INTERNATIONAL STANDARD

ISO 26162

First edition
2012-08-15

Systems to manage terminology, knowledge and content — Design, implementation and maintenance of terminology management systems

Systèmes de gestion de la terminologie, de la connaissance et du contenu — Conception, mise en œuvre et maintenance des systèmes de gestion de la terminologie

Reference number
ISO 26162:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 26162 was prepared by Technical Committee ISO/TC 37, *Terminology and other language and content resources*, Subcommittee SC 3, *Systems to manage terminology, knowledge and content*.

Introduction

Terminological data are collected, managed and stored in a wide variety of terminology management systems (TMSs). TMSs employ a variety of database management systems, ranging from personal computer applications for individual users to client server applications or Web-based applications operated by major companies and governmental agencies. Terminological data collections (TDCs) are based on various kinds of data models and consist of different sets of data categories (Data Category Selections, DCSs). To facilitate co-operation and to prevent duplication of work, it is important to develop standards and guidelines for creating and using TDCs as well as for sharing and exchanging data.

ISO/TC 37 has published the following standards in order to facilitate the exchange of terminological data and to create an integrated approach to be used in analysing an existing TDC and in designing new ones: ISO 704, ISO 12620, ISO 16642.

Having some acquaintance with these documents before reading this International Standard would be beneficial.

This International Standard gives guidance on choosing the relevant data categories, as well as designing and implementing a data model and a user interface for a TMS, taking into account the intended user group. The guidelines described in this International Standard are indispensable for the successful development of a TMS and for avoiding costly errors. This International Standard may be used for choosing the appropriate TMS for a certain purpose.

Systems to manage terminology, knowledge and content — Design, implementation and maintenance of terminology management systems

1 Scope

This International Standard specifies criteria for designing, implementing and maintaining terminology management systems (TMSs).

This International Standard provides information about the rationale for using a TMS, types of users and users' needs, steps in designing and implementing a TMS, as well as the tasks of organizing and managing a terminological data collection (TDC). It also provides guidelines for selecting and using data categories for managing terminology in various environments.

This International Standard is intended for terminologists, software developers and others who are involved in the process of developing or acquiring a TMS.

2 Normative references

The following referenced documents are indispensable for the application 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.

ISO 704, *Terminology work — Principles and methods*

ISO 12620, *Terminology and other language and content resources — Specification of data categories and management of a Data Category Registry for language resources*

ISO 16642, *Computer applications in terminology — Terminology markup framework*

ISO 30042, *Systems to manage terminology, knowledge and content — TermBase eXchange (TBX)*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1 Resources

3.1.1
terminological resource
terminological data collection
TDC
text or data resource consisting of **terminological entries** (3.1.4)

NOTE Adapted from ISO 24613:2008.

3.1.2
terminology management system
TMS
software tool specifically designed for collecting, maintaining, and accessing terminological data

3.1.3
terminological database
TDB
termbase
database comprising a **terminological resource** (3.1.1)

3.1.4
terminological entry
TE
part of a **terminological resource** (3.1.1) that contains the terminological data related to one concept

NOTE Adapted from ISO 1087-2, 2.22. The abbreviated term has been added.

3.1.5
concept orientation
principle applied to terminology management whereby a **terminological entry** (3.1.4) describes one and only one concept or two or more **quasi-equivalent concepts** (3.1.7)

NOTE A concept-oriented entry can contain multiple terms, which are therefore considered semantically equivalent.

3.1.6
equivalent concept
concept in one language that comprises the same characteristics as that covered by a given concept in another language

3.1.7
quasi-equivalent concept
nearly equivalent concept
concept in one language that shares most but not all characteristics with a concept in another language, but that is nevertheless used as an equivalent for that concept in some contexts

3.1.8
entailed term
term used in a text field such as the */definition/* or */context/* that designates a concept that is defined in another **terminological entry** (3.1.4) in the same **terminological resource** (3.1.1)

3.1.9
doublette
terminological entry (3.1.4) that describes the same concept as another entry

NOTE Doublettes are normally detected by TMS by identifying two terms having the same form; however, doublettes are determined based on identical concepts, not on identical terms. Doublettes should not be confused with homographs.

3.1.10
concept system
set of concepts structured according to the relationships among them

[ISO 1087-1:2000, 3.2.11]

3.1.11
concept diagram
graphic representation of a **concept system** (3.1.10)

[ISO 1087-1:2000, 3.2.12]

3.1.12
legacy data
terminological data that are available in an existing file or database, and that are considered for importation into a **TMS** (3.1.2)

NOTE Legacy data may be in the form of previously used databases, word-processing files, comma-delimited text files, SGML, HTML, or XML files, and the like. Conversion of such data to a format that will be compatible with a new TMS can pose serious challenges.

3.1.13
term
word, or several words, that denote a concept

EXAMPLES "Olympics" and "special Olympics" are two terms in the field of sports.

NOTE 1 When the word or words can denote more than one concept, each word/concept pair is a separate term. For example, "port" (shelter for boats) and "port" (computer connection point) are two different terms.

NOTE 2 In terminology theory, terms denote concepts in specific subject fields, and words from the general lexicon are not considered to be terms. In a TDC, however, words from the general lexicon are sometimes recorded in terminological entries, where they are still referred to as "terms".

3.2 Data categories

3.2.1
data category
result of the specification of a data field

[ISO 1087-2:2000, 6.14]

3.2.2
data element
unit of data that, in a certain context, is considered indivisible

[ISO 1087-2:2000, 6.11]

3.2.3
data granularity
degree of precision of data

NOTE For example, the set of individual **data categories** (3.2.1) */part of speech/*, */grammatical gender/,* and */grammatical number/* provides for greater data granularity than does the single data category */grammar/.*

3.2.4
data elementarity
principle whereby a single data field shall contain only one item of information

NOTE For example, including both a full form and an abbreviation in the same field would be a violation of the principle of data elementarity.

3.2.5
term autonomy
principle whereby all terms in a **terminological entry** (3.1.4) can be described by using the same set of **data categories** (3.2.1)

3.2.6
Data Category Registry
DCR
set of standardized **data categories** (3.2.1) to be used as a reference for the definition of linguistic annotation schemes or any other formats in the domain of language resources

[ISO 12620:2009, 3.2.1]

NOTE The ISO/TC 37 DCR contains **data category specifications** (3.2.7), which include historical, descriptive, and administrative information and other metadata.

3.2.7
data category specification
set of attributes used to fully describe a given **data category** (3.2.1)

[ISO 12620:2009, 3.2.2]

NOTE The abbreviation DCS refers to **Data Category Selection** (3.2.8).

3.2.8
Data Category Selection
DCS
set of **data categories** (3.2.1) selected from the **Data Category Registry** (3.2.6)

[ISO 12620:2009, 3.2.3]

3.2.9
complex data category
data category (3.2.1) that has a **conceptual domain** (3.2.11)

[ISO 12620:2009, 3.1.7]

3.2.10
open data category
complex data category (3.2.9) whose **conceptual domain** (3.2.11) is not restricted to an enumerated set of values

[ISO 12620:2009, 3.1.8]

3.2.11
conceptual domain
set of valid **value meanings** (3.2.14)

NOTE 1 Based on ISO/IEC 11179-1:2004, 3.3.6.

NOTE 2 The value meanings in a conceptual domain may be enumerated, further specified by additional constraints or expressed via a description. For instance, the **data category** (3.2.1) */term/* is described by its definition and thus constrained from properly containing, for example, contextual information or grammatical information, but it would be impossible to enumerate all values associated with this data category.

3.2.12
closed data category
complex data category (3.2.9) whose **conceptual domain** (3.2.11) is restricted to a set of identified **simple data categories** (3.2.13)

[ISO 12620:2009, 3.1.13]

3.2.13
simple data category
data category (3.2.1) with no **conceptual domain** (3.2.11)

[ISO 12620:2009, 3.1.12]

3.2.14
value meaning
meaning or semantic content of a value

[ISO/IEC 11179-1:2004, 3.3.39]

NOTE ISO/TC 37 treats enumerated values as simple data categories, that is, as data categories in their own right. The meaning of a value is always viewed in the context of the overall value domain and the closed data category with which it is associated, and is not just a property of the domain value itself.

3.2.15
value domain
set of **permissible values** (3.2.16)

[ISO/IEC 11179-1:2004, 3.3.38]

3.2.16
permissible value
expression of a **value meaning** (3.2.14) allowed in a specific **value domain** (3.2.15)

[ISO/IEC 11179-1:2004, 3.3.28]

3.2.17
thematic domain
class of applications identified by the similarity of the data structures they need to manipulate

[ISO 12620:2009, 3.4.3]

EXAMPLES Terminology, lexicography, morphosyntactic annotation.

3.2.18
thematic domain profile
profile
representation within a **data category specification** (3.2.7) of the **thematic domain** (3.2.17) with which a **data category** (3.2.1) is associated

[ISO 12620:2009, 3.4.4]

NOTE A data category may have several thematic domain profiles, indicating that it is used by several thematic domains.

3.3 Data modelling

3.3.1
data model
graphical and/or lexical representation of data, specifying their properties, structure and inter-relationships

[ISO/IEC 11179-1:2004, 3.2.7]

3.3.2
data modelling
process of structuring and organizing data, typically for implementation in a database management system

3.3.3
data modelling variance
variation in the assignment of **data categories** (3.2.1) to data models as a result of differences in philosophy with respect to the ordering of information in the **terminological entry** (3.1.4)

3.3.4
metamodel
data model (3.3.1) that specifies one or more other data models

[ISO/IEC 11179-1:2004, 3.2.20]

3.3.5
metadata
data that define and describe other data

[ISO/IEC 11179-1:2004, 3.2.16]

3.3.6
global information
GI
technical and administrative information applying to a complete data collection

[ISO 16642:2003, 3.7]

EXAMPLE Title of the data collection, revision history.

3.3.7
complementary information
CI
information supplementary to that described in **terminological entries** (3.1.4) and shared across the **terminological data collection** (3.1.1)

[ISO 16642:2003, 3.1]

EXAMPLE Domain hierarchies, institution descriptions and bibliographical references are typical examples of complementary information.

3.3.8
shared resource
information object that can be accessed from any of the terminological or lexicographical entries in a terminological or lexicographical resource

EXAMPLE Shared resources typically include bibliographical entries, responsibility entries, namespace identifiers, frequently referenced textual material, geographical location lists, and external files, such as graphics or audio files.

3.3.9
language section
LS
part of a **terminological entry** (3.1.4) containing information related to one language

[ISO 16642:2003, 3.9]

3.3.10
term section
TS
part of a **language section** (3.3.9) containing information about a **term** (3.1.13)

[ISO 16642:2003, 3.15]

3.3.11
class
object class
⟨UML⟩ description of a set of objects that share the same members

3.3.12
multiplicity
number of instances of one **class** (3.3.11) linked to one instance of another class in a relationship set

NOTE For instance, an origination date may be the same for many terminological entries, but one terminological entry can only have one origination date.

3.4 Applications

3.4.1
language planning
deliberate efforts to influence human behaviour with respect to the acquisition, structure, or functional allocation of language

EXAMPLE Language planning can include the standardization of spelling and grammatical rules, the specification of official national languages, efforts to create viable neologisms designed to increase the capacity of a language as a vehicle for scientific and commercial communication, and measures designed to protect the language from foreign influences.

3.4.2
descriptive terminology
approach for managing terminology that documents the way that **terms** (3.1.13) are used in contexts without indicating preferred usage

3.4.3
prescriptive terminology
approach for managing terminology that indicates preferred usage

3.4.4
normative terminology
approach for managing terminology that is used in standards work or governmental regulation

3.4.5
translation editor
software that supports the process of creating and revising translations

3.4.6
controlled authoring
authoring that uses limited vocabulary and textual complexity to produce clear documents

3.4.7
localization
l10n
process of taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold

[Localization Industry Standards Association]

4 Terminology management system (TMS)

4.1 General description

A TMS is a software tool specifically designed to collect, maintain, and access terminological data. It is used by translators, terminologists, technical writers, and various other users. TMSs shall be based on the principles for terminology work set down in ISO 704, which means that the terminological resources that are created by using a TMS contain terminological entries designed to document concepts and all the terms (synonyms and equivalents in other languages) that denote those concepts.

TMSs are typically used to elaborate, maintain and disseminate TDCs in a variety of organizational environments, including national bodies, government agencies, interest groups, companies, and individuals. Depending on the organizational framework, the TMS will need to support a variety of necessary features, different tasks and potential users.

When planning a software application for terminology management, it is important to consider the different possible types or categories of TMS.

4.2 Purchased or self-programmed TMS

There is a variety of terminology management tools on the software market; most of the software is commercial, but an increasing array of freeware or open-source software is available for terminology management. If the organizational, functional and methodological needs of the respective application environment are met by the system, it can be used off the shelf. However, in many cases, a customer-specific adaptation and adjustment is necessary. If the commercial software cannot be tailored to specific, identified needs, a TMS will have to be developed in-house. Costs, maintenance, support and interoperability are important parameters to be considered when deciding between buying or leasing a TMS, and developing one in-house.

4.3 Predefined or freely definable TMS

Commercial TMSs can have predefined or freely definable data modelling structures. A TMS with a fixed set of data categories, a defined set of languages and/or an unchangeable data model shall only be used if the setting corresponds exactly with the needs of the organizational environment in which the TMS will be applied. A TMS with a freely definable structure allows users to define their own data categories and their own entry structures so that the software can be adapted to suit users' specific terminological needs and can be modified as future requirements change.

4.4 Desktop, client-server or Web-based TMS

The simplest TMSs are installed on a local computer for a single individual. Sometimes the underlying database files are stored on a LAN computer and can be shared by multiple desktop clients. In large organizations with many users (and different roles), only a TMS with client-server architecture can serve the needs of the organization. These TMSs require a client installation on each local desktop and a server software installation handling the simultaneous access of different clients (with different tasks).

If multiple users all over the world are to have access to the terminological data, the only viable solution is a Web-based TMS; users do not have to install software specific to the TMS on their local computers because a standard Web browser will allow all operations with the TMS. When opting for a Web-based TMS, designers also have to decide whether to host their own server or to utilize a third-party TMS, for instance choosing between services offered by major TMS software suppliers or free, Web-resident collaborative environments. Before opting to outsource the hosting function, they should carefully weigh factors such as cost, host expertise, and reliable data accessibility, as well as data security and confidentiality.

4.5 Stand-alone, integrated or combined TMS

A stand-alone TMS is an autonomous software package designed for managing terminology independently of any other application. Integrating the terminology from a stand-alone TMS into other applications, such as controlled authoring tools or Computer Assisted Translation (CAT) tools, usually requires terminology to be exported from the TMS and imported into the other application. Sometimes the integration can be realized by developing specific plug-ins for each application in which the terminology is required. Because they are not tied to a specific software application, these types of TMS typically offer more flexibility in terms of the adaptability of the data for various applications.

An integrated TMS is one that comprises a set of terminology management functions directly within another software application, such as in a CAT tool. In order to use an integrated TMS, the user has to install and use the TMS package in conjunction with a seamlessly interoperable suite of software. These types of TMS are designed to offer the functions and features specifically required by the software of which they are a part, but, as a consequence, they might neglect to include features required for other applications. For instance, a TMS integrated into a controlled authoring application will contain functions and data types required for controlled authoring, but it would probably lack features required for the translation process.

4.6 Monolingual, bilingual or multilingual TMS

Monolingual TMSs are generally used by authors and readers who are probably not involved in translation or multilingual document production. A bilingual TMS can feature simple term or word pairs, sometimes using a lexicographical approach, where Term A is represented as equal to Term B, without reference to any subject field, definition or context. They may also respect concept orientation, where the terms listed in the two languages pertain specifically to a single concept, and if a term is attributed to more than one concept, then each concept is recorded in a separate entry. A multilingual TMS is usually concept-oriented because of the difficulties involved in managing multiple meanings across multiple languages; some of these TMSs also allow for a bilingual view, for instance for use in translation environments.

4.7 Database or structured text TMS

Since a TMS has to maintain a huge amount of data, a database engine is usually running in the background. A relational database is the ideal architecture for handling typical terminological requirements such as many terms in many languages, multiple data categories, and (unlimited) repeatability of specific types of terminological information. A structured text-oriented TMS also meets many of the typical terminological requirements by tagging the different types of data, such as in XML.

4.8 Single-database or multiple-database TMS

A TMS can store all terminological entries in one single database. Different subject fields, processing stages, customers and other types of information can be classified by attributes, and filters can be used to create a virtual subset of the database. Other TMSs store the terminological data in different physical databases, such as for different subject fields or different customers. Either the user has to select one of the databases before searching and editing, or the software allows for a simultaneous search and access to all databases, sometimes with a definable hierarchy specifying which database will be consulted first. A special case of the latter is a federated TMS that collects the data from different external databases, potentially with different data structures.

5 Project phases

5.1 Overview

As described in 4.2, for some purposes it may be possible to buy or otherwise utilize a TMS; however, in other cases, it may be necessary to develop a new TMS. The process of implementing a TMS, whether it is a third-party or a proprietary solution, comprises some or all of the phases listed below. Some projects, especially agile development projects which may also involve end users, require an iterative process where several phases are repeated until the desired outcome is achieved.

— Conduct a pre-feasibility study (see 5.2).

— Conduct a feasibility study (see 5.3).

— Conduct a use-case analysis (see 5.4).

— Establish system requirements (see 5.5).

— Analyse cost-effectiveness (see 5.6).

— Design the TMS (see 5.7).

— Develop the TMS (see 5.8).

— Deploy the TMS (see 5.9).

— Test the TMS (see 5.10).

— Populate, use and maintain the TMS (see 5.11).

A detailed work and time schedule should be established to set deadlines and distribute the tasks to the personnel involved. Intermediate reports should document the results. The schedule should clearly indicate when decisions will be made to proceed with the project (so-called go/no-go decision) and whether to build a new TMS, buy or opt for an existing TMS, or modify an existing TMS. These decisions may recur at different stages in the process as the requirements are prioritized.

The phases described in this clause can be adjusted according to the scope of the project and the type of organization involved (commercial enterprises, government agencies, educational institutions, translation agencies, and so forth).

5.2 Pre-feasibility study

The pre-feasibility study defines the scope of the project and determines which user groups, organizational units and systems should be involved. Stakeholders and users can include people with specific roles in the organization (writers, translators, and so forth), organizational entities such as publishing departments and translation services, business partners and vendors, and customers. But they can also include persons responsible for related software and systems such as file handling systems, workflow systems, translation software, and content authoring tools.

There may be a decision to deliberately exclude units or parts of the process defined here or to postpone them to a later stage of the project. The pre-feasibility study also provides an initial description of the process of creating the TMS. It justifies the need for introducing a TMS, including identifying any alternatives, and determines the basic requirements of the potential users. The organizational environment required to implement and operate the TMS should be discussed.

5.3 Feasibility study

A feasibility study includes a more refined outline of the tasks to be performed and the personnel and financial resources necessary to establish and operate the TMS. The feasibility study should also consider the technical and organizational integration of the TMS into the overall information management environment of the organization. The feasibility study should enable a final decision to be made on whether or not the TMS shall be implemented.

5.4 Use-case analysis

Conducting a use-case analysis is essential for ensuring that the TMS will meet the needs of the target user group. This phase is described in 6.3.

5.5 System requirements

During this phase, the specific requirements for the projected TMS are identified based on feedback from direct users and other stakeholders. Software feature specifications are then defined to fulfil those requirements. The hardware and software components are chosen according to the prerequisites outlined in the software feature specifications.

Typically, requirements are prioritized based on user feedback since it may not be possible to deliver all requirements (see 6.6).

In addition to user-centred requirements, technical requirements relating to the software infrastructure of the organization shall be met, such as to ensure conformity to company standards, as well as security and maintenance procedures.

The data model for the TMS shall be defined before it is developed or purchased.

5.6 Cost-effectiveness

The costs of implementing and operating the TMS should be estimated, as well as the expected benefits of the system. It should be noted that the financial return on investment will not be realized until after the TMS has been operating for a while. Any business case that is used to determine whether or not the project is approved for funding should include intangible, non-financial benefits (sometimes called “soft benefits”) in addition to financial ones. Soft benefits to consider include improved quality and customer satisfaction, cost avoidance through reduction of work duplication or other sources of inefficiencies, and more efficient processes (such as translation and customer service). If several alternative options are still possible at this phase (such as building a new TMS or buying a commercially available TMS), the costs and benefits should be estimated and compared for each option.

Another factor to consider is the cost of *not* implementing a TMS. The cost-avoidance factors mentioned above fit into this category, but there are other considerations such as the legal costs of trademark infringement that a TMS might help to avoid, the lost opportunities for controlled authoring and the additional difficulties in implementing a content management system without controlled terminology. All the potential current and future applications of terminology should be taken into account in building the business case.

The costs of importing existing data can be significant, particularly if the existing data set needs to be modified prior to import. The facilities for importing the existing data offered by the various systems being considered should be calculated in the cost analysis. It may be possible to request that the supplier of an existing TMS develop any required custom conversion routines.

5.7 System design

This stage involves either searching for and finding software products that can be used to implement the TMS, or designing a new TMS. If several software products are available on the market that may meet the requirements, each will have to be evaluated according to the projected requirements for the TMS – including the cost factor – in order to determine the best solution. If a new TMS is to be created, a detailed concept for the software system and its components has to be developed and the environment and staff for developing the software has to be provided.

Any requirements for importing existing terminological data shall be considered in the design.

5.8 System development

The objective of this stage is to develop the necessary hardware and software components. The TMS and its subsystems are implemented by writing new software or by configuring and/or customizing existing software. It is essential to document and test the components and their function within the whole system as they are implemented.

5.9 System deployment

The deployment stage encompasses several substages and should ideally be integrated with the testing stage.

Initial deployment, especially in large enterprises, should feature a system launch for a small group of selected users. During this initial launch, the TMS should be tested and any shortcomings should be corrected. The individuals involved in this initial deployment should participate in testing activities and should experiment with a small set of terminological entries before working with large projects.

Full enterprise-wide deployment should be undertaken only after completion of the testing phase, as described in 5.10.

5.10 System test

A thorough test of the whole system (all the software features and requirements) shall be performed after initial deployment. Testing should be based on the complete processing of a representative set of terminological entries that is large enough to challenge the system. The test data should be discardable in case of testing failures. Testing shall cover all aspects of terminology workflow and include multiple-use case scenarios. Testing should encompass all program features (input, output, import, export, displays, etc.) and involve representatives of all user roles (terminologists, technical writers, subject field specialists, translators, etc.). Manual testing can be enhanced by the use of automatic methods and routines and can provide initial data that can be used later to analyse the benefits of changes in the system.

Individuals should not be allowed to set up their own stand-alone environment of the same TMS until the testing is complete.

5.11 TMS population, use, and maintenance

As soon as the TMS has been deployed and tested, the system is operational. Full deployment involves instructing system administrators and system users, establishing the required maintenance infrastructure, and integrating the TMS into the overall workflow. It may be necessary to populate the TMS initially by importing existing terminological data. Prior to import, the data should be reviewed; modifications in content, structure, and format may be necessary. This task can take considerable time so the effort should be estimated and appropriate time allowed in the overall schedule. For more information about importing data, see 12.2.

6 User-centred design

6.1 Basic procedures

6.1.1 User-centred design refers to the methodologies and principles of designing a product so that it meets the needs of the product's users. The purpose of this clause is to explain how to include the users in the design process.

The potential users of a TMS include:

— people, such as terminologists, translators, interpreters, subject field specialists, teachers, students, technical writers, advertisers, media designers and scriptwriters, knowledge and information engineers, language engineers, product developers, engineers, and so forth;

— machines and tools that access and process terminological information, such as controlled authoring tools, spelling and grammar checkers, machine translation engines, term extraction tools, computer-assisted translation tools, concordance functions, search engines, text analysers, and so forth.

Every user category has its particular characteristics and specific needs. If the TMS is intended for more than one user category, the common denominator or dominant needs of all users should be determined, and these needs should be prioritized over all others.

6.1.2 Designers should determine:

— whether the goal of the terminology collection is normative, prescriptive, or descriptive;

— whether the TMS should be a general one covering all fields or a specialized one treating only one subject field or a limited number of subject fields and, if it is the latter, which subject fields shall be covered.

6.1.3 Options for delivering information to users with varying needs include providing:

— multiple views of the terminological data (partial views, some fields hidden until requested);

— all information available in all languages, or some information available in only one language;

— different interfaces for various purposes (translator's workstation, technical writing support, interaction with other computerized tools, etc.);

— access on a local (in-house), national, and/or international basis;

— Web interface or another type of interface;

— open public access, or access restricted to a certain environment.

6.1.4 Designers shall determine how the database is going to be used. Some types of information are more important to some users than to others. For example, technical writers need to know the meaning of technical terms and how to write the term correctly (capitalization, hyphenation, etc.), as well as what term to choose when they are aware of synonyms. Translators are typically interested in finding the preferred equivalent for a source term in a given target language. Software programs that interact with the TMS, such as those mentioned above, should be able to access the different types of information they require.

6.2 Steps in a user-centred approach

Adopting a user-centred approach to designing a TMS helps to ensure that the system will be easy to use and will achieve the desired objectives.

The following key steps are described.

— Identify the users and their needs (see 6.3).

— Identify the output products (see 6.4).

— Analyse the tasks to be performed and create use cases (see 6.5).

— List the requirements (see 6.6).

— Conduct a competitive evaluation (see 6.7).

— Prepare and evaluate a design prototype (see 6.8).

— Adjust the design according to user feedback (see 6.9).

— Perform a beta assessment (see 6.10).

6.3 Identifying the users and their needs

The first step in designing a TMS is to determine who will use it and what their needs are. All potential users of the TMS should be surveyed to find out if they will use it, how often they will use it, what tasks they will perform, and what specific needs they have. Designers can also conduct focus group sessions and one-on-one interviews to probe more deeply into the responses. The TMS should be designed to meet these needs, and should be assessed at several stages during its development to determine how successfully it meets these needs and expectations.

The different types of users should be surveyed in a manner that is proportional to their use. For example, if there are 1 000 translators and 100 terminologists who are expected to access the TMS on the Web, then the ratio of translators to terminologists who are surveyed on the Web-based functions should be 10:1. However, if the terminologists will spend twice as much time updating the data collection as translators, the ratio should be adjusted accordingly for questions about the update functions: 10*1:1*2 = 10:2.

Special attention should be given to needs that seem to contradict each other. Sometimes, the perceived needs of one group are opposed to the needs of another group; for instance, technical writers and translators can express mutually exclusive requirements. For example, context sentences (sample sentences containing the term) may be useful for translators but not for writers. In such cases, it may be possible to meet both needs by providing user-specific views of the data, and/or user-specific workflows.

Online surveys can be used to collect feedback from off-site users who cannot attend a face-to-face focus group session. The preparation of a survey should include a pilot test of the survey, in which the survey is tested with a small group, before conducting the survey with the complete user group. This will help ensure that the right questions are asked in the right order. The focus group for the pilot test should be representative of all types of users in order to avoid biased conclusions, and steps should be taken to motivate users to participate in the survey.

To validate needs and the perception that survey-takers have of the needs they express, questions should be included that attempt to evaluate the results that the survey-takers envision will be obtained by implementing their suggestions. It is important to avoid any ambiguity in survey questions.

In addition to identifying the needs of the various user groups, the survey should provide information about the following general determining factors, taking into consideration both present and potential future needs.

1. What is the main purpose of the TMS?

   — For example, if the goal is to *standardize* terminology in one or more languages, then the data collection should be *normative*. This means that it will enforce the use of specific standard terms. In this case, the design and methodology adopted for the TMS should follow the ISO standards for standardizing terminology, such as ISO 704, ISO 1087, and ISO 10241-1. On the other hand, if the goal is to *improve* the use of terminology, then the data collection should be *prescriptive*. This means that it should indicate preferred terms and non-preferred terms and include usage guidelines. If the goal is to *describe* how terms are used (without prescribing how they should be used), then the data collection is *descriptive*. This type of terminological resource does not indicate preferred or non-preferred terms.

   — If the data collection is meant to support a localization service provider that has many different clients, then the integrity of the client's terminology within the TMS is paramount. Data fields should be provided to identify the origin of terms, the client, and the projects where the terms are used, and security measures may be implemented allowing for the protection of the data provided by, or developed for, the client. In some cases, separate data collections should be provided for separate clients.

2. Are all users within the company/organization or are some external?

   — The answer to this question will impact the delivery and access channels for the data. For example, in the case of a network-based TMS, if some users are outside the organization's firewall, special measures should be taken to provide them with access to the TMS. Will all the terminology data be suitable for both internal and external users? If not, it will be necessary to somehow differentiate data that can be distributed externally from data that cannot, such as through the use of a special data field. External groups such as customers and vendors may not have access to the same tools for using the terminology. In this case interoperability of the data is important and shall be facilitated through the support of international interchange standards (see ISO 12620, ISO 16642, and ISO 30042).

   — Whenever multiple versions or copies of the data exist internally and externally, versioning and global updating should be accounted for in workflow and scheduling. It is, however, not advisable to maintain multiple versions of data.

3. Are all users experienced language professionals (writers, translators, terminologists, etc.) or are some from other areas such as sales, management, support services, training, and so forth?

   — A TMS that serves a range of user types may need to display different types of information to different types of users. If many users can create terminological entries, they also may need different interfaces to create those entries. Administrative personnel will not want to record as much detailed information as terminologists, and technical writers may need to have information presented in different ways than that required by translators.

4. Will any other tools, such as a desktop publishing system or a translation memory system, need to access or interact with the TMS?

   — The design phase should identify all tools that may interact with terminological data and determine what data and formats they require. Machine processes can require data that are not typically required by human users, such as the part of speech.

5. Which languages should the TMS support?

   — Unicode (ISO 10646) is the recommended encoding standard for interoperability with other systems and to support the widest possible range of languages. All newly developed TMSs shall be Unicode-compliant.

   — The choice of UTF8 or higher depends on the languages to be implemented and the available operating system.

   — If the TMS includes bidirectional languages (such as Arabic or Hebrew), bidirectional input methods and display formats shall be supported.

6. How many languages should the TMS support, both in terms of content and in terms of the interface?

   — The design of a TMS for monolingual, bilingual, and multilingual terminological resources will be quite different.

   — Monolingual databases are developed primarily as reference tools for writers and other users of the language, whereas bilingual and multilingual databases are developed primarily for translators and multilingual communicators. Monolingual databases typically record more information about term usage and tend to place more importance on definitions. They also more frequently record information about related terms and synonyms. Bilingual and multilingual databases, on the other hand, typically record context sentences, a data category that is often omitted in monolingual resources. These databases should obviously also feature the full range of data categories for translation-oriented terminology: language identifiers, transfer comments, degree of equivalency indicators, regional identifiers, translation-specific comments, and so forth.

   — If the same TMS is intended to be used for monolingual, bilingual, and multilingual applications, then different views should be created for each specific user type and activity. Each view can hide unwanted fields and emphasize important ones.

   — For a bilingual or multilingual TMS, designers should also consider whether there will be one specific source language (pivot language) or whether database search functions will be bidirectional or multidirectional with respect to source and target languages. For example, when the source language is fixed, the user cannot set the search language, and the user interface can be designed such that the data on this source language always appear at the top of the screen. But in a bilingual TMS that supports search on either language, the user interface should allow the user to select the source language, and the search results could dynamically adjust to display the selected source language at the top. In a multilingual TMS that allows search on any language, the user interface should allow the user to select the source language plus one or more target languages. In any case, if it is determined that a majority of users will typically choose one specific source language, then that source language should be set as the default search language to maximize efficiency, but users should be able to change that language if desired.

   — Within a multilingual, concept-oriented TMS, designers should consider how to handle situations involving quasi-equivalent concepts: a term A can be considered equivalent to B in one language and C in another, but the concepts underlying B and C may not be fully equivalent. A decision should be made about how to handle such cases of incomplete concept equivalence, as this decision can impact the core design of the TMS. These partially equivalent concepts may either be included in the same concept entry, with a field available for transfer comments to explain the difference, or they may be recorded in separate terminological entries and linked together as translation candidates.

7. What specific data does each type of user need?

— People looking for monolingual terminological information, such as for controlled authoring, usually need definitions and usage information. Translators obviously need target-language equivalents, but they can often accept context sentences instead of definitions. Project-specific identifiers are necessary if different equivalents for the same source term could be used for different projects. For large data collections that serve different users and purposes, special fields may be necessary to organize the terminology according to their needs, such as project, customer, or product identifiers. Such fields are sometimes described as fulfilling "subsetting" purposes, since they can be essential for exporting subsets of the data.

8. How will users search and retrieve the information?

— Do users need to be able to search for terms from within other software applications, such as from a word processor or translation editor? Will they use the Internet or an intranet? Do they need access from mobile devices? Will they need advanced search filters or is a simple search string sufficient? Are there any performance problems that should be considered with regard to these access media?

— Search functions can allow for finding terms that are similar to the search term (fuzzy match) and finding terms based on the search term without its declinations (truncation). Whole classes of terms can be retrieved by filtering on specific data categories that they share in common, such as the source of the term, a product identifier, a particular part of speech, or even the date that the entry was created or last changed.

6.4 Identifying the output products

Based on information provided by target users, designers shall determine what kinds of output products these users need the TMS to produce. These may range from traditional glossaries (monolingual, bilingual or multilingual), comprising, for example, only terms and definitions, to sophisticated machine-readable output formats requiring very granular data, such as morphological and syntactic data used in machine translation software. Some outputs, such as glossaries and translation dictionaries, may best be configured with a word-based structure, while others, such as controlled authoring dictionaries and search engines, may be more effective with a concept-based structure. Some output products, such as spell checkers, require only the term, while others require a full range of terminological and lexical information.

Different output products also require different export formats, from simple comma-separated lists to detailed XML markup.

In order to accommodate all these factors, designers shall make an inventory of all the anticipated terminological output products and of the formats and data categories required to support those products.

6.5 Performing a task analysis and preparing use cases

This stage involves conducting an in-depth analysis of the tasks that the future users of the TMS currently perform, and then developing use cases for how the users would perform equivalent tasks, as well as any newly required tasks, when using the new TMS.

An effective method of performing a task analysis is to conduct a face-to-face focus group session with the users. This involves asking users to describe the typical tasks they perform that require terminology, and what their current problems are when carrying out these tasks. Designers shall consider factors that could change the way users perform their tasks and how the tasks would be affected. It may be more effective to conduct separate focus sessions with each type of target user.

EXAMPLE 1 Sample analysis of a task performed by a translator.

Task: use prescribed term from project dictionary during translation

Actor: off-site translator

Material used: spreadsheet containing project-specific terminology

Tools used: translation editor

Description: In their translation editor, translators currently only have access to bilingual dictionaries containing basic lexicons. Project-specific dictionaries are currently compiled in spreadsheet format by the translation project manager. The spreadsheets are not imported into the translation editor as integrated project dictionaries because the project managers do not have time to add the required mandatory fields (part of speech and project identifier) and to convert the file into importable XML format. As a result, the translator only uses the project dictionary spreadsheet on an ad hoc basis and terminological consistency is not enforced. There is a need for an automated means of displaying standardized project terms to translators and possibly for automatically checking the target text for compliance with specified terminological usage.

When developing use cases for the new TMS, designers shall identify who performs the tasks, what the prerequisites are, what specific data and tools are involved, the sequential steps that will be taken, and any specific interactions that will occur between the user and the TMS.

EXAMPLE 2 Simplified use case showing how the translator might perform this task in the new TMS.

Task: use prescribed term from project dictionary during translation

Actor: off-site translator

Tools used: translation editor, project dictionary

Prerequisite: project dictionary is available in translation editor

Description:

1. While translating in her translation editor, a translator comes across a term in a sentence she is translating and the term exists in the project dictionary (terms in the active translation segment that are included in the project dictionary should be visually identifiable).

2. Translator selects the term in the sentence.

3. Translator clicks a button in the translation editor user interface to display the terminological entry from the project dictionary (alternative activation modes such as right mouse clicking and keyboard shortcuts should also be possible). The project dictionary opens with the entry for the selected term displayed.

4. Translator selects an equivalent from the list of options in the entry.

5. Translator clicks a button (or alternative activation) to insert the selected equivalent term into the translated sentence.

6. Translator clicks a button (or alternative activation) to close the project dictionary and return to the translation editor.

Using a task modelling tool, it is possible to add conditions and alternative paths for achieving the above task and to use visual aids such as flowcharts. For example, what actions would occur if the project dictionary did not contain the selected term? What actions would occur if none of the available translations were acceptable for the translator? What alternative actions could occur for on-site translators who may have access to different resources? What would happen if the term were an abbreviation of another term or it had an abbreviation? Should the project dictionary always be open and, if so, is a condensed view of the entry required?

6.6 Identifying and prioritizing requirements

Based on the task analysis and use cases, designers shall create a list of the requirements for the TMS. Fundamental broad requirements can include the ability to

— create and update terminological entries directly in the system interface,

— compare data in two or more terminological entries,

— manage doublettes,

— import and export data in predefined standardized formats,

— filter data according to a variety of criteria, and

— publish data on the Web or by other means.

In environments where various computing systems are used for tasks that can be enhanced through the use of terminology, such as authoring, content management, and translation, the components of the TMS environment should be interoperable with these other systems to gain maximum benefit of the terminology data in the enterprise. For instance, the TMS can provide lists of known terms to a term extraction tool so that the tool can effectively identify only “new” terms. The TMS can provide data for an automatic lookup function in a translation environment to improve terminology consistency in translations. The TMS can also be dynamically searched by a localization workflow system to provide relevant terminological dictionaries for a given localization project. In an authoring environment, the TMS can provide data to assist writers to avoid prohibited terms and to use abbreviations and acronyms correctly. As stated earlier, each of these different applications of terminology has different requirements that should be investigated carefully.

Once potential requirements have been identified, designers should return to their focus groups to ask users to prioritize requirements from highest to lowest importance. Those requirements that are mandatory shall be clearly identified, meaning that without them, the TMS will not be developed.

Requirements are typically prioritized in the following categories as a first step: (1) essential, (2) important, and (3) nice to have.

The second step is to assign a value to each item on a scale of 1 to 5, ignoring at this stage whether it is essential, important, or nice to have. This approach allows for the creation of a grid representing the priority of each requirement.

In some cases, there is another prioritization category that reflects some corporate strategy or other strategic or legal requirement. For instance, the company may have a policy for supporting certain accessibility requirements for sensory-impaired users, and these requirements may be mandatory whether or not the current user group needs them.

From the sample use case cited in 6.1.3, it is possible to identify the following design requirements. (The following list is illustrative only and not meant to be comprehensive.)

a) The translation editor and project dictionary should be usable in an off-site environment.

b) Terminological entries should have project identifiers that are recognizable by the translation editor.

c) Terminological entries should be exportable from the data collection in the form of a project dictionary for the translation editor, or directly connected to the translation editor. The project identifier should therefore be usable as an export filter, or as a view filter in an integrated environment. Also, mandatory field requirements for the translation editor dictionary should be met by the TMS.

d) The project dictionary should be accessible directly from the translation editor.

e) The translation editor should automatically highlight terms that are in the project dictionary.

f) The translation editor should search the project dictionary for a term that is highlighted in the source sentence.

g) Data fields should be visible in the project dictionary to help the translator make an appropriate selection when the entry contains multiple translations or multiple senses.

h) The translation editor should be programmed to insert the selected translation into the target sentence. (This type of dependency on an external tool should be identified and resolved with the developers of that tool.)

6.7 Conducting a competitive evaluation

To ensure that the tasks and requirements that have been identified are not overly focussed on internal procedures and that they take into account terminology management practices outside the organization, the design team should research the way other comparable organizations handle similar terminology tasks. The team should evaluate other TMSs to generate ideas on how to implement the functions. Before developing a new TMS, the team should determine whether there is a commercially available TMS that provides the required features and, if so, whether it would be more cost-effective to buy or customize that TMS than to develop a new one.

6.8 Designing and evaluating a prototype

Using the data gathered during the previous stages, the designers shall produce a prototype reflecting how the TMS will look and function. Users should evaluate the prototype in another face-to-face focus group session where they walk through the design and compare it to the tools and processes they currently work with. Prototype design evaluations should be conducted throughout the development of the product. A questionnaire or comment form should be used to collect feedback from users. These data can then be compared to feedback statistics in earlier and later stages to validate the design.

Milestones shall be established in the development schedule when the evolving TMS should be checked against the design specifications. This helps to ensure that the functions are developed according to the original user input.

6.9 Adjusting the design to user feedback

On the basis of the feedback from the design prototype evaluation session, designers shall make adjustments to the design to correct weaknesses or add requested functions. The feedback sessions should be repeated as needed until users are satisfied with the prototype.

6.10 Performing a beta assessment

Before the TMS is deployed, it should be tested with users to determine whether it meets their expectations and, if necessary, some final adjustments can be made, although at this late stage it is impossible to make major changes to the core design. For planning future enhancements, it can also be useful to determine core quantitative measures such as time on task, number of assists, and number of errors, as well as qualitative measures such as user satisfaction.

7 Terminological data categories

7.1 Introduction to data categories

Terminological entries are made up of specific units of information, such as *terms, definitions* and *contexts*, which are called *data categories.* ISO 12620 defines the parameters for specifying data categories for recording information related to terminology and other language and content resources (for instance, terminological resources, lexicographical resources, electronic lexical resources, etc.). The Data Category Selection (DCS) designated by the Terminology thematic domain group, which functions under the supervision of TC 37/SC 3/WG 1, shall specify data categories for recording terminological information in computerized environments and for the interchange and retrieval of terminological information independent of the local software applications or hardware environments in which these data categories are used. The terminology DCS comprises a subset of the global set of data categories defined for use in linguistic resources and contained in the Data Category Registry (DCR, see http://www.isocat.org).

The terminology DCS includes more than 200 data categories for terminological data that can be used as field names in entries, plus additional data categories that are used as field values. For instance, */grammaticalGender/* is a data category that would typically be a field name, and *masculine, feminine, neuter* are data categories that would typically occur as values of this field.

The list of data categories in the terminology DCS is comprehensive because it aims to contain all data categories from a wide variety of existing TDCs. Usually, a given termbase will only use a small subset of the terminology data categories found in the terminology DCS. The data categories chosen for a terminological database (TDB) depend on the specifications of the terminology project, its goals and the user needs. In most databases, usually a small number of data categories are mandatory and the remaining are optional.

Data categories cited in ISO 10241-1 and ISO 12616 include:

- terms (in any desired language);
- term-related information (such as grammatical attributes, register, status):
  - classification of term types (such as spelling variants, full forms, abbreviated forms, and so forth);
- descriptive concept-related information (such as subject field (domain), definitions, contexts, examples, notes, and graphics);
- administrative and bibliographical information:
  - identifiers of various sorts, such as to identify products or projects with which the terms are associated;
  - dates, names of people who created or modified the entry or parts of it;
  - entry status, for example submitted, working, approved;
  - sources of terms, definitions, contexts, notes, etc.

7.2 Principles for selecting and using data categories

7.2.1 Concept orientation

The structure of terminological entries is distinguished from that of lexicographical resources in that each terminological entry contains information about a single concept (or sometimes, especially in multilingual collections, quasi-equivalent concepts), together with all the terms that are used to designate this concept, in as many languages as desired.

It can be beneficial to provide a view of the data that is arranged in a lexical, or word-based, structure, along with the traditional concept-oriented terminological view. Many of the same data categories are used for both views, but they are arranged differently in the display. A lexical view includes all senses, or meanings, of a word, whereas the terminological view displays all terms that denote a single concept or meaning. The lexical view is therefore particularly useful for understanding all the meanings of a polysemous word, and the terminological view is useful when the user needs to identify synonyms and translation equivalents.

Figures 1 and 2 graphically demonstrate the basic differences between a concept view and a word view and the correspondence between them.

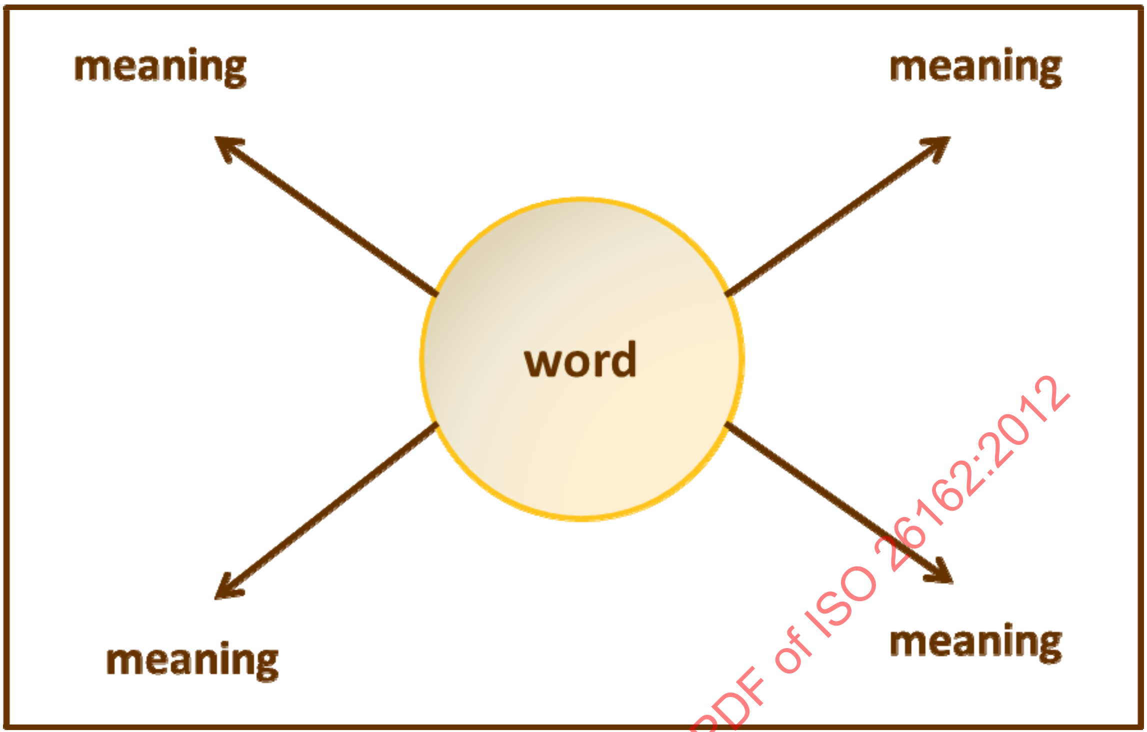

Figure 1 — Word-oriented structure

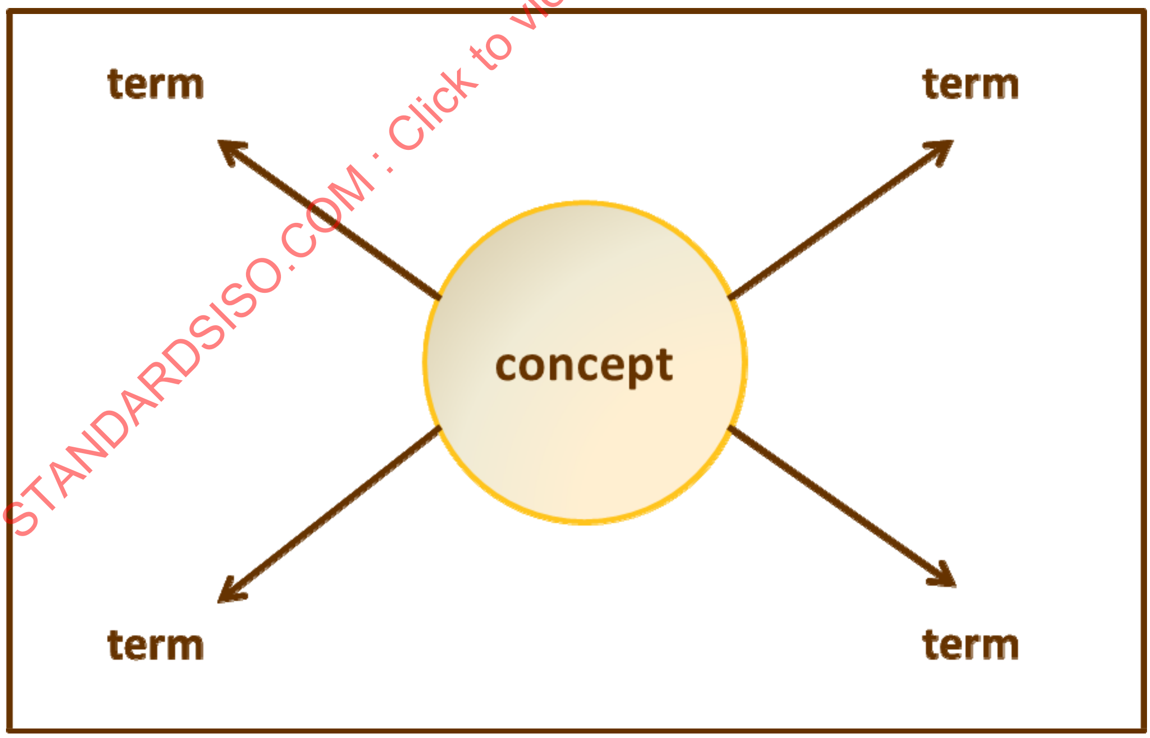

Figure 2 — Concept-oriented structure

7.2.2 Contextual and semantic data categories

Frequently, it can be difficult to discern the meaning of a term based just on the term itself. Thus, bilingual and multilingual glossaries published on the Web that provide only equivalents, without definitions, contexts, or in many cases, scope notes or subject-field references, may not be reliable. Users cannot be assured that a target language term provided with such limited information is accurate for a given context. A well-designed TMS will provide data categories to explicitly store a full range of information types to disambiguate the meaning of terms, such as definitions, subject fields, context sentences, part of speech values, and so forth. Sometimes, however, users do need simple lists of source language and target language equivalents. In this case, a semantic categorization, such as a subject field, should be clearly indicated in the list to clarify in what contexts the terms are relevant. A TMS should be capable of exporting this type of terminological product through the use of an export filter on the semantic data category *(/subject field/* or other).

7.2.3 Term autonomy

According to the principle of term autonomy, it should be possible to provide an equal level of information and description for each term included in a terminological entry. Term autonomy implies that, even if one term is identified as a preferred term, all synonyms and equivalents are also associated with a full set of data categories. The best way to preserve term autonomy is to provide an individual */term/* field for each term to be documented in the entry. The */termType/* data category is then used to indicate whether the term is a full form, an abbreviated form, an acronym, and so forth. The */administrativeStatus/* data category can be used to indicate whether the term is preferred, deprecated, and so forth. Various additional data categories are available to distinguish terms within the same entry and to provide information that tells users which term is the most appropriate for a given context.

In a termbase, due to the principles of term autonomy and concept orientation, a data category */synonym/* should not be necessary. All terms within a concept entry are synonyms, and each term within the concept entry has equal status as a term.

7.2.4 Repeatability

In theory, data categories can be repeated as often as necessary and can be combined with other data categories to record information about the concept or about one of the terms in the concept entry. For example, the data category */term/* should be repeatable in a language section as many times as needed in order to document the synonyms and equivalents for a given language. The */term/* data category can be accompanied by a variety of term-related data categories to describe properties of the term, and textual data categories such as */definition/*, */context/*, and */note/* can be supplemented by a */source/* data category to indicate the source of the information. For instance, in the sample multilingual terminological entry shown in Annex A, the */term/* data category, along with all its associated data categories, is repeated for each term in each language (Japanese, English, and German). The */term/* data category is supplemented by */partOfSpeech/, /grammaticalGender/* (for the German term only), and */termType/.*

On the other hand, depending on the type of the TMS and its purpose, it may be decided that some data categories should not be repeatable in an entry or a part of an entry. For instance, in a prescriptive TMS, the */definition/* data category could be constrained to a single instance in the entry, or as a single instance for a language section, whereas in a highly descriptive TMS, the */definition/* might be repeatable. It is also often agreed that, for a given concept entry, only one subject field applies and only one part of speech value. The TMS should be designed to enforce the property of data repeatability according to the needs of the users and the purpose of the data.

7.2.5 Data granularity

Terminological entries should provide data categories designed to accommodate the appropriate degree of data granularity. For instance, grammatical information should be recorded in various specific data categories such as in the following example:

/partOfSpeech/ - noun
/grammaticalGender/ - feminine
/grammaticalNumber/ - singular

Combining multiple properties in one general data category is not recommended, as in the following example:

/grammar/ - noun, feminine, singular

TMS designs that feature multiple specific data categories are more "granular" than those that use fewer, more general, data categories. Failing to differentiate subtypes of information in this way can lead to situations where data become difficult to retrieve, manipulate, and exchange.

7.2.6 Data elementarity

Data category elementarity refers to the principle whereby a data category shall only hold a single item of information. The example in 7.2.5 of a */grammar/* data category containing three different types of data (part of speech, gender, and number) violates this principle. Likewise, if a term and its abbreviation are both recorded together in a single */term/* field, such as "access control list (ACL)", this principle is violated. Such practices are common in published glossaries but should be avoided in a termbase. Automatic processing of much of the data becomes difficult if not impossible if this principle is not observed. For instance, it would be impossible to search for or extract acronyms from the database effectively if they were combined in the same field as their full forms. Likewise, the presence of two or more terms in the same field will cause problems for CAT tools that are designed to automatically insert terms from terminological entries into documents as they are being translated.

Decisions affecting granularity and elementarity involve weighing the information retrieval advantages afforded by providing more granular information against the sometimes increased effort required to enter information into separate fields. Care should be taken during the design stage to determine what kind of information users will want to record. Failure to provide the appropriate data entry fields for each kind of information can result in people using inappropriate fields or entering several different kinds of information in a single field, but requiring users to fill in more fields than they need can adversely affect usability of the TMS and decrease productivity during data input.

7.3 Types of data categories

7.3.1 Open, closed, and simple data categories

Open data categories can contain any text that adheres to their data category definitions. For instance, a */term/* field should only contain a term, not a definition or a context. Hence the */term/* data category is considered open because the actual terms that can be recorded there are unpredictable. Other examples of open data categories are */definition/* and */context/.*

In contrast, some data categories can only contain one or more of a finite set of permissible values, sometimes called permissible instances. These data categories are called closed data categories because their values comprise a closed set. For instance, when documenting terms in the German language, the data category */grammaticalGender/* can only take the values */masculine/, /feminine/,* or */neuter/.* These permissible values make up the value domain of this data category for the German language. For French, however, only *masculine and feminine* are valid.

Permissible values in value domains are called simple data categories. For example, */feminine/* is a simple data category used as a value for the closed data category */grammaticalGender/*. They are listed and defined inside a particular data category specification for their respective parent data categories. They are called simple because they cannot have content themselves. When designing a terminological resource, it is important to select the specific permissible values associated with closed data categories and to provide them as selectable values (sometimes called picklist values) in the TMS. For instance, a TMS used to document French terms would not need to include the */neuter/* gender value because there is no neuter gender in the French language. The use of picklists enables users to select the appropriate value without having to type it, and this also prevents misspellings or variant forms of the values from being introduced into the TDC. For instance, left to their own devices, users might type *masculine, masc.,* or simply *m.* for a masculine noun. Providing uniform representation of these values ensures consistency throughout the TDC, which is important for ensuring the performance of search filters and other data management tasks.

7.3.2 Mandatory, optional, automatic, and default data categories

Data categories can be either mandatory or optional in a TMS. Typically, a TMS will not allow users to save a terminological entry if a mandatory field is empty. At least one term is required for each terminological entry.

Other typical mandatory fields can include subject-field information, which can be essential for differentiating homographs, or part of speech values, which also serve to differentiate homographs. Designating certain fields to be mandatory can be problematic. For instance, it may take considerable time and effort to find definitions and contexts, to discuss them, and to enter them into the TMS. It may be more productive to allow this information to be added later as it becomes available.

Some data category content is system-generated instead of being filled in manually by the user. For instance, many TMSs automatically assign entry numbers to terminological entries, as well as record creation dates and modification dates, and the names of individuals who created or modified the entry. Part of TMS design involves making decisions about the levels within the entry that will be documented with such administrative information. It is generally insufficient to record such administrative information only for the entry as a whole, because different people can be responsible for different parts of the entry, especially for different language sections.

It is also possible, and even recommended, to set default values for certain data categories. For instance, if a user is going to document terms associated with a single project or source text during a given working session, it may be convenient to allow the user to pre-set the values for the corresponding data categories */source/* or */projectSubset/* so that all terminological entries created during that working session automatically contain those values. Likewise, if it is determined that most of the terms in a TDC are going to be nouns, it saves the user time if the */partOfSpeech/* data category is pre-set to noun, and the user can change this value for non-nouns.

7.3.3 Read-only, read-write, and hidden data categories

In a TMS, it should be possible to set different access levels for users at the field level, depending on the user's needs and role. A field that is visible but cannot be modified is a “read-only” field. A field that is visible and can be modified is a “read-write” field. Fields that are not needed by some users can be hidden from their view.

For instance, it may be desirable for a lead terminologist to have read-write access to all fields. This person may also be responsible for assigning authorization levels to other users. Terminologists creating entries for specific languages should have read-write access to the fields in their language sections. Only specific lead terminologists may have read-write access to the entry status field to set the value to “approved.” Technical writers and translators might have read-write access to only one or several fields necessary to allow them to provide feedback to the terminologists, and read-only rights to the remaining fields. It is frequently desirable to hide administrative data categories such as */date/* and */responsibility/* fields in order to provide a less cluttered view of the data for most users.

7.3.4 Multimedia files

A termbase that describes concepts referring to concrete objects, such as in science and engineering fields, may benefit from having the ability to embed multimedia files such as graphics directly into terminological entries. In this case, fields should be configured to store and display binary objects. A field for graphics shall be available at the concept level to allow for graphics that represent the entire concept independently of language. A field for graphics can also be provided at the language level to accommodate situations where the objects designated by the concept differ in appearance in different cultures. If the same multimedia file is needed for multiple terminological entries, it may be more economical to handle it as a shared resource.

7.3.5 Shared resources

Some data categories point to other resources, such as other files (texts, graphics, audio, or video) or Web sites. These resources are often called shared resources because any one item can be referenced from many terminological entries; thus it is in effect shared by different entries.

Bibliographical entries are another type of shared resource. It is common to document the source of terms, definitions, contexts, notes, and other information, as well as the source of any external resources (graphics, audio, video, etc.). Rather than citing the complete bibliographical information in the entry itself, it is more economical, particularly if the same reference is used in multiple entries, to create a separate bibliographical entry and point to it from entries by using a bibliographical identifier. This practice preserves the principle of data economy because it eliminates the need to repeat the same data in multiple terminological entries. It is also easier to update bibliographical references that occur in multiple entries, because there is only one instance of each bibliographical entry. An exception to this practice is the citation of URLs and URIs referencing Web resources because they are already short, unique identifiers.

Shared resources can be stored in an external database or in a separate location in the termbase itself. It should be possible for users to view shared resources while browsing or searching the termbase, because this information is often crucial to the evaluation of a terminological entry.

7.3.6 Relations between entries

Often, it is necessary to indicate that a term in one terminological entry is somehow related to a term in another entry.

Hierarchical relations, such as superordinate and subordinate, may be desirable in a termbase where users need to record hierarchical concept relations for some purpose, such as for certain normative databases where terminology standardization and the rigour of definitions is paramount, or for software applications that can use such hierarchical data, for instance, applications for knowledge management or retrieval. In this case, a separate data category for each type of hierarchical relation will be required. Such fields should allow the user to uniquely point to the target of the relation, which may be a concept entry, or it may be a term within a concept entry. For example, the following is the XML representation, in TBX (ISO 30042) format, of this type of data category:

```
<descrip type="broaderConceptGeneric" target="cid1234">boat</descrip>
```

This example might occur in a terminological entry for a particular kind of boat, such as a “sail boat,” to point to the entry for the broader generic concept, “boat.” The unique pointer is indicated as the value of the target attribute, which is the identifier of the concept entry that contains the term “boat.” The content of the element, that is “boat,” is not a unique identifier since this word can represent several different concepts and so it may occur as a term in several different concept entries.

Suppose that entry number cid1234 contains several terms, specifically “boat”, “ship”, and “vessel”. In this case, it may also be desirable to point to one of these specific terms within the entry. This will be possible if each term within an entry also has a unique identifier, for instance:

```
Concept: cid1234
    boat: tid1234-1
    vessel: tid1234-2
    ship: tid1234-3
```

In this case, the link would be represented as follows in TBX format, if the user wants to point specifically to the term “boat” within the concept entry:

```
<descrip type="broaderConceptGeneric" target="tid1234-1">boat</descrip>
```

Concept systems can also be reconstructed based on notational numbers designed to represent the position of a given concept within a concept system. Providing links from concept diagrams to terminological entries and vice versa is a highly useful means of visualizing conceptual relationships.

Concept systems can comprise non-hierarchical relations, such as associative relations or temporal relations. There are also non-hierarchical relations where the type of relation is not specified, such as simply */related concept/*. Some termbases use the label “see also” or “see” for this type of relation. The following are sample representations in TBX format:

```
<descrip type="relatedConcept" target="cid2345">sail</descrip>
<ref type="see" target="cid2345">sail</ref>
<ref type="crossReference" target="cid2345">sail</ref>
```

It is recommended that such relations be expressed through designated fields and data categories as shown in the previous examples. This approach optimizes the ability for such relations to be machine-processable. For a TDC that includes hierarchical concept relations, the type of relation needs to be explicitly expressed by using specific data categories such as */broaderConcept/.*

Some termbases do express relations embedded within other text fields, such as in a definition or note. Indeed, terms in definitions are often closely related to the concept being defined, and it is a popular practice to make any such terms into hyperlinks that can be clicked directly to open the related concept entry. The following is an example of two entailed terms in TBX format, in a sample definition for the term “sail boat”.

```
<descrip type="definition">A <hi type="entailedTerm" target="cid1234">boat</hi>
propelled by a <hi type="entailedTerm" target="cid2345">sail</hi>.</descrip>
```

Whatever the type, these relations can be converted to hyperlinks in the user interface, allowing users to click the link to open the target terminological entry directly.

7.4 Data entry structures

Termbases and terminological entries have a logical structure as described in ISO 16642. This structure is reflected in a hierarchical data model. Data categories are “anchored” to various levels in this hierarchical model. This process is called data modelling, as described in 8.2

A termbase consists of global information (information about the entire TDC), complementary information (containing shared resources), and terminological entries. Each terminological entry has a section for concept-related data categories (such as */concept identifier/, /subject field/,* and */definition/*), which are followed by language sections. Each language section contains one or more term sections, which contain the */term/* and term-related data categories (such as the */part of speech/, /grammatical gender/, /register/, /term type/,* and so forth). Finally, term sections can be broken down into term component sections, where components of compound or phrasal terms can be fully documented. These structures provide a framework on which terminological entries and data categories are anchored.

Some data categories can occur in different sections, as allowed by the data model. For example, context sentences (text that contains the term itself) can be viewed as both term-related and concept-related, because they can provide information about the term or the concept. Some databases allow the definition only at the concept level, whereas others allow language-specific definitions in the language section. Notes and administrative data categories can be associated with virtually all other data categories and can therefore occur at any level of the entry.

7.5 Selecting data categories

The different target users of a TMS, including not only people but also machine applications (as described in Clause 6), as well as the different output products that the TMS is expected to deliver, may have different needs with respect to terminological data categories. The data categories cited in 7.1 are widely used, but others may be needed for individual situations, such as for software localization, specific business applications, or terminology standardization and language planning, which is sometimes carried out by governmental organizations. Early in the design process, it is important to identify all the data categories that will be needed for all envisioned purposes of the TMS. The ability of existing TMSs to cover the required data categories shall be considered when deciding whether to purchase or modify an existing TMS rather than developing a new one. Major additions of data categories to existing TMSs can be cost-prohibitive and, due to their design, certain TMSs may not be able to accommodate certain additions or changes.

If there are existing data (sometimes called legacy data) that should be incorporated into the TMS, the set of data categories in these data should be evaluated to determine which ones are required in the new TMS. See 8.5. Care should be taken to harmonize data category names with names selected for the new TMS.

Annex A provides a list of typical data categories, followed by a multilingual terminological entry modelled for Japanese, English, and German.

7.6 Translation-specific data categories

Terms appearing in a single entry in different languages are assumed to be translation equivalents. However, sometimes the concepts they denote are not totally equivalent. In this case, it can be necessary to provide information about the differences between two quasi-equivalent concepts, or the conditions under which they can be considered equivalent, by using the */degreeOfEquivalence/* data category.

The */falseFriend/* data category can be used to indicate a term that is commonly mistaken to be semantically equivalent to another term, often due to morphological similarity, when in fact it has a different meaning. For instance, in French, the term *réaliser* does not mean *to realize* (to become aware of) but rather *to accomplish*. Two terms that are false friends will, of course, be documented in separate entries because they represent different concepts, but a cross-reference using the */falseFriend/* data category can make it explicitly clear to users of the termbase to avoid this potential error. The following is an example of this type of cross-reference in TBX format, as it might occur in the entry of “to realize”, where in this example tid5678 is the unique identifier of the term “réaliser”:

```
<termNote type="falseFriend" target="tid5678">réaliser</termNote>
```

Another useful data category is */transferComment/*, which can be used to provide additional explanations about the conditions affecting the use of quasi-equivalent concepts.

7.7 Prescriptive data categories

A number of data categories are available when there is a need to prescribe terminology. For example, the */administrativeStatus/* data category can be used to indicate if a term is *preferred, admitted*, or *deprecated* for use within a certain working environment such as in a commercial enterprise. Authoritative groups such as standards organizations and language planning departments in governments may use a similar data category, */normativeAuthorization/*, to indicate normative usage restrictions, such as when a term has been defined in formal legislation.

7.8 Workflow-related data categories

Typically, terminological entries are created in stages. These stages can be marked in the entry by using the data category */elementWorkingStatus/,* which takes values such as *starter, working,* and *consolidated*, reflecting the completeness of the entry and its approval status. The data category */language-planningQualifier/* provides workflow states for a terminological entry as it passes through the approval process in language planning environments, such as *suggested*, *nonstandardized*, *proposed*, and *new*. The */processStatus/* data category provides the basic workflow states *unprocessed, provisionallyProcessed,* and *finalized.*

7.9 Standardized data category names and data category concepts

As noted in 7.1, the names of data categories and their meanings have been standardized in the DCR. In order for a TMS to meet ISO standards, it shall implement and use data categories according to their meaning and descriptions in the terminology DCS of the ISO/TC 37 DCR (see 7.1). However, if desired, in a particular TMS, data category names that are different from those in the DCR can be used, provided that a one-to-one mapping to the data category in the DCR is possible. In this case, the TMS administrator can register these alternative names in the DCR. For instance, one TMS can choose the name “sample sentence” for the data category */context/,* or it can choose the name “grammatical category” for the data category */partOfSpeech/*. The support of application-specific data category names also meets the needs of a TMS that is developed in a language other than English, so that for instance a French TMS can have the name “*partie du discours*” for the data category */partOfSpeech/.* Sometimes designers need to add new data categories to the DCR, where they can be proposed for future standardization.

8 Data modelling

8.1 The terminological metamodel

Data models provide formal descriptions of how data will be structured in a TMS, and metamodels are used to provide standard high-level architectures that models used in a single discipline can follow in order to achieve interoperability.

As already mentioned in 7.4, ISO 16642 specifies a framework designed to provide guidance on the basic principles for representing data recorded in a TDC. This framework includes a metamodel and methods for describing a terminological markup language (TML) expressed in XML (eXtensible Markup Language). The ISO 16642 metamodel facilitates the exchange of terminological data and can be used to analyse an existing TDC and to design a new one.

Thus, specific data models that form the basis for any TMS shall be based on the generic metamodel specified in ISO 16642. This terminological metamodel is, in turn, based on the methods and principles of terminology management for creating terminological entries as described in ISO 704.

The metamodel is an abstract conceptual data model that describes the basic hierarchy of information levels to which any TDC shall conform: 1) global information about the collection, 2) any complementary information (shared resources referenced throughout the collection), and 3) a number of terminological entries.

Each entry performs three functions:

— it describes one concept or two or more quasi-equivalent concepts in one or more languages;

— it identifies the terms that designate the concepts;

— it describes the terms themselves.

Each entry can have multiple language sections, and each language section can have multiple term sections. Each data element in an entry can be associated with various kinds of descriptive and administrative information. In addition, there are various shared resources that can be referenced by one or more entries. Such resources include bibliographic references, descriptions of ontologies, and images that illustrate concepts. For more information about shared resources, see 7.3.5.

ISO 16642 also provides a schematic view of the structure of a TDC, which is reproduced in Figure 3.

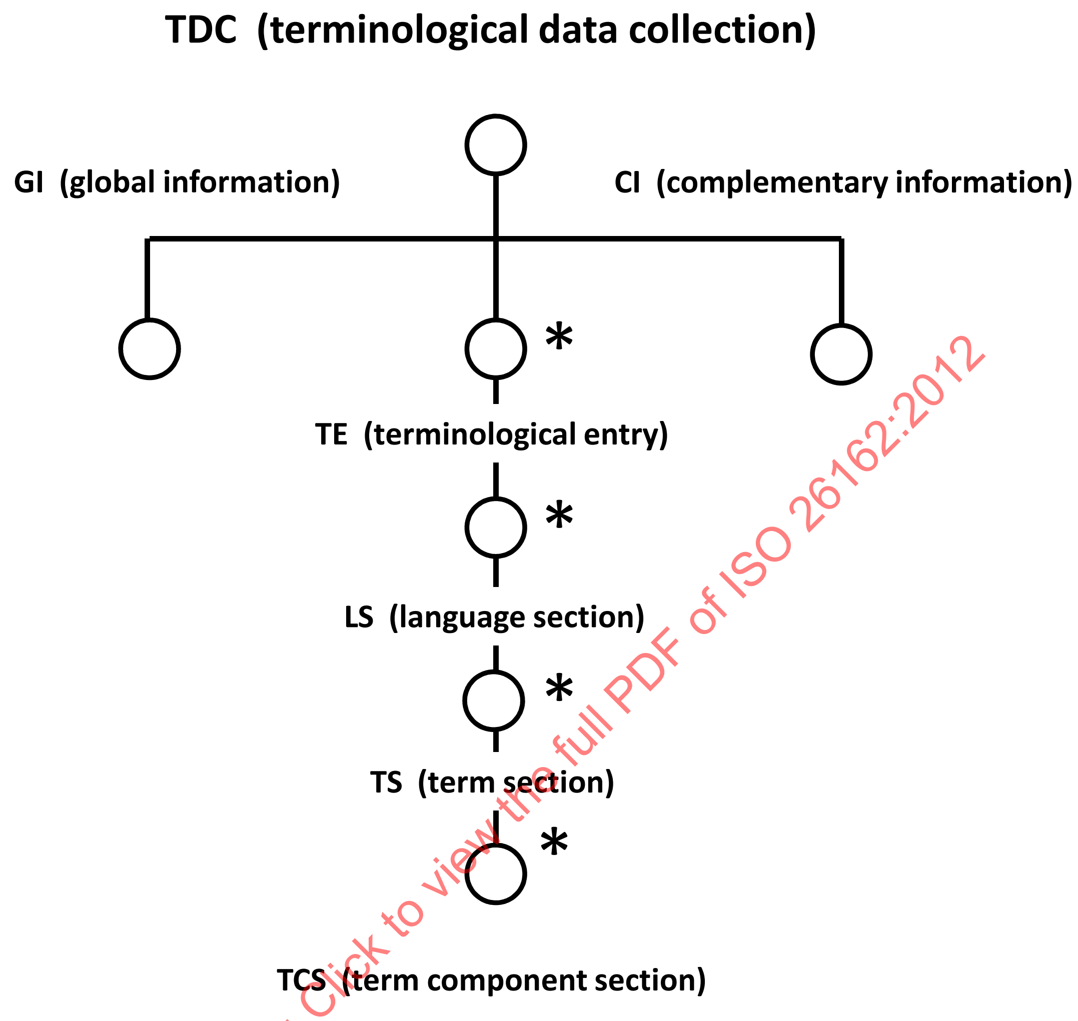

Figure 3 — Schematic view of the terminological metamodel from ISO 16642

8.2 Data modelling for concept orientation

One of the most important characteristics of a terminological entry is its concept orientation. A terminological entry treats one concept. In multilingual entries, usually the concept is the same, no matter which term in which language is used to denote the concept. However, there can be differences in the characteristics of the concept for different languages. In such cases, one may say that language-specific concepts treated in the same entry are quasi-equivalent.

As shown in Figure 4, in the sciences and in technology, it is not uncommon for conceptual consensus to be so strong that the common perception is that the terms in each language indeed designate one and the same concept. In the social sciences and in cultural studies, however, some concepts differ slightly from one language to another, even when there is a consensus that the terms denoting those language-specific concepts are functionally equivalent, which means that they may be used for translation purposes. Furthermore, synonyms can exist within one language, such that each language section can contain multiple terms. These synonyms can have slight differences in meaning or application, making them also quasi-equivalents. One can take several approaches to accurately reflect these slight differences in meaning. One approach is to design a data model that allows for concept variation. Another is to use a homogeneous concept model where differences in meaning are documented at the data category level.

When two terms express the same concept in one language, they are referred to as synonyms and they shall be recorded in the same terminological entry. There may, however, be differences in style or register that distinguish the synonyms, meaning that one of them may be more suitable in certain contexts than the other. These differences shall be recorded in an appropriate data category such as */administrativeStatus/* or */register/.*

Figure 4 gives an example of a multilingual entry comprising terms in two languages which designate a single concept, *protocol stack*. Like so many concepts in modern science and technology, this feature of the Internet architecture has evolved worldwide and is known and discussed across cultures. The terms that designate the concept in most languages are essentially loan translations of the English original. Perhaps the metaphoric image behind the term might have varied had the term evolved individually in the different cultures, but it did not. The term *stack* in English is frequently used in clear contexts as a short form for the full form term, and the paraphrase *stack of interconnected protocols* occurs on occasion as a kind of explanation or synonym. There are virtually no nuanced ambiguities or differences between the languages with respect to the intension of the concept, hence there is no problem in including all the terms in the same entry, nor is there any need to include any transfer comment.

Entry level
Entry number: 7
Subject: Protocols

English

Definition: particular software implementation of a computer networking protocol suite
Source Identifier: http://en.wikipedia.org/wiki/Protocol_stack

protocol stack

Source Identifier: IEE 802.11
Part of Speech: noun
Context: Another standard architectural model that is often used to describe a network protocol stack is the OSI reference model. This model consists of a seven layer protocol stack.
Source Identifier: http://www.cisco.com

stack

Source Identifier: http://www.w3.org/DesignIssues/Stack.html
Part of Speech: noun

stack of interconnected protocols

Source Identifier: http://www.w3.org/DesignIssues/Stack.html
Part of Speech: noun

German

Definition: konzeptionelle Architektur von mehreren Protokollschichten, die direkt aufeinander aufbauen
Source Identifier: http://www.esk.fraunhofer.de

Protokollstapel

Source Identifier: http://www.winfuture-forum.de.html
Part of speech: noun
Grammatical Gender: masculine
Context: Daten, die über ein Netz übertragen werden, werden von einem Netzprotokoll des Protokollstapels nach dem anderen verarbeitet.
Source Identifier: http://de.wikipedia.org/wiki/Protokollstapel

Figure 4 — Multilingual terminological entry documenting a single concept

In some cases, designations in different languages that are deemed to be equivalent for practical purposes do not denote exactly the same concept, as described in 7.6. This phenomenon is illustrated in Figure 5, which demonstrates the bifurcation of the concept underlying the German *wissenschaftlich*, which, when translated into English, becomes either *scientific* (for the empirical sciences) or *scholarly* for arts and the humanities, along with the parallel diversification of English *research*, which becomes in German either *Forschung*, for scholarship that creates new knowledge, or *Recherche*, for the retrieval and integration of existing knowledge.

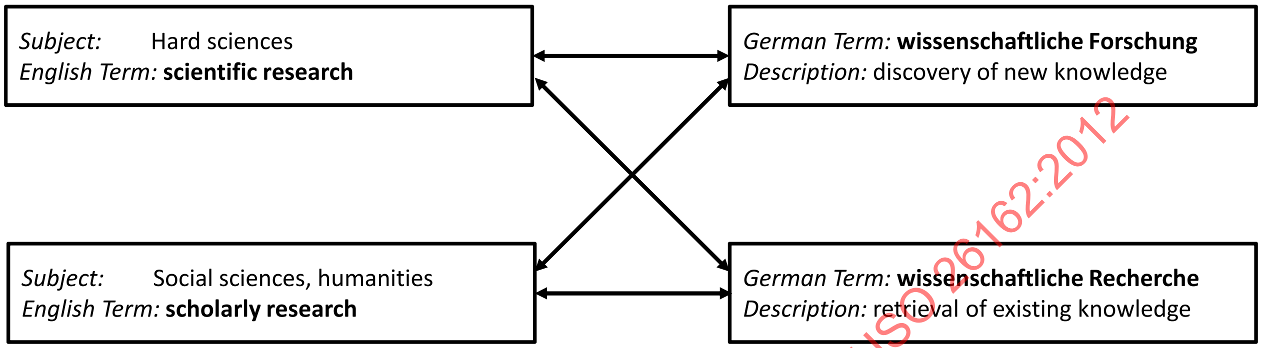

Figure 5 — Example of equivalence relations between two English and two German concepts

Figure 6 demonstrates the combination of these concepts, which results in four different nuanced concepts, potentially documented in four different terminological entries.

Entry 1:
Subject**:** **Hard sciences**
English Term**:** **scientific research**
German Term**:** **wissenschaftliche Forschung**
Description**:** **discovery of new knowledge**

Transfer comment**:**
The concept *scientific research* is partially equivalent to two German concepts, *wissenschaftliche Forschung* and *wissenschaftliche Recherche,* because the term *wissenschaftlich* refers to both the hard sciences and to the social sciences and the humanities.

Entry 3:
Subject**:** **Social sciences, humanities**
English Term**:** **scholarly research**
German Term**:** **wissenschaftliche Forschung**
Description**:** **discovery of new knowledge**

Transfer comment:
The concept *wissenschaftliche Forschung* is partially equivalent to the two concepts in English, *scientific research* and *scholarly research,* because a distinction is made in German between original scholarship that creates new knowledge (*Forschung*) and the retrieval of existing knowledge (*Recherche*).

Entry 2:
Subject**:** **Hard sciences**
English Term**:** **scientific research**
German Term**:** **wissenschaftliche Recherche**
Description**:** **retrieval of existing knowledge**

Transfer comment**:**
The concept *scientific research* is partially equivalent to two German concepts, *wissenschaftliche Forschung* and *wissenschaftliche Recherche,* because the term *wissenschaftlich* refers to both the hard sciences and to the social sciences and the humanities.

Entry 4:
Subject**:** **Social sciences, humanities**
English Term**:** **scholarly research**
German Term**:** **wissenschaftliche Recherche**
Description**:** **retrieval of existing knowledge**

Transfer comment**:**
The concept *wissenschaftliche Recherche* is partially equivalent to the two concepts in English, *scientific research* and *scholarly research*, because a distinction is made in German between original scholarship that creates new knowledge (*Forschung*) and the retrieval of existing knowledge (*Recherche).*

Figure 6 — Entries comprising information about the concepts in Figure 5

By creating four monolingual concept entries, the user can choose which monolingual entries should be linked together to signify quasi-equivalents. This approach is more powerful for documenting concepts at the language level than the approach where all languages are in one entry. However, most commercially-available TMSs do not follow this approach. Transfer comments could be used in multilingual entries to explicate the relationships shown here.

8.3 Application-oriented approaches

A TDC can be managed using a relational database management system or it can be stored as structured documents with markup based on formats that are typically defined using eXtensible Markup Language (XML). Once the choice between a database management system and a file with markup has been made, a data model can be developed. For a relational database, a typical method of describing a data model is an ER (entity-relationship) diagram or a UML (unified modelling language) diagram. For an XML document, typical methods of describing a data model include a DTD (document type definition) or an XML schema, both of which can be represented by means of a tree structure diagram.

To conform to this International Standard, all XML formats and relational database applications for terminological data shall be based on:

— the metamodel specified in ISO 16642,

— a data model defined by means of a UML diagram, an ER diagram, an XML DTD or an XML schema, and

— a DCS derived from the DCR specified in ISO 12620.

When defining the data structure of a TMS, there are two approaches, depending on whether or not the TMS will contain legacy data. The latter situation is described in 8.5.

8.4 Data modelling examples

The first step is to choose data categories for the TMS based on the principles described in Clause 7. The second step is to define the data structure using a UML diagram. The third step is to assign the data categories to their relevant positions in that structure. The resulting model forms the basis for implementing the TMS.

If data are going to be stored in a relational database, the UML diagram can be used as a basis for elaborating the database tables. If data are going to be collected using an XML-based TMS, the UML diagram can be used as a basis for elaborating a DTD or an XML schema.

The following list contains a sample set of data categories that are used in the following sections describing the development of a data model for a hypothetical sample TMS:

— Subject Field;

— Definition;

— Note;

— Term;

— Part of Speech;

— Source Identifier (combinable with *definition, term, context, note*);

— Context;

— Transfer Comment;

— Originating Person;

— Origination Date.

Figure 7 contains a UML diagram covering the above data categories for one possible implementation of a TMS. The boxes signify classes and the lines signify relationships (associations) between the classes. For each class, information on the multiplicity of a class is given, which is to say, the number of occurrences of a class participating in the relationship as well as information on whether the participation is optional or mandatory. In UML the following symbols are used:

1	exactly one occurrence
0..1	zero or one occurrence
1..*	one or more occurrences
0..*	zero, one or more occurrences

In this model, each terminological entry comprises one or more Language Sections. Each terminological entry is assigned to one subject field, but one subject field may be linked to zero, one or more terminological entries. In the example, each language section can have only one */definition/*, which means that there exists a one-to-one relationship between Language Section and */definition/*. The zero means that the */definition/* is optional. The same relationship exists between Language Section and /note/. In many cases, definitions are mandatory and it is often desirable to be able to store several definitions, such as, for instance, different definitions aiming at different target groups. Also, it may be desirable to store several versions of a definition as it goes through several stages of approval. This would require that the information on multiplicity between Language Section and */definition/* be changed (1..1 to 0..*). In this particular model, there is only one */source identifier/* for each */definition/*. In cases where input for a definition is found in a source but a terminologist changes or reformulates the definition, it may be desirable to be able to store both the original */source identifier/* and the name of the terminologist as a source. In the metamodel specified in ISO 16642, a definition may also be at the concept (terminological entry, TE) level, in which case this one */definition/* applies to all Language Sections. This is not the approach taken in the model shown in Figure 7.

Equivalence may be established between concepts in two or more languages on the basis of the characteristics of the concepts in question. If two concepts are fully equivalent, the concepts are identical and can be viewed as a single concept. If nearly equivalent concepts are recorded together in one entry, it means that they share characteristics to a degree that one may use the terms as equivalents in translations. The */transfer comment/* data category gives information about the nature of the equivalence relation between two such closely related terms stored in a single terminological entry. For example, there may be a comment on the specific conceptual characteristics that match between an English and a German term, and another comment between the English term and a French term.

One term may have one or several sources, and one source may be assigned to many terms, which means that there is a many-to-many relationship between */term/* and */source identifier/*. There is also a many-to-many relationship between */term/* and */context/,* since one term can be found in several contexts and one context can contain several terms. In this model there will not always be a */context/*, as the */context/* is optional. For simplicity's sake, the model in Figure 7 shows only one item of information (*/part of speech/*) associated with one */term/*, although a terminological entry could contain a variety of other items. Since each */part of speech/* value is related to many terms, there will exist a one-to-many relationship between the classes */part of speech/* and *TermSection*.

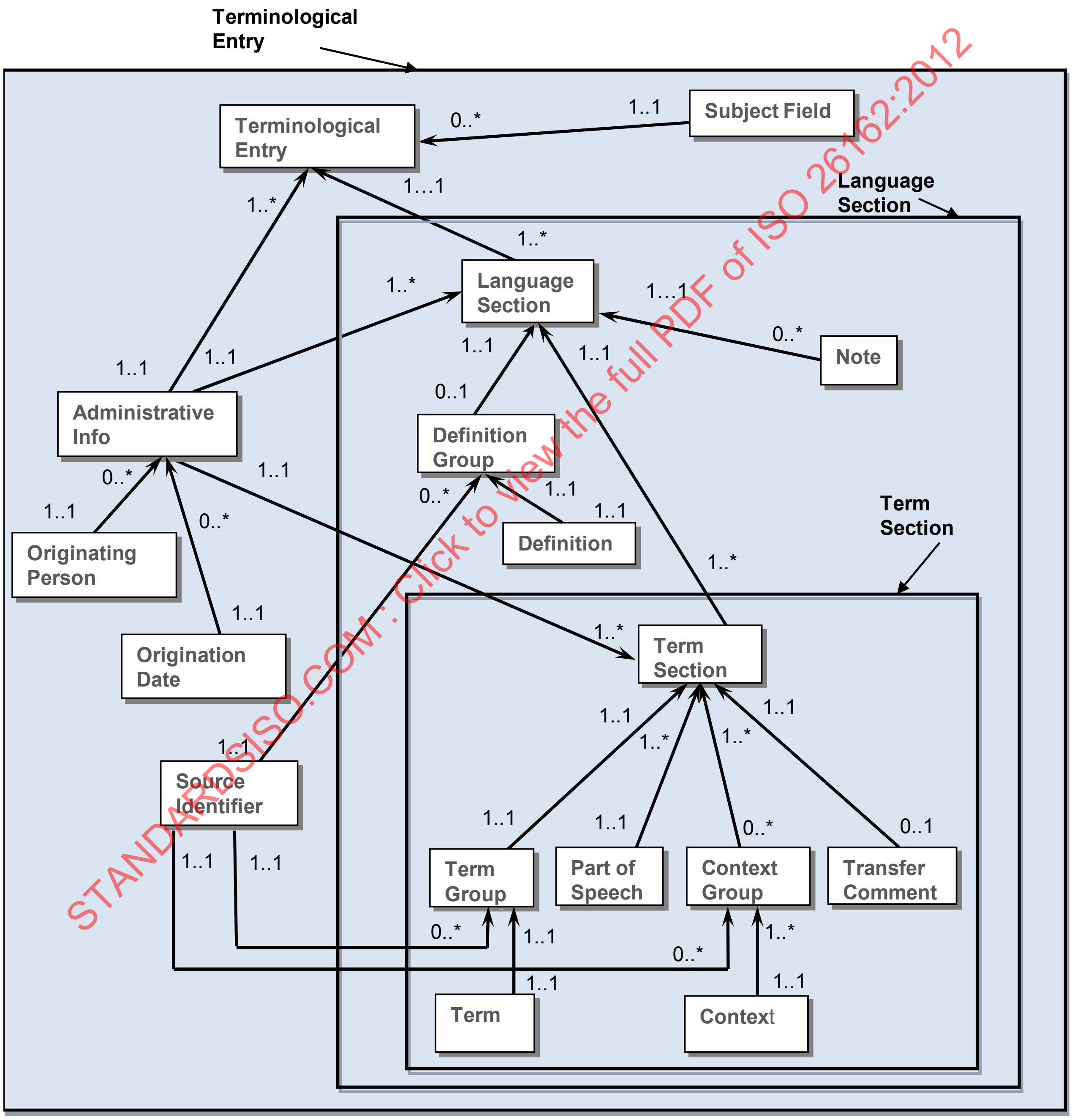

Figure 7 — UML diagram for basic data categories

If the TMS is to be implemented using a relational database management system, the data model in Figure 7 can be used to elaborate the database tables. Figure 8 shows some of the tables corresponding to the model in Figure 7. The names of the tables and columns used here correspond to the names of the terminological metamodel from ISO 16642 and the names of the data categories found in the DCR.

TerminologyEntry

#TE_ID	Subject_field_ID

SubjectField

#Subject_field_ID	Subject_field

LanguageSection

#LS_ID	LANGUAGE	TE_ID

TermSection

#TS_ID	LS_ID	Transfer_Comment

Definition

DEFINITION_ID	DEFINITION	LS_ID	SOURCE_IDENTIFIER_ID

Figure 8 — Table structure corresponding to some of the classes in the model in Figure 7

Figure 9 shows a DTD corresponding to the UML diagram in Figure 7.

The DTD in Figure 9 reflects a structure in which one entry comprises one concept or two or more nearly equivalent concepts with no information about the equivalence relations between concepts. Users should refer to ISO 30042 and the formats in ISO 16642 (including Geneter) for a standard XML vocabulary. For demonstration purposes, the example defines a fictitious XML vocabulary.

```
<!Element    Termbase              (TerminologicalEntry+)>
<!Element    TerminologicalEntry   (SubjectField, LanguageSection+, AdministrativeInfo)>
<!Element    SubjectField          (#PCDATA)>
<!Element    LanguageSection       (DefinitionGroup?,Note*, TermSection+, AdministrativeInfo)>
<!Element    DefinitionGroup       (Definition, SourceIdentifier)>
<!Element    Definition            (#PCDATA)>
<!Element    SourceIdentifier      (#PCDATA)>
<!Element    Note                  (#PCDATA)>
<!Element    TermSection           (TermGroup, PartOfSpeech, ContextGroup*,
                                   TransferComment?, AdministrativeInfo)>
<!Element    TermGroup             (Term, SourceIdentifier)>
<!Element    Term                  (#PCDATA)>
<!Element    PartOfSpeech          (#PCDATA)>
<!Element    ContextGroup          (Context, SourceIdentifier)>
<!Element    Context               (#PCDATA)>
<!Element    TransferComment       (#PCDATA)>
<!Element    AdministrativeInfo    (OriginatingPerson, OriginationDate)>
<!Element    OriginatingPerson     (#PCDATA)>
<!Element    OriginationDate       (#PCDATA)>
```

Figure 9 — DTD

XML tools exist which enable the user to write a DTD or an XML schema, which can then be presented in a diagram (a hierarchical element tree) or to draw a diagram, which can then be presented in XML syntax. Figure 10 contains a hierarchical element tree reflecting the DTD in Figure 9.

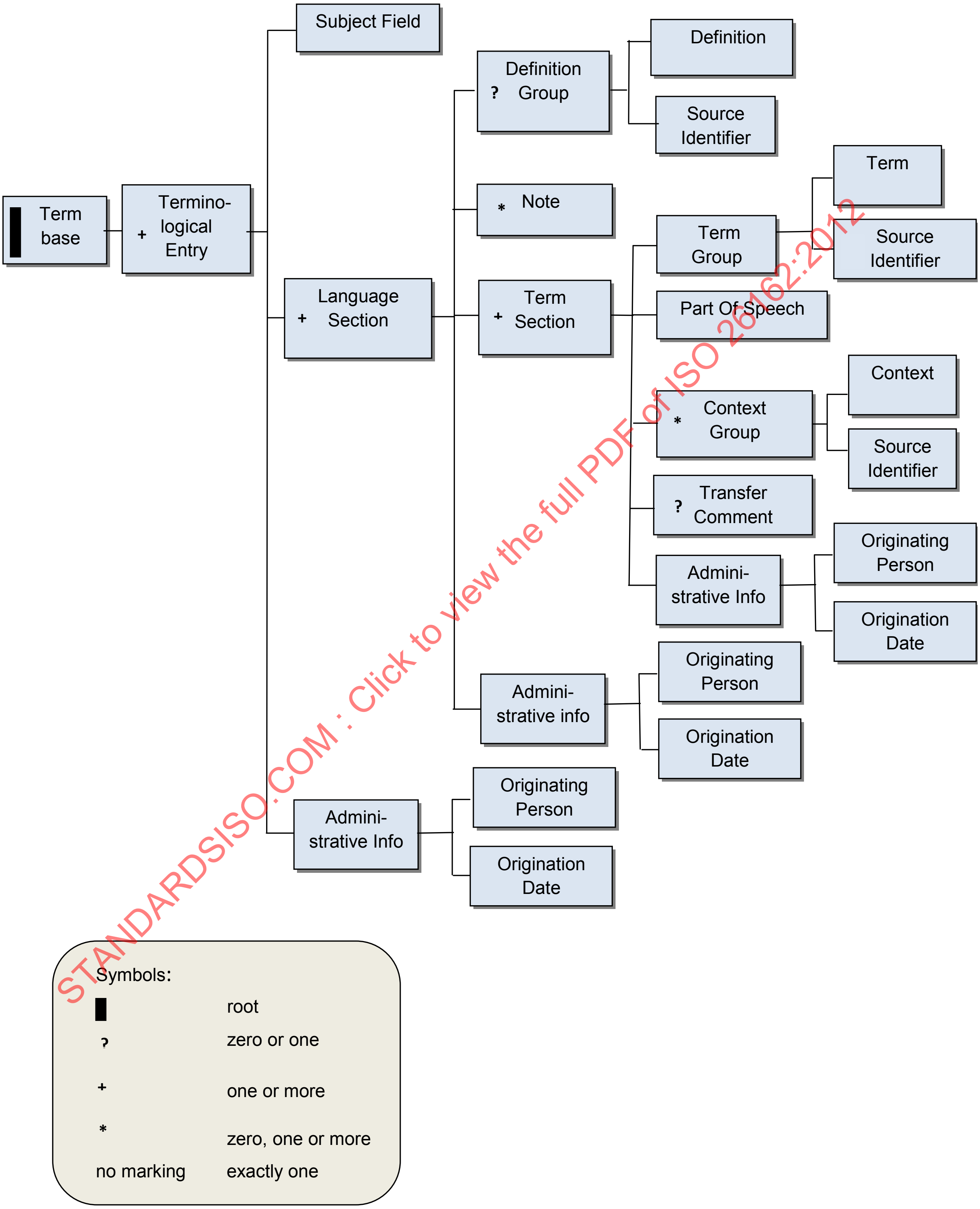

Figure 10 — Hierarchical element tree reflecting the DTD in Figure 9

Figure 11 shows the entry from Figure 4 with XML tags corresponding to the DTD in Figure 9 and the hierarchical element tree in Figure 10.

NOTE For purposes of simplicity and brevity, administrative information is not shown in Figure 4, nor is it reiterated in all the required positions in this example (LanguageSection, TermSection). It is shown once for the terminological entry.

```
<Termbase>
  <TerminologicalEntry>
    <SubjectField>Protocols</SubjectField>
    <LanguageSection xml:lang="en">
        <DefinitionGroup>
            <Definition>particular software implementation of a computer networking protocol
            suite</Definition>
            <SourceIdentifier>http://en.wikipedia.org/wiki/Protocol_stack</SourceIdentifier>
        </DefinitionGroup>
        <TermSection>
            <TermGroup>
                <Term>protocol stack</Term>
                <SourceIdentifier>IEE 802.11</SourceIdentifier>
            </TermGroup>
            <PartOfSpeech>noun</PartOfSpeech>
            <ContextGroup>
              <Context>Another standard architectural model that is often used to describe a network
              protocol stack is the OSI reference model. This model consists of a seven layer
              protocol stack.</Context>
              <SourceIdentifier> http://www.cisco.com/univercd/cc/td/doc/product/iaabu/centri4/user
              /scf4ap1.htm</SourceIdentifier>
            </ContextGroup>
        </TermSection>
        <TermSection>
            <TermGroup>
                <Term>stack</Term>
                <SourceIdentifier>http://www.w3.org/DesignIssues/Stack.html</SourceIdentifier>
            </TermGroup>
            <PartOfSpeech>noun</PartOfSpeech>
        </TermSection>
        <TermSection>
            <TermGroup>
                <Term>stack of interconnected protocols</Term>
                <SourceIdentifier>http://www.w3.org/DesignIssues/Stack.html</SourceIdentifier>
            </TermGroup>
            <PartOfSpeech>noun</PartOfSpeech>
        </TermSection>
    </LanguageSection>
    <LanguageSection xml:lang="de">
        <DefinitionGroup>
            <Definition>konzeptionelle Architektur von mehreren Protokollschichten, die direkt
            aufeinander aufbauen</Definition>
            <SourceIdentifier>http://www.esk.fraunhofer.de/projekte/automotive/TIP.jsp
            </SourceIdentifier>
        </DefinitionGroup>
        <TermSection>
            <TermGroup>
                <Term>Protokollstapel</Term>
                <SourceIdentifier> http://www.winfuture-forum.de.html</SourceIdentifier>
            </TermGroup>
            <PartOfSpeech>noun</PartOfSpeech>
            <ContextGroup>
                <Context>Daten, die über ein Netz übertragen werden, werden von einem Netzprotokoll
                des Protokollstapels nach dem anderen verarbeitet.</Context>
                <SourceIdentifier>http://de.wikipedia.org/wiki/Protokollstapel</SourceIdentifier>
            </ContextGroup>
        </TermSection>
    </LanguageSection>
    <AdministrativeInfo>
        <OriginatingPerson>J.Doe</OriginatingPerson>
        <OriginationDate>2011-03-22</OriginationDate>
    </AdministrativeInfo>
  </TerminologicalEntry>
</Termbase>
```

Figure 11 — Complete entry from Figure 4 expressed using XML

8.5 Accounting for legacy data

If there is a need to incorporate legacy data into the TMS, the structure and content (data categories) of the legacy data will need to be evaluated. Is the existing set of data categories adequate? Do they conform to the principles defined in this and other ISO/TC 37 standards? The effort involved in modifying the data categories in the legacy data to bring them into conformance with the new TMS should be estimated, and weighed against the benefits. Some data categories in the legacy data may not be needed in the new TMS and may therefore be discarded to reduce migration effort and to keep the new TMS uncluttered. Data categories in the new TMS that require significant modifications to the legacy data should be evaluated carefully to confirm whether they are necessary and to see if they can be modified in any way to reduce the effort. It may also be possible to adopt a strategy whereby the non-compliant data categories in the legacy data are imported into the new TMS into temporary read-only fields that are reserved exclusively for the legacy data, while newly entered data use standardized fields.

There are three cases to consider:

a) legacy data are stored in a database management system, in which the structure is made explicit;

b) legacy data are stored in a flat file (for instance word processor or spreadsheet file) with no explicit markup of the data structure, or maybe they only exist in printed form, and even if the printed text is scanned, there will be no explicit logical markup of the data structure in the resulting file;

c) legacy data are stored in an XML or other marked-up file where the data structure is explicitly expressed.

Even if data are already stored in a TMS in which the structure is made explicit, it is likely going to be necessary to define a new structure for the new TMS by means of a UML diagram, a DTD or an XML schema. The old structure may be inadequate; for example, there may be a need to add new data categories, or to split up existing data categories.

If data already exist, but with no explicit markup of the data structure [case b)], data modelling will involve the following three steps:

1) analysis and description of the existing data categories and their structure;

2) making the structure of the data explicit, for instance by means of a parser or some sort of conversion utility;

3) definition of the data structure which should form the basis for implementing the TMS.

The analysis of the data categories in legacy data should be carried out in coordination with the application-specific DCS described in Clause 7. It is possible to develop a UML diagram on the basis of the hierarchical element tree in Figure 10 (representing the DTD in Figure 9). First, all elements of the diagram from Figure 10 are represented as UML classes, and the information on multiplicity, which is known from this diagram, is inserted. This process will result in a first draft diagram.

Not all information about multiplicity can be deduced from the diagram in Figure 10. For instance, one subject field is assigned to many concepts, and one originating person may be responsible for many concepts. Information about multiplicity that is known from data analysis, but not represented in the tree structure, should be added to the UML diagram.

In the metamodel for ISO 16642, definitions (which are a kind of descriptive information) may reside at the level of the terminological entry or at the language section level. In the model in Figure 7, the definition resides at the language section level.

9 Implementing the TMS

Implementing the TMS takes place after the project stages described in Clause 5 have been carried out. The TMS and its subsystems are implemented according to system requirements and design priorities by developing new software or by configuring an existing TMS product.

The systems developers should work closely with the persons responsible for establishing the system requirements and describing the design priorities.

Systems developers will typically also create an overview of the functionalities that they will develop, such as by showing a mock-up of the screens. Figure 12 shows a simplified example of such an overview.

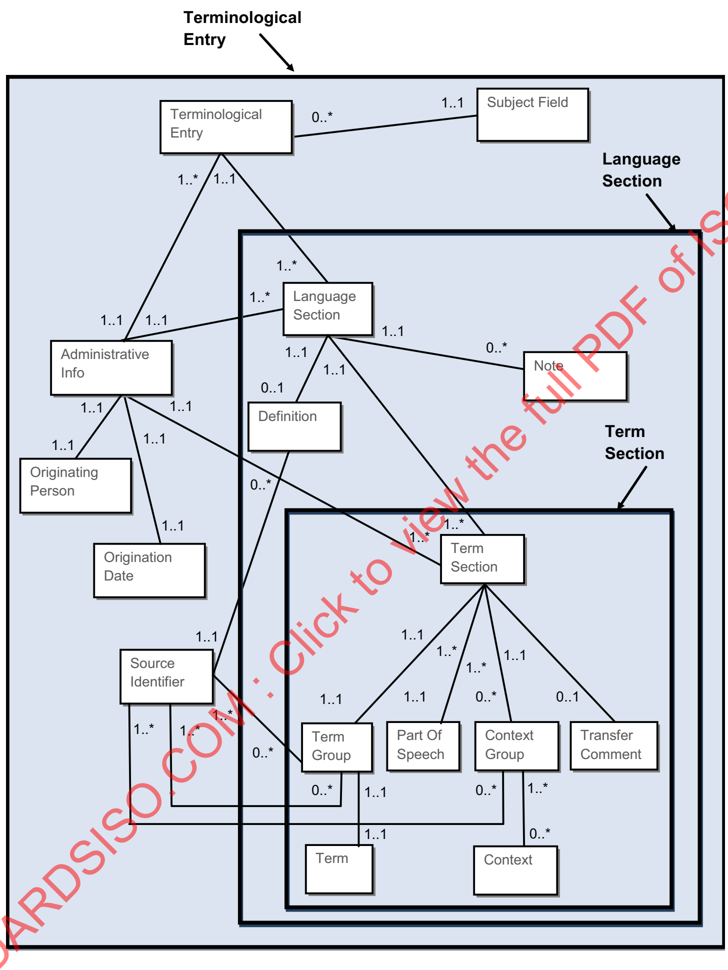

Figure 12 — Example of overview of functionalities in a TMS

Systems developers will use this overview to create, for example, HTML or PowerPoint pages showing the layout of the screens of the TMS.

The next step is to actually develop a first prototype of (parts of) the TMS, and present it to the persons responsible for establishing the system requirements and the design priorities as well as to potential end users.

It is essential to document and test the components and their function within the whole TMS as they are implemented.

10 Deploying the TMS

10.1 Deployment activities

Deployment of the TMS refers to all the activities that occur after the TMS has been developed and tested in order to make it fully operational across the organization. These activities include the following:

— preparing documentation, online help, tutorials, and educational materials;

— developing a support and maintenance plan and engaging the required services;

— identifying any interdependencies with other parts of the organization, other stakeholders, and other software or programs, and ensuring that these interdependencies are synchronized (see Clauses 5 and 6);

— making a pre-announcement to forewarn users that the TMS will be released on a certain date;

— delivering the TMS;

— announcing availability;

— providing training;

— conducting on-going promotional activities;

— monitoring user satisfaction and implementing user feedback.

The deployment tasks should actually have been identified and described in the original plan for developing the TMS, as described in Clause 5.

Some of these tasks should also be completed in parallel with the development phase, such as the preparation of documentation and educational materials and engaging support services.

10.2 Preparing documentation, help, and educational materials

To optimize user acceptance of the TMS, system instructions, prompts, responses, help functions, and training programmes should be clear and comprehensive.

Documentation, online help, and educational materials should be prepared by a technical writer who has expert knowledge of the TMS. The writer is preferably a member of the extended development team and has been involved in the project since the design stage. The writer should start early and follow the various stages of the system's development. Writers should participate in design discussions, development status meetings, usability feedback sessions, and marketing plans in order to be fully informed about all aspects of the TMS.

The writer should review all text that is written by the software developers, such as the error messages and the user interface labels, as well as marketing material, to ensure that the information is clear and consistent. Likewise, other members of the development team should review the documentation created by the writer to ensure technical accuracy.

An editor should perform a final review of the documentation and other end-user materials for style and clarity, especially if several writers are creating the materials.

In addition to information about how to use the TMS, some of the fundamental concepts of terminology management and terminology use should be described.

10.3 Providing support and maintenance

It is critical to have all the support and maintenance services in place before the TMS is deployed. This includes staff, problem submission and problem tracking systems, and a means for delivering fixes and updates. Support services should be arranged through a formal contract, even if provided internally.

Data should be tracked systematically for any problems that arise, because these data are valuable for forecasting future support requirements.

10.4 Meeting stakeholder dependencies

During the software design and planning stage, the various stakeholders and users of the TMS were identified (see Clauses 5 and 6).

Before the TMS is deployed, it is important to review any interdependencies that may exist with these various stakeholders and take the necessary action to resolve them. For instance, if the TMS is intended to be integrated with any translation software or content authoring tools, an integration plan should be in place and there should be no unresolved issues affecting the integration. Or, if translation vendors are required to use the TMS to ensure translation quality, they shall all have access to the TMS as soon as it is deployed and be trained on the TMS.

10.5 Announcing and promoting the TMS

Stakeholders shall be informed some time before the actual launch date so that they can take any necessary measures within their own teams to be ready to use the TMS as soon as it is launched. Announcements should be prepared ahead of time and reviewed by several members of the team.

After the initial announcements, some promotional information should be sent out on a periodic basis to remind people about the TMS.

10.6 Delivering the TMS

Delivery refers to the activities required to actually put the TMS into the hands of the end users. For Web-based applications, this means installing the TMS on the production server. For stand-alone applications, it means packaging the TMS into the delivery media, such as a compressed file that can be sent by e-mail or made available for download on the Web. Delivery functionality shall also be tested, such as accessing the Web sites, downloading the files, decompressing the files, and performing the installation.

For stand-alone applications, if updates need to be provided on a periodic basis, a plan for delivering these updates should be prepared before the TMS is launched.

10.7 Providing training

A training plan should be prepared as part of the original software development plan. The training plan should detail the types of training that will be provided, the audiences, schedules, required materials, and so forth.

Smaller and shorter training sessions that are customized for the various types of users are most effective. Repeat sessions will be required to accommodate people's availability. Providing a downloadable tutorial or pre-recorded training presentation enables users to take the training when it is convenient for them.

10.8 Measuring user satisfaction

After deployment, user satisfaction should be measured periodically, through a survey, for example. The results should be gauged against the original objectives of the TMS, and analysed for trends and patterns. This will help to validate the effectiveness of the TMS and identify areas in need of improvement.

11 User interfaces

11.1 Designing the user interface

Designing the user interface is a very important step. The user interface comprises the screens and functions that are used by people to interact with the TMS, including viewing and updating terminological entries, searching, importing, exporting, creating customized user profiles, backing up the system, and so forth. It also ensures that the terminological information is presented in an adequate, unambiguous and clear way. Although general guidelines and standards for designing user interfaces for software applications should be followed, specific aspects have to be taken into account for a TMS.

When arranging and displaying the content of a TMS, typographical conventions used to save space in printed dictionaries (compression mechanisms) should be avoided since, in online resources, there is less need for a compact display of information than in printed resources.

The display and layout mechanisms described in the next subclauses represent different approaches to designing the user interface of a TMS. The method that is preferred by, and most appropriate to, the target user group shall be chosen, and it shall be consistently applied throughout the whole TMX. Too much variation between different methods will confuse the user; however, different methods for displaying and arranging terminological data may be useful for different user groups and terminology management tasks.

11.2 Displaying terminological data categories

To display terminological data categories, use one of the methods described in this section.

— Field-oriented display with data category name preceding or positioned above the field (see Figure 13).

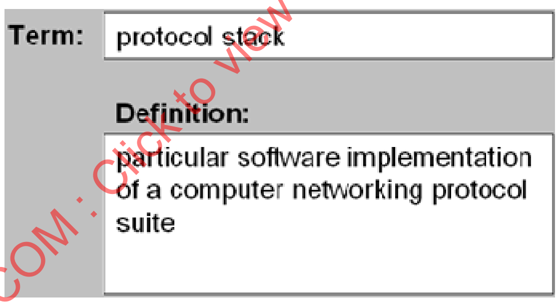

Figure 13 — Data category name preceding and positioned above the field

— Field-oriented display with data category name in message line when pointing with the cursor to the field (see Figure 14).

TerminologicalEntry

#TE_ID	Subject_field_ID

SubjectField

#Subject_field_ID	Subject_field

LanguageSection

#LS_ID	LANGUAGE	TE_ID

TermSection

#TS_ID	LS_ID	Transfer_Comment

Definition

DEFINITION_ID	DEFINITION	LS_ID	SOURCE_IDENTIFIER_ID

Figure 14 — Data category name display in message line

— Table-oriented display with data category name on top of the column (see Figure 15).

Term	Part of Speech	Note
heap stack	noun	
memory stack	noun	
protocol stack	Noun	
stack	noun	
stack of interconnected protocols	noun	

Figure 15 — Table-oriented display with data category name on top of the column

— Structured display with text layout instead of data category name (see Figure 16).

protocol stack *noun*

particular software implementation of a computer networking protocol suite

http://en.wikipedia.org/wiki/Protocol_stack - 22.08.2008

Figure 16 — Using layout for different data categories

11.3 Displaying and arranging terminological entries

The terminological entry itself is probably the most important component of the user interface since it is used most frequently. Fixed templates are commonly used to arrange information on a computer screen, but there are other approaches as well. The content of different data categories shall be presented in different fields. Various layouts are shown in Figures 17 to 21.

```
<!Element   Termbase               (TerminologicalEntry+)>
<!Element   TerminologicalEntry    SubjectField, LanguageSection+, AdministrativeInfo)>
<!Element   SubjectField           (#PCDATA)>
<!Element   LanguageSection        (DefinitionGroup?,Note*,TermSection+,AdministrativeInfo)>
<!Element   LanguageSection        (…|…|…)#IMPLIED>
<!Element   DefinitionGroup        (Definition,SourceIdentifier)>
<!Element   Definition             (#PCDATA)>
<!Element   SourceIdentifier       (#PCDATA)>
<!Element   SourceIdentifier       (…|…|…)#IMPLIED>
<!Element   Note                   (#PCDATA)>
<!Element   TermSection            (TermGroup,PartOfSpeech,ContextGroup*,
                                    TransferComment?,AdministrativeInfo)>
<!Element   TermGroup              (Term,SourceIdentifier)>
<!Element   Term                   (#PCDATA)>
<!Element   PartOfSpeech           (#PCDATA)>
<!Element   PartOfSpeech           (noun|verb|adj|adv|…)#IMPLIED>
<!Element   ContextGroup           (Context,SourceIdentifier)>
<!Element   Context                (#PCDATA)>
<!Element   TransferComment        (#PCDATA)>
<!Element   AdministrativeInfo     (OriginatingPerson,OriginationDate)>
<!Element   OriginatingPerson      (#PCDATA)>
<!Element   OriginationDate        (#PCDATA)>
```

Figure 17 — Fixed and defined layout

ID1234

<protocols>

English

protocol stack ***noun***

particular software implementation of a computing networking protocol suite
http://en.wikipedia.org/wiki/Protocol_stack-22.08.2008

German

Protokollstapel ***noun***

konzeptionelle Architektur von mehreren Protokollschichten, die direkt aufeinander aufbauen
http://www.esk.fraunhofer.de-22.08.2008

Figure 18 — Variable information layout

- *Entry*
  - *ID* ***1234***
  - ***SubjectField*** protocols
  - ***English***
    - ***TermSection***
      - *Term* protocol stack
      - *POS* noun
    - ***Description***
      - ***Definition*** particular software implementation of a computing networking protocol suite
      - *Source* http://en.wikipedia.org/wiki/Protocol_stack-22.08.2008
  - ***German***
    - ***TermSection***
      - *Term* Protokollstapel
      - *POS* noun
    - ***Description***
      - ***Definition*** konzeptionelle Architektur von mehreren Protokollschichten, die direkt aufeinander aufbauen
      - *Source* http://www.esk.fraunhofer.de-22.08.2008

Figure 19 — Hierarchically structured layout

ID1234 *<protocols> ENG:* **protocol stack, *noun, Def:*** particular software implementation of a computing networking protocol suite (http://en.wikipedia.org/wiki/Protocol_stack-22.08.2008) *DEU:* **Protokollstapel, *noun, Def:*** konzeptionelle Architektur von mehreren Protokollschichten, die direkt aufeinander aufbauen http://www.esk.fraunhofer.de-22.08.2008

Figure 20 — Compressed layout

ID	Lang	Subject Field	Term	POS	Definition	Source
1234	ENG	protocols	protocol stack	noun	particular software implementation of a computing networking protocol suite	http://en.wikipedia.org/wiki/Protocol_stack-22.08.2008
1234	DEU	protocols	Protokollstapel	noun	konzeptionelle Architektur von mehreren Protokollschichten, die direkt aufeinander aufbauen	http://www.esk.fraunhofer.de-22.08.2008

Figure 21 — Table-oriented layout

12 Inputting and editing data

12.1 Manually inputting data

A TMS shall permit data to be input manually. This form of data input is practical for adding limited volumes of data. Figure 22 shows a typical example of an interface for manually inputting data.

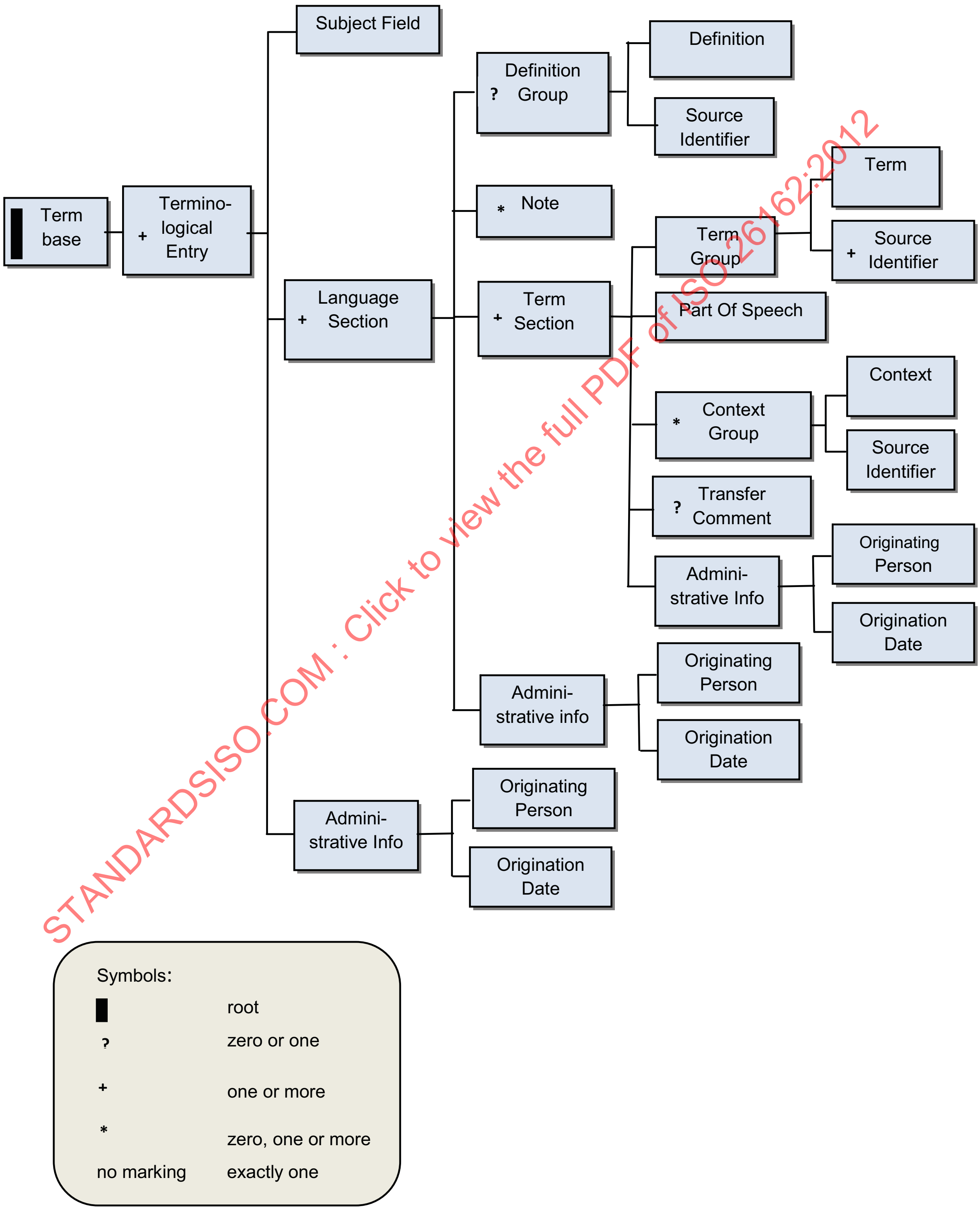

Figure 22 — Example of a data input interface

In the sample entry, the first section consists of the fields *Subject* (for */subjectField/*, a concept-related data category) and *Language*. The next section contains information about the term; this section can be repeated as many times as necessary to document additional (synonymous) terms. The third section contains concept-related information (in addition to *Subject,* which is at the top) – that is, information that is valid for all synonymous terms – for instance in this example*, general definition, reference for definition, location (in reference)* and *comment*.

Ideally, it should be possible for users to customize the data entry interface so that it contains only the fields they need. Customizable entry templates can be made available for this purpose. Several default templates could be made available for selection, based on standard user roles such as terminologist, translator, technical writer, and so forth.

Figure 23 shows a more detailed layout that may be suitable for a terminologist.

```
<Termbase>
  <TerminologicalEntry>
        <SubjectField>Protocols</SubjectField>
    <LanguageSection xml:lang="en">
                <DefinitionGroup>
            <Definition>particular software implementation of a computer networking protocol
            suite</Definition>
                <SourceIdentifier>http://en.wikipedia.org/wiki/Protocol_stack</SourceIdentifier>
                </DefinitionGroup>
                <TermSection>
                        <TermGroup>
                  <Term>protocol stack</Term>
                                <SourceIdentifier>IEE 802.11</SourceIdentifier>
                        </TermGroup>
                        <PartOfSpeech>noun</PartOfSpeech>
                        <ContextGroup>
                <Context>Another standard architectural model that is often used to describe a network
                protocol stack is the OSI reference model. This model consists of a seven layer
                protocol stack.</Context>
                <SourceIdentifier> http://www.cisco.com/univercd/cc/td/doc/product/iaabu/centri4/user
                /scf4apl.htm</SourceIdentifier>
            </ContextGroup>
                </TermSection>
                <TermSection>
                        <TermGroup>
                                <Term>stack</Term>
                  <SourceIdentifier>http://www.w3.org/DesignIssues/Stack.html</SourceIdentifier>
                        </TermGroup>
                        <PartOfSpeech>noun</PartOfSpeech>
                </TermSection>
                <TermSection>
                        <TermGroup>
                  <Term>stack of interconnected protocols</Term>
                  <SourceIdentifier>http://www.w3.org/DesignIssues/Stack.html</SourceIdentifier>
            </TermGroup>
                        <PartOfSpeech>noun</PartOfSpeech>
                </TermSection>
    </LanguageSection>
    <LanguageSection xml:lang="de">
        <DefinitionGroup>
                <Definition>konzeptionelle Architektur von mehreren Protokollschichten, die direkt
            aufeinander aufbauen</Definition>
            <SourceIdentifier>http://www.esk.fraunhofer.de/projekte/automotive/TIP.jsp
            </SourceIdentifier>
        </DefinitionGroup>
        <TermSection>
                <TermGroup>
                                <Term>Protokollstapel</Term>
                  <SourceIdentifier> http://www.winfuture-forum.de.html</SourceIdentifier>
            </TermGroup>
                        <PartOfSpeech>noun</PartOfSpeech>
                        <ContextGroup>
                  <Context>Daten, die über ein Netz übertragen werden, werden von einem Netzprotokoll
                  des Protokollstapels nach dem anderen verarbeitet.</Context>
                  <SourceIdentifier>http://de.wikipedia.org/wiki/Protokollstapel</SourceIdentifier>
                        </ContextGroup>
                </TermSection>
    </LanguageSection>
    <AdminstrativeInfo>
        <OriginatingPerson>J.Doe</OriginatingPerson>
        <OriginationDate>2011-03-22</OriginationDate>
    </AdministrativeInfo>
  </TerminologicalEntry>
</Termbase>
```


Figure 23 — Layout comprising more data categories

The data input interface shall permit repeatability of term-related information, which means that it shall be possible to add an unlimited number of terms, including all term-related information for each term, to a terminological entry. In the interfaces shown in Figures 22 and 23, when the user clicks the button with the plus sign next to the term field, a new set of fields for a new term is added to the entry. This capability supports the principle of term autonomy (*see* 7.2.3).

It should be possible to change the order of the terms in the entry. In the interface in Figure 22, the term order can be changed by means of the *arrow up* and *arrow down* buttons.

In the user interface in Figure 23, it is possible to add a new reference clicking on the pencil icon (see the field *Reference for term* in Figure 23), which will open a page for recording information about a reference. This is an example of a picklist field, where new values must be added frequently, even at the moment of editing the entry; therefore, adding new values should not be restricted to the system administrator. In the user interface in Figure 23, it is also possible to click on the magnifying glass icon to search for specific references. Additionally, it is possible to set up a list of references by using the function accessible from the menu bar. Figure 24 shows an image of the screen for creating a reference to a printed resource. This is an example of a shared resource.

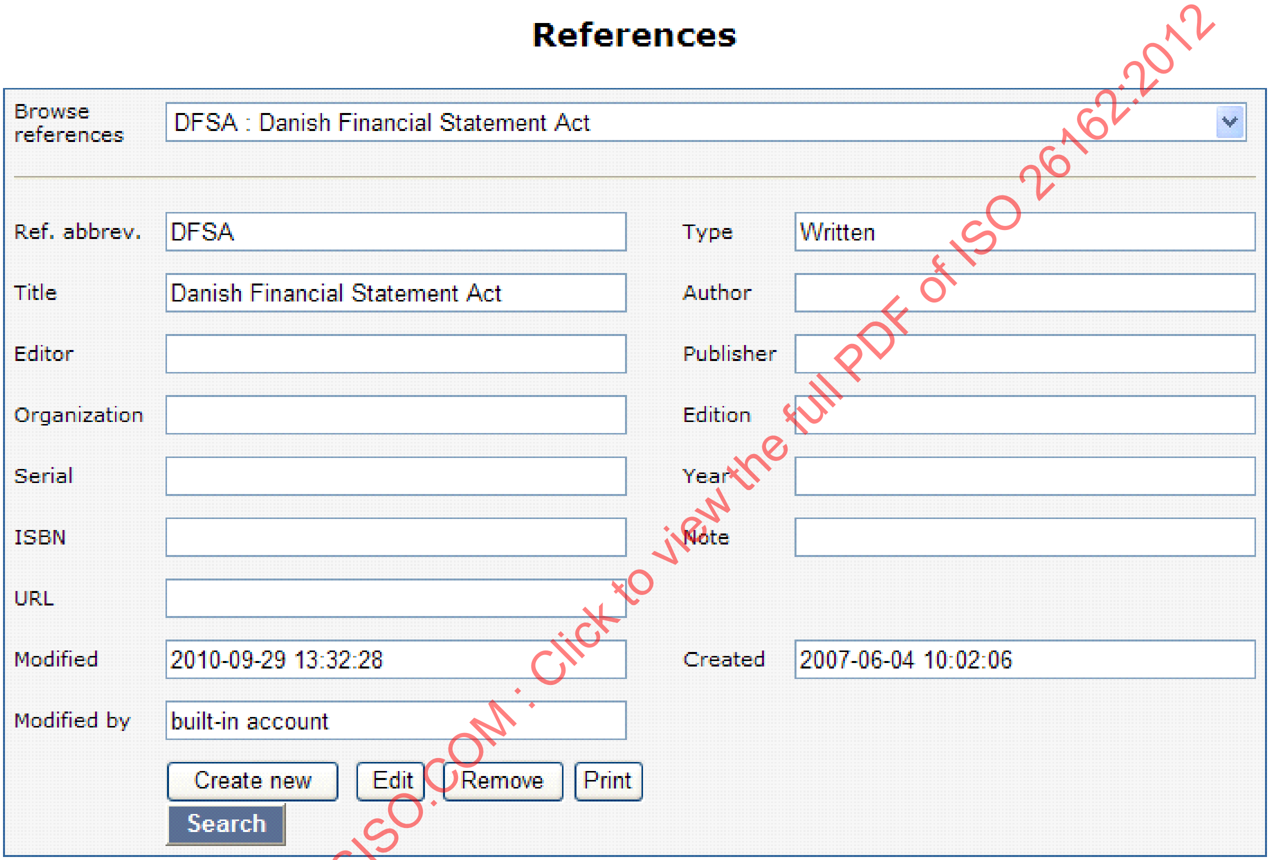

Figure 24 — Screen for creating a reference

The user interface should implement closed data categories such as, for example, */subject field/*, */language/*, */part of speech/* and */source/* reference by means of picklist fields. The use of picklist fields prevents data input errors. The system administrator should be able to define these picklist values when the database is initially configured. The administrator should also be able to add new values and modify the displayed format of the values at any time (see 12.5 concerning display values). The creation and modification of most types of picklist values should be restricted to certain user roles, such as the system administrator, in order to preserve data integrity. Figure 25 shows the picklist values available in the above user interface and the interface for updating the set of language values.

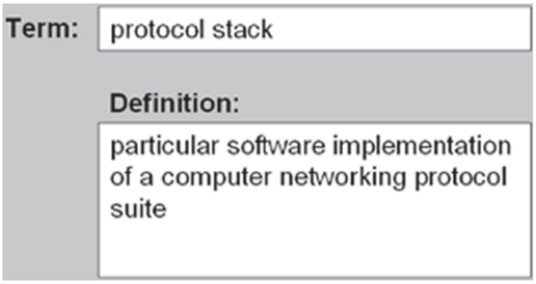

Figure 25 — Menu with picklist values and interface for updating the list of languages

It is important for the TMS to prevent unwanted doublettes during manual data input. At a minimum, a message shall be displayed if the user tries to add a term whose surface form matches an existing term in the database. Of course, it shall nevertheless be possible to create separate terminological entries for terms that have the same surface form, but different meanings, in order to accommodate homographs. If users type in a duplicate term, the system should help them decide whether the term they want to add represents a new concept (that is, it is a homograph) or an unwanted doublette by allowing them to check the existing entries that contain the term with the matching surface form. Figure 26 shows how an interactive doublette check might work. When the user clicks the link for the existing term, the existing entry is displayed.

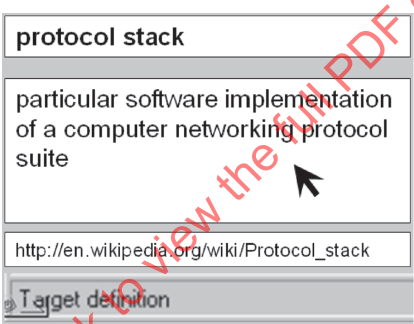

Figure 26 — Interactive doublette check

12.2 Importing data

Existing terminological data in machine-readable form can normally be transferred to a termbase by means of an import function. The data may have to be restructured (see also 8.5).

A TMS should allow for importing industry standard XML files for terminological data, for instance, XML formats that comply with ISO 16642, such as the family of TBX formats (ISO 30042):

a) user-defined XML format;

b) delimited text files.

When importing external data, it is important to ensure the integrity of both existing and incoming terminological entries by guaranteeing that no unintended changes are introduced into the original data. The database should be backed up before carrying out any import.

A TMS may typically offer the following import options:

1) append new entries to the database;

2) append an incoming entry to an existing entry;

3) replace an existing entry with the incoming entry;

4) ignore the incoming entry and keep the existing entry intact;

5) replace data from the database with imported data, and synchronize on entry numbers and perhaps other data categories such as */subject field/.*

Normally, these functions take into consideration only one language, as selected by the importer.

In option 2, if an entry in the import file contains a term that has the same surface form as a term in the database, new data categories may be added to the existing entry. In option 3, if an entry in the import file contains a term that has the same surface form as a term in the database, the existing entry will be replaced (overwritten). This action also assumes that the two entries describe the same concept but, in this case, only the information in the import file is kept. In option 4, if an entry in the import file contains a term that has the same surface form as a term in the database, the entry in the import file will be ignored (skipped) and the existing entry will remain unchanged. This action could be appropriate for one of two scenarios: (1) the two entries describe the same concept, but the information in the database is more accurate and the information in the import file should be ignored, or (2) the two entries describe different concepts, so the existing entry should be conserved, and the entry in the import file should be added, perhaps later in manual entry mode. For this purpose, the import routine should generate a log listing any ignored import entries.

In option 5, the import file contains entry numbers that correspond to the entry numbers in the database. This option is typically used to import data that have been previously exported from the TMS. Sometimes, it is more productive to make global changes or batch edits to large numbers of entries by editing an exported file rather than attempting to do this directly in the database. The imported entries replace the existing ones, but the system checks first to make sure that the entry numbers, and any other specified data categories, are the same before making the replacement.

Advanced TMSs should provide facilities for interactive doublette checking during import, allowing the user to decide whether to import a term whose surface form matches that of another term in the database

The user should be able to import subsets of data categories that are present in the import file and ignore others. For instance, import only selected languages, ignore administrative data categories such as dates, or ignore the definitions in the import file.

12.3 Editing data

Ideally, the user interface for editing data should look like the interface for inputting data, and it should provide the same features.

While editing, it should be possible for the user to see the original entry without changes. The original entry may for example be shown in a pop-up window when the user presses a button (View original).

When an entry is being edited by one user, the system shall lock it in order to prevent other users from editing the same entry.

12.4 Validating data

Data validation functions are needed to minimize errors during data input or editing. Most database systems already offer some of these capabilities, but if special data input software is developed, it is advisable to include validation functions.

EXAMPLE Examples of data validation:

— checking for doublettes: to determine that a term is stored only once;

— spell check: to determine that all terminological entries comply with predefined spelling rules;

— character check: to verify the character types used in an input field;

— completeness check: to ensure that data are present where required;

- format check: to determine that data conform to a specified layout;
- plausibility check: to determine that a value conforms to specified criteria;
- validity check: a check based on known conditions or constraints that apply to a given piece of information or result.

A very important feature is logging changes, which range from information on the date on which an entry or field was created or updated, to the identity of the user responsible for a change, to more detailed logs of previous versions, giving the full history of all information in the database. For example, it may be useful to be able to see all previous versions of a definition.

12.5 Automatically generating or modifying data

Some data can be automatically generated, such as the name of the person who created or modified an entry, and any related dates.

It should be possible to identify a default value for some fields, either for a given session or across the whole TMS. For example, during a work session, if a terminologist is creating multiple concept entries within the same language and subject field, it would increase productivity if the */language/* field and the */subject field/* could be pre-filled with the desired values. A system-wide default value may be desirable for certain fields that take a predominate value across the whole system. For instance, the */part of speech/* field could have the default value *noun* if it is pre-determined that most terms in the database will be nouns. This increases productivity by not requiring the user to select the part of speech for each entry. It should, of course, be possible for the user to change these values at any time.

It shall be possible to make changes to the display of picklist values automatically across the entire TMS. Codes shall be used to identify languages, subject fields, subsets, and other picklist values, internally in the TMS. Each code corresponds to a user-friendly surface form which is visible in the user interface. For instance, a customer subset for Acme Computers could have the internal code AC and the surface form Acme Computers. These codes and surface forms shall be stored only once in the TMS. In individual terminological entries, references to the picklist value are made internally by using the code, but the user sees and selects the surface form. If the company name changes later, such as to Acme Information Technology, the change to the surface form is made only once in the central entry, and the code AC remains unchanged. This enables all the surface forms to be automatically updated in all the terminological entries. See the example in Figure 24 and the information about shared resources in 7.3.5.

12.6 Adding cross-references

It should be possible to add links to other terminological entries, such as a link to the entry for an entailed term that is used in a definition or other text field, even if the term in the text does not have the same surface form as the term in the linked entry. For example, it should be possible to link the plural form “ports” in a definition to the entry that contains the term “port”. Links can point to a term within a concept entry or to the concept entry itself, and they shall not be ambiguous, for example by using the form of the term for link resolution (which would prevent “ports” from being linkable to “port”). Words can be polysemous, in which case each separate meaning of a word comprises a different term, and each shall be recorded in a separate entry. In this case, using the surface form of a term for link resolution will result in a situation where the link points to multiple entries. Unambiguous links shall be established by using more detailed criteria, such as a combination of the term ID and the concept ID.

The TMS should prevent invalid cross-references and links. See also 7.3.6.

12.7 Adding multimedia files

The terminological entry structure should allow links to multimedia files and other resources, for instance graphics, audio, AVI (Audio Video Interface), video, etc. It should also be possible to include hypertext links to Web-based resources. The inclusion of image maps, linking areas of an image to terminological entries, may be desirable. Figure 27 shows an entry that includes a multimedia file. Clicking on the icon displays a larger version of the file.

13 Search functions

13.1 Database search features

As in all database or information retrieval systems, the search functions in a TMS should enable users to rapidly access desired information.

Most typical users are interested in finding information about a term or a concept. For example, they want to read the definitions of a term in order to understand its meanings, they need to look up a target language equivalent for a text that they are translating, they want to confirm a word's spelling or check the contexts in which it is used, or they need to know if a term has abbreviations or synonyms.

Terminologists and administrators of the TMS may require more complex searches for data management purposes, for example to find entries that were updated on a specific date or by a specific person, to find entries that contain doublettes, to find entries where specific fields are empty, and so forth. Terminologists interested in completing the coverage of the TMS can benefit from systematic searches.

Typical user scenarios include:

— translators or localizers searching for a target language equivalent;

— subject field experts or technical writers searching for a definition;

— translators or writers interested in information about term usage;

— language planners or persons responsible for the corporate language of a company who need information about all terms representing a concept including their normative status;

— terminologists who need any information about a concept and its relationships to other concepts;

— database managers who need to find terminological entries according to a wide range of parameters, for database maintenance purposes.

All types of searching, filtering and browsing capabilities required by all users of the TMS shall be identified during the design stage (see Clause 6).

It should be possible to store search queries and filtering criteria as well as other user-specific retrieval settings (such as the layout of the search results, see Clause 14) in a user profile for reuse at a later date.

Term	Part of Speech	Note
heap stack	noun	
memory stack	noun	
protocol stack	noun	
stack	noun	
stack of interconnected protocols	noun	

Courtesy Wikipedia, as per http://creativecommons.org/licenses/by-sa/3.0/)

Figure 27 — Display of multimedia file

