INTERNATIONAL STANDARD

ISO/IEC 13250-5

First edition
2015-04-01

Information technology — Topic Maps —

Part 5: **Reference model**

Technologies de l'information — Plans relatifs à des sujets —

Partie 5: Modèle de réference

Reference number
ISO/IEC 13250-5:2015(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

ISO/IEC 13250-5 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 34, *Document description and processing languages.*

ISO/IEC 13250 consists of the following parts, under the general title *Information technology — Topic Map*:

— *Part 2: Data model*

— *Part 3: XML syntax*

— *Part 4: Canonicalization*

— *Part 5: Reference model*

— *Part 6: Compact syntax*

Introduction

The Topic Maps family of standards is designed to facilitate the gathering of all the information about a subject at a single location. The information about a subject includes its relationships to other subjects; such relationships may also be treated as subjects (subject-centric).

ISO/IEC 13250-2:2006 [1] provides a foundation for syntaxes and notations, such as those defined in ISO/IEC 13250-3 Topic Maps XML Syntax [2] and ISO/IEC 13250-4 Topic Maps Canonicalization [3]. Of necessity, ISO/IEC 13250-2:2006 [1] makes ontological commitments in terms of how particular subjects are identi_ed (topics, associations, occurrences), what properties are required, the tests to be used to determine whether two or more proxies represent the same subject, and other matters.

This part of ISO/IEC 13250 defines TMRM (Topic Maps Reference Model), which is more abstract and has fewer ontological commitments. Its purpose is to serve as a minimal, conceptual foundation for subject-centric data models such as ISO/IEC 13250-2:2006 [1], and to supply ontologically neutral terminology for disclosing these. It de_nes what is required to enable the mapping of different subject-centric data models together to meet the overall goal of the Topic Maps standards, that each subject has a single location for all the information about it.

TMRM also provides a formal foundation for related Topic Maps standards such as the ISO/IEC 19756 Topic Maps Constraint Language (TMCL) [4].

Information technology — Topic Maps — Part 5: Reference Model

1 Scope

This part of ISO/IEC 13250 specifies a formal model for subject maps, minimal access functionality and information retrieval from subject maps and a constraint framework governing the interpretation of subject maps.

Particular formalisms to constrain subject maps are not covered by this part of ISO/IEC 13250.

2 Subjects

A subject is defined in the Topic Maps family of standards as something which '[...] can be anything whatsoever, regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever' (*ISO/IEC 13250-2:2006* 5.3.1). According to the TMRM, there is only one representative for subjects: subject proxies (*proxies*).

3 Subject Proxies and Maps

Proxies consist of properties. These are key/value pairs which—in turn—may contain references to other proxies. This recursive relationship is defined via two postulated sets. One is the finite set of proxy *labels*, $\mathcal{L}$. The second set is $\mathcal{V}$, a finite set of values. It contains values (such as numbers, strings, etc.), and all the labels in $\mathcal{L}$.

A *property* is the pair $\langle k, v\rangle \in \mathcal{L} \times \mathcal{V}$. The first component of this pair is called the *key*, the other the *value* of the property. The set of all such properties is denoted as $\mathcal{P}$. Keys in properties are always labels, the values in properties may be labels or any other value from the value set $\mathcal{V}$.

EXAMPLE 1 Given the label `shoesize` and the integer 43, then $\langle \texttt{shoesize}, \texttt{43}\rangle$ is a property.

A *proxy* is a finite set of properties, $\{p_1, \ldots, p_n\}$, with $p_i \in \mathcal{P}$ (see Fig. 1). The multiset of all keys of a proxy x is retrievable via the function $keys(x)$, i.e. keys can occur more than once in a proxy with different values. The multiset of all values is $values(x)$. Particular values may appear more than once in one proxy.

EXAMPLE 2 A particular person may be represented by the following proxy: $\{\langle \texttt{shoesize}, \texttt{43}\rangle, \langle \texttt{beardcolor}, \texttt{white}\rangle, \langle \texttt{beardlength}, \texttt{verylong}\rangle\}$

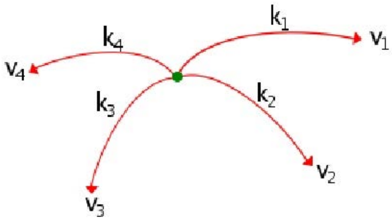

Figure 1 – Proxy Structure

The set of all proxies $\mathcal{X}$ is the set of all subsets of $\mathcal{P}$, $\mathcal{X} = 2^{\mathcal{P}}$.

The connection between proxies and their labels is modeled with a partial, injective function $\tilde{}: \mathcal{X} \mapsto \mathcal{V}$. It returns the label for a given proxy x whereby two different proxies never share the same label. The function is extended to values in that $\tilde{v} = v$.

NOTE 1 A proxy is defined by the totality of its properties. Individual properties can provide a basis for mapping multiple proxies of the same subject to each other.

NOTE 2 One proxy may contain several properties which all share the same key but have different values; or share the same value, but have differing keys.

A *subject map* (*map*) is a finite set of *proxies*. The set of all such maps is denoted as $\mathcal{M}$. As maps are simply sets of proxies, *generic merging of maps* is achieved via set union, $m \cup m'$.

NOTE 3 The model of subject maps described herein assumes no particular implementation technology or strategy.

4 Ontological Commitments

This part of ISO/IEC 13250 deliberately leaves undefined the methods whereby subject proxies are derived or created. No specific mechanism of subject identification is inherent in or mandated by this part of ISO/IEC 13250, nor does it predefine any subject proxies.

NOTE 1 Any subject proxy design choices would be specific to a particular application domain and would exclude equally valid alternatives that might be appropriate or necessary in the contexts of various requirements.

Two types of relationships, *ako* (subclass of) and *isa* (instance of), are defined. These predicates are always interpreted *relative* to a given map m:

a) Two proxies c, c' can be in a *subclass-superclass* relationship, $\text{ako}_m \subseteq m \times m$. In such a case, the same relationship can be stated either c *is a subclass of* c' or c' *is a superclass of* c.

   ako_m is reflexive and transitive. *Reflexive* means that any proxy is a subclass of itself, regardless whether the proxy is used as a class in the map or not: $x \text{ ako}_m\, x$ for all $x \in m$.

   Transitive means that if a proxy c is a subclass of another, c', and that subclasses c'', then c is also a subclass of c'', i.e. if $c \text{ ako}_m\, c'$ and $c' \text{ ako}_m\, c''$ then also $c \text{ ako}_m\, c''$ must be true.

NOTE 2 Circular subclass relationships may exist in a map.

b) Two proxies a, c can be in an *isa* relationship, $\text{isa}_m \subseteq m \times m$. In such a case, the same relationship can be stated either *a is an instance of c* or *c is the type of a*.

The *isa* relationship is non-reflexive, i.e. $x \text{ isa}_m x$ for no $x \in m$, so that no proxy can be an instance of itself. Additionally, whenever a proxy a is an instance of another c, then a is an instance of any superclass of c: if $x \text{ isa}_m c$ and $c \text{ ako}_m c'$, then $x \text{ isa}_m c'$ is true.

NOTE 3 Other definitions of the above relationships are possible with different properties than defined by the TMRM. Such definitions would appear in legends and would be distinguished from those defined in the TMRM.

NOTE 4 This part of ISO/IEC 13250 does not mandate any particular way of representing such relationships inside a map. One option is to model such a relationship simply with a property using a certain key (say `type`). An alternative way is to provide a proxy for each such relationship. Such relationship proxies could, for example, have properties whose keys are `instance` and `class`, or respectively `subclass` and `superclass`.

5 Navigation

Given a map m and particular proxies $x, y \in m$ in it, the following *primitive navigation operators* are defined:

a) A postfix operator $\downarrow$ to return the multiset of *all local keys* of a given proxy:

$$x\downarrow = keys(x) \tag{1}$$

b) A postfix operator $\uparrow_m$ to retrieve the multiset of *remote keys* of a proxy inside m. These are those where the given proxy (more precisely its label) *is* the value in another proxy:

$$x\uparrow_m = [k \mid \exists y \in m \colon \langle k, \tilde{x} \rangle \in y] \tag{2}$$

This is easily generalized to an operator for all values:

$$v\uparrow_m = [k \mid \exists y \in m \colon \langle k, \tilde{v} \rangle \in y] \tag{3}$$

c) A postfix operator $\rightarrow k$ to retrieve the multiset of *local values for a particular key* $k \in \mathcal{L}$:

$$x \rightarrow k = [v \mid \exists \langle k, v \rangle \in x] \tag{4}$$

Using the predicate ako_m the operator can be generalized to honor subclasses of the key k:

$$x{\rightarrow_m}k^* = [v \mid \exists \langle k', v \rangle \in x \colon k' \text{ ako}_m k] \tag{5}$$

d) A postfix operator $\leftarrow_m k$ which navigates to all proxies in the given map which use a given value v together with a certain key $k \in \mathcal{L}$:

$$v{\leftarrow_m}k = [x \in m \mid \exists \langle k, \tilde{v} \rangle \in x] \tag{6}$$

Using the predicate ako_m the operator can be generalized to honor subclasses of the key k:

$$v{\leftarrow_m}k^* = [x \in m \mid \exists \langle k', \tilde{v} \rangle \in x \colon k' \text{ ako}_m k] \tag{7}$$

It is straightforward to generalize all these navigation operators on individual proxies and values to multisets of them. As a consequence the result of one postfix can be used as startpoint for another postfix, enabling the building of *postfix chains*. This primitive path language is denoted as $\mathcal{P}_{\mathcal{M}}$.

NOTE 1 $\mathcal{P}_{\mathcal{M}}$ only serves as a minimal baseline for functionality to be provided by conforming implementations. It can be also used as the basis for a formal semantics for higher-level query and constraint languages. Annex A describes one.

6 Merging

Generic merging of maps only combines two (or more) proxy sets. Application-specific merging includes a second aspect. A mechanism has to be found to state whether—in a given map—two proxies are *regarded to be about the same subject*. Then all such equivalent proxies have to be actually merged.

NOTE 1 How *subject sameness* is determined and how the actual *proxy merging* is effectively done is not constrained by this part of ISO/IEC 13250. Such a process may be defined as having inputs that consist only of the proxies to be merged. Alternatively, the inputs may also include other information that may appear either inside the map or elsewhere in the merging process's environment.

NOTE 2 Given the appropriate expressitivity of the used constraint language, such equivalence and the consequent merging process can be described with a constraint.

Merging is modeled with a partial function $\bowtie: \mathcal{X} \times \mathcal{X} \times \mathcal{E} \mapsto \mathcal{X}$. It takes two proxies and an—otherwise unconstrained—environment $\mathcal{E}$ as parameters and produces a new proxy. In the special case that the environment has no influence in this process, $\bowtie$ is an infix operator $\mathcal{X} \times \mathcal{X} \mapsto \mathcal{X}$ between proxies.

NOTE 3 The reason for including $\mathcal{E}$ as a term in the definition of merging is to account for the fact that merging criteria may be defined as being dependent on conditions external to the maps. When environmental differences affect the results of merging, a single interchangeable subject map may be realized as different subject maps in different environments.

The fact that $\bowtie$ is *partial* means that it may be applicable to some pairs of proxies but not to others. Where a merging result is defined, the proxies shall be merged.

The operator $\bowtie$ is commutative and associative:

$$x \bowtie x' = x' \bowtie x \quad (8)$$

$$(x \bowtie x') \bowtie x'' = x \bowtie (x' \bowtie x'') \quad (9)$$

Additionally, $\bowtie$ is idempotent, as proxies merged with themselves should not result in a different one:

$$x \bowtie x = x \quad (10)$$

The operator $\bowtie$ factors a given proxy set into equivalence classes: two proxies x, y from a given map $m \in \mathcal{M}$ are then in the same class if $x \bowtie y$ is defined. The set of equivalence classes is written $m/\bowtie$. Every such class can be merged into a single proxy by combining all its members by applying $\bowtie$. Given a set of proxies $c = \{x_1, \ldots, x_n\}$, the merging of members of an equivalence class is defined as:

$$\bowtie c = x_1 \bowtie x_2 \bowtie \ldots \bowtie x_n \quad (11)$$

Implied with this is a *change set*, i.e. a prescription to replace proxies $x_1, \ldots, x_n$ in m with $\bowtie c$. We denote the change set of c as Δ. When it is applied to a map m, then every occurrence of a label of x_i in some property is replaced with the label of $\bowtie c$.

Given a map m and a particular $\bowtie$, the *merged view of the map* $m|_{\bowtie}$ is defined as:

$$m|_{\bowtie} = \left(\bigcup_{i=1}^{n} \bowtie c_i \right) \Delta_{c_1} \ldots \Delta_{c_n} \tag{12}$$

with all $c_i \in m/\bowtie$ and the Δ_{c_i} the corresponding change sets. Obviously the individual change sets have to be applied to all newly merged proxies.

One such merging step may result in proxies being created which again may be mergeable. The process can be repeated until a *fully merged map* is computed, so that $m|_{\bowtie} = m$. Fully merged maps are symbolized as $m|_{\bowtie}^{*}$. Since the sizes of newly merged maps are weakly monotonously decreasing, the process to produce a fully merged map takes always a finite number of steps.

7 Constraints

Subject maps are structures which are used to encode assertional knowledge. To interpret a map, be it for modelling, retrieval or modification, some background information about the map may be necessary. That information is provided in form of *constraints*, so that a given map m either satisfies a constraint, or not. A constraint language is a formalism which allows the expression of such constraints.

NOTE 1 Constraints may conditionally or unconditionally require the existence of certain proxies in maps, the existence of properties in proxies, and/or values in properties. Constraints may also prohibit the existence of any of the foregoing.

EXAMPLE 1 A constraint language may allow the expression of constraints such as *all instances of the concept* `person` *must have at least one* `shoesize` property or *any* `shoesize` *property must have an integer value between 10 and 50.*

NOTE 2 The precise ways in which constraints may be expressed are not constrained by this part of ISO/IEC 13250. Different constraint languages will differ in expressitivity and, consequently, in computational complexity.

This part of ISO/IEC 13250 imposes two requirements on any constraint language $\mathcal{C}$:

a) $\mathcal{C}$ shall define the *application of a constraint to a map* in the form of a binary operator $\otimes : \mathcal{M} \times \mathcal{C} \mapsto \mathcal{M}$. The operator $\otimes$ is used to define the *satisfaction relation* $\models \subseteq \mathcal{M} \times \mathcal{C}$ between a map and a constraint.

   A particular map m is said to *satisfy a constraint*, $m \models c$, if the application of the constraint results in a non-empty map:

$$m \models c \iff m \otimes c \neq \emptyset \tag{13}$$

b) $\mathcal{C}$ shall define a *merging operator* $\oplus : \mathcal{M} \times \mathcal{M} \mapsto \mathcal{M}$ as binary operator between two maps. Definitions of $\oplus$ shall be commutative, associative and idempotent.

NOTE 3 The provision of $\oplus$ and $\otimes$ may be done in any manner that is sufficiently expressive. $\oplus$ can but does not need to be defined in terms of $\bowtie$. Annex A demonstrates one way of defining $\otimes$.