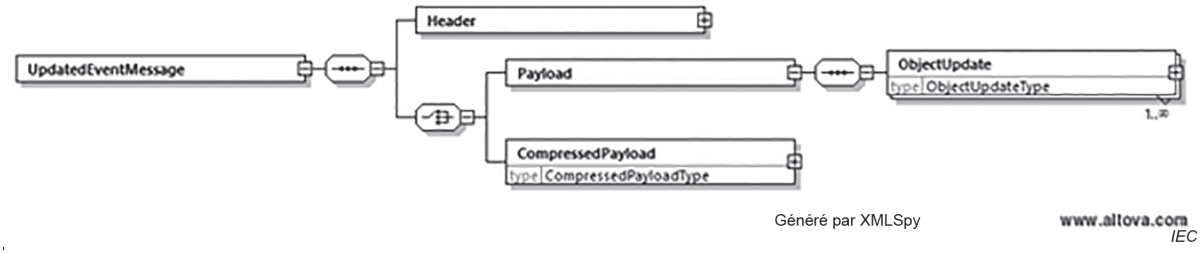


Figure 30 – Structure de l'élément <UpdatedEventMessage>

7.5 Élément <Header>

7.5.1 Généralités

Un message IEC 61968-100 autre qu'un message de défaut (5.5) doit contenir un élément <Header>.

L'élément <Header> contient des informations qui spécifient la manière dont le reste du message doit être traité.

La structure précise de l'élément <Header> doit dépendre de son utilisation dans le contexte d'un message de demande (7.5.3), d'un message de réponse (7.5.4) ou d'un message d'événement (7.5.5).

7.5.2 Sous-éléments de Message header (en-tête de message)

Le présent article décrit les différents éléments qui peuvent être présents dans un élément <Header> du message et leur sémantique associée.

NOTE Parmi les éléments du Tableau 4, <Source>, <User> et <Property> sont des éléments facultatifs qui peuvent faciliter l'intégration des systèmes. Les valeurs attribuées à ces éléments ne sont pas normatives et ont une signification uniquement dans la mise en œuvre locale (8.2.3).

Tableau 4 – Éléments dans l'élément <Header> du message

Élément d'en-tête	Plage de cardinalités		Type de données XSD	Signification
<Verb>	1	1	VerbType	Les éléments <Verb> et <Noun> doivent ensemble définir l'interprétation des données fournies dans les éléments <Reply> et <Payload> (7.5.3, 7.5.4, 7.5.5).
<Noun>	1	1	xs:string	
<Timestamp>	1	1	xs:dateTime	Cet élément doit indiquer à quel moment le message a été produit (il ne s'agit pas de l'heure de transmission du message, car l'envoi réel peut avoir lieu plus tard en raison de transactions ou de la mise en file d'attente des messages côté client). L'horodatage doit être conforme à l'ISO 8601. Il doit inclure soit la lettre "Z" pour indiquer le temps universel coordonné (TUC) soit un décalage horaire explicite.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Élément d'en-tête	Plage de cardinalités		Type de données XSD	Signification
<MessageID>	1	1	UUID	Cet élément doit être un identifiant unique universel (UUID) de 128 bits généré selon la RFC 4122. Il n'a aucune signification normative autre que celle de fournir un identifiant unique au message. Cet élément, accompagné de l'élément <Timestamp> est utile pour tracer ou consigner les messages au fur et à mesure de leur traitement par le système demandeur, le système répondant ou d'autres systèmes.
<CorrelationID>	1	1	UUID	Cet élément doit être un identifiant unique universel (UUID) de 128 bits généré selon la RFC 4122 (7.5.3.5, 7.5.4.5, 7.5.5.5).
<ReplyAddress>	0	1	xs:string	Cet élément doit être présent uniquement dans les messages de demande (5.2). Si un système répondant reçoit un message de demande qui contient cet élément, il doit vérifier que tous les messages de réponse ou messages d'événement induit sont acheminés vers l'adresse spécifiée. Si cet élément n'est pas fourni, tous les messages de réponse ou messages d'événement induit sont acheminés vers une adresse par défaut définie par le système répondant. L'adresse est en général une URL dans le cas d'un transport de services Web (9.2) ou le nom d'une file d'attente ou d'une rubrique dans le cas d'un transport JMS (9.3)
<AckRequired>	0	1	xs:boolean	Cet élément doit être présent uniquement dans les messages de demande (5.2). Cet élément a pour objet de permettre au système demandeur de déterminer que le message de demande a été délivré avec succès. Si cet élément est présent et que sa valeur est true, un message d'accusé de réception simple explicite (5.3.6, 6.5) doit être immédiatement envoyé par le système répondant dès la réception d'une demande. Si cet élément est omis ou qu'il est présent et que sa valeur est false, il n'est exigé d'envoyer aucun message d'accusé de réception simple. Ce type de message est un accusé de réception au niveau de l'application et ne dépend pas du transport sous-jacent.
<MessageSequence>	0	1	MessageSequenceType	Cet élément doit être présent uniquement dans les messages de réponse et messages d'événement induit (7.5.4.6, 7.5.5.6).
<SequenceNumber>	0	1	xs:unsignedInt	Si cet élément est présent, un système répondant doit lui attribuer la valeur 1 pour la première réponse ou le message d'événement induit à transmettre. Il doit être incrémenté pour chaque message suivant envoyé.
<LastMessage>	0	1	xs:boolean	Voir 7.5.4.6 et 7.5.5.6.
<Source>	0	1	xs:string	Si cet élément est présent, il doit être utilisé pour identifier l'origine du contenu informationnel. Les valeurs attribuées à cet élément ne sont pas normatives et ont une signification uniquement dans l'organisme local.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Élément d'en-tête	Plage de cardinalités		Type de données XSD	Signification
<User>	0	1	UserType	Si cet élément est présent, il doit identifier l'utilisateur est (éventuellement) l'organisme associé. Il peut s'avérer utile à des fins d'audit. Les valeurs attribuées à cet élément ne sont pas normatives et ont une signification uniquement dans l'organisme local.
<UserID>	1	1	xs:string	Cet élément doit identifier l'utilisateur lié à la création du message. Les valeurs attribuées à cet élément ne sont pas normatives et ont une signification uniquement dans la mise en œuvre locale.
<Organisation>	0	1	xs:string	Si cet élément est présent, il doit identifier le nom de l'organisme auquel l'utilisateur est associé. Les valeurs attribuées à cet élément ne sont pas normatives et ont une signification uniquement dans la mise en œuvre locale.
<Comment>	0	1	xs:string	Cet élément peut contenir du texte. Il n'a aucune signification normative.
<Property>	0	plusieurs	MessagePropertyType	Cet élément est composé d'un ensemble de paires nom/valeur qui peuvent être utilisées pour personnaliser le comportement du système au-delà de celui défini par la présente Norme internationale. Il peut être utilisé pour assurer certaines fonctionnalités spécifiques à la mise en œuvre. Les valeurs attribuées à ces éléments ne sont pas normatives et ont une signification uniquement dans la mise en œuvre locale. Pour les besoins de l'interopérabilité, il convient d'éviter d'utiliser l'élément <Property> dans toute la mesure du possible.
<Name>	1	1	xs:string	Nom de la propriété
<Value>	0	1	xs:string	Valeur associée de la propriété

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.5.3 En-tête des messages de demande

7.5.3.1 Généralités

Le présent article définit l'utilisation des sous-éléments <Header> dans les messages de demande (5.2).

7.5.3.2 Structure

La Figure 31 décrit la structure de l'élément <Header> telle que définie par le fichier Message.xsd pour les messages de demande (5.2).

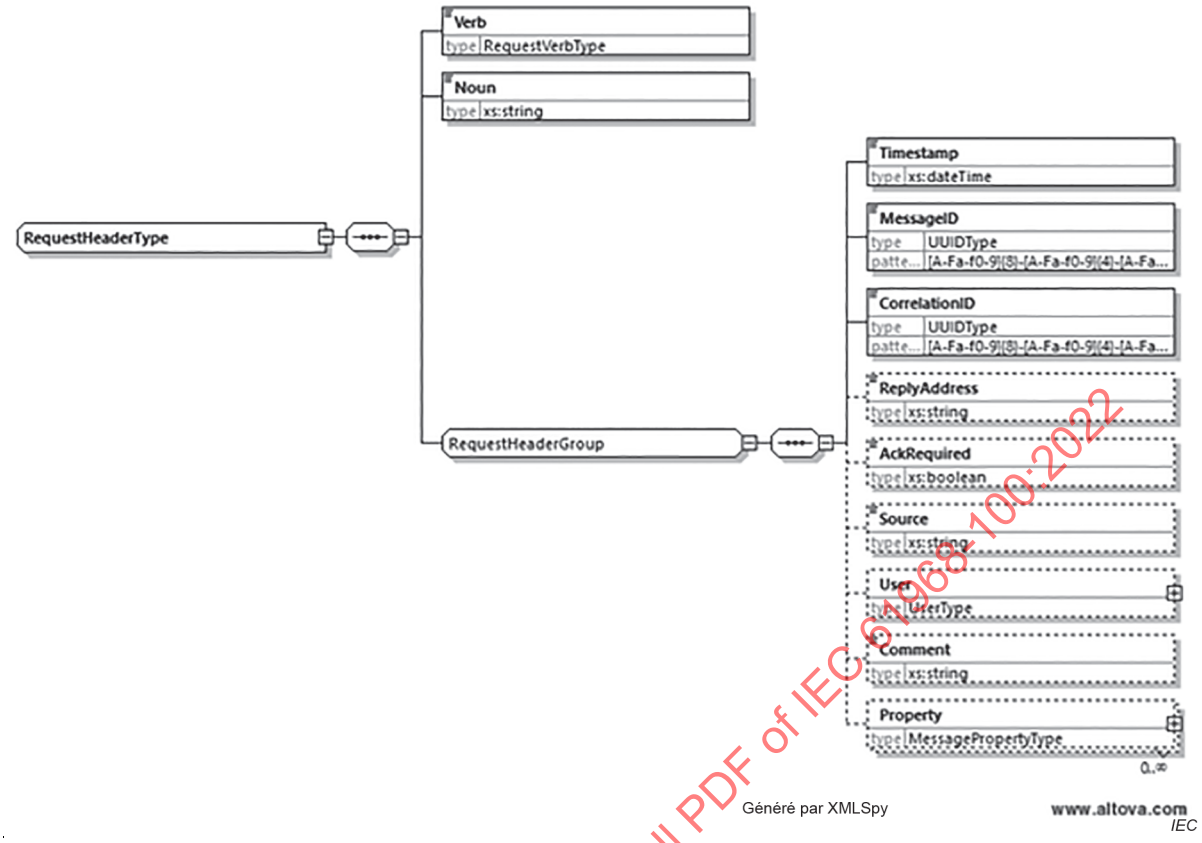


Figure 31 – Structure de l'élément <Header> pour les messages de demande

NOTE À la Figure 31, RequestHeaderGroup indique un groupe nommé d'éléments XML référencés dans les différentes parties du fichier Message.xsd. Il n'indique pas un élément XML.

7.5.3.3 Élément <Header><Verb>

La valeur de l'élément <Header><Verb> doit être l'une des valeurs qui figurent dans le Tableau 5.

Tableau 5 – Verbes utilisés dans les messages de demande

Verbe	Signification	Voir Article/paragraphe
cancel	Spécifie une demande d'annulation d'une ou de plusieurs transactions préalablement demandées dans le système répondant.	8.5
change	Spécifie une demande de remplacement de l'ensemble des données relatives à un ou plusieurs objets existants dans le système répondant par les données spécifiques à l'application fournies dans la section <Payload> du message.	8.3.3
create	Spécifie une demande d'instanciation d'un ou de plusieurs nouveaux objets dans le système répondant avec les données spécifiques à l'application fournies dans la section <Payload> du message.	8.3.2
delete	Spécifie une demande de suppression dans leur ensemble d'un ou de plusieurs objets existants dans le système répondant.	8.3.4
execute	Spécifie une demande d'exécution d'une transaction composite dans le système répondant telle que spécifiée dans la section <Payload><OperationSet> du message.	7.7.3
get	Spécifie une demande d'extraction d'un ensemble d'objets du système répondant telle que spécifiée dans la section <Payload> du message.	8.4
update	Spécifie une demande de modification de tout ou partie des données relatives à un ou plusieurs objets existants sur le système répondant.	8.3.5

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.5.3.4 Élément <Header><Noun>

L'élément <Header> doit contenir un sous-élément <Noun>.

Si l'élément <Verb> est défini sur cancel, le sous-élément <Noun> doit être le même que celui présent dans le message de demande d'origine en cours d'annulation.

Si <Verb> est défini sur change ou create, le sous-élément <Noun> doit spécifier le nom de l'élément remplacé par l'élément <Payload><AbstractPayload> (7.7.5, 8.3).

Si <Verb> est défini sur delete, le sous-élément <Noun> doit alors spécifier le nom de l'élément remplacé par l'élément <Payload><AbstractPayload> qui a été spécifié lors de l'instanciation initiale de l'objet (8.3.2.1).

Si <Verb> est défini sur execute, le sous-élément <Noun> doit être OperationSet (7.7).

Si <Verb> est défini sur get, le sous-élément <Noun> doit spécifier le nom de l'élément remplacé par l'élément <Payload><AbstractPayload> du message de réponse attendu (5.3, 7.7.5, 8.4).

Si <Verb> est défini sur update, le sous-élément <Noun> doit spécifier le nom de l'élément remplacé par l'élément <Payload><ObjectUpdate><AbstractPayload>.

7.5.3.5 Élément <Header><CorrelationID>

L'élément <Header> doit contenir un sous-élément <CorrelationID>. Cela permet à un système demandeur de mettre en corrélation les messages de réponse et d'événement induit suivants avec le message de demande d'origine.

Un nouveau sous-élément <CorrelationID> doit être généré pour chaque message de demande séparé.

7.5.4 En-tête des messages de réponse

7.5.4.1 Généralités

Le présent article définit l'utilisation de l'élément <Header> dans les messages de réponse (5.3).

7.5.4.2 Structure

La Figure 32 décrit la structure de l'élément <Header> telle que définie par le fichier Message.xsd pour les messages de réponse (5.3).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

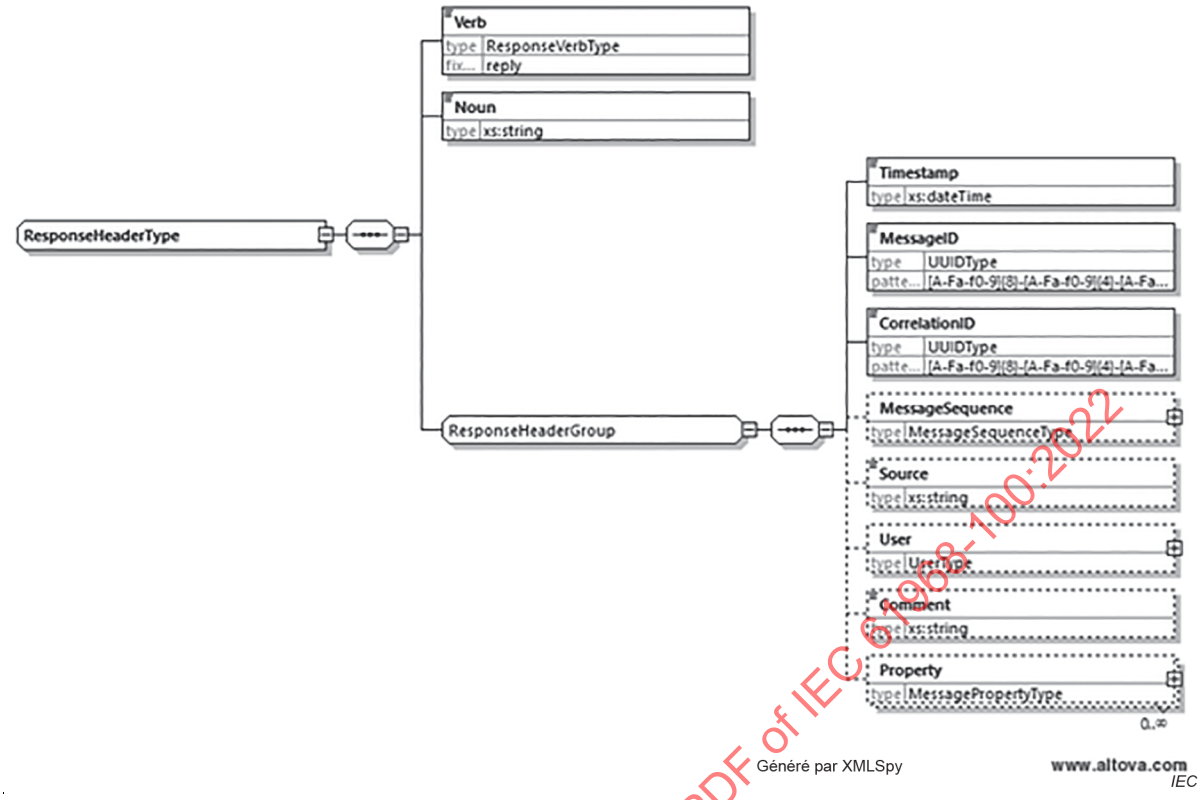


Figure 32 – Structure de l'élément <Header> pour les messages de réponse

NOTE À la Figure 32, ResponseHeaderGroup indique un groupe nommé d'éléments XML référencés dans les différentes parties du fichier Message.xsd. Il n'indique pas un élément XML.

7.5.4.3 Élément <Header><Verb>

La valeur de l'élément `<Header><Verb>` doit être `reply` pour un message de réponse.

7.5.4.4 Élément <Header><Noun>

L'élément `<Header>` doit contenir un sous-élément `<Noun>`.

L'élément `<Noun>` contenu dans un message de réponse doit être identique à l'élément `<Noun>` du message de demande d'origine pour lequel ce message de réponse est généré.

7.5.4.5 Élément <Header><CorrelationID>

L'élément `<Header>` dans le message de réponse doit contenir un élément `<CorrelationID>` identique à l'élément `<CorrelationID>` du message de demande d'origine pour lequel ce message de réponse est généré (7.5.5.5).

Cela permet à un système demandeur de mettre en correspondance les messages de réponse reçus avec le message de demande d'origine.

En présence de plusieurs messages de réponse (6.2.2), chacun d'eux doit contenir la même valeur pour leurs éléments `<CorrelationID>` respectifs.

7.5.4.6 Élément <Header><MessageSequence>

Un seul message de demande peut générer plusieurs messages de réponse (4.7.2, 6.2.2).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

L'élément <MessageSequence> doit être inclus lorsque plusieurs messages de réponse (5.3) doivent être envoyés et que le système répondant peut soit identifier la séquence de messages soit déterminer le dernier message de ce type de séquence.

Les règles précises qu'un système répondant doit édicter lors de la génération d'un message de réponse dans un ensemble possible de plusieurs messages sont les suivantes:

1) Si le système répondant génère exactement un message, l'élément <Header><MessageSequence> doit alors être ignoré.
2) Sinon, si le système répondant génère un message réputé être le dernier d'une séquence de ce type de messages de réponse, l'élément <Header><MessageSequence> doit alors être inclus et la valeur true doit être attribuée à l'élément <Header><MessageSequence><LastMessage>.
3) Sinon, le système répondant génère un message qui est soit réputé ne pas être le dernier d'une séquence de ce type de messages de réponse soit qui peut être le dernier, mais ce dont le système répondant n'a pas conscience. L'élément <Header><MessageSequence> doit être inclus et la valeur false doit être attribuée à l'élément <Header><MessageSequence><LastMessage>.

Ces règles permettent à un système de réception d'ordonner correctement les messages reçus et de déterminer à quel moment il les a reçus, même s'ils ont été livrés dans le désordre.

7.5.5 En-tête des messages d'événement

7.5.5.1 Généralités

Le présent article définit l'utilisation de l'élément <Header> dans les messages d'événement (5.4).

7.5.5.2 Structure

La Figure 33 décrit la structure de l'élément <Header> telle que définie par le fichier Message.xsd pour les messages d'événement (5.4).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

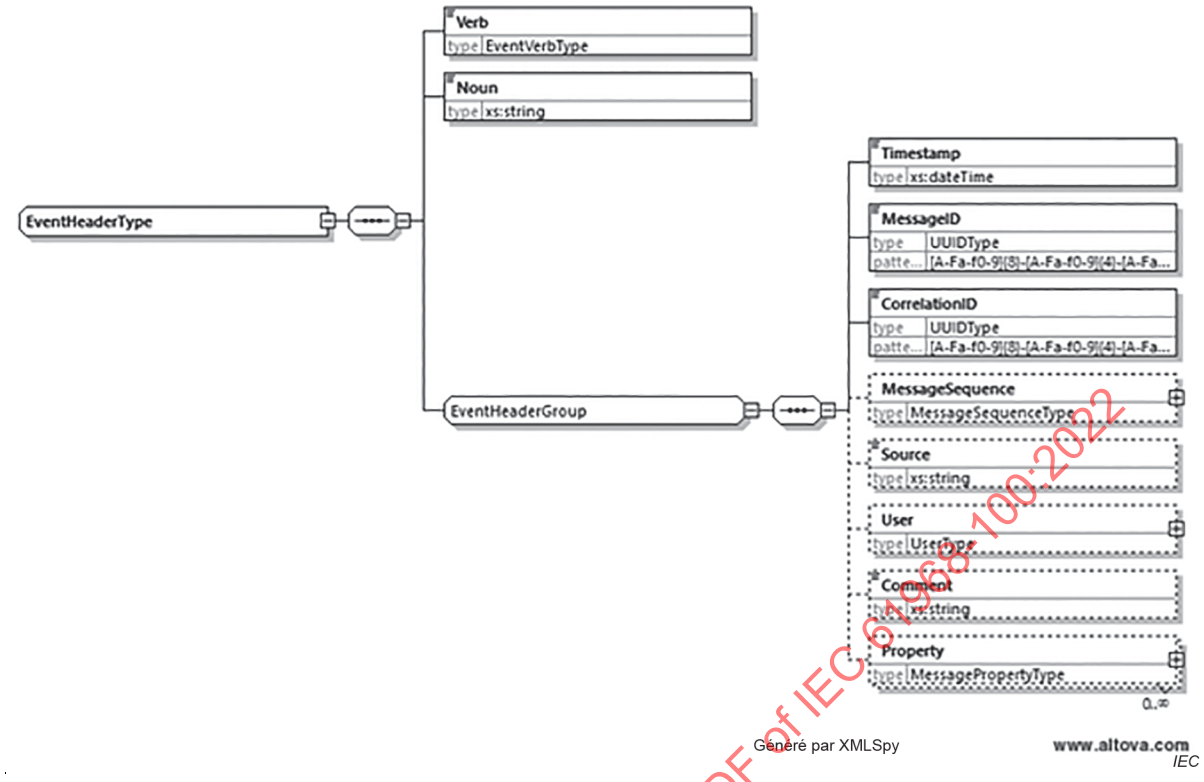


Figure 33 – Structure de l'élément <Header> pour les messages d'événement

NOTE À la Figure 33, EventHeaderGroup indique un groupe nommé d'éléments XML référencés dans les différentes parties du fichier Message.xsd. Il n'indique pas un élément XML.

7.5.5.3 Élément <Header><Verb>

<Header><Verb> doit appartenir à l'un des ensembles énumérés qui figurent dans le Tableau 6.

Tableau 6 – Verbes utilisés dans les messages d'événement

Verbe	Signification
canceled	Utilisé pour publier une notification selon laquelle une ou plusieurs actions ont été abandonnées.
changed	Utilisé pour publier une notification selon laquelle les données relatives à un ou plusieurs objets existants ont été totalement remplacées.
created	Utilisé pour publier une notification de l'instanciation d'un ou de plusieurs objets.
deleted	Utilisé pour publier une notification selon laquelle un ou plusieurs objets existants ont été totalement supprimés.
executed	Utilisé pour publier une notification selon laquelle l'exécution d'une transaction complexe qui utilise l'élément <Payload><OperationSet> a eu lieu.
updated	Utilisé pour publier une notification selon laquelle les données relatives à un ou plusieurs objets ont été totalement ou en partie modifiées.

7.5.5.4 Élément <Header><Noun>

L'élément <Header> doit contenir un sous-élément <Noun>.

Si l'élément <Verb> est défini sur canceled, le sous-élément <Noun> doit être le même que celui présent dans le message de demande d'origine en cours d'annulation.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Si l'élément <Verb> est défini sur changed, created ou deleted, le sous-élément <Noun> doit spécifier le nom du profil qui apparaît dans la section AbstractPayloadType de l'élément <Payload>.

Si <Verb> est défini sur executed, le sous-élément <Noun> doit être OperationSet.

Si <Verb> est défini sur updated, le sous-élément <Noun> doit spécifier le nom du profil qui apparaît dans la section AbstractPayloadType de l'élément <Payload><ObjectUpdate>.

7.5.5.5 Élément <Header><CorrelationID>

7.5.5.5.1 Message d'événement non sollicité

Un message d'événement non sollicité (6.3) doit contenir un élément <CorrelationID>. Ce sous-élément <CorrelationID> doit être recréé pour chaque nouveau message d'événement non sollicité.

7.5.5.5.2 Message d'événement induit

Un message d'événement induit (6.4) doit contenir un élément <CorrelationID>. La valeur de cet élément doit être définie en fonction de celle du message de demande d'origine qui est directement responsable de la création de l'événement induit.

7.5.5.6 Élément <Header><MessageSequence>

Un seul message de demande peut générer plusieurs messages d'événements induits (6.4).

L'élément <MessageSequence> doit être inclus lorsque plusieurs messages d'événements induits (5.4) doivent être envoyés et que le système répondant peut soit identifier la séquence de messages soit déterminer le dernier message de ce type de séquence.

Les règles précises qu'un système répondant doit édicter lors de la génération d'un message d'événement induit dans un ensemble possible de plusieurs messages sont les suivantes:

1) Si le système répondant génère exactement un message, l'élément <Header><MessageSequence> doit alors être ignoré.
2) Sinon, si le système répondant génère un message réputé être le dernier d'une séquence de ce type de messages d'événements induits, l'élément <Header><MessageSequence> doit alors être inclus et la valeur true doit être attribuée à l'élément <Header><MessageSequence><LastMessage>.
3) Sinon, le système répondant génère un message qui est soit réputé ne pas être le dernier d'une séquence de ce type de messages d'événements induits soit qui peut être le dernier, mais ce dont le système répondant n'a pas conscience. L'élément <Header><MessageSequence> doit être inclus et la valeur false doit être attribuée à l'élément <Header><MessageSequence><LastMessage>.

Ces règles permettent à un système de réception d'ordonner correctement les messages reçus et de déterminer à quel moment il les a tous reçus, même s'ils ont été livrés dans le désordre.

7.6 Élément <Reply> (Réponse)

7.6.1 Généralités

L'élément <Reply> doit être utilisé uniquement dans le contexte des archétypes de message de réponse ou de message de défaut (5.3, 5.5).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.6.2 Structure

La structure de l'élément <Reply> est représentée à la Figure 34.

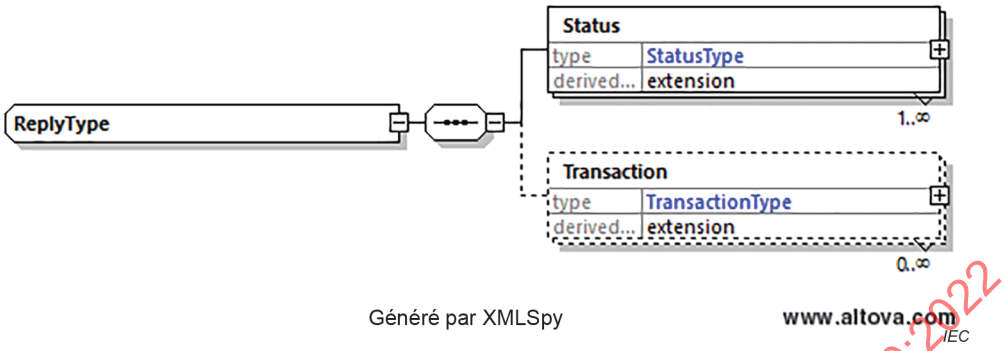


Figure 34 – Structure de l'élément <Reply>

7.6.3 Sous-éléments de Reply

Le Tableau 7 décrit les différents sous-éléments qui peuvent être présents dans un élément <Reply> et leur sémantique associée.

Tableau 7 – Sous-éléments de l'élément <Reply>

Élément	Plage de cardinalités		Type de données XSD	Signification
<Status>	1	plusieurs	StatusType	Voir 7.6.4
<Transaction>	0	plusieurs	TransactionType	Voir 7.6.4.10

7.6.4 Sous-élément <Reply><Status>

7.6.4.1 Généralités

L'élément <Reply><Status> doit être utilisé pour renvoyer des notifications informationnelles et d'erreur provenant du système répondant à un système demandeur.

Un élément <Reply> doit contenir au moins un sous-élément <Status>. Chaque notification informationnelle ou d'erreur séparée doit être attribuée à une instance <Reply><Status> distincte.

Chaque élément <Reply><Status> doit contenir un seul sous-élément <Level> et un seul sous-élément <Code> (7.6.4.6, 7.6.4.7).

7.6.4.2 Réponse avec une notification de succès

Une notification selon laquelle l'ensemble du message de demande d'origine a été réalisé avec succès doit être indiquée par un élément <Reply><Status><Code> défini sur 0.0 et un élément <Reply><Status><Level> défini sur OK. Ce type de notification de succès doit être envoyé uniquement s'il n'y a aucune notification informationnelle ou d'erreur à acheminer (7.6.4.3).

7.6.4.3 Réponse avec des notifications de succès, d'erreur ou d'autres informations

Un élément <Reply><Status><Code> autre que 0.0 indique une notification informationnelle ou d'erreur.

NOTE Si une réponse est composée de plusieurs messages, un système demandeur peut déterminer les cas qui ont abouti uniquement lorsque tous les messages de réponse ont été reçus.

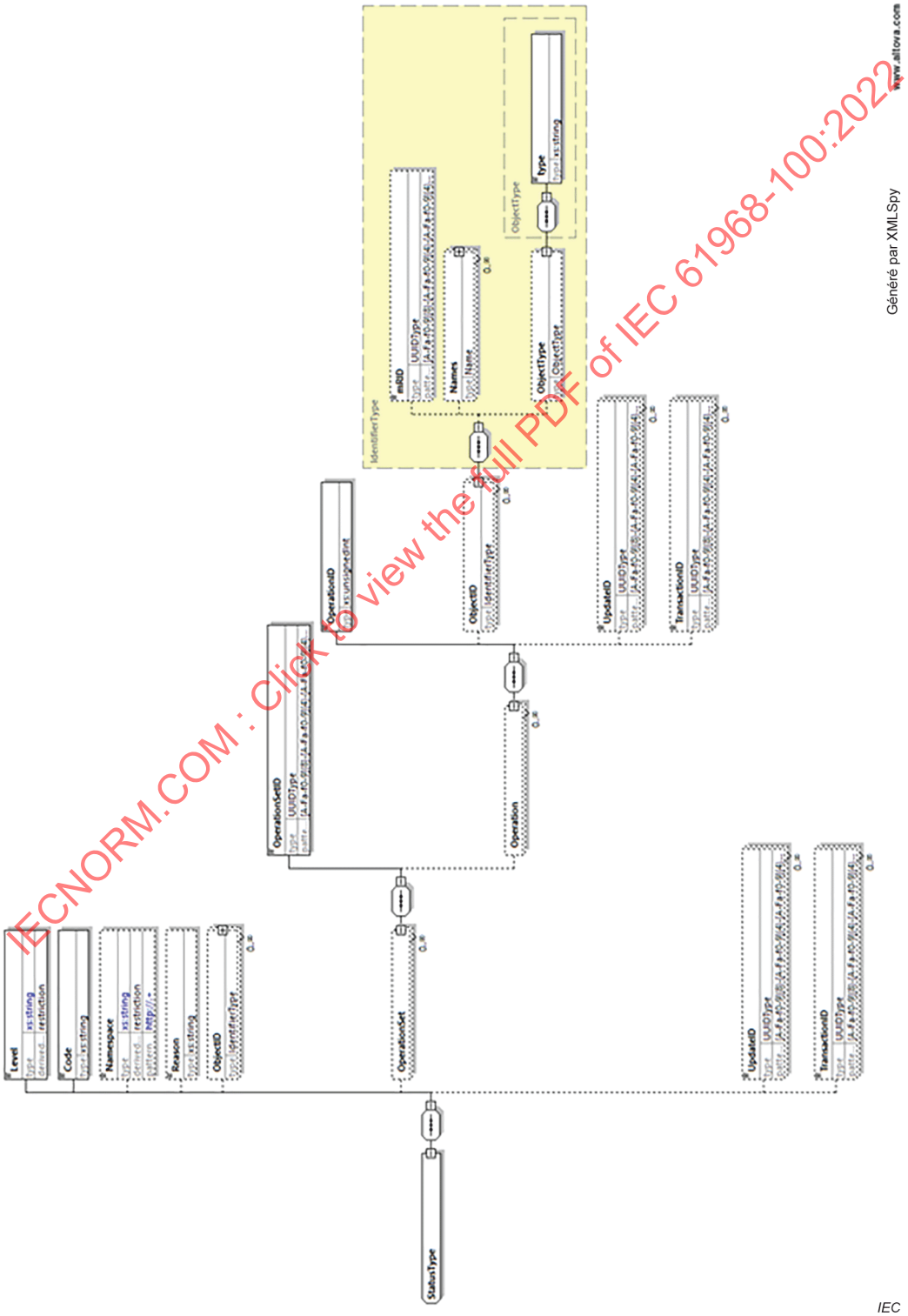
IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

EXEMPLE D.9.4 donne un exemple de message de réponse pour les cas qui ne comportent aucune indication explicite de succès. Un seul élément `<Reply><Status>` est inclus.

EXEMPLE D.11.3 donne un exemple de messages de réponse dans lequel une notification d'erreur explicite est incluse.

7.6.4.4 Structure

La structure du schéma XML correspondant à l'élément `<Reply><Status>` est représentée à la Figure 35.



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 35 – Structure de l'élément `<Reply><Status>`

7.6.4.5 Sous-éléments <Reply><Status>

Le Tableau 8 décrit les différents sous-éléments qui peuvent être présents dans un sous-élément <Reply><Status> du message et leur sémantique associée.

Tableau 8 – Sous-éléments de l'élément <Reply><Status>

Élément	Plage de cardinalités		Type de données XSD	Signification
<Level>	1	1	énumération	Voir 7.6.4.6.
<Code>	1	1	xs:string	Voir 7.6.4.7.
<Namespace>	0	1	URIType	Voir 7.6.4.8.
<Reason>	0	1	xs:string	Si cet élément est présent, il doit contenir un texte libre qui clarifie la notification informationnelle ou d'erreur. Cet élément n'a aucune signification normative.
<ObjectID>	1	plusieurs	IdentifierType	Si ces éléments sont présents, ils doivent spécifier les objets auxquels cette notification d'erreur ou d'état fait référence (7.6.4.9, 8.6.5).
<OperationSet>	1	plusieurs		Cet élément doit être utilisé uniquement si la réponse est envoyée dans le cadre d'une demande OperationSet.
<OperationSetID>	1	1	UUIDType	Cet élément doit contenir la valeur de <Payload><OperationSet><OperationSetID> de la demande d'origine. Cela permet à un système demandeur de déterminer l'ensemble d'opérations précis pour lequel la notification est générée (7.7.3).
<Operation>	0	plusieurs		
<OperationID>	1	1	xs:unsignedInt	Si la réponse est envoyée après une demande OperationSet, cet élément doit contenir la valeur de <Payload><OperationSet><Operation><OperationID> de la demande d'origine. Cela permet à un système demandeur de déterminer l'opération exacte pour laquelle la notification est générée (7.7.3).
<ObjectID>	1	plusieurs	IdentifierType	
<mRID>	0	1	UUIDType	Si la réponse est envoyée après une demande OperationSet, ces éléments doivent contenir les valeurs des sous-éléments correspondants de l'élément <Payload><OperationSet><Operation> <ObjectID> ou de l'élément correspondant dans l'élément AbstractPayload de la demande d'origine. Cela permet à un système demandeur de déterminer l'opération exacte pour laquelle la notification d'erreur est générée.
<Names>	0	plusieurs	Nom	
<ObjectType>	0	1	ObjectType	
<type>	1	1	xs:string	
<UpdateID>	1	plusieurs		Si la réponse est envoyée après une opération de mise à jour dans une demande OperationSet, cet élément doit contenir la valeur de l'élément <Payload><OperationSet><Operation><ObjectUpdate><UpdateID> de la demande d'origine. Cela permet à un système demandeur de déterminer l'opération de mise à jour exacte pour laquelle la notification d'erreur est générée (7.7.3, 8.3.5).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Élément	Plage de cardinalités		Type de données XSD	Signification
<TransactionID>	1	plusieurs		Si la réponse est envoyée après une demande OperationSet pour annuler une demande en cours, cet élément doit contenir les ID de transaction pertinents.
<UpdateID>	0	1	UUIDType	Si la réponse est envoyée après une demande de mise à jour d'un objet, cet élément doit contenir la valeur de l'élément <Payload><ObjectUpdate><UpdateID> de la demande d'origine. Cela permet à un système demandeur de déterminer l'opération de mise à jour exacte pour laquelle la notification d'erreur est générée (7.7.3, 8.3.5).
<TransactionID>	1	plusieurs	UUIDType	Si la réponse est envoyée après une demande d'annulation d'une opération en cours, cet élément doit contenir les ID de transaction pertinents (7.9, 8.5).

7.6.4.6 Élément <Reply><Status><Level>

L'élément <Reply><Status><Level> est une énumération, et l'une des valeurs qui figurent dans le Tableau 9 doit lui être attribuée. Les éléments <Reply><Status><Code> et <Reply><Status><Reason> peuvent contenir de plus amples informations explicatives.

Tableau 9 – Valeurs de l'élément <Reply><Status><Level>

Valeur	Signification
OK	Cette valeur doit indiquer que l'opération a abouti.
WARNING	Cette valeur doit indiquer que l'opération a abouti, mais qu'il peut y avoir des questions exigeant d'être examinées.
ERROR	Cette valeur doit indiquer que l'opération n'a pas abouti.
UNKNOWN	Cette valeur doit indiquer que le système répondant ne peut pas certifier que l'opération a abouti.

7.6.4.7 Élément <Reply><Status><Code>

L'élément <Reply><Status><Code> doit indiquer le code d'état spécifique.

Sauf si une valeur dépendant de l'application est attribuée à l'élément <Reply><Status><Code> (7.6.4.8), l'un des ensembles normatifs de valeurs qui figurent dans le Tableau 10 doit être attribué à l'élément <Reply><Status><Code>. Ce tableau présente également les valeurs correspondantes de l'élément <Reply><Status><Level>.

Tableau 10 – Valeurs normatives de l'élément <Reply><Status><Code>

Valeur	Signification	Valeur <Level> correspondante	Notes d'utilisation
0.0	Le message ne signale aucune erreur	OK	Si plusieurs messages de réponse sont générés (6.2.2), les autres messages peuvent signaler des codes d'erreur ou d'état.
0.3	Accusé de réception simple	OK	Réservé aux messages d'accusé de réception simple (5.3.6).
1.1	Des éléments IEC 61968-100 obligatoires sont absents ou non valides	ERROR	
1.2	Le message n'a pas réussi la validation de schéma XML	ERROR	

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Valeur	Signification	Valeur <Level> correspondante	Notes d'utilisation
1.3	Espace de nom IEC 61968-100 spécifié non pris en charge	ERROR	Utilisé lorsqu'une édition particulière de l'IEC 61968-100 (telle qu'indiquée par l'espace de nom dans le message de demande) n'est pas prise en charge par le système répondant.
1.4	Le message ne peut pas être analysé comme un document XML	ERROR	Voir 5.5.
2.1	Paramètre non pris en charge	ERROR	
2.2	Élément d'en-tête de message non valide	ERROR	
2.3	Élément de réponse de message non valide	ERROR	
2.4	Élément de charge utile de message non valide	ERROR	Utilisé pour spécifier qu'un élément de charge utile spécifique à la Partie 100 (c'est-à-dire non spécifique à l'application) n'est pas valide (<CompressedPayload><CompressionKind>, par exemple). Pour les erreurs de données spécifiques à l'application dans la charge utile, utiliser le code 5.2.
2.5	Objet référencé pas identifié de manière unique	ERROR	Utilisé lorsque la référence à l'objet n'identifie pas de manière unique un seul objet (par exemple, une référence <Names><name> est utilisée lorsque de nombreux objets sont associés à ce nom).
3.1	Limite du système dépassée	ERROR	Utilisé pour les messages qui dépassent la limite de capacité du système (lorsque les limites de taille de message ou les limites d'infrastructure telles que la mémoire ou l'espace disque sont dépassées, par exemple).
3.2	Limite d'application dépassée	ERROR	Utilisé pour les messages qui dépassent une limite imposée par l'application (trop d'éléments d'un certain type, par exemple).
4.1	Délai de la demande expiré	ERROR	Utilisé lorsque la demande ne peut pas être traitée dans le temps imparti.
4.2	Demande annulée	ERROR	Utilisé lorsqu'un processus système ou un utilisateur sur le système répondant annule activement une demande avant son exécution. Pour les demandes annulées par un système demandeur, voir 8.5.
4.3	Service non disponible	ERROR	Utilisé lorsqu'une autre application ou un autre service n'est pas disponible pour traiter cette demande.
5.0	Erreur d'application générale	ERROR	Une application peut utiliser ce code pour indiquer qu'une erreur s'est produite pendant le traitement.
5.1	Impossible d'annuler la transaction	ERROR	Utilisé uniquement pour les réponses à une demande de transaction d'annulation (8.5).
5.2	Données spécifiques à l'application spécifiées non prises en charge	ERROR	Utilisé lorsque le système répondant ne prend pas en charge les données d'application spécifiées. Par exemple, il est demandé à un système répondant de créer un objet dont le type n'est pas pris en charge par le système ou un code IEC 61968-9 ReadingType ou EndDeviceControls est spécifié. Peut également être utilisé lorsqu'une combinaison d'éléments de données spécifiques à l'application n'est pas valide.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Valeur	Signification	Valeur <Level> correspondante	Notes d'utilisation
5.3	L'objet référencé n'existe pas	ERROR	L'objet référencé correspondant à ce code peut être soit un objet de données spécifique à l'application soit un objet IEC 61968-100 général (TransactionID ou OperationID, par exemple). L'élément approprié de <StatusCode> (<ObjectID>, <OperationSetID>, etc.) peut être utilisé pour spécifier plusieurs informations relatives à l'objet référencé.
5.4	L'objet référencé existe déjà	ERROR	L'objet référencé correspondant à ce code peut être soit un objet de données spécifique à l'application soit un objet IEC 61968-100 général (TransactionID ou OperationID, par exemple). L'élément approprié de <StatusCode> (<ObjectID>, <OperationSetID>, etc.) peut être utilisé pour spécifier plusieurs informations relatives à l'objet référencé.
5.5	L'objet en cours de modification n'est pas identifié de manière unique	ERROR	Utilisé pour un message <ObjectIdentificationRevisionSets> qui tente d'identifier un objet et le <mRID> n'identifie pas un objet simple de manière unique (8.7.6).
5.6	Données spécifiques à l'application obligatoires manquantes	ERROR	Une application peut utiliser ce code d'erreur pour indiquer qu'il manque certaines données nécessaires à la réalisation de la demande spécifiée.
5.7	Impossible d'assurer l'intégrité transactionnelle	ERROR	Une ou plusieurs opérations d'un OperationSet ont été annulées ou répétées afin de maintenir l'intégrité transactionnelle. Utilisé lorsqu'un OperationSet exige l'application de l'intégrité transactionnelle.
5.8	Le <mRID> ajouté n'est pas unique	ERROR	Utilisé pour un message <ObjectIdentificationRevisionSets> qui tente d'ajouter un <mRID> qui existe déjà pour un autre objet (8.7.6).
5.9	La suppression du dernier <mRID> restant est interdite	ERROR	Utilisé pour un message <ObjectIdentificationRevisionSets> qui tente de supprimer le dernier identificateur <mRID> restant d'un objet (8.7.6).
5.10	<mRID> ou <Name> en cours de suppression n'existe pas	ERROR	Utilisé pour un message <ObjectIdentificationRevisionSets> qui tente de supprimer un <mRID> ou un <Name> inexistant d'un objet donné (8.7.6).
5.11	L'élément <ObjectType> ne correspond pas au type de l'objet	ERROR	Utilisé lorsqu'un <mRID> ou un <Name> est associé à un <ObjectType> et qu'il n'existe aucun objet de ce type avec l'identificateur donné.
5.12	Relation d'objet non valide	ERROR	Utilisé pour indiquer qu'au moins deux objets sont associés dans une charge utile spécifique à l'application dont la relation n'est pas valide. Utiliser l'élément <ObjectID> dans la réponse pour spécifier les objets spécifiques en question. Par exemple, une demande de suppression d'un objet IEC 61968-9 Compteur d'un UsagePoint génère cette erreur si le Compteur spécifié n'est pas installé au niveau de cet UsagePoint.
6.0	Violation des règles commerciales	ERROR, WARNING ou UNKNOWN	Une application peut utiliser ce code d'état pour indiquer que certaines règles commerciales ont été violées.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Valeur	Signification	Valeur <Level> correspondante	Notes d'utilisation
6.1	Demande rejetée selon les règles commerciales	ERROR	Utilisé pour indiquer lorsqu'une règle commerciale interdit l'exécution d'une demande. Par exemple, une demande de déconnexion d'un compteur peut être rejetée du fait de l'existence d'une règle commerciale qui interdit les opérations de déconnexion en hiver.
6.2	Demande mise en attente selon les règles commerciales	WARNING	Utilisé pour indiquer le cas dans lequel une règle commerciale exige la mise en attente de l'opération demandée avant exécution. Par exemple, une demande de déconnexion d'un compteur peut être mise en attente du fait de l'existence d'une règle commerciale qui exige l'examen et l'agrément d'un utilisateur autorisé avant exécution.
6.3	Demande libérée de la règle commerciale de mise en attente	OK	Utilisé pour indiquer qu'une demande a été libérée d'une règle commerciale précédente de mise en attente et peut désormais être exécutée. Par exemple, une demande de déconnexion d'un compteur précédemment mise en attente a été examinée et agréée, et peut désormais être exécutée.
6.4	Demande reprogrammée selon les règles commerciales	WARNING	Utilisé pour indiquer le cas dans lequel une règle commerciale exige la reprogrammation de l'opération demandée pour une date et/ou une heure différentes. Par exemple, il peut s'avérer nécessaire de reprogrammer sur 8:00 AM une demande de déconnexion d'un compteur envoyée à 1:00 AM du fait de l'existence d'une règle commerciale qui exige l'exécution de ce type d'opérations pendant les heures normales de travail.
7.0	Violation de sécurité	ERROR	Une application peut utiliser ce code d'état pour indiquer qu'une tentative d'atteinte à certaines mesures de sécurité a été détectée.
99.x	Autres états qui ne figurent pas dans le présent tableau	OK, ERROR, WARNING ou UNKNOWN	Cette série de codes peut être utilisée pour étendre l'ensemble des codes d'état en tant que candidats à l'inclusion dans l'édition suivante de l'IEC 61968-100. Il convient de ne pas utiliser cette série pour les codes spécifiques à l'application.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.6.4.8 Élément <Reply><Status><Namespace>

Une valeur dépendante de l'application ne figurant pas dans le Tableau 10 peut être attribuée à l'élément <Reply><Status><Code>.

Pour éviter toute ambiguïté, ce type de valeur doit être associé à un espace de nom spécifique à l'application. Cet espace de nom doit être attribué à l'élément <Reply><Status><Namespace>.

Pour éviter les codes d'état équivoques définis par les normes publiées par le comité d'études 57 de l'IEC, l'espace de nom doit se présenter sous la forme suivante:

http://iec.ch/TC57/*<year-of-publication>*/*<publication-id>*#.

EXEMPLE http://iec.ch/TC57/2013/61968-9# est un exemple de ce type d'espace de nom.

Si aucun élément <Reply><Status><Namespace> n'est spécifié, l'élément <Reply><Status><Code> doit alors être l'une des valeurs qui figurent dans le Tableau 10.

Pour les autres mises en œuvre spécifiques au projet, l'élément <Reply><Status><Namespace> ne contient aucun ensemble normatif de valeurs. Il revient au développeur d'application concerné de définir les valeurs pertinentes.

7.6.4.9 Élément <Reply><Status><ObjectID>

L'élément <Reply><Status><ObjectID>, s'il est présent, doit identifier l'objet auquel fait référence la notification (8.6.5).

Un seul élément <Reply><Status> peut contenir plusieurs sous-éléments <ObjectID>. Chaque instance <ObjectID> doit indiquer un objet distinct auquel l'état est associé.

EXEMPLE Dans l'exemple présenté à la Figure 36, les éléments <Reply><Status><ObjectID> indiquent les objets à l'origine de l'échec des demandes. Les notifications d'erreur sont renvoyées uniquement pour les cas considérés comme ayant échoué d'une certaine manière (7.6.4.3).

```
<Reply>
  <Status>
    <Level>ERROR</Level>
    <Code>2.4</Code>
    <Namespace>http://iec.ch/TC57/2013/61968-9#</Namespace>
    <Reason>no such meter</Reason>
    <ObjectID>
      <Names>
        <name>M1003</name>
      </Names>
     <ObjectType>
       <type>Meter</type>
     </ObjectType>
    </ObjectID>
    <ObjectID>
      <Names>
        <name>M1004</name>
      </Names>
      <ObjectType>
        <type>Meter</type>
      </ObjectType>
    </ObjectID>
  </Status>
  <Status>
    <Level>ERROR</Level>
    <Code>2.12</Code>
    <Namespace>http://iec.ch/TC57/2013/61968-9#</Namespace>
    <Reason>invalid usagepoint</Reason>
    <ObjectID>
      <Names>
        <name>UP2</name>
      </Names>
      <ObjectType>
        <type>UsagePoint</type>
      </ObjectType>
    </ObjectID>
  </Status>
</Reply>
```

Figure 36 – Exemple d'indications d'erreur dans un élément <Reply>

7.6.4.10 Élément <Reply><Status><OperationSet>

L'élément <Reply><Status><OperationSet>, s'il est présent, doit identifier l'ensemble d'opérations auquel fait référence la notification (7.7.3.6).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.6.4.11 Élément <Reply><Status><UpdateID>

L'élément <Reply><Status><UpdateID>, s'il est présent, doit identifier la mise à jour d'objet auquel fait référence la notification (7.7.4).

7.6.4.12 Élément <Reply><Status><TransactionID>

L'élément <Reply><Status><TransactionID>, s'il est présent, doit identifier la transaction à laquelle fait référence la notification (7.9).

7.6.5 Combinaison des éléments <Reply><Status><ObjectID>, <OperarionSet>, <UpdateID> et <TransactionID>

Une notification d'état peut contenir les éléments suivants:

- aucun ou plusieurs éléments <ObjectID> (7.6.4.9);
- aucun ou plusieurs éléments <OperationSet> (7.6.4.10);
- aucun ou plusieurs éléments <UpdateID> (7.6.4.11);
- aucun ou plusieurs éléments <TransactionID> (7.6.4.12).

Un élément <Reply><Status> qui ne contient aucun de ces éléments d'identification doit être interprété comme faisant référence à l'ensemble de la demande.

Si un élément <Reply><Status> contient plusieurs identifications qui utilisent le même nom d'élément (deux éléments <ObjectID>, par exemple), celles-ci doivent être interprétées comme faisant référence à l'une des parties identifiées de la demande d'origine ("OR" logique).

Si un élément <Reply><Status> contient plusieurs identifications qui utilisent des noms d'élément différents (un élément <ObjectID> et un élément <TransactionID>, par exemple), celles-ci doivent être interprétées comme faisant référence aux parties de la demande d'origine qui correspondent à tous les critères spécifiés ("AND" logique).

NOTE Ce cas s'apparente à la règle du produit vectoriel (8.4.2).

7.6.6 Élément <Reply><Transaction>

L'élément <Reply><Transaction> est utilisé pour l'envoi d'identificateurs de transaction d'un système répondant à un système demandeur (7.9, 8.5).

7.7 Élément <Payload>

7.7.1 Généralités

L'élément <Payload> a pour objet d'acheminer les données spécifiques à l'application.

7.7.2 Sous-éléments de Payload (charge utile)

Le Tableau 11 décrit les différents sous-éléments qui peuvent être présents dans un élément <Payload> et leur sémantique associée.

Un seul des sous-éléments <ObjectID>, <Transaction>, <OperationSet>, <ObjectUpdate> ou <AbstractPayloadType> peut être présent (7.4.2, 7.4.3, 7.4.4).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Tableau 11 – Sous-éléments de l'élément <Payload>

Élément	Plage de cardinalités		Type de données XSD	Signification
<ObjectID>	1	plusieurs	IdentifierType	Cet élément doit être présent uniquement dans les messages de demande (5.2) ou les messages d'événement (5.4). Cet élément doit être utilisé pour fournir des ID d'objet pour les opérations de récupération ou de suppression qui ne sont pas par ailleurs spécifiées à l'aide d'une charge utile spécifique au type (8.6.5).
<Transaction>	1	plusieurs	TransactionType	Cet élément doit être présent uniquement dans les messages de demande (5.2) ou les messages d'événement (5.4). Cet élément doit être utilisé pour fournir des ID de transaction pour les opérations d'annulation (8.5.3).
<OperationSet>	1	plusieurs	RequestOperationSetType, ResponseOperationSetType ou EventOperationSetType	Voir 7.7.3
<ObjectUpdate>	1	plusieurs	UpdatePayloadType	Voir 7.7.4
< ... >	1	plusieurs	AbstractPayloadType	Voir 7.7.5

7.7.3 Élément <Payload><OperationSet>

7.7.3.1 Généralités

Un OperationSet permet de combiner plusieurs Opérations dans un seul message (7.7.3.6). Cet élément peut s'avérer utile lorsqu'il est nécessaire de vérifier que ces Opérations sont réalisées dans un ordre particulier ou de vérifier l'intégrité transactionnelle entre les différentes Opérations.

La structure précise du sous-élément <Payload><OperationSet> dépend de son utilisation dans le contexte d'un message de demande (7.7.3.3), d'un message de réponse (7.7.3.4) ou d'un message d'événement (7.7.3.5).

EXEMPLE Les Articles D.5 et D.13 contiennent des exemples de message qui explicitent l'utilisation d'OperationSets.

7.7.3.2 Sous-éléments de OperationSet

Le Tableau 12 spécifie les différents sous-éléments qui peuvent être présents dans un élément <Payload><OperationSet> et leur sémantique associée.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Tableau 12 – Sous-éléments de l'élément <Payload><OperationSet>

Élément	Plage de cardinalités		Type de données XSD	Signification
<OperationSetID>	1	1	UUIDType	Cet élément doit être utilisé pour mettre en corrélation <Payload><OperationSetID> avec l'élément <OperationSetID> dans la réponse.
<EnforceMsgSequence>	0	1	xs:boolean	Cet élément peut être présent uniquement dans les messages de demande (5.2) ou d'événement (5.4). Si sa valeur est true, le système de réception doit alors traiter les sous-éléments <Payload><OperationSet><Operation> dans l'ordre strictement croissant des sous-éléments <Payload><OperationSet><Operation><OperationID>. Si cet élément est absent ou que sa valeur est "false", le système répondant est libre d'exécuter les Operations spécifiées dans l'OperationSet dans l'ordre qui lui paraît approprié.
<EnforceTransactionalIntegrity>	0	1	xs:boolean	Cet élément peut être présent uniquement dans les messages de demande (5.2) ou d'événement (5.4). Les Operations d'un OperationSet peuvent également constituer une seule transaction "ACID" (atomique, cohérente, isolée et durable) de sorte qu'elles doivent toutes être réalisées avec succès ou que la défaillance de l'une des Operations doive provoquer leur annulation. La fonctionnalité d'intégrité transactionnelle peut être activée ou désactivée en fonction de la valeur attribuée à cet élément (true ou false respectivement).
<Operation>	1	plusieurs	OperationType	Voir 7.7.3.6

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.7.3.3 Structure des messages de demande

Pour les messages de demande (5.2), le contenu du sous-élément <Payload><OperationSet> est représenté à la Figure 37.

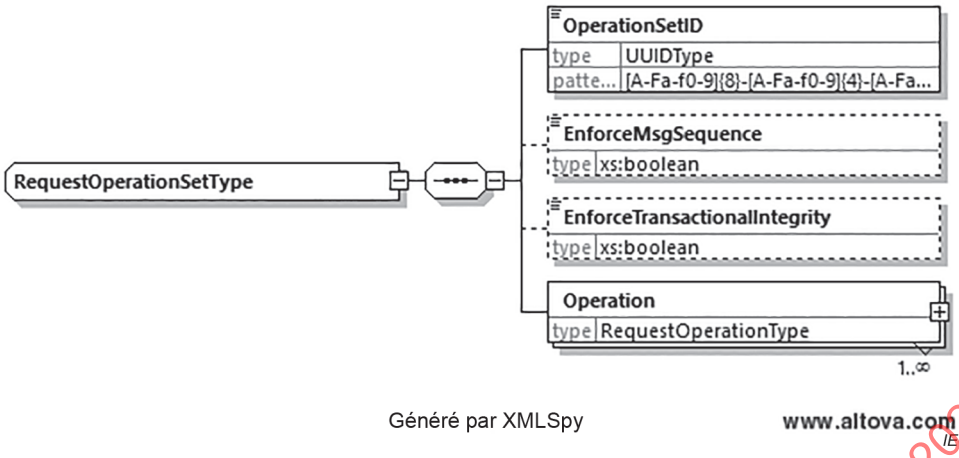


Figure 37 – Structure de l'élément <Payload><OperationSet> pour les messages de demande

7.7.3.4 Structure des messages de réponse

Pour les messages de réponse (5.3), le contenu du sous-élément <Payload><OperationSet> est représenté à la Figure 38.

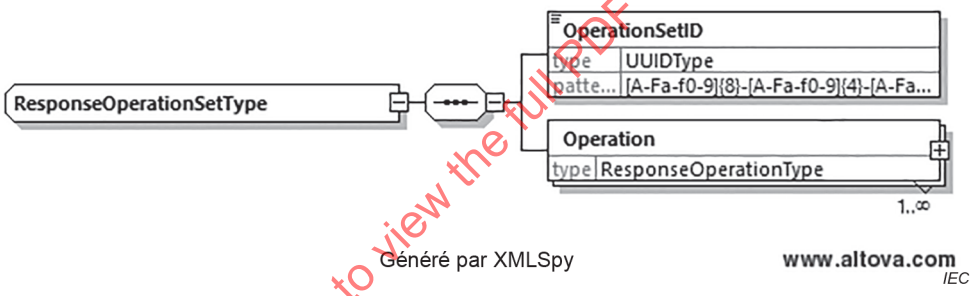


Figure 38 – Structure de l'élément <Payload><OperationSet> pour les messages de réponse

7.7.3.5 Structure des messages d'événement

Pour les messages d'événement (5.4), le contenu du sous-élément <Payload><OperationSet> est représenté à la Figure 39.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

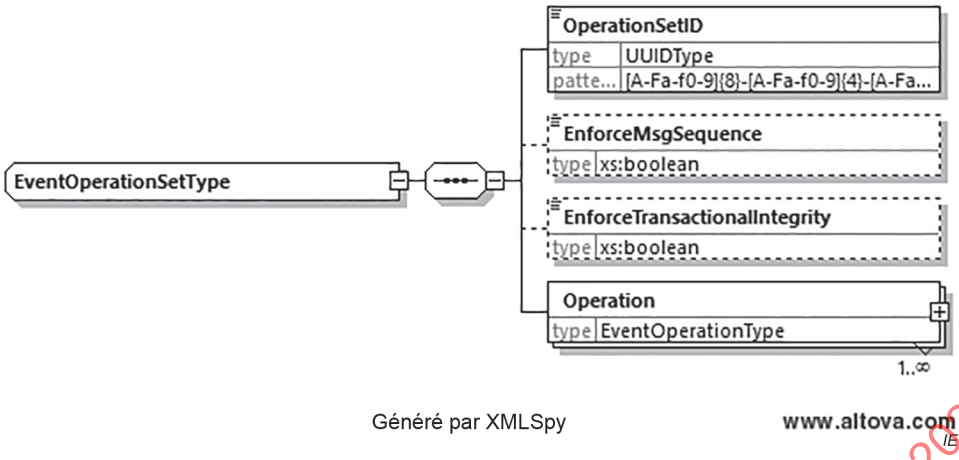


Figure 39 – Structure de l'élément <Payload><OperationSet> pour les messages d'événement

7.7.3.6 Élément <Payload><OperationSet><Operation>

7.7.3.6.1 Généralités

Chaque élément <Payload><OperationSet><Operation> doit contenir des sous-éléments <Verb>, <Noun> et <OperationID> (7.7.3).

7.7.3.6.2 Sous-éléments de Operation (Opération)

Le Tableau 13 décrit les différents sous-éléments qui peuvent être présents dans un élément <Payload><OperationSet><Operation> et leur sémantique associée.

Un seul des sous-éléments <ObjectID>, <Transaction>, <ObjectUpdate> ou <AbstractPayloadType> peut être présent. Le choix du sous-élément dépend de l'élément <Payload><OperationSet><Operation><Verb> (7.7.3.6.6).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Tableau 13 – Sous-éléments de l'élément <Payload><OperationSet><Operation>

Élément	Plage de cardinalités		Type de données XSD	Signification
<OperationID>	1	1	UUIDType	Voir 7.6.4.5.
<Verb>	1	1	VerbType	Les éléments <Verb> et <Noun> doivent spécifier ensemble la sémantique du message et l'interprétation des données de demande fournies dans le sous-élément <Payload><OperationSet><Operation>AbstractPayloadType (7.5.3.4, 7.5.4.4, 7.5.5.4).
<Noun>	1	1	xs:string	
<Comment>	0	1	xs:string	Cet élément peut contenir du texte. Il n'a aucune signification normative.
<Property>	0	plusieurs	MessagePropertyType	Cet élément doit être composé d'un ensemble de paires nom/valeur qui peuvent être utilisées pour personnaliser le comportement du système au-delà de celui défini par la présente Norme internationale. Il peut être utilisé pour assurer certaines fonctionnalités spécifiques à la mise en œuvre. Les valeurs attribuées à ces éléments ne sont pas normatives et ont une signification uniquement dans la mise en œuvre locale. Pour les besoins de l'interopérabilité, il convient d'éviter d'utiliser l'élément <Property> dans toute la mesure du possible.
<Name>	1	1	xs:string	Nom de la propriété
<Value>	0	1	xs:string	Valeur associée de la propriété
<ObjectID>	1	plusieurs	IdentifierType	Utilisé pour fournir les ID d'objet pour les opérations d'annulation ou les ID d'objet pour les opérations de récupération ou de suppression qui ne sont pas par ailleurs spécifiées à l'aide d'une charge utile spécifique au type (8.6.5).
<Transaction>	1	plusieurs	TransactionType	Cet élément doit être présent uniquement dans les messages de demande (5.2) ou les messages d'événement (5.4). Cet élément doit être utilisé pour fournir des ID de transaction pour les opérations d'annulation (8.5.3).
<ObjectUpdate>	1	plusieurs	UpdatePayloadType	Cet élément doit être présent uniquement dans les messages de demande (5.2) ou les messages d'événement (5.4). Voir 7.7.4
< ... >	1	plusieurs	AbstractPayloadType	L'élément <Operation> peut contenir des données spécifiques à l'application. Ces données dépendent de la sémantique précise de la demande et sont laissées à l'appréciation du groupe de travail concerné ou autre développeur d'application à définir (7.7.5).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.7.3.6.3 Structure des messages de demande

Pour les messages de demande (5.2), le contenu du sous-élément <Payload><OperationSet> <Operation> est représenté à la Figure 40.

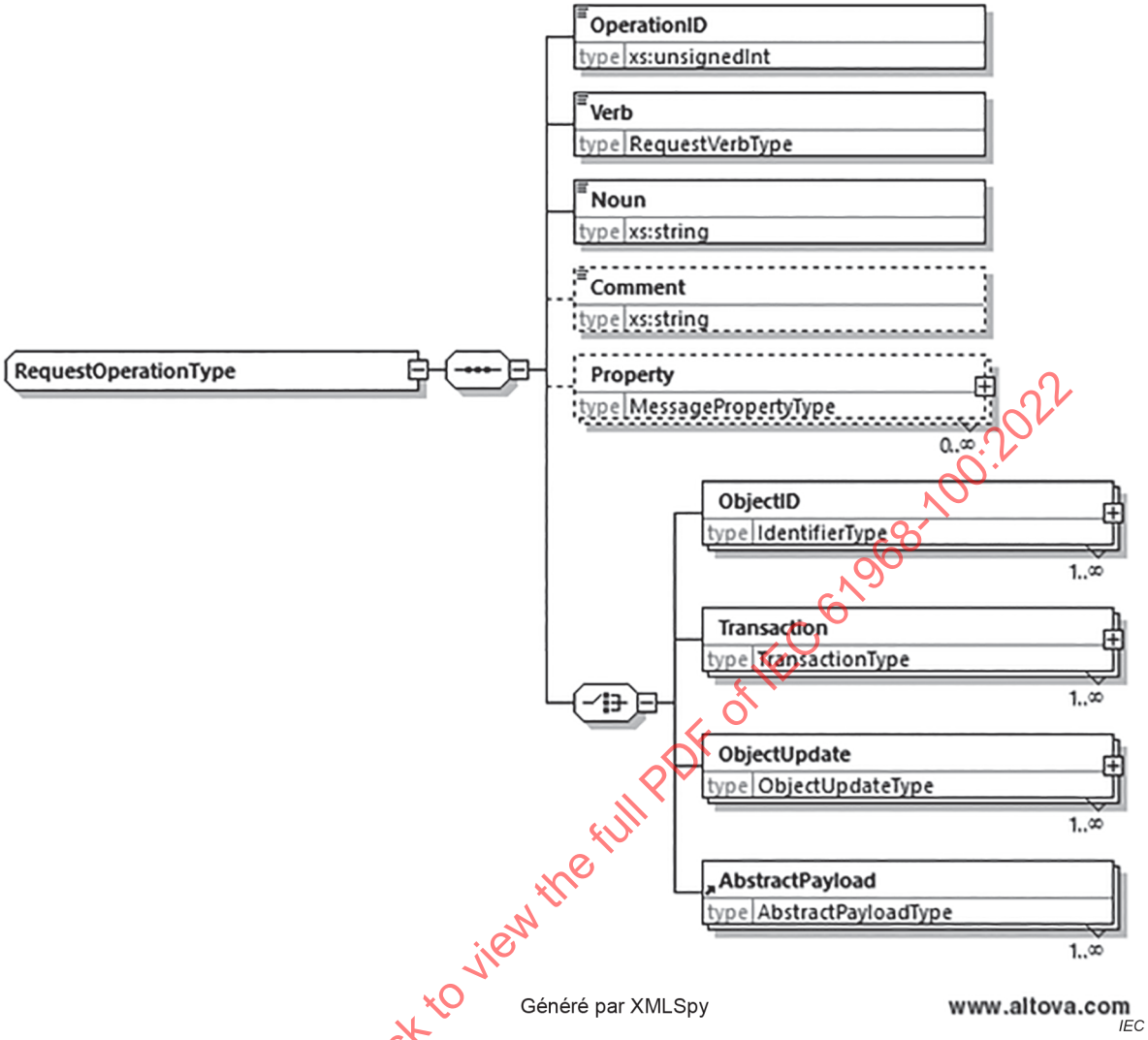


Figure 40 – Structure de l'élément `<Payload><OperationSet><Operation>` pour les messages de demande

7.7.3.6.4 Structure des messages de réponse

Pour les messages de réponse (5.3), le contenu du sous-élément `<Payload><OperationSet><Operation>` est représenté à la Figure 41.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

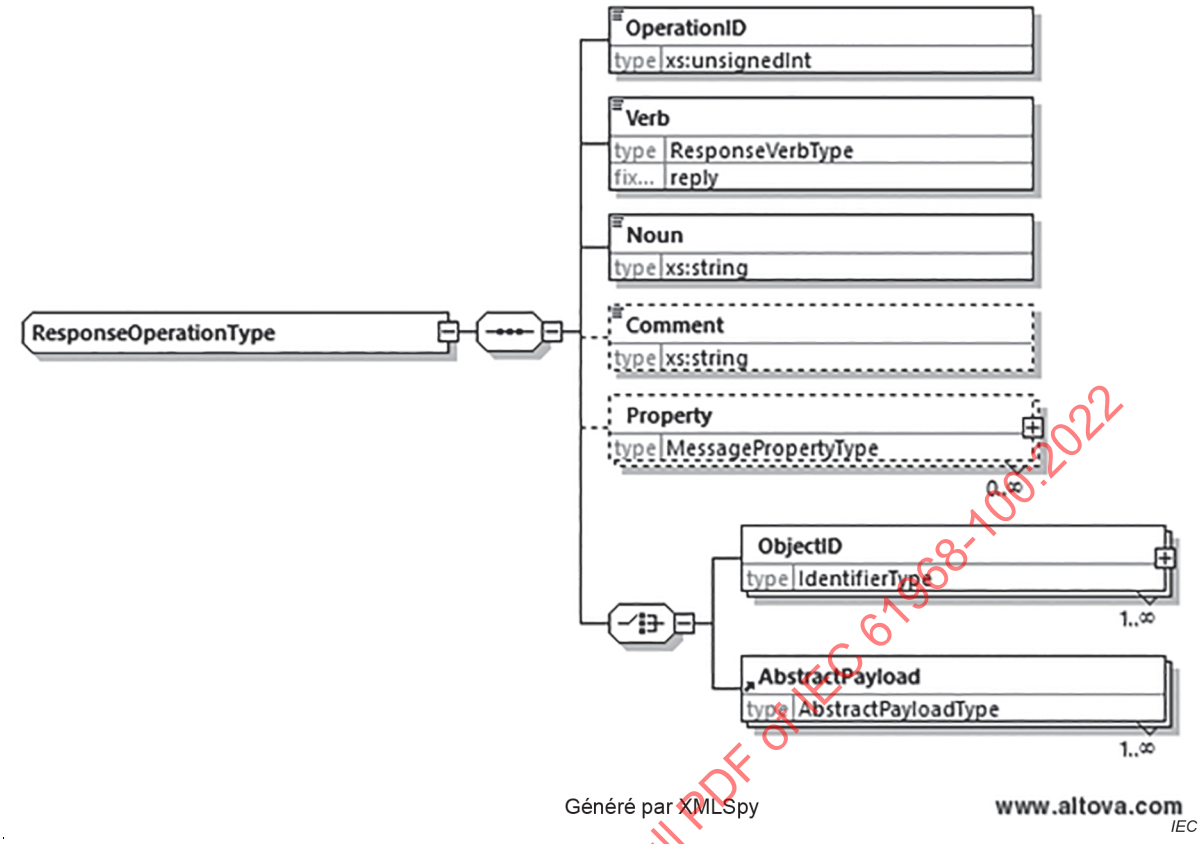


Figure 41 – Structure de l'élément `<Payload><OperationSet><Operation>` pour les messages de réponse

7.7.3.6.5 Structure des messages d'événement

Pour les messages d'événement (5.4), le contenu du sous-élément `<Payload><OperationSet><Operation>` est représenté à la Figure 42.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

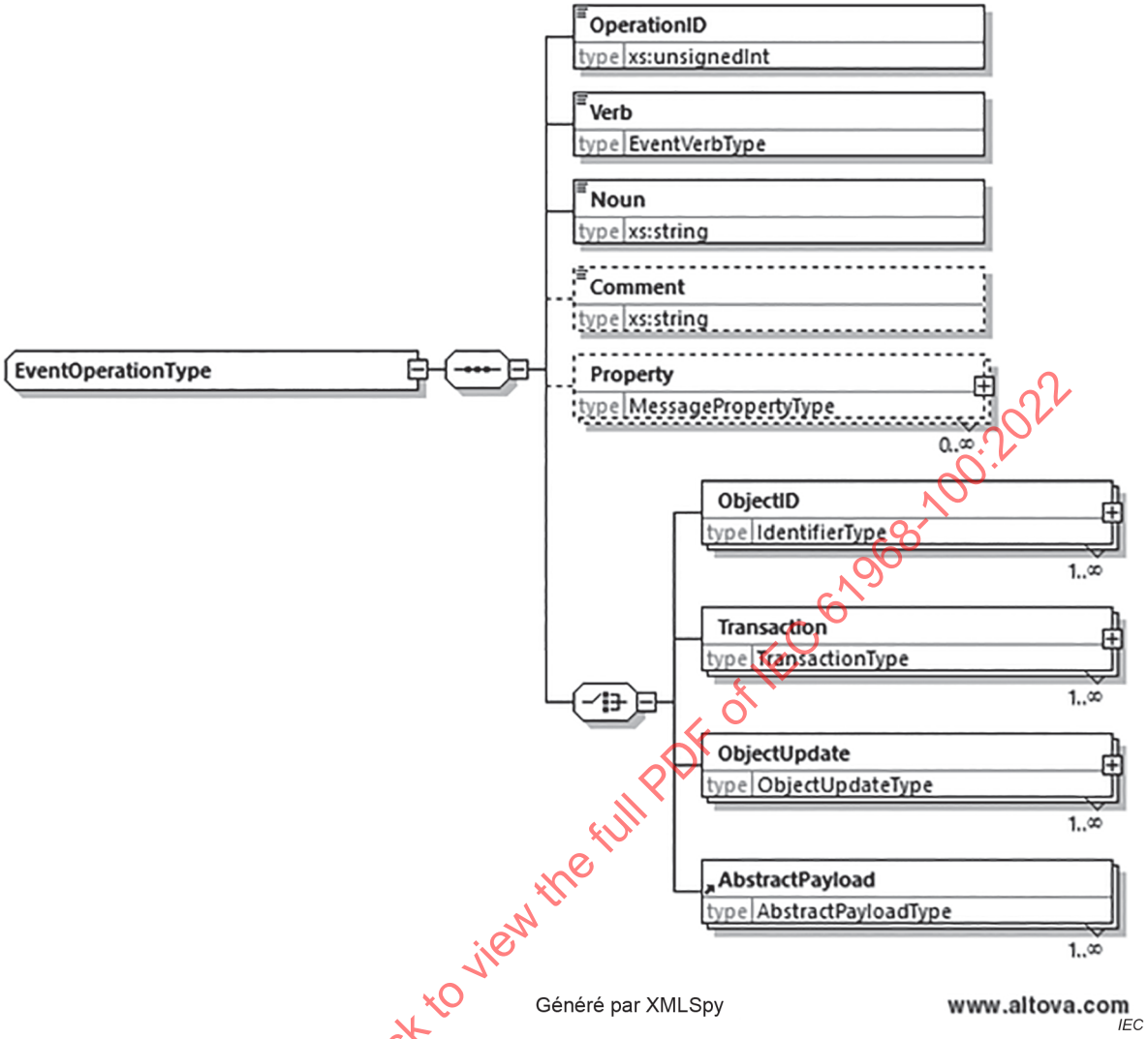


Figure 42 – Structure de l'élément <Payload><OperationSet><Operation> pour les messages d'événement

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

7.7.3.6.6 Élément <Payload><OperationSet><Operation><Verb>

Pour un message de demande (5.2) dans lequel <Header><Verb> est 'execute', les verbes des sous-éléments <Payload><OperationSet><Operation><Verb> doivent être ceux du Tableau 5.

Pour un message de réponse (5.3) dans lequel <Header><Verb> est ''execute'', les verbes des sous-éléments <Payload><OperationSet><Operation><Verb> doivent être ''reply''.

Pour un message d'événement (5.4) dans lequel <Header><Verb> est ''execute'', les verbes des sous-éléments <Payload><OperationSet><Operation><Verb> doivent être ceux du Tableau 6.

7.7.4 Élément <Payload><ObjectUpdate>

7.7.4.1 Généralités

L'élément <Payload><ObjectUpdate> doit être utilisé pour apporter des modifications partielles (ajouts, modifications ou remplacements) en tout ou partie à un objet existant (8.3.5).

Chaque élément <Payload><ObjectUpdate> doit contenir des sous-éléments <UpdateID>, <UpdateAction> et <AbstractPayload>

EXEMPLE L'Annexe E contient des exemples de messages qui présentent l'utilisation du sous-élément <Payload><ObjectUpdate>.

7.7.4.2 Structure

Le contenu du sous-élément <Payload><ObjectUpdate> est décrit à la Figure 43.

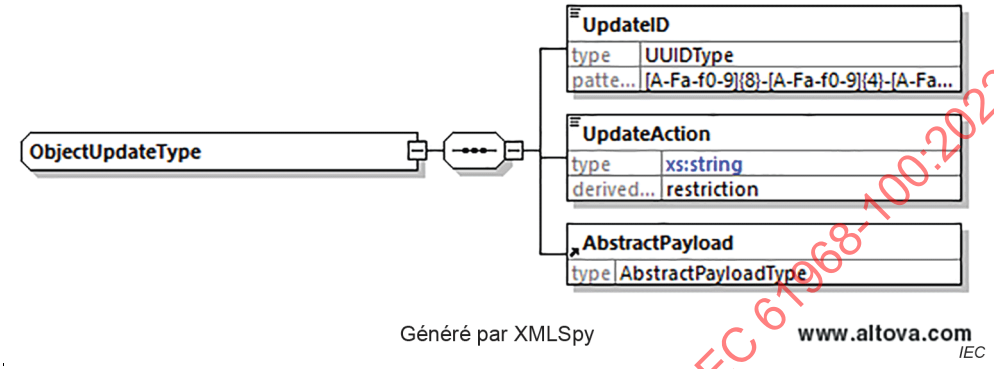


Figure 43 – Structure de l'élément <Payload><ObjectUpdate>

7.7.4.3 Sous-éléments UpdatePayloadType

Le Tableau 14 décrit les différents sous-éléments qui doivent être présents dans un élément <Payload> <ObjectUpdate> et leur sémantique associée.

Tableau 14 – Sous-éléments de l'élément <Payload><ObjectUpdate>

Élément	Plage de cardinalités		Type de données XSD	Signification
<UpdateID>	1	1	xs:unsignedInt	Voir 7.6.4.5
<UpdateAction>	1	1	UpdateActionType	Voir 7.7.4.4
< ... >	1	1	AbstractPayloadType	Cet élément peut contenir des données spécifiques à l'application. Ces données dépendent de la sémantique précise de la demande et sont laissées à l'appréciation du groupe de travail concerné ou autre développeur d'application à définir. En d'autres termes, ces données n'ont aucune signification normative dans le contexte de la présente Norme internationale. Toutefois, elles peuvent comporter une sémantique normative en ce qui concerne d'autres normes. Voir 7.7.5

7.7.4.4 UpdateAction

La valeur de l'élément <Payload><ObjectUpdate><UpdateAction> doit être l'une des valeurs qui figurent dans le Tableau 15.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Tableau 15 – Sous-éléments de l'élément <Payload><ObjectUpdate><UpdateAction>

UpdateAction	Signification	Article/paragraphe
add	Spécifie une demande d'ajout d'un ou de plusieurs éléments à un objet existant.	8.3.5.2
modify	Spécifie une demande de remplacement d'un ou de plusieurs éléments à l'intérieur d'un objet existant.	8.3.5.3
remove	Spécifie une demande de suppression d'un ou de plusieurs éléments d'un objet existant.	8.3.5.4

7.7.5 Données AbstractPayloadType

Un message IEC 61968-100 peut acheminer des données spécifiques à l'application (8.2).

Si les données spécifiques à l'application sont présentes, elles doivent être fournies dans les sections AbstractPayloadType de l'élément <Payload> (7.7), de l'élément <Payload><OperationSet><Operation> (7.7.3.6) ou de l'élément <Payload><OperationUpdate> (7.7.4). En d'autres termes, le sous-élément AbstractPayLoadType est remplacé par les données spécifiques à l'application (8.2.2).

La présente Norme internationale ne prescrit pas ces données spécifiques à l'application, car elles sont en général définies de manière normative dans d'autres parties de l'IEC 61968. Toutefois, cela ne dispense pas d'utiliser l'IEC 61968-100 à d'autres fins de messagerie non liées à l'IEC 61968.

L'Article 8 définit les conventions normatives auxquelles ces données spécifiques à l'application doivent satisfaire.

7.7.6 Charges utiles abstraites multiples

Un seul message peut contenir plusieurs éléments <AbstractPayload> (7.7.5). Dans un tel cas, les données spécifiques à l'application contenues dans les éléments <AbstractPayload> doivent avoir le même nom d'élément au niveau le plus élevé. Toutefois, cela ne peut pas être assuré par une validation XSD seule.

EXEMPLE L'Article G.5 donne un exemple de message qui contient plusieurs éléments AbstractPayloadType.

7.8 Élément <CompressedPayload>

7.8.1 Généralités

L'élément <CompressedPayload> permet de compresser une charge utile avant de l'acheminer dans un message IEC 61968-100.

Les éléments <Payload> et <CompressedPayload> sont mutuellement exclusifs.

L'élément <CompressedPayload>, s'il est présent, doit contenir les mêmes informations que l'élément <Payload> équivalent. Ces données doivent être compressées, puis codées en représentation base64 selon la RFC 4648.

7.8.2 Structure

La structure de l'élément <CompressedPayload> est représentée à la Figure 44.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

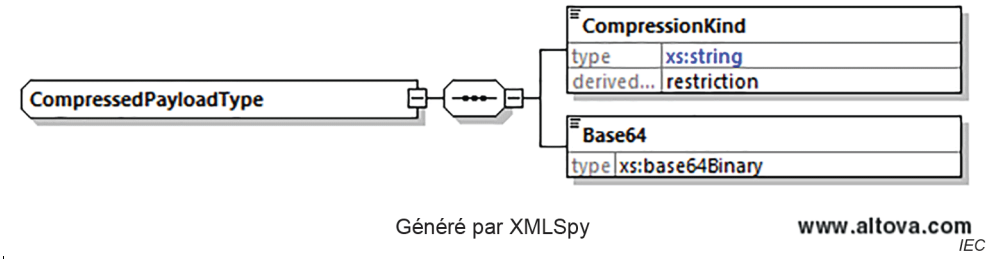


Figure 44 – Structure de l'élément <CompressedPayload>

7.8.3 Éléments CompressedPayload

Le Tableau 16 spécifie les différents sous-éléments qui doivent être présents dans un élément <CompressedPayload>.

Tableau 16 – Sous-éléments de l'élément <CompressedPayload>

Élément	Plage de cardinalités		Type de données XSD	Signification
<CompressionType>	1	1	énumération	Voir 7.8.4
<Base64>	1	1	xs:base64Binary	Voir 7.8.5

7.8.4 Sous-élément <CompressedPayload><CompressionType>

Le sous-élément <CompressedPayload><CompressionType> doit spécifier la méthode de compression qui a été appliquée aux données de charge utile. L'une des valeurs du Tableau 17 doit être attribuée à cet élément.

Tableau 17 – Valeurs de l'élément <CompressedPayload><CompressionType>

Valeur de l'élément <CompressionType>	Signification
zip	Compression ZIP telle que définie dans l'IEC 21320-1:2015
gzip	Compression GZIP telle que définie par la RFC 1952
other	Autre mécanisme de compression

7.8.5 Sous-élément <CompressedPayload><Base64>

Le sous-élément <CompressedPayload><Base64> doit contenir la représentation base64 des données compressées.

7.9 Type de transaction

Le type de transaction est utilisé dans le cadre des transactions d'annulation (8.5).

Une instance d'un type de transaction peut se produire dans le cadre d'un élément <CancelRequestMessage><Payload><Transaction> (7.4.2.2), d'un élément <Payload><OperationSet><Operation><Transaction> (7.7.3.6.2) ou d'un élément <Reply><Status> (7.6.4.12).

La structure du type de transaction est décrite à la Figure 45.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

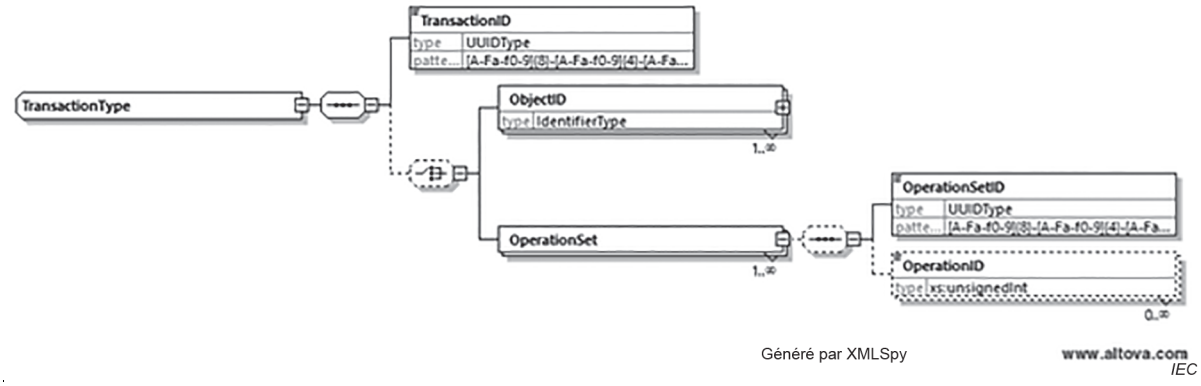


Figure 45 – Structure de TransactionType

Le Tableau 18 décrit les éléments d'un TransactionType.

Tableau 18 – Éléments d'un type de transaction

Élément	Plage de cardinalités		Type de données XSD	Signification
<TransactionID>	1	1	UUIDType	ID transaction
<ObjectID>	0	plusieurs	IdentifierType	Cet élément doit être utilisé pour identifier les objets auxquels la transaction fait référence (8.6.5).
<OperationSet>	0	plusieurs		Cet élément doit être utilisé pour identifier les OperationSets auxquels la transaction fait référence (7.7.3).
<OperationSetID>	1	1	UUIDType	Voir 7.6.4.5
<OperationID>	0	plusieurs	xs:unsignedInt	Cet élément doit être utilisé pour identifier le sous-ensemble des Operations dans un OperationSet auquel la transaction fait référence (7.7.3).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8 Contraintes et conventions

8.1 Généralités

Le présent article détaille les contraintes et conventions qui visent à faciliter la cohérence et l'interopérabilité entre toutes les parties et applications qui utilisent le présent document.

8.2 Données spécifiques à l'application

8.2.1 Généralités

Le présent document ne prescrit pas le type des données spécifiques à l'application transmises dans un message (7.7.5).

Le contenu des sections `AbstractPayloadType` est généralement défini selon un profil XSD spécifique à l'application qui a été spécifié par le groupe de travail ou l'organisme local concerné en fonction de l'application en cours (4.3).

Les informations détaillées ces profils spécifiques à l'application ne relèvent pas du domaine d'application du présent document. Toutefois, il convient de satisfaire aux exigences formulées dans le protocole d'accord entre l'IEC et l'UN/CEFACT [1] [1].

Un espace de nom XML doit être attribué à cet effet à ce type de profil XSD. Le choix de l'espace de nom XML n'est pas normatif et ne relève pas du domaine d'application de la présente Norme internationale.

8.2.2 AbstractPayloadType

La sémantique des données spécifiques à l'application n'est pas normative en ce qui concerne le présent document. Toutefois, ces données peuvent comporter une sémantique normative en ce qui concerne d'autres normes de l'application ou du domaine.

Le profil spécifique à l'application doit hériter du type de données abstraites XSD `AbstractPayloadType` défini dans le fichier `AbstractPayloadType.xsd` (voir l'Article A.3). Cet `AbstractPayloadType` est défini comme une directive XSD `<xs:substitutionGroup>`. Cela implique qu'il convient de remplacer les sections `AbstractPayloadType` par les données spécifiques à l'application correspondantes.

Un profil spécifique à l'application peut être défini par l'intégration du fichier `AbstractPayloadType.xsd` à l'aide d'une directive `<xs:import>`. L'héritage XSD doit être réalisé au moyen des directives `<xs:extension>` et `<xs:substitutionGroup>`.

EXEMPLE Des exemples de profils XSD qui présentent l'utilisation d'AbstractPayloadType sont donnés à l'Annexe C. Des exemples de messages correspondant aux messages XML peuvent être consultés à l'Annexe D et à l'Annexe E.

8.2.3 Personnalisation spécifique à l'application

Pour assurer la personnalisation locale, un organisme peut définir ses propres profils XSD spécifiques à l'application. Ces personnalisations sont spécifiques à l'organisme local et n'ont aucune signification normative.

Si un profil spécifique à l'organisme est exigé, un espace de nom XML spécifique à l'organisme doit lui être attribué afin de souligner qu'une extension non normalisée est utilisée.

Un organisme peut également utiliser les éléments `<Header><Revision>`, `<Header><Source>`, `<Header><User>` et `<Header><Property>` pour procéder à d'autres personnalisations.

8.2.4 Données RDF

Le fichier XSD normatif indiqué à l'Article A.5 peut être utilisé dans le contexte d'un AbstractPayloadType pour transporter une charge utile RDF conformément à l'IEC 61970-552:2016.

8.3 Demandes et notifications de gestion d'objets

8.3.1 Généralités

Une demande de gestion d'objets doit avoir lieu dans le contexte d'un modèle d'échange de messages de demande/réponse (6.2).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

1 Les chiffres entre crochets renvoient à la Bibliographie.

Un message de demande avec le verbe `create`, `change`, `delete` ou `update` envoyé d'un système demandeur à un système répondant doit être interprété comme une demande adressée au système répondant d'instancier, de remplacer, de supprimer ou de modifier, respectivement, l'objet spécifique à l'application correspondant (8.3.2, 8.3.3, 8.3.4, 8.3.5).

Le présent article décrit les cas aboutis pour les différentes demandes de gestion d'objets. Pour les cas non aboutis, voir 7.6.4.3.

Le présent article ne s'applique pas à la création, la modification, la suppression ou la mise à jour des identificateurs (éléments `<mRID>` ou `<Names>`) d'un objet spécifique à l'application (8.7).

8.3.2 Message de demande de création

8.3.2.1 Création d'un message de demande de création

En l'absence d'erreur, un message de demande de création doit permettre d'instancier un ou plusieurs objets spécifiques à l'application sur le système répondant.

Un message de demande de création doit être indiqué par un `<Header><Verb>` dont la valeur est `create` (7.5.3.3).

Les informations connexes nécessaires à la création du ou des nouveaux objets doivent être contenues dans l'élément `<Payload>` du message de demande de création (7.7).

EXEMPLE Le paragraphe D.2.3 donne un exemple de message de de demande de création.

8.3.2.2 Dénomination d'objet

Un ou plusieurs identificateurs doivent être donnés à l'objet qui vient d'être instancié afin de pouvoir être référencé dans les demandes suivantes.

L'objet qui vient d'être instancié doit être identifié par spécification des champs `<mRID>` ou `<Name>` correspondants dans l'élément `<Payload>` spécifique à l'application fourni (7.7, 8.6).

Un objet peut être connu sous plusieurs noms.

8.3.2.3 Instanciation réussie d'un objet

Si la demande de création d'un objet particulier a abouti et que l'objet qui vient d'être instancié ne contient aucune information autre que celle de la demande, le système répondant peut renvoyer une copie identique de l'objet dans l'élément `<Payload>` d'un message de réponse (7.7).

Si la demande de création d'un objet particulier a abouti sur le système répondant, l'objet qui vient d'être instancié peut être complété par des informations autres que celles contenues dans la demande. Par exemple, le nouvel objet peut contenir des identificateurs d'objet supplémentaires, des noms ou des informations supplémentaires. Dans un tel cas, le système répondant peut renvoyer une copie de l'objet amélioré dans l'élément `<Payload>` d'un message de réponse (7.7).

Le système répondant peut, à son appréciation, choisir de ne pas renvoyer une copie de l'objet qui vient d'être instancié. Par exemple, le système répondant peut ne pas disposer de toutes les informations relatives à l'objet, et l'assemblage d'une copie complète de l'objet par l'interrogation d'autres systèmes peut s'avérer extrêmement coûteux en matière de calculs informatiques.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8.3.3 Message de demande de modification

8.3.3.1 Création d'un message de demande de modification

Une demande de modification doit être utilisée pour remplacer un ou plusieurs objets spécifiques à l'application existants sur le système répondant. Les objets spécifiés sont remplacés, dans leur ensemble, par les nouvelles données.

Un message de demande de modification doit être indiqué par un `<Header><Verb>` dont la valeur est `change` (7.5.3.3).

Les objets pour lesquels la demande de modification est formulée doivent être identifiés par spécification des champs `<mRID>` ou `<Name>` correspondants dans l'élément `<Payload>` spécifique à l'application fourni (7.7, 8.6).

Les données spécifiques à l'application destinées à remplacer le ou les objets existants doivent être placées dans l'élément `<Payload>` du message de demande de modification (7.7).

8.3.3.2 Remplacement réussi d'un objet

Si la demande de remplacement d'un objet particulier a abouti, l'objet remplacé peut être associé à des informations autres que celles contenues dans la demande. Par exemple, l'objet remplacé peut contenir des identificateurs d'objet supplémentaires, des noms ou d'autres informations supplémentaires. Dans un tel cas, le système répondant peut renvoyer une copie de l'objet remplacé dans l'élément `<Payload>` d'un message de réponse (7.7).

Le système répondant peut, à son appréciation, choisir de ne pas renvoyer une copie de l'objet remplacé. Par exemple, la génération d'une copie complète de l'objet remplacé peut être extrêmement coûteuse pour le système répondant en matière de calculs informatiques.

8.3.4 Message de demande de suppression

8.3.4.1 Généralités

Une demande de suppression doit être utilisée pour supprimer du système répondant un ou plusieurs objets spécifiques à l'application dans leur ensemble. Un message de demande de suppression doit être indiqué par un `<Header><Verb>` dont la valeur est `delete` (7.5.3.3).

8.3.4.2 Suppression réussie d'un objet

Il convient que l'élément `<Payload>` du message de demande de suppression ne contienne aucune valeur autre que les champs `<mRID>` ou `<Name>`. Le système répondant doit ignorer tous les autres champs.

Les objets qui doivent être supprimés dans leur ensemble doivent être identifiés selon l'une des méthodes suivantes:

- Soit au moyen des éléments `<Payload><ObjectID><mRID>` ou `<Payload><ObjectID><Name>` pour spécifier les objets concernés (8.6.5);
- Soit par spécification des champs `<mRID>` ou `<Name>` correspondants dans l'élément `<Payload>` spécifique à l'application fourni (7.7, 8.6).

EXEMPLE Le paragraphe D.14.3 donne un exemple de message de demande de suppression.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8.3.5 Message de demande de mise à jour

8.3.5.1 Généralités

Un message de demande de mise à jour doit être utilisé pour apporter des modifications à un ou plusieurs objets spécifiques à l'application existants sur le système répondant sans remplacer cet objet dans son ensemble.

La demande de ce type de mise à jour d'objet est contenue dans un élément `<Payload><ObjectUpdate>`.

Les objets pour lesquels la demande de mise à jour est formulée doivent être identifiés par spécification des champs `<mRID>` ou `<Name>` correspondants dans l'élément `<Payload><ObjectUpdate><AbstractPayload>` spécifique à l'application fourni (7.7, 8.6).

La demande à réaliser est déterminée en fonction de la valeur de l'élément `<Payload><ObjectUpdate><UpdateAction>` (7.7.4.4, 8.3.5.2, 8.3.5.3, 8.3.5.4).

8.3.5.2 Ajout d'éléments à un objet existant

Si la valeur de l'élément `<Payload><ObjectUpdate><UpdateAction>` (7.7.4.4) est `add`, tous les éléments spécifiques à l'application (autres que les champs `<mRID>` ou `<Name>` utilisés pour identifier cet objet) contenus dans l'élément `<Payload><ObjectUpdate><AbstractPayload>` doivent être ajoutés à l'objet spécifié sur le système répondant.

EXEMPLE Des exemples d'ajout d'éléments à un objet peuvent être consultés aux Articles E.5, E.10, E.11, E.17 et E.20.

8.3.5.3 Modification d'éléments dans un objet existant

Si la valeur de l'élément `<Payload><ObjectUpdate><UpdateAction>` (7.7.4.4) est `modify`, tous les éléments spécifiques à l'application (autres que les champs `<mRID>` ou `<Name>` utilisés pour identifier cet objet) contenus dans l'élément `<Payload><ObjectUpdate><AbstractPayload>` doivent alors être utilisés pour remplacer ceux appartenant à l'objet spécifié sur le système répondant.

EXEMPLE Des exemples de modification d'éléments dans un objet existant peuvent être consultés aux Articles E.3, E.6, E.7, E.12, E.15 et E.18.

8.3.5.4 Suppression d'éléments d'un objet existant

Si la valeur de l'élément `<Payload><ObjectUpdate><UpdateAction>` (7.7.4.4) est `remove`, tous les éléments spécifiques à l'application (autres que les champs `<mRID>` ou `<Name>` utilisés pour identifier cet objet) contenus dans l'élément `<Payload><ObjectUpdate><AbstractPayload>` doivent alors être supprimés de l'objet spécifié sur le système répondant.

Dans certains cas, il peut s'avérer nécessaire d'utiliser un élément autre que `<mRID>` ou `<Name>` pour identifier les éléments à supprimer. Un exemple de ce type de cas est donné à l'Article E.13.

Toutes les valeurs fournies pour un élément en cours de suppression doivent être ignorées.

NOTE Il peut s'avérer nécessaire de fournir une valeur fictive pour un élément en cours de suppression pour que la charge utile procède à la validation en fonction du schéma XSD.

EXEMPLE Des exemples de suppression d'éléments d'un objet existant peuvent être consultés aux Articles E.4, E.8, E.9, E.13, E.14, E.19 et E.21.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8.3.6 Notifications de gestion d'objet

Une notification de gestion d'objet doit être formulée comme un message d'événement (5.4). Ce type de notification doit avoir lieu dans le contexte du modèle d'échange de messages d'événement non sollicité (6.3).

Un message d'événement avec le verbe `created`, `changed`, `deleted` ou `updated` envoyé par un système d'envoi à un système de réception doit être interprété comme une notification selon laquelle l'objet spécifié sur le système d'envoi a été instancié, remplacé, supprimé ou modifié, respectivement.

EXEMPLE Un exemple de notification de gestion d'objet peut être consulté à l'Article D.3.

8.4 Demandes de récupération

8.4.1 Formuler une demande de récupération

Un message de demande dont `<Header><Verb>` a la valeur `get` doit être interprété comme une requête de récupération d'un ou de plusieurs objets spécifiques à l'application dont le type est spécifié par le nom de la demande.

Les paramètres de requête spécifique à l'application (4.3) utilisés pour spécifier les objets correspondants à renvoyer doivent être identifiés selon l'une des méthodes suivantes:

- Soit au moyen des éléments `<Payload><ObjectID><mRID>` ou `<Payload><ObjectID><Names>` (8.6.5);
- Soit par spécification des champs de sélection appropriés dans un profil de requête spécifique à l'application. Les paramètres de requête sont alors spécifiés dans l'élément `<AbstractPayload>`.

En règle générale, il convient qu'une demande d'extraction soit idempotente. En d'autres termes, il convient qu'elle ne modifie pas l'état des objets demandés. Toutefois, une demande d'extraction peut donner lieu à la création de nouvelles données dans le système répondant. Par exemple, une requête IEC 61968-9:2013 get(MeterReadings) sur demande peut amener les données à être lues sur le réseau AMI dans l'un des nouveaux objets MeterReading.

EXEMPLE Les paragraphes D.7.3, D.8.3, D.9.3, D.11.2 et D.13.2 donnent des exemples de messages de demande de récupération.

8.4.2 Spécification de plusieurs contraintes dans un message de demande

8.4.2.1 Généralités

Un fichier XSD donné qui définit un profil de requête spécifique à l'application peut permettre à un message de demande de spécifier plusieurs contraintes de paramètre de sélection (7.7). Le présent article définit le mode selon lequel ces contraintes de paramètre peuvent être définies afin d'obtenir les opérations "AND" et "OR" logiques souhaitées, décrites de 8.4.2.2 à 8.4.2.5.

Les règles énoncées de 8.4.2.2 à 8.4.2.4 sont ensemble appelées règle du produit vectoriel.

8.4.2.2 "OR" logique en fonction des critères spécifiés

Si une demande spécifie plusieurs contraintes de paramètre de sélection appartenant à la même classe de critères (c'est-à-dire qu'elles partagent le même nom d'élément XML au niveau le plus élevé), elles doivent être interprétées comme une opération "OR" logique.

EXEMPLE L'Article G.2 contient un exemple de spécification d'opérations "OR" logiques en fonction des critères spécifiés.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8.4.2.3 "AND" logique en fonction des critères spécifiés

Lorsque plusieurs contraintes de paramètre de sélection de niveau supérieur sont spécifiées dans une demande et qu'elles se rapportent toutes à différentes classes de critères (c'est-à-dire qu'elles ne partagent pas le même nom d'élément XML au niveau le plus élevé), elles doivent alors être interprétées comme une opération "AND" logique.

EXEMPLE L'Article G.3 contient un exemple de spécification d'opérations "AND" logiques en fonction des critères spécifiés.

8.4.2.4 Contraintes multiples

Si une demande spécifie des combinaisons de différents critères de paramètre de sélection, cela doit être interprété comme une demande de renvoi de données pour toutes les combinaisons des critères spécifiés.

EXEMPLE L'Article G.4 contient un exemple de spécification d'une combinaison des opérations "AND" et "OR" logiques en fonction des critères spécifiés.

8.4.2.5 Spécification de plusieurs profils spécifiques à l'application dans un seul message

Si un message de demande de récupération contient plusieurs profils spécifiques à l'application (7.7.6), cela offre un moyen équivalent de spécifier plusieurs critères de sélection liés par une opération "OR" logique.

EXEMPLE L'Article G.5 contient un exemple de spécification de plusieurs profils AbstractPayloadType dans un seul message.

8.4.3 Réponse à une demande de récupération

Après une tentative de réalisation de la demande de récupération, le système répondant doit répondre par un ou plusieurs messages de réponse qui contiennent les données demandées accompagnées des conditions informationnelles ou d'indications d'erreur (5.3.5, 6.2.2, 7.7).

8.5 Annulation de transactions

8.5.1 Généralités

Un message de demande d'annulation envoyé par un système demandeur à un système répondant doit être interprété comme une demande adressée à ce dernier de mettre fin aux transactions spécifiées.

NOTE 1 Dans ce contexte, une "transaction" fait référence à une seule opération indivisible.

NOTE 2 Les annulations posent un problème par nature, car il peut être impossible pour un système de réception d'annuler une transaction. Si l'annulation concerne une demande dont l'action doit avoir lieu ultérieurement ou qui définit une opération idempotente, il n'y a alors généralement aucun problème. Toutefois, si la demande d'annulation concerne une transaction qui a déjà commencé ou qui a certains effets externes (un message IEC 61968-9:2013 create(EndDeviceControls), par exemple), il peut ne pas toujours être possible d'annuler toutes les transactions demandées. Des problèmes de synchronisation du réseau peuvent également compliquer l'ensemble, car la transaction d'origine peut être déjà en cours lors de la réception de la demande d'annulation.

8.5.2 Modèle d'échange de message

Un modèle d'échange de message dans lequel une demande est en partie ou entièrement annulée est représenté à la Figure 46. Ici, il est pris pour hypothèse que la demande sous-jacente fait partie intégrante d'un modèle d'échange de message d'événement induit (6.4).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

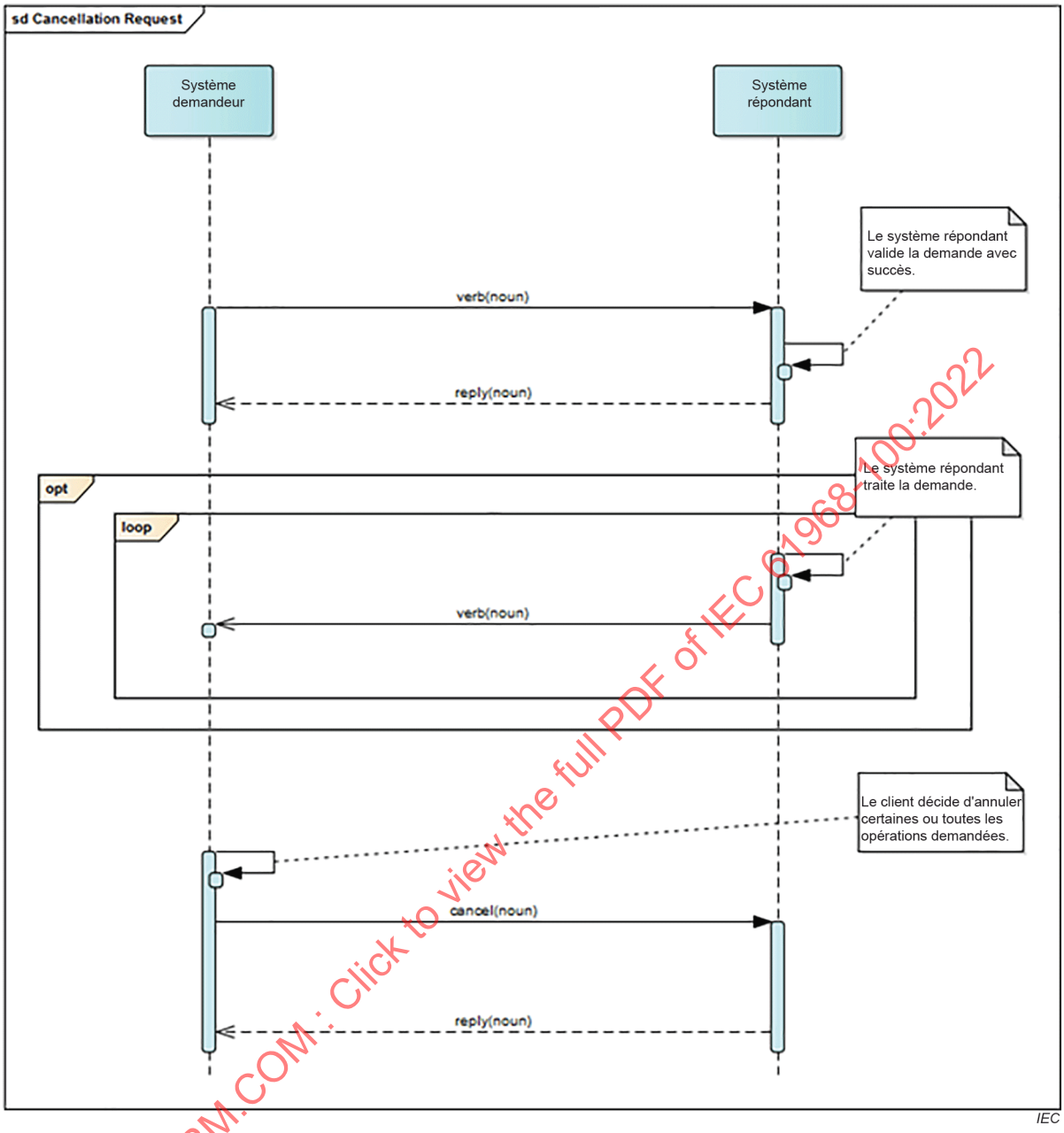


Figure 46 – Modèle d'échange de message pour une demande d'annulation

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8.5.3 ID transaction

Une transaction disponible pour l'annulation est identifiée par un ID transaction. Un ID transaction doit être spécifié en tant qu'identifiant unique universel (UUID) de 128 bits généré selon la RFC 4122.

Lorsqu'un système répondant reçoit une demande qu'il estime pouvoir être annulée, il doit générer un ID transaction et le renvoyer au système demandeur. Pour demander l'annulation suivante de la transaction d'origine, le système demandeur doit fournir cet ID transaction dans le message de demande d'annulation suivant.

Un système répondant peut, à son appréciation, renvoyer plusieurs ID transaction pour une demande donnée. Cela permet une granularité plus fine pour les parties d'annulation de la demande.

EXEMPLE L'Article D.16 donne un exemple de l'utilisation de plusieurs ID transaction.

Toutes les opérations ne peuvent pas être annulées. Le système répondant a la possibilité de décider pour quelles demandes et à quel niveau de granularité il fournit les ID transaction.

8.5.4 Réception d'un ID transaction

Un message de réponse peut contenir aucun ou plusieurs ID transaction. Ce type d'ID transaction doit être placé dans un élément `<Reply><Transaction><TransactionID>` (8.6.5).

Chaque ID transaction doit être spécifié avec un élément `<Reply><Transaction><ObjectID><mRID>` ou un ou plusieurs éléments `<Reply><Transaction><ObjectID><Names>` (8.6). Cela permet au système demandeur d'identifier l'objet auquel l'ID transaction s'applique.

EXEMPLE Les paragraphes D.15.2 et D.16.1 donnent des exemples de messages de réponse qui contiennent des ID transaction.

8.5.5 Envoi d'un message de demande d'annulation

Un message de demande d'annulation doit comporter un verbe d'en-tête `cancel`. Le nom d'en-tête doit être le même que celui de la demande d'origine à laquelle fait référence le message d'annulation.

Les transactions à annuler doivent être spécifiées par les ID transaction respectifs. Ces derniers doivent être placés dans des éléments `<Payload><Transaction><TransactionID>` séparés.

EXEMPLE Les paragraphes D.15.3 et D.16.2 donnent des exemples de message de demande d'annulation.

8.5.6 Réponse à un message de demande d'annulation

Le système répondant doit répondre à un message de demande d'annulation par un message de réponse (5.3).

Ce message de réponse doit spécifier l'état de la demande d'annulation dans l'élément `<Reply><Status><Code>` (7.6.4.7). Ce dernier peut prendre l'une des trois valeurs possibles suivantes.

`0.0` – Cette valeur doit indiquer que la demande d'annulation a été réalisée avec succès.

`0.4` – Cette valeur doit indiquer que la demande d'annulation a été reçue et acceptée. L'opération d'annulation réelle doit être placée en file d'attente pour son exécution. Après que le système répondant a réalisé l'opération d'annulation, il doit informer le système demandeur du succès de l'opération ou de l'annulation par l'envoi d'un message de réponse suivant (5.3) qui contient un élément `<Reply><Status><Code>` approprié.

`2.16` – Cette valeur doit indiquer que la demande d'annulation peut ne pas être réalisée, car un ID transaction non valide a été spécifié.

Une autre valeur de l'élément `<Reply><Status><Code>` doit indiquer qu'une autre erreur a été détectée dans la demande d'annulation.

8.6 Identification d'objet

8.6.1 Généralités

Bien que le présent document ne formule aucun avis en ce qui concerne les données spécifiques à l'application qui sont transmises dans les messages de la Partie 100, il spécifie un moyen cohérent de référencer tous les objets CIM ou spécifiques à l'application à partir d'un élément `<Reply>` ou `<Payload>` d'un message.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

La façon précise selon laquelle un objet donné peut être référencé est appelée identification.

8.6.2 Règles d'identification d'objet

Les classes et attributs de dénomination d'objets CIM sont spécifiés dans l'IEC 61970-301.

Un objet doit être identifié de manière unique au moyen d'un seul élément <mRID> (8.6.3) ou d'un ou plusieurs éléments <Name> (8.6.4.2). Ces moyens d'identification ne sont pas mutuellement exclusifs. Il convient d'utiliser la méthode <mRID> de dénomination dans toute la mesure du possible.

L'ensemble des identifiants faisant référence à un objet peut être étendu ou modifié (8.7).

Au moins un identifiant unique doit être attribué à un objet lors de son instanciation (8.3.2). Cet identifiant peut par la suite être utilisé pour référencer l'objet, par exemple, lorsque l'objet doit être demandé, mis à jour ou supprimé (8.6.5).

8.6.3 Identification d'objets à l'aide d'un élément n <mRID>

Il convient d'utiliser un identifiant de ressource principal (mRID) pour identifier un objet.

Un mRID doit être spécifié en tant qu'identifiant unique universel (UUID) de 128 bits généré selon la RFC 4122.

Un objet donné ne doit pas avoir plusieurs mRID.

8.6.4 Identification d'objets à l'aide d'un élément <Name>

8.6.4.1 Généralités

Un élément <Name><name> peut être utilisé pour identifier un objet.

Le contenu de l'élément <Name><name> n'est pas normatif.

Si l'élément <Name><name> n'identifie pas de manière unique l'objet, le nom peut être en outre qualifié en spécifiant un élément <Name><NameType><name>. Le contenu cet élément <Name><NameType><name> n'est pas normatif.

Si cela génère toujours une potentielle ambiguïté, le nom peut en outre être qualifié en spécifiant un élément <Name><NameType><NameTypeAuthority><name>. Le contenu cet élément n'est pas normatif.

8.6.4.2 Caractère unique des éléments <Name>

Tous les identifiants <Name> ne doivent pas être uniques. Toutefois, si un objet n'est pas associé à un identifiant <mRID>, au moins l'un de ses identifiants <Name> doit alors être unique.

Un élément <Name> peut être attribué à des fins autres que l'identification d'objet. Il n'est pas nécessaire que ce type d'élément <Name> soit unique.

8.6.4.3 Structure

La structure de l'élément <Name> est représentée à la Figure 47.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

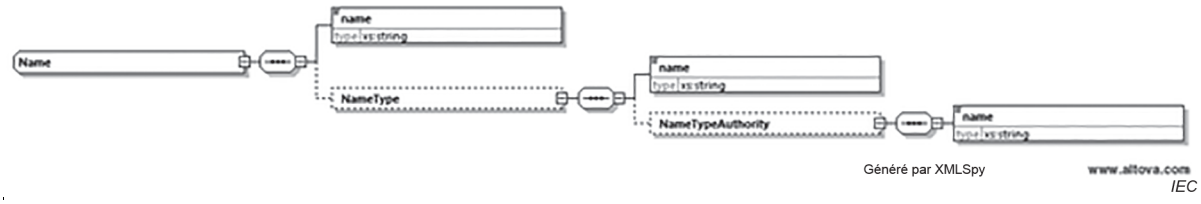


Figure 47 – Schéma de dénomination général

8.6.5 Référencement d'objets

8.6.5.1 Généralités

Il peut s'avérer nécessaire de faire directement référence à un objet à partir d'un message. Pour ce faire, le sous-élément `<ObjectID>` des éléments `<Reply><Status>` et `<Payload>` est utilisé. Ces exemples incluent:

- l'utilisation de l'élément `<Reply><Status><ObjectID>` pour faire référence à un objet à partir d'un message de réponse afin d'associer les notifications d'erreur ou d'état (7.6.4.6).
- l'utilisation de l'élément `<Payload><ObjectID>` pour faire référence à un objet à partir d'une demande "create", "change", "delete" ou "update" (8.3.2, 8.3.3, 8.3.4, 8.3.5)
- l'utilisation de l'élément `<Payload><ObjectID>` pour faire référence à un objet à partir d'un message de demande d'annulation (8.5)

8.6.5.2 Structure

Les éléments `<ObjectID>` partagent une structure similaire représentée à la Figure 48.

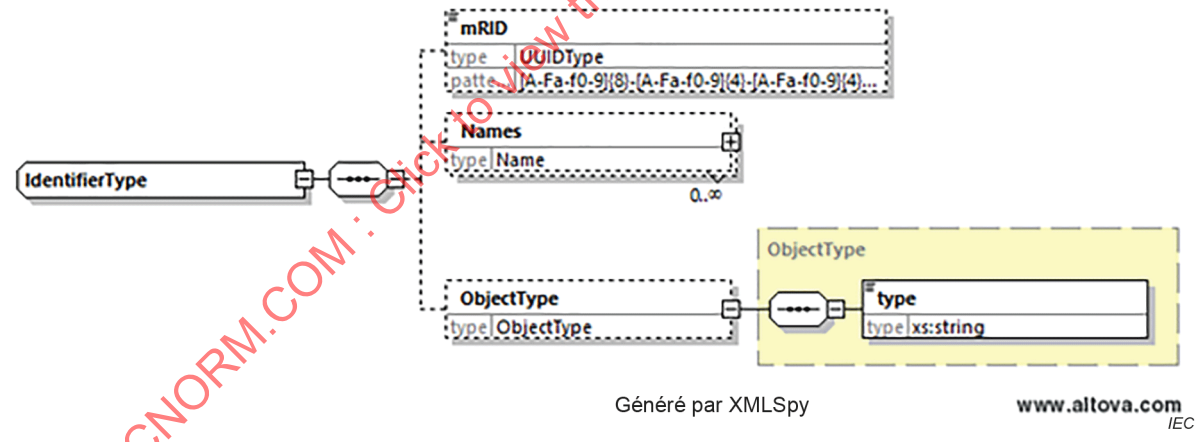


Figure 48 – Structure d'un élément <ObjectID>

Le Tableau 19 spécifie les différents sous-éléments qui peuvent être présents dans un élément `<ObjectID>>` et leur sémantique associée.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Tableau 19 – Sous-éléments d'un élément <ObjectID>

Élément	Plage de cardinalités		Type de données XSD	Signification
<mRID>	0	1	UUIDType	Le mRID de l'objet référencé (8.6.3).
<Names>	0	plusieurs	Nom	Le nom de l'objet référencé (8.6.4).
<ObjectType>	0	1	ObjectType	
<type>	1	1	xs:string	Cet élément doit être utilisé pour identifier le type d'objet auquel le nom fait référence Cela offre un moyen supplémentaire de lever l'ambiguïté des objets nommés. La valeur de cet élément doit être le nom de la classe réelle (et pas celui d'une classe parent) de l'objet référencé.

8.7 Profil ObjectIdentificationRevisionSets

8.7.1 Généralités

Le profil ObjectIdentificationRevisionSets permet d'ajouter (8.7.4), de supprimer (8.7.5) ou de modifier les identifiants (8.6) d'un objet existant.

Ce type de demande doit avoir lieu dans le contexte d'un modèle d'échange de messages de demande/réponse (6.2).

Le XSD du profil ObjectIdentificationRevisionSets est défini dans le fichier `LegacyObjectIdentificationRevisionSets.xsd`. Le contenu de ce fichier est présenté à l'Article A.4.

Le profil ObjectIdentificationRevisionSets offre la seule méthode qui permet d'ajouter, de supprimer ou de modifier les identifiants d'un objet existant.

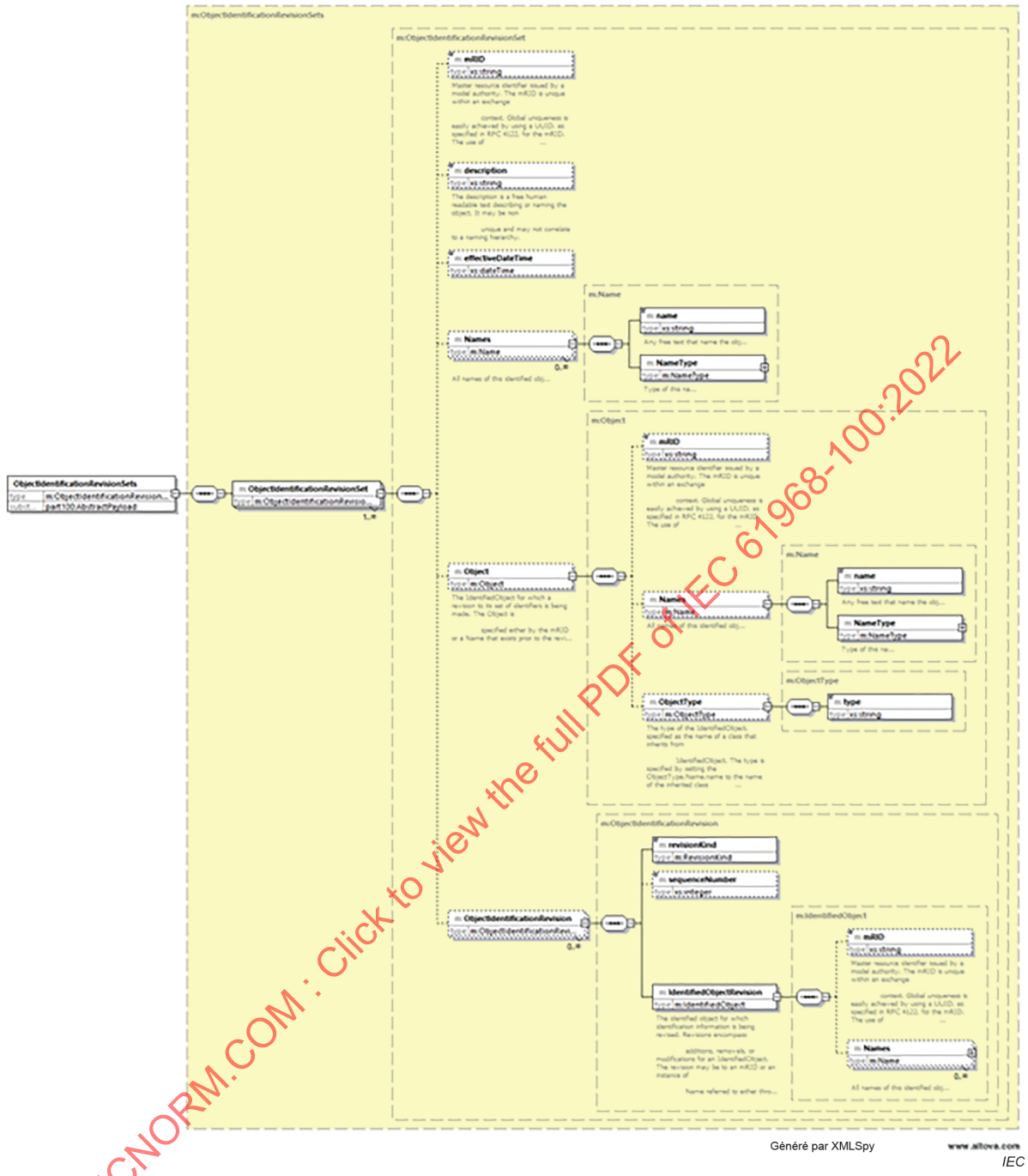
Un identifiant doit être modifié en supprimant l'identifiant existant et en ajoutant un nouvel identifiant.

EXEMPLE Des exemples d'utilisation de messages ObjectIdentificationRevisionSets pour ajouter ou supprimer des identifiants peuvent être consultés à l'Annexe F.

8.7.2 Structure

Le profil ObjectIdentificationRevisionSets est décrit à la Figure 49.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure 49 – Profil ObjectIdentificationRevisionSets

Le Tableau 20 décrit les différents sous-éléments qui peuvent être présents dans un message ObjectIdentificationRevisionSets et leur sémantique associée.

Tableau 20 – Éléments d'un profil ObjectIdentificationRevisionSets

Élément	Plage de cardinalités		Type de données XSD	Signification
<mRID>	0	1	xs:string	Permet de donner une identité à ObjectIdentificationRevisionSets. .
<description>	0	1	xs:string	
<Names>	0	plusieurs	Nom	
<effectiveDateTime>	0	1	xs:dateTime	Spécifie l'heure à laquelle il convient d'ajouter ou de supprimer l'identifiant. S'il est omis, la valeur par défaut indique que l'opération doit avoir lieu immédiatement.
<Object>	0	1	Objet	Spécifie l'objet existant dont le ou les identifiants doivent être ajoutés ou supprimés.
<mRID>	0	1	xs:string	Le mRID existant de l'objet référencé (8.6.3, 8.7.4, 8.7.5).
<Names>	0	1	Nom	Un nom existant de l'objet référencé (8.6.4, 8.7.4, 8.7.5).
<ObjectType>	0	1	ObjectType	
<type>	1	1	xs:string	Cet élément doit être utilisé pour identifier le type d'objet auquel le nom fait référence Cela offre un moyen supplémentaire de lever l'ambiguïté des objets nommés. La valeur de cet élément doit être le nom de la classe réelle (et pas celui d'une classe parent) de l'objet référencé.
<ObjectIdentificationRevision>	0	plusieurs	ObjectIdentificationRevision	Spécifie les identifiants ajoutés ou supprimés
<revisionKind>	1	1	RevisionKind	add ou remove
<sequenceNumber>	0	1	xs:integer	Spécifie l'ordre dans lequel il convient de réaliser plusieurs demandes d'ajout ou de suppression.
<IdentifiedObjectRevision>			ObjectIdentifiedRevision	
<mRID>	0	1	UUIDType	Le mRID ajouté ou supprimé de l'objet référencé (8.6.3, 8.7.4, 8.7.5).
<Names>	0	plusieurs	Nom	Le ou les noms ajoutés ou supprimés de l'objet référencé (8.6.4, 8.7.4, 8.7.5).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

8.7.3 Utilisation du profil ObjectIdentificationRevisionSets

Un message de demande avec un <Header><Noun> dont la valeur est ObjectIdentificationRevisionSets doit être interprété comme une demande d'ajout, de suppression ou de modification d'un ou de plusieurs identificateurs associés à un objet donné (8.7.4, 8.7.5).

Un OperationSet (7.7.3) peut également être utilisé pour combiner plusieurs de ces opérations dans un seul message.

8.7.4 Ajout d'un ou de plusieurs nouveaux identifiants

Le profil ObjectIdentificationRevisionSets peut être utilisé pour attribuer de nouveaux identifiants à un objet existant.

`<Header><Verb>` doit être `create`.

L'élément `<Payload><ObjectIdentificationRevisionSets><ObjectIdentificationRevisionSet><ObjectIdentificationRevision><revisionKind>` doit être `add`.

L'un des éléments `<Payload><ObjectIdentificationRevisionSets><ObjectIdentificationRevisionSet><Object><mRID>` ou `<Name>` doit être spécifié. Cela permet de définir l'objet existant dont l'ensemble d'identifiants doit être étendu.

Les éléments `<Payload><ObjectIdentificationRevisionSets><ObjectIdentificationRevisionSet><ObjectIdentificationRevision><IdentifiedObjectRevision><mRID>` ou `<Name>` doivent également être spécifiés. Ils doivent spécifier les identifiants supplémentaires par lesquels l'objet doit être connu.

Dans la mesure où un objet peut ne pas avoir plusieurs mRID (8.6.3), un `<mRID>` peut uniquement être ajouté à un objet qui n'est associé à aucun mRID existant.

8.7.5 Suppression d'un ou de plusieurs identifiants

Le profil ObjectIdentificationRevisionSets peut être utilisé pour supprimer un ou plusieurs identifiants d'un objet existant.

Le mRID existant (8.6.3) d'un objet peut être supprimé, même si cette pratique est déconseillée.

`<Header><Verb>`doit être `create`.

L'élément `<Payload><ObjectIdentificationRevisionSets>`

`<ObjectIdentificationRevisionSet><ObjectIdentificationRevision><revisionKind>` doit être `remove`.

L'un des éléments `<Payload><ObjectIdentificationRevisionSets><ObjectIdentificationRevisionSet><Object><mRID>` ou `<Name>` doit être spécifié. Cela permet de définir l'objet existant dont les identifiants doivent être supprimés.

Les éléments `<Payload><ObjectIdentificationRevisionSets><ObjectIdentificationRevisionSet><ObjectIdentificationRevision><Identifier><mRID>` ou `<Name>` doivent également être spécifiés. Ils doivent indiquer les identifiants qui doivent être supprimés.

8.7.6 Erreurs possibles concernant les demandes ObjectIdentificationRevisionSet

Les codes `<Reply><Status><Code>` qui figurent dans le Tableau 10 doivent être utilisés pour communiquer une notification de succès ou d'erreur rencontrée lors de l'exécution d'un message de demande ObjectIdentificationRevisionSets.

9 Transport des messages

9.1 Généralités

Le présent article détaille les considérations relatives à toutes les applications IEC 61968-100 qui échangent des messages sur les services Web SOAP (9.2) et JMS (9.3). D'autres services de transport peuvent également être utilisés, mais leur utilisation n'est pas normative et ne relève pas du domaine d'application de la présente Norme internationale.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

9.2 Transport sur les services Web SOAP

9.2.1 Généralités

Le présent article décrit l'utilisation des services Web SOAP (Simple Object Access Protocol) 1.1 sur HTTP pour le transport de messages IEC 61968-100.

9.2.2 Codage des messages IEC 61968-100 à l'intérieur des messages SOAP

Un message IEC 61968-100 doit être acheminé dans l'élément `<soap:Envelope><soap:Body>` du message SOAP.

`SOAPAction` et les autres en-têtes HTTP n'ont aucune valeur normative en ce qui concerne la présente Norme internationale. Il revient à l'intégrateur système de définir leurs valeurs.

9.2.3 Exemple de message SOAP

La Figure 50 représente comment l'exemple de message IEC 61968-100 de la Figure D.28 peut être acheminé à l'intérieur d'un message SOAP.

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header/>
  <soap:Body>
    <GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Header>
        <Verb>get</Verb>
        <Noun>MeterReadings</Noun>
        <Timestamp>2018-10-02T14:54:31-06:00</Timestamp>
        <MessageID>1f92de80-b835-4074-b42e-233164063a39</MessageID>
        <CorrelationID>6d6fe897-b588-40a2-ba30-55e4d838b6f4</CorrelationID>
        <Comment>Request for historical meter read data</Comment>
      </Header>
      <Payload>
        <GetMeterReadings xmlns="http://iec.ch/TC57/2011/GetMeterReadings#"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <EndDevice>
            <Names>
              <name>M001</name>
            </Names>
          </EndDevice>
          <ReadingType>
            <Names>
      <!--bulkQuantity forward electricitySecondaryMetered energy (kWh) -->
              <name>0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0</name>
            </Names>
          </ReadingType>
          <TimeSchedule>
            <scheduleInterval>
              <end>2018-07-01T09:50:00-06:00</end>
              <start>2018-07-01T09:00:00-06:00</start>
            </scheduleInterval>
          </TimeSchedule>
        </GetMeterReadings>
      </Payload>
    </GetRequestMessage>
  </soap:Body>
</soap:Envelope>
```

Figure 50 – Exemple de message SOAP

9.2.4 Mapping des messages IEC 61968-100 aux messages SOAP

L'utilisation de SOAP sur HTTP exige d'échanger tous les messages au niveau transport dans un modèle d'échange de messages de demande/réponse synchrone.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Les modèles d'échange de messages définis en 6.2, 6.3 et 6.4 permettent à un système de réception d'envoyer des messages de réponse asynchrones à un système d'envoi. Pour permettre au transport SOAP sous-jacent d'envoyer un message de réponse quel que soit le modèle d'échange de messages au niveau de l'application, un message d'accusé de réception simple (5.3.6) doit être envoyé dans la réponse HTTP synchrone (6.5).

EXEMPLE Des exemples de messages d'accusé de réception simple peuvent être consultés en D.3.4, D.6.4, D.8.6, D.8.8 et D.12.6.

9.2.5 Fichiers WSDL

9.2.5.1 Généralités

Le présent document définit de manière normative deux types de fichiers WSDL utilisés pour le transport de messages IEC 61968-100 sur SOAP. Il s'agit de fichiers WSDL génériques et WSDL fortement typés. Ces deux types de fichiers WSDL doivent être générés à l'aide des modèles de fichiers qui font partie intégrante de la présente Norme internationale (Annexe B).

Par rapport au fichier WSDL générique, un fichier WSDL fortement typé présente l'avantage d'offrir un contrôle de type plus strict. En particulier, l'élément AbstractPayloadType (7.7.5) est remplacé par un profil XSD spécifique.

Un système conforme doit mettre en œuvre une interface de services Web définie par des modèles de fichier WSDL génériques ou fortement typés.

Une mise en œuvre peut définir ses services Web en fonction de la tâche d'intégration en cours. À cet effet, il peut être nécessaire de fusionner ou de modifier les fichiers WSDL générés de manière normative.

9.2.5.2 Fichiers WSDL génériques

Pour un système client SOAP, un fichier WSDL générique spécifie le point d'extrémité HTTP auquel il convient d'envoyer tous les messages SOAP.

Pour un système de serveur SOAP, un fichier WSDL générique permet à un point d'extrémité SOAP d'accepter les messages IEC 61968-100 et de les diffuser en interne en fonction de leurs éléments racines respectifs.

Un fichier WSDL générique doit être créé au moyen du modèle de fichier `GenericMessageWSDLTemplate.txt` dont le contenu figure à l'Article B.2.

Pour générer un fichier WSDL générique à partir de ce modèle de fichier, il est en premier lieu nécessaire de copier le modèle de fichier dans un nouvel emplacement, puis de modifier son contenu selon le Tableau 21. Il convient de remplacer globalement chaque variable énumérée dans tout le fichier.

Tableau 21 – Noms de variable du modèle de fichier WSDL générique

Nom de variable	Signification	Exemple
`{PATHPREFIX}`	Dossier dans lequel se trouve le fichier Message.xsd	../XSDs
`{SOAPADDRESS}`	Point d'extrémité HTTP vers lequel il convient d'acheminer les messages SOAP	http://example.com

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

9.2.5.3 Fichiers WSDL fortement typés

Pour un système client SOAP, un fichier WSDL fortement typé spécifie le point d'extrémité HTTP auquel il convient d'envoyer tous les messages SOAP.

Pour un système de serveur SOAP, un fichier WSDL fortement typé permet à un point d'extrémité SOAP d'accepter les messages IEC 61968-100 reçus et de les diffuser en interne en fonction de leurs éléments racines et noms de profil respectifs.

La création d'un fichier WSDL fortement typé est un processus en deux étapes. En premier lieu, un fichier de profil XSD de message fortement typé doit être créé en copiant et modifiant le modèle de fichier `StrongMessageXSDTemplate.txt` dont le contenu figure à l'Article B.3. Cette modification doit être conforme au Tableau 22. Il convient de remplacer globalement chaque variable énumérée dans tout le fichier.

Tableau 22 – Noms de variable du modèle XSD de message fortement typé

Nom de variable	Signification	Exemple
{GETPROFILENAMESPACE}	Espace de nom pour les demandes "get'' (8.4)	http://iec.ch/TC57/2011/GetMeterReadings#
{GETPROFILENAME}	Nom de profil pour les demandes "get'' (8.4)	GetMeterReadings
{PROFILENAMESPACE}	Espace de nom pour toutes les autres demandes et réponses	http://iec.ch/TC57/2011/MeterReadings#
{PROFILENAME}	Nom de profil pour toutes les autres demandes et réponses	MeterReadings
{PROFILEPATHPREFIX}	Dossier dans lequel se trouvent les fichiers XSD spécifiques à la partie	../Part9XSDs
{PART100PATHPREFIX}	Dossier dans lequel se trouve le fichier Message.xsd	../XSDs

En second lieu, le fichier WSDL fortement typé doit être créé en copiant et modifiant le modèle de fichier `StrongMessageWSDLTemplate.txt` dont le contenu figure à l'Article B.4. Cette modification doit être conforme au Tableau 23. Il convient de remplacer globalement chaque variable énumérée dans tout le fichier.

Tableau 23 – Noms de variables du modèle WSDL fortement typé

Nom de variable	Signification	Exemple
{GETPROFILENAME}	Nom de profil pour les demandes "get'' (8.4)	GetMeterReadings
{PROFILENAME}	Nom de profil pour toutes les autres demandes et réponses	MeterReadings
{PATHPREFIX}	Dossier dans lequel se trouve le fichier Message.xsd	../XSDs
{SOAPADDRESS}	Point d'extrémité HTTP vers lequel il convient d'acheminer les messages SOAP	http://example.com

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

9.3 Transport sur JMS

9.3.1 Généralités

Le présent article décrit l'utilisation de JMS (Java Message Service) pour le transport des messages IEC 61968-100.

Selon la spécification JMS, "JMS offre aux programmes Java un moyen commun de créer, d'envoyer, de recevoir et de lire les messages d'un système de messagerie d'entreprise". Il décrit un ensemble d'interfaces plutôt qu'une mise en œuvre sous-jacente. Il est généralement laissé à l'appréciation de l'implémenteur de choisir la mise en œuvre conforme.

Le présent article décrit les considérations normatives et informatives dont doit tenir compte une application conforme à l'IEC 61968-100 lors de l'utilisation de JMS pour transporter des messages.

9.3.2 Modèle d'échange de messages

En règle générale, il convient d'utiliser les files d'attente JMS pour envoyer et recevoir des messages dans le cadre du modèle d'échange de messages de demande/réponse (6.2) ou du modèle d'échange de messages induit (6.4).

Il convient en général d'utiliser les rubriques JMS pour envoyer et recevoir des messages dans le cadre du modèle d'échange de messages d'événement non sollicité (6.3). Les rubriques JMS peuvent également être adaptées au modèle de message d'événement induit (6.4) lorsque plusieurs parties sont concernées par l'événement réel.

Outre ces considérations, il revient à l'implémenteur de décider de la manière dont il convient de mettre en place l'infrastructure de messagerie en ce qui concerne les files d'attente et rubriques JMS.

9.3.3 Messages d'accusé de réception au niveau de l'application

JMS permet au système répondant de transmettre automatiquement un accusé de réception au niveau du transport dès la réception d'un message. JMS assure également la persistance des messages. Ensemble, l'accusé de réception automatique des messages et la persistance des messages peuvent contribuer, au niveau transport, à veiller à ce qu'un système répondant reçoive bien un message transmis.

Toutefois, s'il est nécessaire pour une application de vérifier qu'un message transmis a été reçu par son application homologue, le seul moyen est que cette dernière renvoie un accusé de réception au niveau de l'application. Cela peut être particulièrement le cas en présence de plusieurs sauts entre un système et son homologue.

Par conséquent, en règle générale, il convient qu'une application attribue la valeur `true` à l'indicateur `<AckRequired>` dans l'en-tête de message IEC 61968-100 (7.5) afin de forcer l'application homologue à renvoyer un message d'accusé de réception simple (6.5). La réception de ce type de message assure que le message d'origine a été reçu par son homologue.

NOTE Ce type de message d'accusé de réception simple est nécessaire uniquement en cas d'envoi d'un message d'événement (5.4) ou d'un message de demande (5.3) dont un <Header><Verb> a la valeur "reply" (7.5.4.3). Dans tous les autres cas, le système répondant renvoie un message de réponse (6.2, 6.4).

9.3.4 En-têtes JMS

9.3.4.1 Généralités

JMS fournit plusieurs champs d'en-tête de niveau de transport qui peuvent être définis à la convenance d'une application. Ces champs incluent JMSMessageID, JMSCorrelationID, JMSReplyTo et JMSTimestamp.

9.3.4.2 Champ d'en-tête JMSMessageID

La spécification JMS indique que "le champ d'en-tête JMSMessageID contient une valeur qui identifie de manière unique chaque message envoyé par un fournisseur". JMS impose la restriction selon laquelle l'en-tête `JMSMessageID` doit commencer par le préfixe "`ID:`".

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Le champ d'en-tête JMSMessageID joue donc un rôle similaire à celui du sous-élément <MessageID> de l'élément IEC 61968-100 <Header>. Par conséquent, pour les applications conformes à l'IEC 61968-100, il convient d'attribuer au champ d'en-tête JMSMessageID la valeur du sous-élément <MessageID> (7.5) accompagné du préfixe "ID:".

9.3.4.3 Champ d'en-tête JMSCorrelationID

La spécification JMS indique que "un client peut utiliser le champ d'en-tête JMSCorrelationID pour associer un message à un autre. Une utilisation classique consiste à associer un message de réponse à son message de demande". JMS impose la restriction selon laquelle le champ d'en-tête JMSCorrelationID doit commencer par le préfixe "ID:".

Le champ d'en-tête JMSCorrelationID joue donc un rôle similaire à celui du sous-élément <CorrelationID> de l'élément IEC 61968-100 <Header>. Par conséquent, pour les applications conformes à l'IEC 61968-100, il convient d'attribuer au champ d'en-tête JMSCorrelationID la valeur du sous-élément <CorrelationID> (7.5) accompagné du préfixe "ID:".

9.3.4.4 Champ d'en-tête JMSReplyTo

La spécification JMS indique que "le champ d'en-tête JMSReplyTo contient une Destination fournie par un client lorsqu'un message est envoyé. Il s'agit de la destination à laquelle il convient d'envoyer une réponse au message".

Le champ d'en-tête JMSReplyTo joue donc un rôle similaire à celui du sous-élément <ReplyAddress> de l'élément IEC 61968-100 <Header>. Par conséquent, pour les applications conformes à l'IEC 61968-100, il convient d'attribuer au champ d'en-tête JMSReplyTo la même valeur que le sous-élément <ReplyAddress> (7.5).

9.3.4.5 Champ d'en-tête JMSTimestamp

La spécification JMS indique que "le champ d'en-tête JMSTimestamp contient l'heure à laquelle un message a été transféré à un fournisseur pour être envoyé".

Le champ d'en-tête JMSTimestamp joue donc un rôle similaire à celui du sous-élément <Timestamp> de l'élément IEC 61968-100 <Header>. Toutefois, ce champ d'en-tête JMSTimestamp est défini automatiquement par le producteur du message JMS lorsque l'opération d'envoi est appelée et n'est pas sous le contrôle direct de l'application. Par conséquent, un système de réception peut signaler des différences entre les valeurs du champ d'en-tête JMSTimestamp et l'élément <Header><Timestamp>.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annexe A
(normative)

Fichiers XSD

A.1 Généralités

Les fichiers XSD qui figurent dans la présente annexe font partie du présent document.

A.2 Fichier Message.xsd

Le contenu du fichier `Message.xsd` (7.3) est représenté à la Figure A.1.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://iec.ch/TC57/2021/schema/message"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://iec.ch/TC57/2021/schema/message"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="1.0.0">
  <xs:annotation>
    <xs:documentation>
        This is the Message.xsd file which forms an integral part of the
International Standard
        IEC 61968-100:2021, "IEC IMPLEMENTATION PROFILES FOR APPLICATION
INTEGRATION".
        IEC 61968-100:2021 is available from the International
Electrotechnical Commission at http://www.iec.ch
        and contains a full description of the contents of this file.
    </xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="AbstractPayloadType.xsd"/>
  <xs:element name="CanceledEventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="canceled"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CanceledEventResponseMessage" type="SimpleAckReplyType"/>
  <xs:element name="CancelRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
              <xs:element name="Verb" type="RequestVerbType" fixed="cancel"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CancelResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ChangedEventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="changed"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ChangedEventResponseMessage" type="SimpleAckReplyType"/>
  <xs:element name="ChangeRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="change"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
    </xs:choice>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="ChangeResponseMessage">
 <xs:complexType>
  <xs:sequence>
    <xs:element name="Header" type="ResponseHeaderType"/>
    <xs:element name="Reply" type="ReplyType"/>
    <xs:choice minOccurs="0">
      <xs:element name="Payload">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
    </xs:choice>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="CreatedEventMessage">
 <xs:complexType>
  <xs:sequence>
    <xs:element name="Header">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Verb" type="EventVerbType" fixed="created"/>
          <xs:element name="Noun" type="xs:string"/>
          <xs:group ref="EventHeaderGroup"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:choice>
      <xs:element name="Payload">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
    </xs:choice>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="CreatedEventResponseMessage" type="SimpleAckReplyType"/>
<xs:element name="CreateRequestMessage">
 <xs:complexType>
  <xs:sequence>
    <xs:element name="Header">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Verb" type="RequestVerbType" fixed="create"/>
          <xs:element name="Noun" type="xs:string"/>
          <xs:group ref="RequestHeaderGroup"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:choice>
      <xs:element name="Payload">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CreateResponseMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Header" type="ResponseHeaderType"/>
      <xs:element name="Reply" type="ReplyType"/>
      <xs:choice minOccurs="0">
        <xs:element name="Payload">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
              <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="DeletedEventMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Header">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Verb" type="EventVerbType" fixed="deleted"/>
            <xs:element name="Noun" type="xs:string"/>
            <xs:group ref="EventHeaderGroup"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:choice>
        <xs:element name="Payload">
          <xs:complexType>
            <xs:choice>
              <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
              <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
        <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="DeletedEventResponseMessage" type="SimpleAckReplyType"/>
<xs:element name="DeleteRequestMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Header">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Verb" type="RequestVerbType" fixed="delete"/>
            <xs:element name="Noun" type="xs:string"/>
            <xs:group ref="RequestHeaderGroup"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:choice>
        <xs:element name="Payload">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DeleteResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecutedEventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="executed"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet" type="EventOperationSetType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecutedEventResponseMessage" type="SimpleAckReplyType"/>
  <xs:element name="ExecuteRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="execute"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet" type="RequestOperationSetType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecuteResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="ResponseHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Reply" type="ReplyType"/>
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet"
type="ResponseOperationSetType" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="FaultMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="get"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="UpdatedEventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="updated"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="UpdatedEventResponseMessage" type="SimpleAckReplyType"/>
  <xs:element name="UpdateRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="update"/>
              <xs:element name="Noun" type="xs:string"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="UpdateResponseMessage">
    <xs:complexType>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="CompressedPayloadType">
    <xs:sequence>
      <xs:element name="CompressionKind">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="zip"/>
            <xs:enumeration value="gzip"/>
            <xs:enumeration value="other"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Base64" type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="EventVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:group ref="EventHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="EnforceMsgSequence" type="xs:boolean" minOccurs="0"/>
      <xs:element name="EnforceTransactionalIntegrity" type="xs:boolean"
minOccurs="0"/>
      <xs:element name="Operation" type="EventOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="EventVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
        <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
        <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="IdentifierType">
    <xs:sequence>
      <xs:element name="mRID" type="UUIDType" minOccurs="0"/>
      <xs:element name="Names" type="Name" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="ObjectType" type="ObjectType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ObjectType">
    <xs:sequence>
      <xs:element name="type" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MessagePropertyType">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Value" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MessageSequenceType">
    <xs:sequence>
      <xs:element name="SequenceNumber" type="xs:unsignedInt" minOccurs="0"/>
      <xs:element name="LastMessage" type="xs:boolean" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Name">
    <xs:sequence>
      <xs:element name="mRID" type="xs:string" minOccurs="0"/>
      <xs:element name="name" type="xs:string" minOccurs="0"/>
      <xs:element name="NameAuthority" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="mRID" type="xs:string" minOccurs="0"/>
            <xs:element name="name" type="xs:string" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="NameType" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="mRID" type="xs:string" minOccurs="0"/>
            <xs:element name="name" type="xs:string" minOccurs="0"/>
            <xs:element name="NameTypeAuthority" minOccurs="0">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="mRID" type="xs:string" minOccurs="0"/>
                  <xs:element name="name" type="xs:string" minOccurs="0"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ObjectUpdateType">
    <xs:sequence>
      <xs:element name="UpdateID" type="UUIDType"/>
      <xs:element name="UpdateAction">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="add"/>
            <xs:enumeration value="modify"/>
            <xs:enumeration value="remove"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element ref="AbstractPayload"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ReplyType">
    <xs:sequence>
      <xs:element name="Status" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="StatusType"/>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="Transaction" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="TransactionType"/>
          </xs:complexContent>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="RequestVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:group ref="RequestHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="EnforceMsgSequence" type="xs:boolean" minOccurs="0"/>
      <xs:element name="EnforceTransactionalIntegrity" type="xs:boolean"
minOccurs="0"/>
      <xs:element name="Operation" type="RequestOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="RequestVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
        <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
        <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:group ref="ResponseHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="Operation" type="ResponseOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element ref="AbstractPayload" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="SimpleAckReplyType">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <xs:sequence>
      <xs:element name="Header" type="ResponseHeaderType"/>
      <xs:element name="Reply">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Status">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Level" type="xs:string" fixed="OK"/>
                  <xs:element name="Code" type="xs:string" fixed="0.3"/>
                  <xs:element name="Reason" type="xs:string" minOccurs="0"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="StatusType">
    <xs:sequence>
      <xs:element name="Level">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="OK"/>
            <xs:enumeration value="WARNING"/>
            <xs:enumeration value="ERROR"/>
            <xs:enumeration value="UNKNOWN"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Code" type="xs:string"/>
      <xs:element name="Namespace" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="http://.+"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Reason" type="xs:string" minOccurs="0"/>
      <xs:element name="ObjectID" type="IdentifierType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="OperationSet" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="OperationSetID" type="UUIDType"/>
            <xs:element name="Operation" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="OperationID" type="xs:unsignedInt"/>
                  <xs:element name="ObjectID" type="IdentifierType"
minOccurs="0" maxOccurs="unbounded"/>
                  <xs:element name="UpdateID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element name="TransactionID" type="UUIDType"
minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="UpdateID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="TransactionID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="TransactionType">
    <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="TransactionID" type="UUIDType"/>
      <xs:choice minOccurs="0">
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="OperationSet" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence minOccurs="0">
              <xs:element name="OperationSetID" type="UUIDType"/>
              <xs:element name="OperationID" type="xs:unsignedInt"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UserType">
    <xs:sequence>
      <xs:element name="UserID" type="xs:string"/>
      <xs:element name="Organization" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="EventVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="canceled"/>
      <xs:enumeration value="changed"/>
      <xs:enumeration value="created"/>
      <xs:enumeration value="deleted"/>
      <xs:enumeration value="executed"/>
      <xs:enumeration value="updated"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="RequestVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="cancel"/>
      <xs:enumeration value="change"/>
      <xs:enumeration value="create"/>
      <xs:enumeration value="delete"/>
      <xs:enumeration value="get"/>
      <xs:enumeration value="execute"/>
      <xs:enumeration value="update"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ResponseVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="reply"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="UUIDType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Fa-f0-9]{8}-[A-Fa-f0-9]{4}-[A-Fa-f0-9]{4}-[A-Fa-
f0-9]{4}-[A-Fa-f0-9]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:group name="EventHeaderGroup">
    <xs:sequence>
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="MessageSequence" type="MessageSequenceType"
minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="RequestHeaderGroup">
    <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="ReplyAddress" type="xs:string" minOccurs="0"/>
      <xs:element name="AckRequired" type="xs:boolean" minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="ResponseHeaderGroup">
    <xs:sequence>
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="MessageSequence" type="MessageSequenceType"
minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
</xs:schema>
```

Figure A.1 – Contenu du fichier Message.xsd

A.3 Fichier `AbstractPayloadType.xsd`

Le contenu du fichier `AbstractPayloadType.xsd` est représenté à la Figure A.2. Ce fichier peut être inclus par les profils XSD spécifiques à l'application (8.2.2, Annexe C).

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://iec.ch/TC57/2021/schema/message"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified" version="1.0.0">
  <xs:annotation>
    <xs:documentation>
        This is the AbstractPayloadType.xsd file which forms an integral
        part of the International Standard IEC 61968-100:2021,
        "IEC IMPLEMENTATION PROFILES FOR APPLICATION INTEGRATION".
        IEC 61968-100:2021 is available from the International
        Electrotechnical Commission at http://www.iec.ch
        and contains a full description of the contents of this file.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType name="AbstractPayloadType"/>
  <xs:element name="AbstractPayload" type="AbstractPayloadType"
    abstract="true"/>
</xs:schema>
```

Figure A.2 – Contenu du fichier AbstractPayloadType.xsd

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

A.4 Fichier LegacyObjectIdentificationRevisionSets.xsd

Le contenu du fichier `LegacyObjectIdentificationRevisionSets.xsd` (8.7) est représenté à la Figure A.3.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:part100="http://iec.ch/TC57/2021/schema/message"
  xmlns:a="http://langdale.com.au/2005/Message#"
xmlns:sawsdl="http://www.w3.org/ns/sawsdl"
targetNamespace="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#"
elementFormDefault="qualified"
  attributeFormDefault="unqualified"
xmlns="http://langdale.com.au/2005/Message#"
xmlns:m="http://iec.ch/TC57/2020/ObjectIdentificationRevisionSets#">
  <xs:annotation>
    <xs:documentation>
        This is the LegacyObjectIdentificationRevisionSets.xsd file which
forms an integral
        part of the International Standard IEC 61968-100:2021,
        "IEC IMPLEMENTATION PROFILES FOR APPLICATION INTEGRATION".
        IEC 61968-100:2021 is available from the International
        Electrotechnical Commission at http://www.iec.ch
        and contains a full description of the contents of this file.
    </xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
schemaLocation="AbstractPayloadType.xsd" xmlns="" />
  <xs:element name="ObjectIdentificationRevisionSets"
type="m:ObjectIdentificationRevisionSets"
substitutionGroup="part100:AbstractPayload"
    xmlns="" />
  <xs:complexType name="ObjectIdentificationRevisionSets" xmlns="">
    <xs:complexContent>
      <xs:extension base="part100:AbstractPayloadType">
        <xs:sequence>
          <xs:element name="ObjectIdentificationRevisionSet"
type="m:ObjectIdentificationRevisionSet" minOccurs="1"
            maxOccurs="unbounded" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType xmlns="" name="Identifier"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject">
    <xs:annotation>
      <xs:documentation>This is a root class to provide common identification
for all classes needing identification and
        naming attributes.</xs:documentation>
      <xs:documentation />
    </xs:annotation>
    <xs:sequence>
      <xs:element name="mRID" minOccurs="0" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.mRID">
        <xs:annotation>
          <xs:documentation>Master resource identifier issued by a model
authority. The mRID is unique within an exchange
            context. Global uniqueness is easily achieved by using a UUID, as
specified in RFC 4122, for the mRID. The use of
            UUID is strongly recommended.</xs:documentation>
          <xs:documentation>For CIMXML data files in RDF syntax conforming to
IEC 61970-552, the mRID is mapped to rdf:ID or
            rdf:about attributes that identify CIM object
elements.</xs:documentation>
        </xs:annotation>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      </xs:element>
      <xs:element name="Names" minOccurs="0" maxOccurs="unbounded"
type="m:Name"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.Names">
        <xs:annotation>
          <xs:documentation>All names of this identified
object.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="Name"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#Name">
    <xs:annotation>
      <xs:documentation>The Name class provides the means to define any number
of human readable names for an object. A
        name is &lt;b&gt;not&lt;/b&gt; to be used for defining inter-object
relationships. For inter-object relationships
        instead use the object identification 'mRID'.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="name" minOccurs="1" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#Name.name">
        <xs:annotation>
          <xs:documentation>Any free text that name the
object.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="NameType" minOccurs="1" maxOccurs="1"
type="m:NameType"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#Name.NameTyp
e">
        <xs:annotation>
          <xs:documentation>Type of this name.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="NameType"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameType">
    <xs:annotation>
      <xs:documentation>Type of name. Possible values for attribute 'name' are
implementation dependent but standard
        profiles may specify types. An enterprise may have multiple IT systems
each having its own local name for the same
        object, e.g. a planning system may have different names from an EMS.
An object may also have different names within
        the same IT system, e.g. localName as defined in CIM version 14. The
definition from CIM14 is:</xs:documentation>
      <xs:documentation>The localName is a human readable name of the object.
It is a free text name local to a node in a
        naming hierarchy similar to a file directory structure. A power system
related naming hierarchy may be: Substation,
        VoltageLevel, Equipment etc. Children of the same parent in such a
hierarchy have names that typically are unique
        among them.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="description" minOccurs="0" maxOccurs="1"
type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameType.des
cription">
        <xs:annotation>
          <xs:documentation>Description of the name type.</xs:documentation>
        </xs:annotation>
      </xs:element>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="name" minOccurs="1" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameType.nam
e">
        <xs:annotation>
          <xs:documentation>Name of the name type.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="NameTypeAuthority" minOccurs="0" maxOccurs="1"
type="m:NameTypeAuthority"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameType.Nam
eTypeAuthority">
        <xs:annotation>
          <xs:documentation>Authority responsible for managing names of this
type.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="NameTypeAuthority"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameTypeAuth
ority">
    <xs:annotation>
      <xs:documentation>Authority responsible for creation and management of
names of a given name type; typically an
        organization or an enterprise system.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="description" minOccurs="0" maxOccurs="1"
type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameTypeAuth
ority.description">
        <xs:annotation>
          <xs:documentation>Description of the name type
authority.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="name" minOccurs="1" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#NameTypeAuth
ority.name">
        <xs:annotation />
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="Object"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject">
    <xs:annotation>
      <xs:documentation>This is a root class to provide common identification
for all classes needing identification and
        naming attributes.</xs:documentation>
      <xs:documentation />
    </xs:annotation>
    <xs:sequence>
      <xs:element name="mRID" minOccurs="0" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.mRID">
        <xs:annotation>
          <xs:documentation>Master resource identifier issued by a model
authority. The mRID is unique within an exchange
            context. Global uniqueness is easily achieved by using a UUID, as
specified in RFC 4122, for the mRID. The use of
            UUID is strongly recommended.</xs:documentation>
          <xs:documentation>For CIMXML data files in RDF syntax conforming to
IEC 61970-552, the mRID is mapped to rdf:ID or
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            rdf:about attributes that identify CIM object
elements.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Names" minOccurs="0" maxOccurs="1" type="m:Name"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.Names">
        <xs:annotation>
          <xs:documentation>All names of this identified
object.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ObjectType" minOccurs="0" maxOccurs="1"
type="m:ObjectType"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.ObjectType">
        <xs:annotation>
          <xs:documentation>The type of the IdentifiedObject, specified as the
name of a class that inherits from
            IdentifiedObject. The type is specified by setting the
ObjectType.Name.name to the name of the inherited class
            (e.g., Customer, Switch, Meter, etc.).</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="ObjectIdentificationRevision"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevision">
    <xs:annotation>
      <xs:documentation>Each instance of an ObjectIdentificationRevision
identifies:</xs:documentation>
      <xs:documentation>1. The IdentifiedObject for which one or more
identifiers are being added, removed, or modified.
        The IdentifiedObject is specified by the association to
IdentifiedObject with the target role name of 'Object'.
      </xs:documentation>
      <xs:documentation>2. One or more identifiers that are being added,
removed or modified. These identifiers may be an
        mRID or an instance of Name. The identifiers are specified by the
association to IdentifiedObject with the target
        role name of 'Identifier'.</xs:documentation>
      <xs:documentation>3. The type of revision (add, remove, or modify). The
type of revision is specified by the
        ObjectIdentificationRevision.ObjectIdentificationRevisionKind, with
enumerated choices of 'add', 'remove', or 'modify'.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="revisionKind" minOccurs="1" maxOccurs="1"
type="m:RevisionKind"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevision.revisionKind">
        <xs:annotation />
      </xs:element>
      <xs:element name="sequenceNumber" minOccurs="0" maxOccurs="1"
type="xs:integer"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevision.sequenceNumber">
        <xs:annotation />
      </xs:element>
      <xs:element name="Identifier" minOccurs="1" maxOccurs="1"
type="m:Identifier"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevision.Identifier">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        <xs:annotation>
          <xs:documentation>The identifier that is being added to, removed
from, or modified for an IdentifiedObject. The
            identifier may be an mRID or an instance of
Name.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="ObjectIdentificationRevisionSet"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevisionSet">
    <xs:annotation />
    <xs:sequence>
      <xs:element name="mRID" minOccurs="0" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.mRID">
        <xs:annotation>
          <xs:documentation>Master resource identifier issued by a model
authority. The mRID is unique within an exchange
            context. Global uniqueness is easily achieved by using a UUID, as
specified in RFC 4122, for the mRID. The use of
            UUID is strongly recommended.</xs:documentation>
          <xs:documentation>For CIMXML data files in RDF syntax conforming to
IEC 61970-552, the mRID is mapped to rdf:ID or
            rdf:about attributes that identify CIM object
elements.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="description" minOccurs="0" maxOccurs="1"
type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.description">
        <xs:annotation>
          <xs:documentation>The description is a free human readable text
describing or naming the object. It may be non
            unique and may not correlate to a naming
hierarchy.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="effectiveDateTime" minOccurs="0" maxOccurs="1"
type="xs:dateTime"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevisionSet.effectiveDateTime">
        <xs:annotation />
      </xs:element>
      <xs:element name="Names" minOccurs="0" maxOccurs="unbounded"
type="m:Name"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#IdentifiedOb
ject.Names">
        <xs:annotation>
          <xs:documentation>All names of this identified
object.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Object" minOccurs="0" maxOccurs="1" type="m:Object"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevisionSet.Object">
        <xs:annotation>
          <xs:documentation>The IdentifiedObject for which a revision to its
set of identifiers is being made. The Object is
            specified either by the mRID or a Name that exists prior to the
revision.</xs:documentation>
        </xs:annotation>
      </xs:element>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="ObjectIdentificationRevision" minOccurs="0"
maxOccurs="unbounded" type="m:ObjectIdentificationRevision"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectIdenti
ficationRevisionSet.ObjectIdentificationRevision">
        <xs:annotation />
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType xmlns="" name="ObjectType"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectType">
    <xs:annotation>
      <xs:documentation>Identifies the specific type of the object that is
generically identified as an IdentifiedObject.
        This type will always be the name of a CIM class that inherits from
IdentifiedObject and is specified by setting the
        ObjectType.type to the name of the class (e.g., Customer, Switch,
Meter, etc.).</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="type" minOccurs="1" maxOccurs="1" type="xs:string"

sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#ObjectType.t
ype">
        <xs:annotation />
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType xmlns="" name="RevisionKind"
sawsdl:modelReference="http://iec.ch/TC57/CIM_17v39_13v13a_03v17a#RevisionKind
">
    <xs:annotation />
    <xs:restriction base="xs:string">
      <xs:enumeration value="add">
        <xs:annotation />
      </xs:enumeration>
      <xs:enumeration value="remove">
        <xs:annotation />
      </xs:enumeration>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Figure A.3 – Contenu du fichier LegacyObjectIdentificationRevisionSets.xsd

A.5 Fichier `RDF.xsd`

Le contenu du fichier `RDF.xsd` est représenté à la Figure A.4. Ce fichier est adapté à la transmission de données RDF (8.2.4). Il peut également être utilisé pour la transmission de données XML qui ne correspondent à aucune définition de schéma XML.

Les données qui correspondent à ce profil peuvent donc apparaître dans la section `AbstractPayload` d'un élément `<Payload>` (7.7.5).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:a="http://langdale.com.au/2005/Message#"
xmlns:sawsdl="http://www.w3.org/ns/sawsdl"
xmlns="http://langdale.com.au/2005/Message#"
xmlns:m="http://iec.ch/TC57/2011/RDFData#"
xmlns:part100="http://iec.ch/TC57/2021/schema/message"
xmlns:ns1="http://iec.ch/TC57/2021/schema/message"
targetNamespace="http://iec.ch/TC57/2011/RDFData#"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation>
        This is the RDF.xsd file which forms an integral part of the
International Standard
        IEC 61968-100:2021,
        "IEC IMPLEMENTATION PROFILES FOR APPLICATION INTEGRATION".
        IEC 61968-100:2021 is available from the International
Electrotechnical Commission at http://www.iec.ch
        and contains a full description of the contents of this file.
    </xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
schemaLocation="../XSDs/AbstractPayloadType.xsd"/>
  <xs:element name="RDFData" type="m:RDFData"
substitutionGroup="part100:AbstractPayload"/>
  <xs:complexType name="RDFData">
    <xs:complexContent>
      <xs:extension base="part100:AbstractPayloadType">
        <xs:sequence>
          <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

Figure A.4 – Contenu du fichier RDF.xsd

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annexe B
(normative)

Modèles de fichier

B.1 Généralités

Les modèles de fichier qui figurent dans la présente annexe font partie du présent document. Ces fichiers doivent être modifiés à la main pour générer les fichiers WSDL correspondants (9.2.5).

B.2 Fichier GenericMessageWsdlTemplate.txt

Le contenu du fichier `GenericMessageWsdlTemplate.txt` (9.2.5.2) est représenté à la Figure B.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  This is the GenericMessageXsdTemplate.txt file which forms an integral part
  of the International Standard  IEC 61968-100:2021,
 "IEC IMPLEMENTATION PROFILES FOR APPLICATION INTEGRATION".
  IEC 61968-100:2021 is available from the International Electrotechnical
  Commission at http://www.iec.ch
  and contains a full description of the contents of this file.

  This template file may be used to generate a generic WSDL file by globally
  substitutingthe following variable names:

  Variable Name          Example Value
  ===================================
  {PATHPREFIX}           ../XSDs
  {SOAPADDRESS}          http://example.com
-->
<wsdl:definitions
    xmlns:tns="http://iec.ch/TC57/2021/wsdl/IEC61968-100_Ed2_Generic"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:msg="http://iec.ch/TC57/2021/schema/message"
    name="IEC61968-100_Ed2_Generic"
    targetNamespace="http://iec.ch/TC57/2021/wsdl/IEC61968-100_Ed2_Generic">
  <wsdl:types>
    <xs:schema targetNamespace="http://iec.ch/TC57/2021/wsdl/IEC61968-
100_Ed2_Generic" elementFormDefault="qualified">
      <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
schemaLocation="{PATHPREFIX}/Message.xsd"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="CancelRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:CancelRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="CancelResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:CancelResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ChangeRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:ChangeRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="ChangeResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:ChangeResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="CreateRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:CreateRequestMessage"/>
  </wsdl:message>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
  <wsdl:message name="CreateResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:CreateResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="DeleteRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:DeleteRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="DeleteResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:DeleteResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecuteRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:ExecuteRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecuteResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:ExecuteResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="GetRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:GetRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="GetResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:GetResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="UpdateRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:UpdateRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="UpdateResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:UpdateResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="CanceledEventMessage">
    <wsdl:part name="EventMessage" element="msg:CanceledEventMessage"/>
  </wsdl:message>
  <wsdl:message name="CanceledEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:CanceledEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ChangedEventMessage">
    <wsdl:part name="EventMessage" element="msg:ChangedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="ChangedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:ChangedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="CreatedEventMessage">
    <wsdl:part name="EventMessage" element="msg:CreatedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="CreatedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:CreatedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="DeletedEventMessage">
    <wsdl:part name="EventMessage" element="msg:DeletedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="DeletedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:DeletedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecutedEventMessage">
    <wsdl:part name="EventMessage" element="msg:ExecutedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecutedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:ExecutedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="FaultMessage">
    <wsdl:part name="FaultMessage" element="msg:FaultMessage"/>
  </wsdl:message>
  <wsdl:message name="UpdatedEventMessage">
    <wsdl:part name="EventMessage" element="msg:UpdatedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="UpdatedEventResponseMessage">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <wsdl:part name="ResponseMessage"
element="msg:UpdatedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:portType name="IEC61968-100_Ed2_Generic_Port">
    <wsdl:operation name="Cancel">
      <wsdl:input name="CancelInput" message="tns:CancelRequestMessage"/>
      <wsdl:output name="CancelOutput" message="tns:CancelResponseMessage"/>
      <wsdl:fault name="CancelFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Change">
      <wsdl:input name="ChangeInput" message="tns:ChangeRequestMessage"/>
      <wsdl:output name="ChangeOutput" message="tns:ChangeResponseMessage"/>
      <wsdl:fault name="ChangeFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Create">
      <wsdl:input name="CreateInput" message="tns:CreateRequestMessage"/>
      <wsdl:output name="CreateOutput" message="tns:CreateResponseMessage"/>
      <wsdl:fault name="CreateFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Delete">
      <wsdl:input name="DeleteInput" message="tns:DeleteRequestMessage"/>
      <wsdl:output name="DeleteOutput" message="tns:DeleteResponseMessage"/>
      <wsdl:fault name="DeleteFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Execute">
      <wsdl:input name="ExecuteInput" message="tns:ExecuteRequestMessage"/>
      <wsdl:output name="ExecuteOutput" message="tns:ExecuteResponseMessage"/>
      <wsdl:fault name="ExecuteFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Get">
      <wsdl:input name="GetInput" message="tns:GetRequestMessage"/>
      <wsdl:output name="GetOutput" message="tns:GetResponseMessage"/>
      <wsdl:fault name="GetFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Update">
      <wsdl:input name="UpdateInput" message="tns:UpdateRequestMessage"/>
      <wsdl:output name="UpdateOutput" message="tns:UpdateResponseMessage"/>
      <wsdl:fault name="UpdateFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Canceled">
      <wsdl:input name="CanceledInput" message="tns:CanceledEventMessage"/>
      <wsdl:output name="CanceledOutput"
message="tns:CanceledEventResponseMessage"/>
      <wsdl:fault name="CanceledFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Changed">
      <wsdl:input name="ChangedInput" message="tns:ChangedEventMessage"/>
      <wsdl:output name="ChangedOutput"
message="tns:ChangedEventResponseMessage"/>
      <wsdl:fault name="ChangedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Created">
      <wsdl:input name="CreatedInput" message="tns:CreatedEventMessage"/>
      <wsdl:output name="CreatedOutput"
message="tns:CreatedEventResponseMessage"/>
      <wsdl:fault name="CreatedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Deleted">
      <wsdl:input name="DeletedInput" message="tns:DeletedEventMessage"/>
      <wsdl:output name="DeletedOutput"
message="tns:DeletedEventResponseMessage"/>
      <wsdl:fault name="DeletedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Executed">
      <wsdl:input name="ExecutedInput" message="tns:ExecutedEventMessage"/>
      <wsdl:output name="ExecutedOutput"
message="tns:ExecutedEventResponseMessage"/>
      <wsdl:fault name="ExecutedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Updated">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <wsdl:input name="UpdatedInput" message="tns:UpdatedEventMessage"/>
      <wsdl:output name="UpdatedOutput"
message="tns:UpdatedEventResponseMessage"/>
      <wsdl:fault name="UpdatedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="IEC61968-100_Ed2_Generic_Binding" type="tns:IEC61968-
100_Ed2_Generic_Port">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Cancel">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Cancel" style="document"/>
      <wsdl:input name="CancelInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CancelOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CancelFault">
        <soap:fault name="CancelFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Change">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Change" style="document"/>
      <wsdl:input name="ChangeInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ChangeOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ChangeFault">
        <soap:fault name="ChangeFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Create">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Create" style="document"/>
      <wsdl:input name="CreateInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CreateOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CreateFault">
        <soap:fault name="CreateFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Delete">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Delete" style="document"/>
      <wsdl:input name="DeleteInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="DeleteOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="DeleteFault">
        <soap:fault name="DeleteFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Execute">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Execute" style="document"/>
      <wsdl:input name="ExecuteInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ExecuteOutput">
        <soap:body use="literal"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      </wsdl:output>
      <wsdl:fault name="ExecuteFault">
        <soap:fault name="ExecuteFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Get">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Get" style="document"/>
      <wsdl:input name="GetInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="GetOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="GetFault">
        <soap:fault name="GetFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Update">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Update" style="document"/>
      <wsdl:input name="UpdateInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="UpdateOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="UpdateFault">
        <soap:fault name="UpdateFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Canceled">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Canceled" style="document"/>
      <wsdl:input name="CanceledInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CanceledOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CanceledFault">
        <soap:fault name="CanceledFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Changed">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Changed" style="document"/>
      <wsdl:input name="ChangedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ChangedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ChangedFault">
        <soap:fault name="ChangedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Created">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Created" style="document"/>
      <wsdl:input name="CreatedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CreatedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CreatedFault">
        <soap:fault name="CreatedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <wsdl:operation name="Deleted">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Deleted" style="document"/>
      <wsdl:input name="DeletedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="DeletedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="DeletedFault">
        <soap:fault name="DeletedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Executed">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Executed" style="document"/>
      <wsdl:input name="ExecutedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ExecutedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ExecutedFault">
        <soap:fault name="ExecutedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Updated">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Updated" style="document"/>
      <wsdl:input name="UpdatedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="UpdatedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="UpdatedFault">
        <soap:fault name="UpdatedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="IEC61968-100_Ed2_Generic">
    <wsdl:port name="IEC61968-100_Ed2_Generic_Port" binding="tns:IEC61968-
100_Ed2_Generic_Binding">
      <!-- The SOAP address below must be changed for a particular
implementation. -->
      <soap:address location="{SOAPADDRESS}"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Figure B.1 – Contenu du fichier GenericMessageXsdTemplate.txt

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

B.3 Fichier `StrongMessageXsdTemplate.txt`

Le contenu du fichier `StrongMessageXsdTemplate.txt` (9.2.5.3) est représenté à la Figure B.2.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  This is the StrongMessageXsdTemplate.txt file which forms an integral part
of the International Standard
  IEC 61968-100:2021, "IEC IMPLEMENTATION PROFILES FOR APPLICATION
INTEGRATION".
  IEC 61968-100:2021 is available from the International Electrotechnical
Commission at http://www.iec.ch
  and contains a full description of the contents of this file.

  This template file may be used to generate a strongly-typed message XSD file
by globally substituting
  the following variable names:

  Variable Name          Example Value
  ====================================
  {GETPROFILENAMESPACE}  http://iec.ch/TC57/2011/GetMeterReadings#
  {GETPROFILENAME}       GetMeterReadings
  {PROFILENAMESPACE}     http://iec.ch/TC57/2011/MeterReadings#
  {PROFILENAME}          MeterReadings
  {PROFILEPATHPREFIX}    ../Part9XSDs
  {PART100PATHPREFIX}    ../XSDs
-->
<xs:schema xmlns="http://iec.ch/TC57/2021/schema/message"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:obj1="{PROFILENAMESPACE}"
    xmlns:obj2="{GETPROFILENAMESPACE}"
    targetNamespace="http://iec.ch/TC57/2021/schema/message"
    elementFormDefault="qualified" attributeFormDefault="unqualified"
version="1.0.0">
  <xs:import namespace="{PROFILENAMESPACE}"
schemaLocation="{PROFILEPATHPREFIX}/{PROFILENAME}.xsd"/>
  <xs:import namespace="{GETPROFILENAMESPACE}"
schemaLocation="{PROFILEPATHPREFIX}/{GETPROFILENAME}.xsd"/>
  <xs:include schemaLocation="{PART100PATHPREFIX}/AbstractPayloadType.xsd"/>
  <xs:element name="Canceled{PROFILENAME}EventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="canceled"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Canceled{PROFILENAME}EventResponseMessage"
type="SimpleAckReplyType"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
  <xs:element name="Cancel{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="cancel"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Cancel{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Changed{PROFILENAME}EventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="changed"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Changed{PROFILENAME}EventResponseMessage"
type="SimpleAckReplyType"/>
  <xs:element name="Change{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="change"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Change{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Created{PROFILENAME}EventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="created"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Created{PROFILENAME}EventResponseMessage"
type="SimpleAckReplyType"/>
  <xs:element name="Create{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="create"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Create{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Deleted{PROFILENAME}EventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="deleted"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Deleted{PROFILENAME}EventResponseMessage"
type="SimpleAckReplyType"/>
  <xs:element name="Delete{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="delete"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Delete{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecutedEventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="executed"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet" type="EventOperationSetType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecutedEventResponseMessage" type="SimpleAckReplyType"/>
  <xs:element name="ExecuteRequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="execute"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet" type="RequestOperationSetType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExecuteResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
              <xs:element name="Noun" type="xs:string" fixed="OperationSet"/>
              <xs:group ref="ResponseHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Reply" type="ReplyType"/>
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="OperationSet"
type="ResponseOperationSetType" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="FaultMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Get{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="get"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="obj2:{GETPROFILENAME}"
maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Get{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
        <xs:choice minOccurs="0">
          <xs:element name="Payload">
            <xs:complexType>
              <xs:choice>
                <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
                <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Updated{PROFILENAME}EventMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="EventVerbType" fixed="updated"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="EventHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Updated{PROFILENAME}EventResponseMessage"
type="SimpleAckReplyType"/>
  <xs:element name="Update{PROFILENAME}RequestMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verb" type="RequestVerbType" fixed="update"/>
              <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
              <xs:group ref="RequestHeaderGroup"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="Payload">
            <xs:complexType>
              <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
                <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="CompressedPayload" type="CompressedPayloadType"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Update{PROFILENAME}ResponseMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header" type="ResponseHeaderType"/>
        <xs:element name="Reply" type="ReplyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="CompressedPayloadType">
    <xs:sequence>
      <xs:element name="CompressionKind">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="zip"/>
            <xs:enumeration value="gzip"/>
            <xs:enumeration value="other"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Base64" type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="EventVerbType"/>
      <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
      <xs:group ref="EventHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="EnforceMsgSequence" type="xs:boolean" minOccurs="0"/>
      <xs:element name="EnforceTransactionalIntegrity" type="xs:boolean"
minOccurs="0"/>
      <xs:element name="Operation" type="EventOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EventOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="EventVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
        <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
        <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="IdentifierType">
    <xs:sequence>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <xs:element name="mRID" type="UUIDType" minOccurs="0"/>
      <xs:element name="Names" type="Name" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="ObjectType" type="ObjectType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ObjectType">
    <xs:sequence>
      <xs:element name="type" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MessagePropertyType">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Value" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MessageSequenceType">
    <xs:sequence>
      <xs:element name="SequenceNumber" type="xs:unsignedInt" minOccurs="0"/>
      <xs:element name="LastMessage" type="xs:boolean" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Name">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="NameType" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="NameTypeAuthority" minOccurs="0">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="name" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ObjectUpdateType">
    <xs:sequence>
      <xs:element name="UpdateID" type="UUIDType"/>
      <xs:element name="UpdateAction">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="add"/>
            <xs:enumeration value="modify"/>
            <xs:enumeration value="remove"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element ref="AbstractPayload"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ReplyType">
    <xs:sequence>
      <xs:element name="Status" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="StatusType"/>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="Transaction" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="TransactionType"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="RequestVerbType"/>
      <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
      <xs:group ref="RequestHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="EnforceMsgSequence" type="xs:boolean" minOccurs="0"/>
      <xs:element name="EnforceTransactionalIntegrity" type="xs:boolean"
minOccurs="0"/>
      <xs:element name="Operation" type="RequestOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequestOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="RequestVerbType"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="Transaction" type="TransactionType"
maxOccurs="unbounded"/>
        <xs:element name="ObjectUpdate" type="ObjectUpdateType"
maxOccurs="unbounded"/>
        <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseHeaderType">
    <xs:sequence>
      <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
      <xs:element name="Noun" type="xs:string" fixed="{PROFILENAME}"/>
      <xs:group ref="ResponseHeaderGroup"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseOperationSetType">
    <xs:sequence>
      <xs:element name="OperationSetID" type="UUIDType"/>
      <xs:element name="Operation" type="ResponseOperationType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResponseOperationType">
    <xs:sequence>
      <xs:element name="OperationID" type="xs:unsignedInt"/>
      <xs:element name="Verb" type="ResponseVerbType" fixed="reply"/>
      <xs:element name="Noun" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element ref="obj1:{PROFILENAME}" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
  <xs:complexType name="SimpleAckReplyType">
    <xs:sequence>
      <xs:element name="Header" type="ResponseHeaderType"/>
      <xs:element name="Reply">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Status">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Level" type="xs:string" fixed="OK"/>
                  <xs:element name="Code" type="xs:string" fixed="0.3"/>
                  <xs:element name="Reason" type="xs:string" minOccurs="0"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="StatusType">
    <xs:sequence>
      <xs:element name="Level">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="OK"/>
            <xs:enumeration value="WARNING"/>
            <xs:enumeration value="ERROR"/>
            <xs:enumeration value="UNKNOWN"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Code" type="xs:string"/>
      <xs:element name="Namespace" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="http://.+"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Reason" type="xs:string" minOccurs="0"/>
      <xs:element name="ObjectID" type="IdentifierType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="OperationSet" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="OperationSetID" type="UUIDType"/>
            <xs:element name="Operation" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="OperationID" type="xs:unsignedInt"/>
                  <xs:element name="ObjectID" type="IdentifierType"
minOccurs="0" maxOccurs="unbounded"/>
                  <xs:element name="UpdateID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element name="TransactionID" type="UUIDType"
minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="UpdateID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="TransactionID" type="UUIDType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="TransactionType">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <xs:sequence>
      <xs:element name="TransactionID" type="UUIDType"/>
      <xs:choice minOccurs="0">
        <xs:element name="ObjectID" type="IdentifierType"
maxOccurs="unbounded"/>
        <xs:element name="OperationSet" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence minOccurs="0">
              <xs:element name="OperationSetID" type="UUIDType"/>
              <xs:element name="OperationID" type="xs:unsignedInt"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UserType">
    <xs:sequence>
      <xs:element name="UserID" type="xs:string"/>
      <xs:element name="Organization" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="EventVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="canceled"/>
      <xs:enumeration value="changed"/>
      <xs:enumeration value="created"/>
      <xs:enumeration value="deleted"/>
      <xs:enumeration value="executed"/>
      <xs:enumeration value="updated"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="RequestVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="cancel"/>
      <xs:enumeration value="change"/>
      <xs:enumeration value="create"/>
      <xs:enumeration value="delete"/>
      <xs:enumeration value="get"/>
      <xs:enumeration value="execute"/>
      <xs:enumeration value="update"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ResponseVerbType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="reply"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="UUIDType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Fa-f0-9]{8}-[A-Fa-f0-9]{4}-[A-Fa-f0-9]{4}-[A-Fa-
f0-9]{4}-[A-Fa-f0-9]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:group name="EventHeaderGroup">
    <xs:sequence>
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="MessageSequence" type="MessageSequenceType"
minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="RequestHeaderGroup">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <xs:sequence>
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="ReplyAddress" type="xs:string" minOccurs="0"/>
      <xs:element name="AckRequired" type="xs:boolean" minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="ResponseHeaderGroup">
    <xs:sequence>
      <xs:element name="Timestamp" type="xs:dateTime"/>
      <xs:element name="MessageID" type="UUIDType"/>
      <xs:element name="CorrelationID" type="UUIDType"/>
      <xs:element name="MessageSequence" type="MessageSequenceType"
minOccurs="0"/>
      <xs:element name="Source" type="xs:string" minOccurs="0"/>
      <xs:element name="User" type="UserType" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="Property" type="MessagePropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
</xs:schema>
```

Figure B.2 – Contenu du fichier StrongMessageXsdTemplate.txt

B.4 Fichier `StrongMessageWsdlTemplate.txt`

Le contenu du fichier `StrongMessageWsdlTemplate.txt` (9.2.5.3) est représenté à la Figure B.3.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  This is the StrongMessageWsdlTemplate.txt file which forms an integral part
of the International Standard IEC 61968-100:2021,
"IEC IMPLEMENTATION PROFILES FOR APPLICATION INTEGRATION".
  IEC 61968-100:2021 is available from the International Electrotechnical
Commission at http://www.iec.ch
  and contains a full description of the contents of this file.

  This template file may be used to generate a strongly-typed WSDL file by
globally substituting
  the following variable names:

  Variable Name          Example Value
  ====================================
  {GETPROFILENAME}       GetMeterReadings
  {PROFILENAME}          MeterReadings
  {PATHPREFIX}           ../Templates
  {SOAPADDRESS}          http://example.com
-->
<wsdl:definitions xmlns:tns="http://iec.ch/TC57/2021/wsdl/IEC61968-
100_Ed2_Strong/{PROFILENAME}"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:msg="http://iec.ch/TC57/2021/schema/message"
    name="IEC61968-100_Ed2_Strong"
    targetNamespace="http://iec.ch/TC57/2021/wsdl/IEC61968-
100_Ed2_Strong/{PROFILENAME}">
  <wsdl:types>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <xs:schema targetNamespace="http://iec.ch/TC57/2021/wsdl/{PROFILENAME}"
elementFormDefault="qualified">
      <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
schemaLocation="{PATHPREFIX}/{PROFILENAME}Message.xsd"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="Cancel{PROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:Cancel{PROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="Cancel{PROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Cancel{PROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Change{PROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:Change{PROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="Change{PROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Change{PROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Create{PROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:Create{PROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="Create{PROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Create{PROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Delete{PROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:Delete{PROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="Delete{PROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Delete{PROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecuteRequestMessage">
    <wsdl:part name="RequestMessage" element="msg:ExecuteRequestMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecuteResponseMessage">
    <wsdl:part name="ResponseMessage" element="msg:ExecuteResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="{GETPROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:{GETPROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="{GETPROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:{GETPROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Update{PROFILENAME}RequestMessage">
    <wsdl:part name="{PROFILENAME}RequestMessage"
element="msg:Update{PROFILENAME}RequestMessage"/>
  </wsdl:message>
  <wsdl:message name="Update{PROFILENAME}ResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Update{PROFILENAME}ResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Canceled{PROFILENAME}EventMessage">
    <wsdl:part name="{PROFILENAME}EventMessage"
element="msg:Canceled{PROFILENAME}EventMessage"/>
  </wsdl:message>
  <wsdl:message name="Canceled{PROFILENAME}EventResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Canceled{PROFILENAME}EventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Changed{PROFILENAME}EventMessage">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <wsdl:part name="{PROFILENAME}EventMessage"
element="msg:Changed{PROFILENAME}EventMessage"/>
  </wsdl:message>
  <wsdl:message name="Changed{PROFILENAME}EventResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Changed{PROFILENAME}EventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Created{PROFILENAME}EventMessage">
    <wsdl:part name="{PROFILENAME}EventMessage"
element="msg:Created{PROFILENAME}EventMessage"/>
  </wsdl:message>
  <wsdl:message name="Created{PROFILENAME}EventResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Created{PROFILENAME}EventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Deleted{PROFILENAME}EventMessage">
    <wsdl:part name="{PROFILENAME}EventMessage"
element="msg:Deleted{PROFILENAME}EventMessage"/>
  </wsdl:message>
  <wsdl:message name="Deleted{PROFILENAME}EventResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Deleted{PROFILENAME}EventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecutedEventMessage">
    <wsdl:part name="EventMessage" element="msg:ExecutedEventMessage"/>
  </wsdl:message>
  <wsdl:message name="ExecutedEventResponseMessage">
    <wsdl:part name="ResponseMessage"
element="msg:ExecutedEventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="Updated{PROFILENAME}EventMessage">
    <wsdl:part name="{PROFILENAME}EventMessage"
element="msg:Updated{PROFILENAME}EventMessage"/>
  </wsdl:message>
  <wsdl:message name="Updated{PROFILENAME}EventResponseMessage">
    <wsdl:part name="{PROFILENAME}ResponseMessage"
element="msg:Updated{PROFILENAME}EventResponseMessage"/>
  </wsdl:message>
  <wsdl:message name="FaultMessage">
    <wsdl:part name="FaultMessage" element="msg:FaultMessage"/>
  </wsdl:message>
  <wsdl:portType name="IEC61968-100_Ed2_Strong_Port">
    <wsdl:operation name="Cancel">
      <wsdl:input name="CancelInput"
message="tns:Cancel{PROFILENAME}RequestMessage"/>
      <wsdl:output name="CancelOutput"
message="tns:Cancel{PROFILENAME}ResponseMessage"/>
      <wsdl:fault name="CancelFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Change">
      <wsdl:input name="ChangeInput"
message="tns:Change{PROFILENAME}RequestMessage"/>
      <wsdl:output name="ChangeOutput"
message="tns:Change{PROFILENAME}ResponseMessage"/>
      <wsdl:fault name="ChangeFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Create">
      <wsdl:input name="CreateInput"
message="tns:Create{PROFILENAME}RequestMessage"/>
      <wsdl:output name="CreateOutput"
message="tns:Create{PROFILENAME}ResponseMessage"/>
      <wsdl:fault name="CreateFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Delete">
      <wsdl:input name="DeleteInput"
message="tns:Delete{PROFILENAME}RequestMessage"/>
      <wsdl:output name="DeleteOutput"
message="tns:Delete{PROFILENAME}ResponseMessage"/>
      <wsdl:fault name="DeleteFault" message="tns:FaultMessage"/>
    </wsdl:operation>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
    <wsdl:operation name="Execute">
      <wsdl:input name="ExecuteInput" message="tns:ExecuteRequestMessage"/>
      <wsdl:output name="ExecuteOutput" message="tns:ExecuteResponseMessage"/>
      <wsdl:fault name="ExecuteFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Get">
      <wsdl:input name="GetInput"
message="tns:{GETPROFILENAME}RequestMessage"/>
      <wsdl:output name="GetOutput"
message="tns:{GETPROFILENAME}ResponseMessage"/>
      <wsdl:fault name="GetFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Update">
      <wsdl:input name="UpdateInput"
message="tns:Update{PROFILENAME}RequestMessage"/>
      <wsdl:output name="UpdateOutput"
message="tns:Update{PROFILENAME}ResponseMessage"/>
      <wsdl:fault name="UpdateFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Created">
      <wsdl:input name="CreatedInput"
message="tns:Created{PROFILENAME}EventMessage"/>
      <wsdl:output name="CreatedOutput"
message="tns:Created{PROFILENAME}EventResponseMessage"/>
      <wsdl:fault name="CreatedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Changed">
      <wsdl:input name="ChangedInput"
message="tns:Changed{PROFILENAME}EventMessage"/>
      <wsdl:output name="ChangedOutput"
message="tns:Changed{PROFILENAME}EventResponseMessage"/>
      <wsdl:fault name="ChangedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Canceled">
      <wsdl:input name="CanceledInput"
message="tns:Canceled{PROFILENAME}EventMessage"/>
      <wsdl:output name="CanceledOutput"
message="tns:Canceled{PROFILENAME}EventResponseMessage"/>
      <wsdl:fault name="CanceledFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Deleted">
      <wsdl:input name="DeletedInput"
message="tns:Deleted{PROFILENAME}EventMessage"/>
      <wsdl:output name="DeletedOutput"
message="tns:Deleted{PROFILENAME}EventResponseMessage"/>
      <wsdl:fault name="DeletedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Executed">
      <wsdl:input name="ExecutedInput" message="tns:ExecutedEventMessage"/>
      <wsdl:output name="ExecutedOutput"
message="tns:ExecutedEventResponseMessage"/>
      <wsdl:fault name="ExecutedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Updated">
      <wsdl:input name="UpdatedInput"
message="tns:Updated{PROFILENAME}EventMessage"/>
      <wsdl:output name="UpdatedOutput"
message="tns:Updated{PROFILENAME}EventResponseMessage"/>
      <wsdl:fault name="UpdatedFault" message="tns:FaultMessage"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="IEC61968-100_Ed2_Strong_Binding" type="tns:IEC61968-
100_Ed2_Strong_Port">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Cancel">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Cancel" style="document"/>
      <wsdl:input name="CancelInput">
        <soap:body use="literal"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      </wsdl:input>
      <wsdl:output name="CancelOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CancelFault">
        <soap:fault name="CancelFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Change">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Change" style="document"/>
      <wsdl:input name="ChangeInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ChangeOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ChangeFault">
        <soap:fault name="ChangeFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Create">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Create" style="document"/>
      <wsdl:input name="CreateInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CreateOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CreateFault">
        <soap:fault name="CreateFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Delete">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Delete" style="document"/>
      <wsdl:input name="DeleteInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="DeleteOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="DeleteFault">
        <soap:fault name="DeleteFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Execute">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Execute" style="document"/>
      <wsdl:input name="ExecuteInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ExecuteOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ExecuteFault">
        <soap:fault name="ExecuteFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Get">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Get" style="document"/>
      <wsdl:input name="GetInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="GetOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="GetFault">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        <soap:fault name="GetFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Update">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Update" style="document"/>
      <wsdl:input name="UpdateInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="UpdateOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="UpdateFault">
        <soap:fault name="UpdateFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Canceled">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Canceled" style="document"/>
      <wsdl:input name="CanceledInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CanceledOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CanceledFault">
        <soap:fault name="CanceledFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Changed">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Changed" style="document"/>
      <wsdl:input name="ChangedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ChangedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ChangedFault">
        <soap:fault name="ChangedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Created">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Created" style="document"/>
      <wsdl:input name="CreatedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="CreatedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="CreatedFault">
        <soap:fault name="CreatedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Deleted">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Deleted" style="document"/>
      <wsdl:input name="DeletedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="DeletedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="DeletedFault">
        <soap:fault name="DeletedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Executed">
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Generic/Executed" style="document"/>
      <wsdl:input name="ExecutedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ExecutedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ExecutedFault">
        <soap:fault name="ExecutedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Updated">
      <soap:operation soapAction="http://iec.ch/TC57/2021/IEC61968-
100_Ed2_Strong/Updated" style="document"/>
      <wsdl:input name="UpdatedInput">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="UpdatedOutput">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="UpdatedFault">
        <soap:fault name="UpdatedFault" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="IEC61968-100_Ed2_Strong">
    <wsdl:port name="IEC61968-100_Ed2_Strong_Port" binding="tns:IEC61968-
100_Ed2_Strong_Binding">
      <!-- The SOAP address below must be changed for a particular
implementation. -->
      <soap:address location="{SOAPADDRESS}"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Figure B.3 – Contenu du fichier StrongMessageWsdlTemplate.txt

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annexe C
(informative)

Exemple de profils XSD héritant d'AbstractPayloadType

C.1 Profil GetMeterReadings

Le profil représenté à la Figure C.1 est un extrait du profil GetMeterReadings.xsd défini dans l'IEC 61969-9:2013. Il a été modifié afin d'hériter du type XSD `AbstractPayloadType`. Les modifications exigées sont mises en évidence en gras.

Les données qui correspondent à ce profil peuvent donc apparaître dans la section `AbstractPayload` d'un élément `<Payload>` (7.7.5).

NOTE Le code source disponible à l'adresse https://github.com/TC57WG14/Part-100-Ed-2-XSD-Updater peut être utile pour procéder à la conversion automatique des fichiers de profil XSD existants pour le présent document.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:m="http://iec.ch/TC57/2011/GetMeterReadings#"
    xmlns:part100="http://iec.ch/TC57/2021/schema/message"
    xmlns:sawsdl="http://www.w3.org/ns/sawsdl"
    targetNamespace="http://iec.ch/TC57/2011/GetMeterReadings#"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
     schemaLocation="AbstractPayloadType.xsd"/>
  <xs:annotation>
      <xs:documentation/>
  </xs:annotation>
  <xs:element name="GetMeterReadings" type="m:GetMeterReadings"
       substitutionGroup="part100:AbstractPayload"/>
  <xs:complexType name="GetMeterReadings">
    <xs:complexContent>
      <xs:extension base="part100:AbstractPayloadType">
      <xs:sequence>
        <xs:element name="EndDevice" type="m:EndDevice" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="EndDeviceGroup" type="m:EndDeviceGroup"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="MeterReadings" type="m:MeterReadings" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="Reading" type="m:Reading" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="ReadingQuality" type="m:ReadingQuality"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ReadingType" type="m:ReadingType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="TimeSchedule" type="m:TimeSchedule" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="TransformerTank" type="m:TransformerTank"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="UsagePoint" type="m:UsagePoint" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="UsagePointGroup" type="m:UsagePointGroup"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType . . .>
  . . . . .
  </xs:complexType>
</xs:schema>
```

Figure C.1 – Exemple de profil XSD présentant l'héritage d'`AbstractPayloadType`

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

C.2 Profil BinaryData

Le profil XSD représenté à la Figure C.2 est adapté à la transmission de données binaires arbitraires. Les données elles-mêmes sont au format base64 (RFC 4648).

Les données qui correspondent à ce profil peuvent donc apparaître dans la section `AbstractPayload` d'un élément `<Payload>` (7.7.5).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
     xmlns:m="http://iec.ch/TC57/2011/JsonData#"
     xmlns:part100="http://iec.ch/TC57/2021/schema/message"
     targetNamespace="http://iec.ch/TC57/2021/schema/binary"
     elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://iec.ch/TC57/2021/schema/message"
     schemaLocation="AbstractPayloadType.xsd"/>
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:element name="Binary" type="m:Binary"
      substitutionGroup="part100:AbstractPayload"/>
  <xs:complexType name="Binary">
    <xs:complexContent>
      <xs:extension base="part100:AbstractPayloadType">
        <xs:sequence>
          <xs:element name="Base64" type="m:Base64" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="Base64">
    <xs:restriction base="xs:base64Binary"/>
  </xs:simpleType>
</xs:schema>
```

Figure C.2 – Exemple de profil XSD adapté à la transmission de données binaires

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Annexe D
(informative)

Exemple de messages IEC 61968-100:2021

D.1 Généralités

Les exemples de messages de cette annexe présentent l'utilisation de l'IEC 61968-100:2021 pour l'envoi et la réception des données spécifiques à l'application définies selon l'IEC 61968-9:2013. Des modèles et exemples de messages similaires peuvent également être appliqués à d'autres parties de l'IEC 61968.

Le transport HTTP qui utilise le fichier WSDL générique est par hypothèse utilisé tout au long de cette annexe (9.2.5.2). Certains messages font donc l'objet d'un message d'accusé de réception simple.

La plupart des exemples de la présente annexe utilisent des mRID pour l'identification d'objet. Voir l'Annexe F pour des exemples d'utilisation des éléments `<Name><name>`, `<Name><NameType><name>` et `<Name><NameType><NameTypeAuthority><name>` pour l'identification d'objet.

D.2 Messages relatifs à l'instanciation d'objet

D.2.1 Généralités

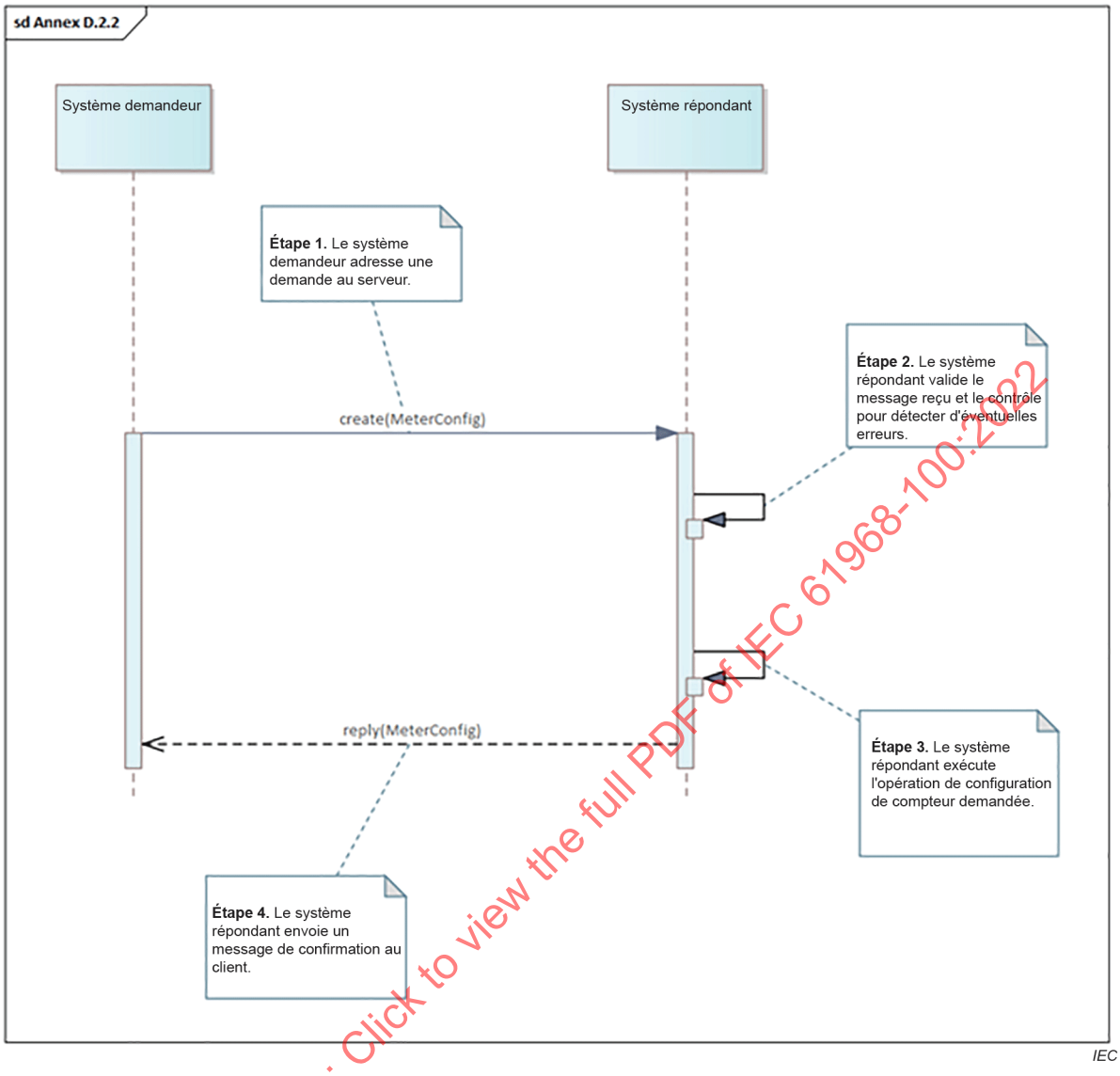
Cet exemple qui utilise le modèle d'échange de messages de demande/réponse présente l'instanciation d'un compteur simple sur un système répondant.

Le système demandeur envoie une demande "d'instanciation de compteur" au système répondant. Le système répondant tente alors d'ajouter le compteur donné avec toutes les informations détaillées fournies à sa base de données interne des compteurs connus. Une notification de succès ou d'échec de cette tentative est ensuite renvoyée au système demandeur.

D.2.2 Diagramme de séquence

Le diagramme de séquence pour cet exemple est représenté à la Figure D.1.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022



IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

Figure D.1 – Exemple de diagramme de séquence pour une demande d'instanciation de compteur

Une demande d'instanciation d'un objet compteur est formulée comme un message de demande IEC 61968-9:2013 create(MeterConfig) qui est envoyé directement par le système demandeur au système répondant (étape 1). Le système répondant vérifie alors la demande pour détecter d'éventuelles erreurs (étape 2) et, en prenant pour hypothèse qu'aucun problème n'a été détecté dans la demande, procède à l'instanciation demandée du compteur (étape 3). Ensuite, il répond au système demandeur par un message de confirmation IEC 61968-9:2013 reply(MeterConfig) (étape 4).

D.2.3 Demande d'instanciation de compteur (message de demande)

Le message de demande envoyé par le système demandeur au système répondant (étape 1) est un message IEC 61968-9:2013 create(MeterConfig) (voir la Figure D.2). Dans cet exemple, la demande spécifie un seul compteur avec un seul registre et deux canaux.

```
<CreateRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>create</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:10-06:00</Timestamp>
    <MessageID>fbb1c893-83c9-47b2-8cca-b8f8859315e6</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a169-ce38fba5d40f</CorrelationID>
    <Comment>Request to instantiate a meter</Comment>
  </Header>
  <Payload>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <ConnectDisconnectFunction>
        <enabled>true</enabled>
        <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
        <isRemoteAutoReconOp>true</isRemoteAutoReconOp>
        <Names>
          <name>CDF1</name>
        </Names>
      </ConnectDisconnectFunction>
      <Meter>
        <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
        <formNumber>2S</formNumber>
        <electronicAddress>
          <mac>98-5F-D3-42-05-39</mac>
        </electronicAddress>
        <ConnectDisconnectFunction ref="CDF1"/>
        <SimpleEndDeviceFunction ref="SEDF1"/>
        <EndDeviceInfo>
          <phaseCount>1</phaseCount>
          <ratedVoltage>240.0</ratedVoltage>
          <capability>
            <communication>true</communication>
            <connectDisconnect>true</connectDisconnect>
            <electricMetering>true</electricMetering>
            <onRequestRead>true</onRequestRead>
          </capability>
        </EndDeviceInfo>
        <Names>
          <name>M001</name>
          <NameType>
            <name>MeterID</name>
          </NameType>
        </Names>
      </Meter>
      <SimpleEndDeviceFunction>
        <programID>ER017A</programID>
        <Names>
          <name>SEDF1</name>
        </Names>
        <Registers>
          <Channels>
            <Names>
              <name>CH1</name>
            </Names>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
          <Channels>
            <Names>
              <name>CH2</name>
            </Names>
            <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          <Names>
            <name>REG1</name>
          </Names>
        </Registers>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </Payload>
</CreateRequestMessage>
```

Figure D.2 – Exemple de message de demande d'instanciation de compteur

D.2.4 Réponse d'instanciation de compteur (étape 4, cas de réussite)

En prenant pour hypothèse que la demande d'instanciation a abouti, le système répondant peut répondre par un message de réponse tel que celui représenté à la Figure D.3.

```
<CreateResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:12-06:00</Timestamp>
    <MessageID>ffb209a3-8f1e-4e5e-a8a1-a52a33eb81a4</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a169-ce38fba5d40f</CorrelationID>
    <Comment>Response to a meter instantiation request</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
</CreateResponseMessage>
```

Figure D.3 – Exemple de message de réponse d'instanciation de compteur (cas de réussite)

D.2.5 Réponse d'instanciation de compteur (étape 4, cas d'échec)

La demande d'instanciation de compteur peut échouer, par exemple, si le compteur spécifié est déjà instancié dans le système répondant. Le système répondant répond donc par un message tel que celui représenté à la Figure D.4.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-02T13:23:12-06:00</Timestamp>
    <MessageID>da1eea46-b239-4685-9c2c-e0300084ddc7</MessageID>
    <CorrelationID>f71ca7f6-ea63-435c-a169-ce38fba5d40f</CorrelationID>
    <Comment>Response to a repeated meter instantiation request</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>ERROR</Level>
      <Code>5.4</Code>
      <Reason>Referenced object already exists</Reason>
      <ObjectID>
        <Names>
          <name>M001</name>
        </Names>
        <ObjectType>Meter</ObjectType>
      </ObjectID>
    </Status>
  </Reply>
</CreateResponseMessage>
```

Figure D.4 – Exemple de message de réponse d'instanciation de compteur (cas d'échec)

D.3 Messages relatifs à la notification d'événement de l'instanciation d'objet

D.3.1 Généralités

Cet exemple représente un système répondant qui consigne de manière autonome un ensemble d'instanciations de compteur à un système demandeur intéressé.

D.3.2 Diagramme de séquence

Le diagramme de séquence pour cet exemple est représenté à la Figure D.5.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

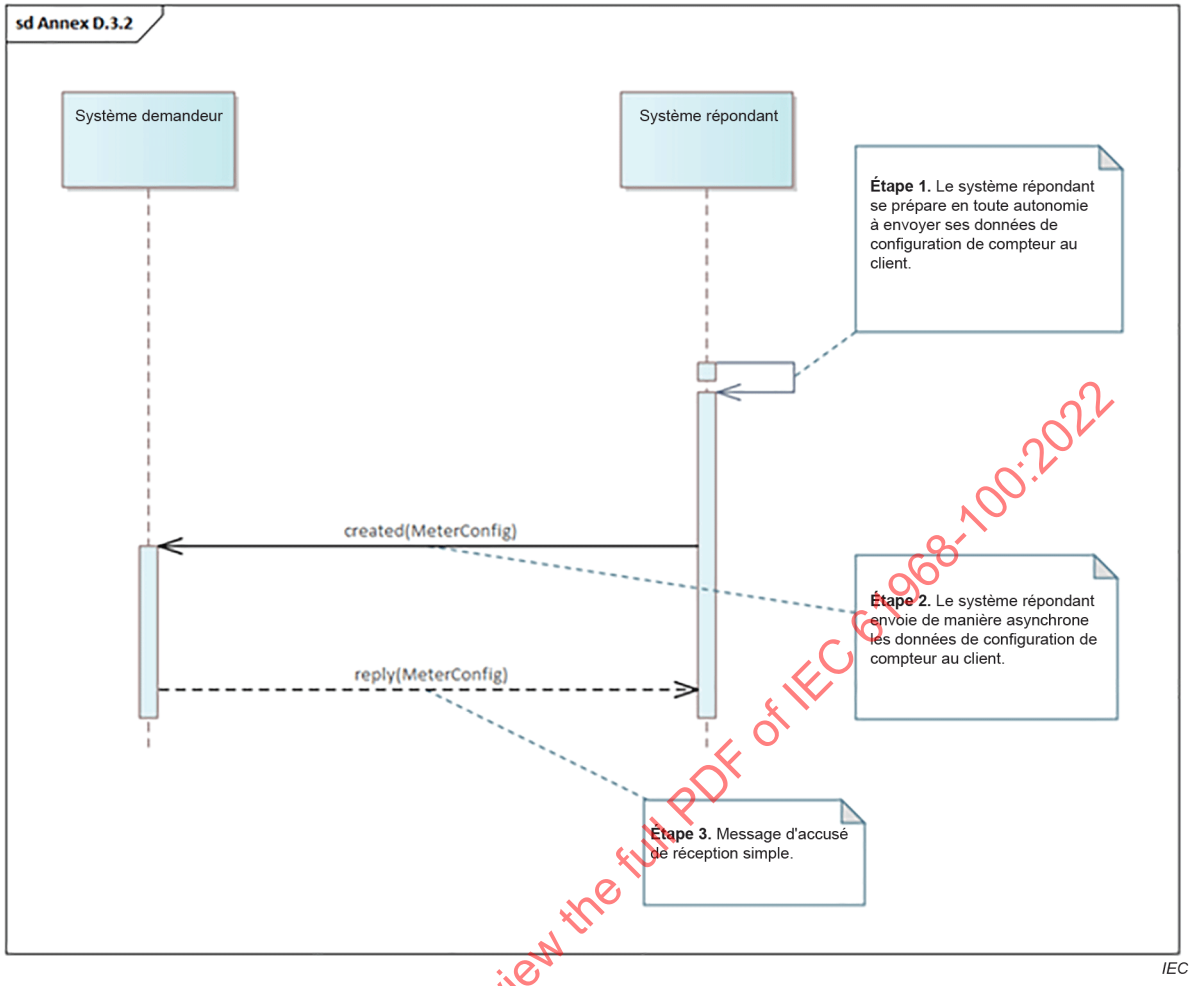


Figure D.5 – Exemple de diagramme de séquence pour la publication d'une instanciation de compteur

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

D.3.3 Publication d'une instanciation de compteur (message d'événement)

Le message d'événement envoyé par le système répondant au système demandeur est un message IEC 61968-9:2013 created(MeterConfig) (voir la Figure D.6). Ce message peut être envoyé par un système répondant après avoir été instancié avec les données MeterConfig présentées dans l'exemple décrit à l'Article D.2.

```
<CreatedEventMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>created</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-17T00:00:02-06:00</Timestamp>
    <MessageID>e0c9a3e6-38cd-429d-b4f8-1c0fb435102e</MessageID>
    <CorrelationID>7976ddcd-9244-4f88-b22d-65d0e1ffe81f</CorrelationID>
    <Comment>Reporting a meter instantiation</Comment>
  </Header>
  <Payload>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <ConnectDisconnectFunction>
        <enabled>true</enabled>
        <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
        <isRemoteAutoReconOp>true</isRemoteAutoReconOp>
        <Names>
          <name>CDF1</name>
        </Names>
```

```
</ConnectDisconnectFunction>
<ConnectDisconnectFunction>
  <enabled>false</enabled>
  <isRemoteAutoDisconOp>false</isRemoteAutoDisconOp>
  <isRemoteAutoReconOp>false</isRemoteAutoReconOp>
  <Names>
    <name>CDF2</name>
  </Names>
</ConnectDisconnectFunction>
<Meter>
  <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
  <formNumber>2S</formNumber>
  <electronicAddress>
    <mac>98-5F-D3-42-05-39</mac>
  </electronicAddress>
  <ConnectDisconnectFunction ref="CDF1"/>
  <SimpleEndDeviceFunction ref="SEDF1"/>
  <EndDeviceInfo>
    <phaseCount>1</phaseCount>
    <ratedVoltage>240.0</ratedVoltage>
    <capability>
      <communication>true</communication>
      <connectDisconnect>true</connectDisconnect>
      <electricMetering>true</electricMetering>
      <onRequestRead>true</onRequestRead>
    </capability>
  </EndDeviceInfo>
  <Names>
    <name>M001</name>
    <NameType>
      <name>MeterID</name>
    </NameType>
  </Names>
</Meter>
<Meter>
  <mRID>83009321-7883-4ea6-bc0e-71ddcd274f69</mRID>
  <formNumber>12S</formNumber>
  <electronicAddress>
    <mac>9A-5F-D3-42-04-39</mac>
  </electronicAddress>
  <ConnectDisconnectFunction ref="CDF1"/>
  <SimpleEndDeviceFunction ref="SEDF1"/>
  <EndDeviceInfo>
    <phaseCount>3</phaseCount>
    <ratedVoltage>240.0</ratedVoltage>
    <capability>
      <communication>true</communication>
      <connectDisconnect>true</connectDisconnect>
      <electricMetering>true</electricMetering>
      <onRequestRead>true</onRequestRead>
    </capability>
  </EndDeviceInfo>
  <Names>
    <name>M002</name>
  </Names>
</Meter>
<Meter>
  <mRID>99dcbdde-3249-11e8-b467-0ed5f89f718b</mRID>
  <formNumber>2S</formNumber>
  <electronicAddress>
    <mac>9A-5F-D3-42-04-40</mac>
  </electronicAddress>
  <ConnectDisconnectFunction ref="CDF1"/>
  <SimpleEndDeviceFunction ref="SEDF1"/>
  <EndDeviceInfo>
    <phaseCount>1</phaseCount>
    <ratedVoltage>240.0</ratedVoltage>
    <capability>
      <communication>true</communication>
      <connectDisconnect>true</connectDisconnect>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
      <electricMetering>true</electricMetering>
      <onRequestRead>true</onRequestRead>
    </capability>
  </EndDeviceInfo>
  <Names>
    <name>M003</name>
  </Names>
</Meter>
<Meter>
  <mRID>645707ea-37d6-11e8-b467-0ed5f89f718b</mRID>
  <formNumber>2S</formNumber>
  <electronicAddress>
    <mac>9A-5F-D3-42-04-4A</mac>
  </electronicAddress>
  <ConnectDisconnectFunction ref="CDF2"/>
  <SimpleEndDeviceFunction ref="SEDF2"/>
  <EndDeviceInfo>
    <phaseCount>1</phaseCount>
    <ratedVoltage>240.0</ratedVoltage>
    <capability>
      <communication>true</communication>
      <connectDisconnect>true</connectDisconnect>
      <electricMetering>true</electricMetering>
      <onRequestRead>true</onRequestRead>
    </capability>
  </EndDeviceInfo>
  <Names>
    <name>M004</name>
  </Names>
</Meter>
<Meter>
  <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
  <formNumber>12S</formNumber>
  <electronicAddress>
    <mac>9A-5F-D3-42-04-4B</mac>
  </electronicAddress>
  <ConnectDisconnectFunction ref="CDF2"/>
  <SimpleEndDeviceFunction ref="SEDF2"/>
  <EndDeviceInfo>
    <phaseCount>1</phaseCount>
    <ratedVoltage>240.0</ratedVoltage>
    <capability>
      <communication>true</communication>
      <connectDisconnect>true</connectDisconnect>
      <electricMetering>true</electricMetering>
      <onRequestRead>true</onRequestRead>
    </capability>
  </EndDeviceInfo>
  <Names>
    <name>M005</name>
  </Names>
</Meter>
<SimpleEndDeviceFunction>
  <programID>ER017A</programID>
  <Names>
    <name>SEDF1</name>
  </Names>
  <Registers>
    <Channels>
      <Names>
        <name>CH1</name>
      </Names>
      <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
    </Channels>
    <Channels>
      <Names>
        <name>CH2</name>
      </Names>
      <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
    </Channels>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
          <Names>
            <name>REG1</name>
          </Names>
        </Registers>
      </SimpleEndDeviceFunction>
      <SimpleEndDeviceFunction>
        <programID>ER017A</programID>
        <Names>
          <name>SEDF2</name>
        </Names>
        <Registers>
          <touTierName>overnight</touTierName>
          <Channels>
            <Names>
              <name>CH1</name>
            </Names>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.1.0.0.0.3.72.0"/>
          </Channels>
          <Channels>
            <Names>
              <name>CH2</name>
            </Names>
            <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.1.0.0.0.3.72.0"/>
          </Channels>
          <Names>
            <name>REG1</name>
          </Names>
          <touTier>
            <end>06:00:00</end>
            <start>22:00:00</start>
          </touTier>
        </Registers>
        <Registers>
          <touTierName>daytime</touTierName>
          <Channels>
            <Names>
              <name>CH1</name>
            </Names>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.2.0.0.0.3.72.0"/>
          </Channels>
          <Channels>
            <Names>
              <name>CH2</name>
            </Names>
            <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.2.0.0.0.3.72.0"/>
          </Channels>
          <Names>
            <name>REG2</name>
          </Names>
          <touTier>
            <end>22:00:00</end>
            <start>06:00:00</start>
          </touTier>
        </Registers>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </Payload>
</CreatedEventMessage>
```

Figure D.6 – Exemple de publication d'une instanciation de compteur (message d'événement)

D.3.4 Publication d'une instanciation de compteur (message d'accusé de réception simple)

Après avoir envoyé le message created(MeterConfig) (étape 2), le système répondant peut attendre que le système demandeur confirme la réception au moyen d'un message d'accusé de réception simple (étape 3) (voir la Figure D.7).

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<CreateResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-01-17T00:00:04-06:00</Timestamp>
    <MessageID>5be39bcc-5e55-4aaf-b10f-ab432bc91da7</MessageID>
    <CorrelationID>7976ddcd-9244-4f88-b22d-65d0e1ffe81f</CorrelationID>
    <Comment>Reporting a meter instantiation (simple ack)</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.3</Code>
      <Reason>simple ack</Reason>
    </Status>
  </Reply>
</CreateResponseMessage>
```

Figure D.7 – Exemple de publication d'une instanciation de compteur (message d'accusé de réception simple)

D.4 Messages relatifs aux demandes d'informations concernant une instanciation d'objet

D.4.1 Généralités

Cet exemple représente un système demandeur qui formule la requête d'un objet à un système répondant.

Dès la réception de ce type de demande, le système répondant extrait les données demandées et les renvoie au système demandeur.

D.4.2 Diagramme de séquence

Le diagramme de séquence pour cet exemple est représenté à la Figure D.8.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

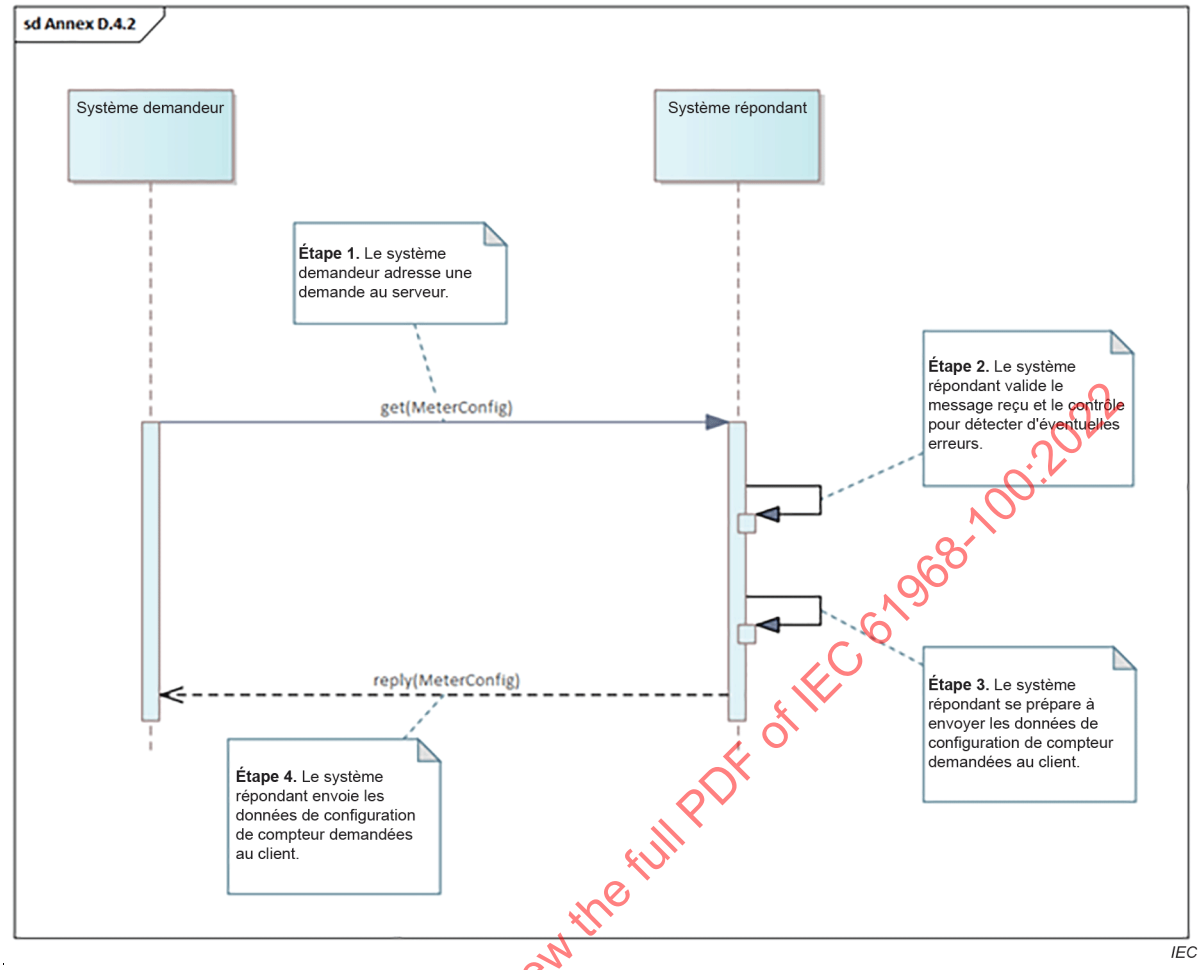


Figure D.8 – Exemple de diagramme de séquence pour les demandes d'informations concernant une instanciation d'objet

Une demande de données de configuration de compteur est formulée comme un message de demande IEC 61968-9:2013 get(MeterConfig) qui est envoyé directement par le système demandeur au système répondant (étape 1). Le système répondant vérifie alors la demande pour détecter d'éventuelles erreurs (étape 2) et, en prenant pour hypothèse qu'aucun problème n'a été détecté dans la demande, procède à l'extraction des données de configuration de compteur demandées (étape 3). Ensuite, il répond au système demandeur avec les données demandées dans un message IEC 61968-9:2013 reply(MeterConfig) (étape 4).

D.4.3 Demande d'informations concernant une instanciation de compteur (message de demande)

Le message de demande (étape 1) envoyé par le système demandeur au système répondant est un message IEC 61968-9:2013 get(MeterConfig) (voir la Figure D.9). Dans cet exemple, la demande concerne toutes les données relatives au compteur M001.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
<GetRequestMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>get</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-03-01T08:04:56-06:00</Timestamp>
    <MessageID>e94d334d-a1ee-4a30-b780-9e4c775ccc5f</MessageID>
    <CorrelationID>705a9743-f3bd-4f5a-b1d9-a7f98bb6bbcd</CorrelationID>
    <Comment>Request for information concerning a meter
instantiation</Comment>
  </Header>
  <Payload>
    <GetMeterConfig xmlns="http://iec.ch/TC57/2011/GetMeterConfig#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Meter>
        <Names>
          <name>M001</name>
        </Names>
      </Meter>
    </GetMeterConfig>
  </Payload>
</GetRequestMessage>
```

Figure D.9 – Exemple de demande d'informations concernant une instanciation de compteur (message de demande)

D.4.4 Demande d'informations concernant une instanciation de compteur (message de réponse)

Le message de réponse (étape 4) envoyé par le système répondant au système demandeur est un message IEC 61968-9:2013 reply(MeterConfig) (voir la Figure D.10).

```
<GetResponseMessage xmlns="http://iec.ch/TC57/2021/schema/message"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Verb>reply</Verb>
    <Noun>MeterConfig</Noun>
    <Timestamp>2018-03-01T08:04:58-06:00</Timestamp>
    <MessageID>2eacb133-59ec-48e0-a07c-51206789d087</MessageID>
    <CorrelationID>705a9743-f3bd-4f5a-b1d9-a7f98bb6bbcd</CorrelationID>
    <Comment>Response to a request for information concerning a meter
instantiation</Comment>
  </Header>
  <Reply>
    <Status>
      <Level>OK</Level>
      <Code>0.0</Code>
    </Status>
  </Reply>
  <Payload>
    <MeterConfig xmlns="http://iec.ch/TC57/2011/MeterConfig#"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <ConnectDisconnectFunction>
        <enabled>true</enabled>
        <isRemoteAutoDisconOp>true</isRemoteAutoDisconOp>
        <isRemoteAutoReconOp>true</isRemoteAutoReconOp>
        <Names>
          <name>CDF1</name>
        </Names>
      </ConnectDisconnectFunction>
      <Meter>
        <mRID>3c67f1fe-3240-11e8-b467-0ed5f89f718b</mRID>
        <formNumber>2S</formNumber>
        <electronicAddress>
          <mac>98-5F-D3-42-05-39</mac>
        </electronicAddress>
        <ConnectDisconnectFunction ref="CDF1"/>
        <SimpleEndDeviceFunction ref="SEDF1"/>
```

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022

```
        <EndDeviceInfo>
          <phaseCount>1</phaseCount>
          <ratedVoltage>240.0</ratedVoltage>
          <capability>
            <communication>true</communication>
            <connectDisconnect>true</connectDisconnect>
            <electricMetering>true</electricMetering>
            <onRequestRead>true</onRequestRead>
          </capability>
        </EndDeviceInfo>
        <Names>
          <name>M001</name>
          <NameType>
            <name>MeterID</name>
          </NameType>
        </Names>
      </Meter>
      <SimpleEndDeviceFunction>
        <programID>ER017A</programID>
        <Names>
          <name>SEDF1</name>
        </Names>
        <Registers>
          <Channels>
            <Names>
              <name>CH1</name>
            </Names>
            <ReadingType ref="0.0.0.1.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
          <Channels>
            <Names>
              <name>CH2</name>
            </Names>
            <ReadingType ref="0.0.2.4.1.1.12.0.0.0.0.0.0.0.3.72.0"/>
          </Channels>
          <Names>
            <name>REG1</name>
          </Names>
        </Registers>
      </SimpleEndDeviceFunction>
    </MeterConfig>
  </Payload>
</GetResponseMessage>
```

Figure D.10 – Exemple de demande d'informations concernant une instanciation de compteur (message de réponse)

D.5 Messages utilisant un OperationSet pour assurer l'intégrité transactionnelle

D.5.1 Généralités

Cet exemple utilise un OperationSet pour assurer l'intégrité transactionnelle (7.7.3).

D.5.2 Diagramme de séquence

Le diagramme de séquence pour cet exemple est représenté à la Figure D.11.

IECNORM.COM : Click to view the full PDF of IEC 61968-100:2022