

ISO/IEC JTC 1/SC34 N0393**ISO/IEC JTC 1/SC34****Information Technology --****Document Description and Processing Languages**

TITLE: Topic Maps Model (TMM)
SOURCE: [Steven R. Newcomb](#), [Sam Hunting](#), [Jan Algermissen](#) and [Patrick Durusau](#)
PROJECT: Topic Maps
PROJECT EDITORS: [Michel Biezunski](#), [Martin Bryan](#), [Steven R. Newcomb](#)
STATUS: Editor's Draft, Revision 2.30
ACTION: For review and comment
DATE: 28 March 2003
SUMMARY:
DISTRIBUTION: SC34 and Liaisons
REFER TO:
SUPERCEDES:
REPLY TO: Dr. James David Mason
(ISO/IEC JTC1/SC34 Chairman)
Y-12 National Security Complex
Information Technology Services
Bldg. 9113 M.S. 8208
Oak Ridge, TN 37831-8208 U.S.A.
Telephone: +1 865 574-6973
Facsimile: +1 865 574-1896
E-mail: <mailto:mxm@y12.doe.gov>
<http://www.y12.doe.gov/sgml/sc34/sc34oldhome.htm>

Ms. Sara Hafele Desautels, ISO/IEC JTC 1/SC 34 Secretariat
American National Standards Institute
25 West 43rd Street
New York, NY 10036
Tel: +1 212 642-4937
Fax: +1 212 840-2298
E-mail: sdesaute@ansi.org

28 March 2003**JTC 1/SC 34 N 0393****Topic Maps Model (TMM)**

Version 2.30, 2003/03/28

Go to <http://www.isotopicmaps.org/TMMM/TMMM-latest.html> to view or contribute to the current editors' working revision.

The previous officially published version (version 1.0) was [ISO/IEC JTC1/SC34/N0344](#).

Table of Contents

- [0 Introduction](#)
- [1 Scope](#)
- [2 Glossary](#)
 - [2.1 a-topic](#)
 - [2.2 assertion](#)
 - [2.3 built-in](#)
 - [2.4 c-topic](#)
 - [2.5 casting](#)
 - [2.6 conferred](#)
 - [2.7 fully merged](#)
 - [2.8 intra-assertion](#)
 - [2.9 merging](#)
 - [2.10 OP](#)
 - [2.11 Other Property \(OP\)](#)
 - [2.12 property](#)
 - [2.13 property class](#)
 - [2.14 property instance](#)
 - [2.15 r-topic](#)
 - [2.16 reified](#)
 - [2.17 role](#)
 - [2.18 role player](#)
 - [2.19 SDD](#)
 - [2.20 SIDP](#)
 - [2.21 SLUO](#)
 - [2.22 situation](#)
 - [2.23 situation feature](#)
 - [2.24 subject](#)
 - [2.25 Subject Identity Discrimination Property \(SIDP\)](#)
 - [2.26 Subject Location Uniqueness Objective \(SLUO\)](#)
 - [2.27 Syntax Deserialization Definition \(SDD\)](#)
 - [2.28 TM Application](#)
 - [2.29 TM Application Definition](#)
 - [2.30 TMA](#)
 - [2.31 TMA-defined](#)
 - [2.32 TMM-defined](#)
 - [2.33 topic](#)
 - [2.34 topic demander](#)
 - [2.35 topic map](#)
 - [2.36 Topic Map Application \(TM Application, TMA\)](#)
 - [2.37 Topic Maps Model \(TMM\)](#)
 - [2.38 treated as a set](#)
 - [2.39 value component](#)
- [3 Subjects, topics, and properties](#)
 - [3.1 Subjects and topics](#)
 - [3.2 Properties of topics](#)
- [4 Relationships and assertions](#)
 - [4.1 Substantive aspects of relationships](#)
 - [4.2 TMM-defined properties](#)
- [5 Situations and Property Values](#)
- [6 TM Application Definitions](#)
 - [6.1 Components of TM Application Definitions](#)
 - [6.2 Other Constraints on TM Application Definitions](#)
 - [6.3 Included TM Application Definitions](#)
- [7 Requirements for Syntax Deserialization Definitions \(SDDs\)](#)
- [8 Requirements for Implementations](#)
- [9 Fully Merged Topic Maps](#)
 - [9.1 Construct the Topic Map](#)
 - [9.2 Validate assertion instances for conformance to definitions](#)
 - [9.3 Confer Values on Properties](#)
 - [9.4 Validate the values of the properties of topics](#)
 - [9.5 Merge topics](#)
 - [9.6 Conditionally stop or repeat](#)

[10 Conformance](#)

0 Introduction

Topic maps are bodies of information that consist of "topics", each of which is a surrogate for a single [subject](#). If every topic in a topic map is the only surrogate for its subject, then users can find all information about that subject in a single location. The Topic Maps Model — the information structure of all topic maps that is defined by this International Standard — constrains the definitions of Topic Maps Applications in order to enable the achievement of this "[Subject Location Uniqueness Objective \[SLUO\]](#)". It specifies a foundation for lossless and uniform treatment of heterogeneous topic map information.

The Topic Maps Model meets the following requirements:

1. It enables an unbounded number of different Topic Map Applications to be created and used.
2. It enables metrics to be developed for arbitrary sets of Topic Map Applications.
3. It enables Topic Map Applications to be expressed as topic maps.
4. It enables the conformance of Topic Map Applications to this International Standard to be verified.
5. It enables rigorous specification and auditing of the process whereby an interchangeable topic map is understood as a set of [subjects](#). It enables specification of conventions for referring to members of that set of subjects by referring to components of interchangeable topic maps.
6. It enables determination of whether two topic maps are identical.
7. It facilitates the specification and determination of [subject](#) identity by humans, as well as machines.
8. When two or more topic maps are merged automatically, the Topic Maps Model:
 - a. enables the merging process to be consistent across Topic Map Applications and their implementations, and
 - b. preserves the integrity of the information contained in the merged topic maps in the resulting single topic map.

The Topic Maps Model meets the above requirements by:

1. Defining the uniform structure, and the [subjects](#) which are the substantive aspects, of all relationships between subjects, and by defining, naming, and constraining the properties that reflect that structure.
2. Requiring [TM Applications](#) to define explicitly the properties of topics that are required to determine the identities of their [subjects](#), the rules for assigning values to them, and the rules for comparing them.

1 Scope

This International Standard specifies:

1. the information structure of all topic maps;
2. certain common properties of topics, and constraints on the values of those properties;
3. constraints on the definitions of Topic Map Applications;
4. the definition of the term "fully merged" as it applies to topic maps; and
5. other definitions and specifications that support the foregoing.

2 Glossary

2.1 a-topic

An "assertion" topic. The subject of an a-topic is the relationship that is represented by an assertion; an a-topic is the

nexus of a specific assertion.

2.2 assertion

A statement of a relationship between [subjects](#), each of which is called a "role player" in the relationship. An assertion represents a relationship as a specific set of topics that collectively connect the topics (x-topics) whose subjects are the role players to an "[a-topic](#)" whose subject is the relationship itself. Every assertion also includes topics whose subjects are: (i) the type of the relationship (the t-topic), if any, (ii) for each role, the fact that a role player is "cast" in the role in the relationship (the c-topics), and (iii) the roles themselves (r-topics). Within any given assertion, the a-topic, the t-topic (if any), the c-topics, the r-topics, and the x-topics directly or indirectly have each other as the values of their TMM-defined OPs.

2.3 built-in

As in "built-in property instance value [component]": required by a TM Application Definition and/or Syntax Deserialization Definition to be assigned to a property instance *regardless* of the [situation](#) of the topic. "Built-in" is the opposite of "[conferred](#)".

2.4 c-topic

In an [assertion](#), a topic whose subject is one of the castings of the assertion.

2.5 casting

The fact that a subject plays a specific role in a relationship; the subject of a c-topic.

2.6 conferred

As in "conferred property instance value [component]": required by a TM Application Definition and/or Syntax Deserialization Definition to be assigned to a property instance *on account of* the [situation](#) of the topic. "Conferred" is the opposite of "[built-in](#)".

2.7 fully merged

The condition of a topic map in which every [subject](#) is represented by only one topic. See [9](#).

2.8 intra-assertion

Pertaining to the components of a single [assertion](#) instance.

2.9 merging

1. The process whereby two topics become a single topic, on account of the fact that they are deemed to have the same [subject](#), in the service of the [Subject Location Uniqueness Objective](#). See [9.5](#).
2. The process whereby a topic map in which more than one topic has the same [subject](#) becomes a "fully merged" topic map, in which every [reified](#) subject is [reified](#) by only one topic. See [9](#).

2.10 OP

[Other Property](#).

2.11 Other Property (OP)

A property class or property instance that is not a [Subject Identity Discrimination Property \(SIDP\)](#).

2.12 property

1. A [property class](#).
2. A [property instance](#).

2.13 property class

A class of name/value pairs defined by either the Topic Maps Model ("[TMM-defined](#)") or by a TM Application ("[TMA-defined](#)") as being instantiable as components of topics.

Note 1: Instances of such classes are called "[property instances](#)".

Note 2: The instances of a given property class may represent the identities of the [subjects](#) of the topics of which they are components, or information about those subjects, such as their relationships with other subjects, or both.

2.14 property instance

A uniquely named component of a topic; a name/value pair that is an instance of a [property class](#). It may be either [built-in](#) or [conferred](#).

2.15 r-topic

A "role" topic. The subject of an r-topic is a role in a relationship and/or relationship type.

2.16 reified

Represented by a surrogate. (Topics *reify* [subjects](#).)

2.17 role

A role in a relationship.

2.18 role player

1. A [subject](#) that plays a role in a relationship.
2. An x-topic.

2.19 SDD

[Syntax Deserialization Definition](#).

2.20 SIDP

[Subject Identity Discrimination Property](#).

2.21 SLUO

[Subject Location Uniqueness Objective](#)

2.22 situation

A topic's status as an a-topic, c-topic, r-topic, t-topic, and/or x-topic in one or more [assertions](#), and the status of each of the topics whose [subjects](#) are aspects of those assertions, recursively.

2.23 situation feature

A path from a situated topic to one of the topics in its [situation](#) that has defined characteristics that trigger the conferring of a value or [value component](#) on one of its properties.

2.24 subject

Any thing whatsoever, regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever.

2.25 Subject Identity Discrimination Property (SIDP)

1. A [property instance](#) which specifies the [subject](#) of the topic of which it is a component, and which independently serves as the only basis for any applicable merging rule(s). The opposite of [Other Property \(OP\)](#).
2. A [property class](#) designed so that each instance independently and comprehensively specifies the [subject](#) of a topic, and in terms of which any applicable merging rules are defined. The opposite of [Other Property \(OP\)](#).

2.26 Subject Location Uniqueness Objective (SLUO)

The objective of the topic map paradigm, which is to enable everything that is known about a [subject](#) to be accessible from one place.

2.27 Syntax Deserialization Definition (SDD)

A definition of the mapping process whereby each kind of construct found in instances of a specific syntax must, when encountered, deterministically result in the addition of topics with certain property values to the topic map that the syntactic instance is intended to represent.

2.28 TM Application

[Topic Map Application](#)

2.29 TM Application Definition

The definition of a [TM Application](#).

2.30 TMA

[Topic Map Application](#)

2.31 TMA-defined

Defined by a Topic Map Application (the opposite of [TMM-defined](#)).

2.32 TMM-defined

Defined for all topic maps by this Topic Maps Model (TMM) in [4.2](#), regardless of their governing TM Applications (the opposite of [TMA-defined](#)).

2.33 topic

A non-empty set of property instances that serves as a surrogate of a [subject](#).

2.34 topic demander

A construct in an instance of an interchangeable topic map which can be referenced as a way of referencing the topic to which, by convention, it corresponds in the deserialized topic map that the interchangeable topic map is intended to represent.

2.35 topic map

A body of information consisting of a non-empty set of topics.

2.36 Topic Map Application (TM Application, TMA)

A defined set of assertion types, properties, merging rules, and their supporting definitions that is intended to govern the expression and merging behaviors of a set of topic maps. It is not a "software application"; it is a world view, expressed in a way that enables it to be implemented uniformly.

2.37 Topic Maps Model (TMM)

This International Standard.

2.38 treated as a set

Regarded (as a list) in such a way as to ignore both the order of the items in the list, and any duplicates that appear in the list.

When the values of two property instances of the same property class include lists that are required to be treated as sets, their equality or inequality can be determined by any procedure that yields the same results as the following procedure:

1. Make each item in each list unique in that list, i.e., delete all duplicate items from each list.
2. Apply the same comprehensively deterministic order-normalization algorithm to both lists.
3. Compare each item in one list for identity with the item in the corresponding position in the other list.

If all the pairs of items in corresponding positions in the two lists are identical, the two sets are equal. If any pair of corresponding items are not identical to each other, or if the lengths of the two lists are not the same, the two sets are not equal.

2.39 value component

A distinct portion of a single property value, such as a member of a complex value or a member of set.

3 Subjects, topics, and properties

3.1 Subjects and topics

In a [topic map](#), all [subjects](#) that are defined, and/or about which any information is conveyed, are represented by [topics](#). The number of topics in a topic map is finite; in order to create a topic map it is necessary to choose the subjects that will be represented in it.

3.2 Properties of topics

Topics consist of [property instances](#). Each is an instance of a [property class](#). A single topic can be comprised of properties defined by multiple TM Applications. In a topic, no two properties can be instances of the same class (can have the same name).

Some properties are [Subject Identity Discrimination Properties \(SIDPs\)](#). All others are called "Other Properties (OPs)".

Some property instances are "[built-in](#)". All others are "[conferred](#)".

Some properties are "[TMM-defined](#)". All other properties are "[TMA-defined](#)".

3.2.1 Property names

Every property has a name; the name is different from that assigned to all other properties, assertion types, and roles.

Note 3: All these names share a single namespace in order to minimize human error.

3.2.2 Property Values

Property instances exist only if values have been assigned to them.

Note 4: A property instance may exhibit a null or empty value if such a value has been explicitly assigned to it.

The definitions of property classes specify the types of the values of their instances. The value type of a property may be simple (without substructure) or complex (having substructure nested within it). The values may be or include lists and lists [treated as sets](#).

3.2.3 Subject identity discrimination properties (SIDPs)

Every topic has at least one SIDP instance. Each SIDP instance independently specifies the [subject](#) of the topic, for all purposes of subject identification. SIDP values are the only basis for automatically recognizing when two topics have the same [subject](#) or different subjects, and should therefore either be merged or left unmerged. No topic can have more than one SIDP instance whose property class is defined by any single TM Application.

Note 5: However, a single topic may have multiple SIDP instances if the class of each of them is defined by a different TM Application.

3.2.4 Other properties (OPs)

The values of OPs do not influence the automated merging process defined by the Topic Maps Model.

4 Relationships and assertions

[Subjects](#) have relationships with each other. These relationships are themselves subjects, and they have substantive aspects which are also subjects.

4.1 Substantive aspects of relationships

4.1.1 Assertion topics (a-topics)

When a topic represents a relationship among subjects, such a topic is called an "assertion topic" or "a-topic". Every a-topic is the nexus of a set of topics, called an "[assertion](#)". Each member of the set (including the a-topic) represents a subject which is a substantive aspect of the relationship.

An a-topic cannot also be a c-topic, r-topic, or t-topic; the subjects of a-, c-, r-, and t- topics are all mutually exclusive (no topic can have more than one subject).

4.1.2 Role player topics (x-topics)

When a relationship exists among subjects, each subject so related is called a "role player". In the context of any specific assertion, each topic that represents a role player in that assertion is called an "x-topic".

All subjects, without exception, are eligible to be role players in relationships; therefore all topics are eligible to be x-topics, regardless of whether or not they are a-topics, t-topics, r-topics, c-topics.

Note 6: The "x" in "x-topic" is intended to connote this global eligibility ("x" can be anything).

4.1.3 Role topics (r-topics)

Each role player plays a specific role in the relationship, and each such specific role is itself a subject. In the context of any specific assertion or assertion type, each topic that represents a role is called an "r-topic".

4.1.4 Casting topics (c-topics)

The fact that a specific role player (or the fact that no role player) plays a specific role in a specific relationship is called a "casting". Topics whose subjects are castings are called "c-topics".

Within a single assertion, no two castings can cast role players in the same role.

Note 7: However, a set or group of subjects can be a subject, and it can therefore be a role player. The Topic Maps Model neither provides nor constrains set/group semantics; such semantics are defined entirely by TM Applications.

4.1.5 Assertion type topics (t-topics)

When a relationship has been classified explicitly as an instance of a specific type of relationship, the type is itself a subject. Such a subject, which may be called a "relationship type" or an "assertion type", must be defined in such a way as to specify the roles and the significance (semantics) of all instances of the type. Assertion type definitions (relationship type definitions) may also specify the qualifications that should be met by the players of the roles. Topics whose subjects are assertion types are called "t-topics".

Note 8: The notion of role-playing implies a theatrical metaphor which may be helpful in understanding relationships/assertions. An assertion type is like a written play. The playwright ~ the definer of the [TM Application](#) in which the assertion type is defined ~ defines the roles to be played in all performances of the play, and characterizes each of the roles in various ways. An instance of an assertion is like a specific performance of a play, in which the specific "actors" who play the roles are specific subjects. The "castings" of an assertion are like the entries that appear in the playbill (or *dramatis personae*) that specifies which actors have been cast in which roles. The SIDPs of topics are like the costumes of the actors; they allow the audience to see and identify the subjects that are playing the roles.

4.2 TMM-defined properties

The topics in the set of topics that comprise an assertion refer to each other by means of certain of their properties. These properties, called "[TMM-defined](#) properties", are the only properties defined for all topic maps by the Topic Maps Model. (All other properties are "[TMA-defined](#) properties".) The behavior of implementations must not be inconsistent with the assertion structure and the rules for merging assertions, as that structure and those merging rules are expressed in this International Standard in terms of the TMM-defined properties.

Note 9: Some implementations may be adapted to usage scenarios that do not require support for all TMM-defined properties. Moreover, with respect to the TMM-defined properties that they do support, there is no requirement that implementations provide the values of all TMM-defined properties with equal speed or efficiency. However, users of topic map engine implementations that support the TMM-defined properties comprehensively can enjoy interoperability and convenience advantages in multi-TM-Application and multi-implementation processing contexts.

4.2.1 Table of TMM-defined Other Property (OP) Classes

Table 1: The following table lists the names of the TMM-defined Other Properties (OP) categorized by the assertion aspects that have them. The basic value constraints for each property are shown along with references to sections of this International Standard where additional constraints are specified.

Substantive Aspect of Relationship	Aspect Name	SIDP or OP?	Name of Property or Property Component[1] of Substantive Aspect	Value Type of Property or Property Component	Value Constraints	Other Relevant Constraints
assertion	a-topic	SIDP	a-sidp	1 complex	4.2.2.1.1	4.2.2.1.2

<u>Substantive Aspect of Relationship</u>	<u>Aspect Name</u>	<u>SIDP or OP?</u>	<u>Name of Property or Property Component[1] of Substantive Aspect</u>	<u>Value Type of Property or Property Component</u>	<u>Value Constraints</u>	<u>Other Relevant Constraints</u>
			a-sidp.t	? t-topic	4.2.2.2.1	
			a-sidp.castingPairs{ }	2+ complexes treated as a set	4.2.2.3.1	
			a-sidp.castingPairs{?}.r	1 r-topic	4.2.2.4.1	
			a-sidp.castingPairs{?}.x	? x-topic	4.2.2.5.1	
		OP	a-castings{ }	2+ c-topics treated as a set	4.2.2.6.1	4.2.2.6.2
		OP	a-roles{ }	2+ r-topics treated as a set	4.2.2.8.1	4.2.2.8.2
		OP	a-type	? t-topic	4.2.2.7.1	4.2.2.7.2
casting	c-topic		c-sidp	1 complex	4.2.3.1.1	4.2.3.1.2
			c-sidp.a	1 a-topic	4.2.3.2.1	
			c-sidp.r	1 r-topic	4.2.3.3.1	
			c-sidp.x	? x-topic	4.2.3.4.1	
		OP	c-role	1 r-topic	4.2.3.5.1	4.2.3.5.2
		OP	c-rolePlayer	? x-topic	4.2.3.6.1	4.2.3.6.2
		OP	c-assertion	1 a-topic	4.2.3.7.1	4.2.3.7.2
		OP	c-otherCastings{ }	1+ c-topics treated as a set	4.2.3.8.1	4.2.3.8.2
role	r-topic	OP	r-assertions{ }	0+ a-topics treated as a set	4.2.4.2.1	4.2.4.2.2
		OP	r-castings{ }	0+ c-topics treated as a set	4.2.4.1.1	4.2.4.1.2
		OP	r-otherATRoles{ }	0+ r-topics treated as a set	4.2.4.4.1	4.2.4.4.2
		OP	r-type	? t-topic	4.2.4.3.1	4.2.4.3.2
assertion type	t-topic	OP	t-assertions{ }	0+ a-topics treated as a set	4.2.5.1.1	4.2.5.1.2
		OP	t-roles{ }	2+ r-topics treated as a set	4.2.5.2.1	4.2.5.2.2
role player	x-topic	OP	x-castings{ }	0+ c-topics treated as a set	4.2.6.1.1	4.2.6.1.2

Legend: { } A list treated as a set.

{?} Every member of the set.

0+ 0 or more value components.

1 1 value component is required.

1+ 1 or more value components are required.

2+ 2 or more value components are required.

? May or may not have a single value component.

[1] All property names begin with the string "IS13250::", which is omitted here to save space. For example, "a-sidp" is really "IS13250::a-sidp".

4.2.2 Constraints relevant to TMM-defined properties of a-topics

4.2.2.1 IS13250::a-sidp property

4.2.2.1.1 Constraints on values of IS13250::a-sidp properties.

The value is a composite that comprehensively specifies the relationship that is subject of the a-topic, in order to support the automatic determination of whether the subject is the same as, or different from, the subject of any other a-topic, in accordance with the Topic Maps Model's merging rule for a-topics.

The values of the [components](#) of the IS13250::a-sidp property (IS13250::a-sidp.t and IS13250::a-sidp.castingPairs{ }) must meet the constraints specified for them (see [4.2.2.2](#) and [4.2.2.3](#)).

4.2.2.1.2 Other constraints relevant to IS13250::a-sidp properties.

Any topic that has an IS13250::a-sidp property is regarded as an a-topic.

Every a-topic:

1. must have all of the TMM-defined properties whose names begin with IS13250::a-.
2. must not have any properties whose names begin with IS13250::c-, IS13250::r-, or IS13250::t-.
3. must not have any TMM-defined or TMA-defined SIDPs other than IS13250::a-sidp.

4.2.2.2 IS13250::a-sidp.t property component

4.2.2.2.1 Constraints on values of IS13250::a-sidp.t property components.

The value must be the same as the value of the IS13250::a-type property of the same topic (see [4.2.2.7](#)).

Note 10: Implementations are not required to store the same information redundantly, but merely to provide access to the information, however it may be stored, under multiple rubrics. For example, the redundancy of the IS13250::a-type property with the IS13250::a-sidp.t property component stems from the distinction between their anticipated uses: the purpose of the IS13250::a-type property is to answer the question, "What is the assertion type of this assertion?", while the purpose of the IS13250::a-sidp property is to constrain and facilitate comparisons between the subjects of a-topics in support of assertion-merging operations.

Note 11: Whenever one property value or [value component](#) is required to be the same as (or to correspond to) another property value or value component, all of the constraints that apply to each of them actually apply to both. This International Standard specifies the applicable constraints non-redundantly, except for the "must be the same as (or correspond to)" constraints, which are redundantly stated everywhere they are applicable. The purpose of this limited redundancy is to facilitate the reader's understanding.

4.2.2.3 IS13250::a-sidp.castingPairs{ } property component

4.2.2.3.1 Constraints on values of IS13250::a-sidp.castingPairs{ } property components.

The value is a set of pairs of topics. Each pair must uniquely correspond to one of the castings of the assertion of which the a-topic is the nexus; each pair consists of the r-topic and x-topic (if any) of the casting to which the pair corresponds. The number of pairs must be the same as the number of members in the set of c-topics that constitute the value of the a-castings{ } property.

The values of the components of each member pair (IS13250::a-sidp.castingPairs{?}.r and IS13250::a-sidp.castingPairs{?}.x) must meet the constraints specified for them (see [4.2.2.4](#) and [4.2.2.5](#)).

4.2.2.4 IS13250::a-sidp.castingPairs{?}.r property component

Note 12: The symbol {?} in IS13250::a-sidp.castingPairs{?}.r signifies that the r-topic property component is a component of each member of the set of pairs that is the value of the IS13250::a-sidp.castingPairs{ } property component.

4.2.2.4.1 Constraints on values of IS13250::a-sidp.castingPairs{?}.r property components.

The value must be the same as the value of the IS13250::c-role property of the c-topic whose subject is the casting to which the pair uniquely corresponds.

4.2.2.5 IS13250::a-sidp.castingPairs{?}.x property component

4.2.2.5.1 Constraints on values of IS13250::a-sidp.castingPairs{?}.x property components.

The value must be the same as the value of the IS13250::c-rolePlayer property of the c-topic whose subject is the casting to which the pair uniquely corresponds.

4.2.2.6 IS13250::a-castings{ } property

4.2.2.6.1 Constraints on values of IS13250::a-castings{ } properties.

The value is the set of two or more c-topics whose subjects are the castings of the assertion of which the a-topic is the nexus.

Each of the c-topics in the value must uniquely correspond to one of the pairs in the value of the IS13250::a-sidp.castingPairs{ } property component (see [4.2.2.3.1](#), [4.2.2.4](#) and [4.2.2.5](#)).

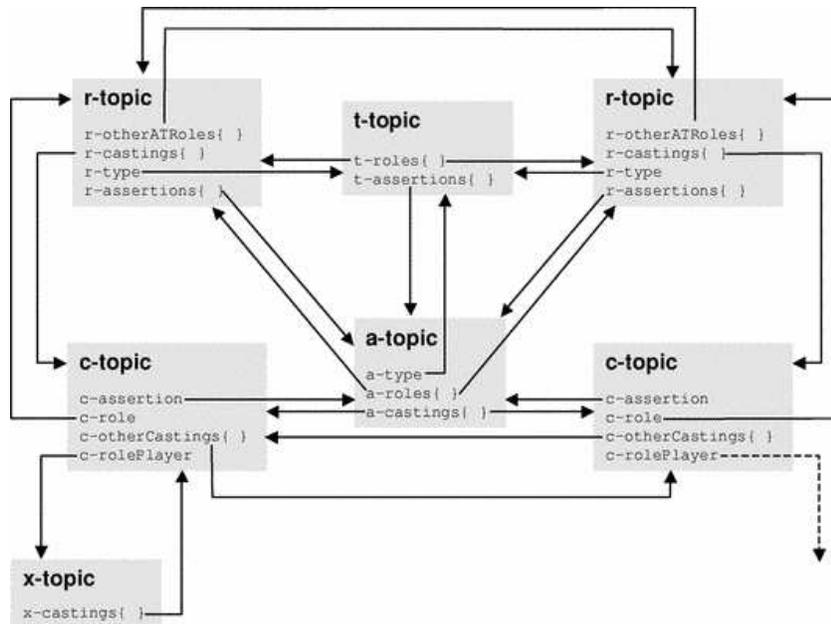
4.2.2.6.2 Other constraints relevant to IS13250::a-castings{ } properties.

Each of the c-topics in the value of the IS13250::a-castings{ } property:

1. must have, as the value of its IS13250::c-role property, an r-topic that is included in the value of the IS13250::a-roles{ } property of the a-topic.
2. must reciprocally have the a-topic as the value of its IS13250::c-assertion property.

Note 13: Figure 1:

The following diagram is provided as an aid to the reader's understanding of many of the [intra-assertion](#) consistency constraints in this Section [4.2](#), including those immediately preceding this [Figure 1](#). The diagram shows an instance of an assertion in which there are two roles, one of which has a role player.



Any topic that has an IS13250::a-castings{ } property is regarded as an a-topic (see [4.2.2.1.2](#)).

4.2.2.7 `IS13250::a-type` property

4.2.2.7.1 Constraints on values of `IS13250::a-type` properties.

The value, if any, is the t-topic whose subject is the type of the assertion which is the subject of the a-topic.

The value may be empty. If the value is empty, the type of the assertion is unspecified.

The value must be the same as the value of the `IS13250::a-sidp.t` property component of the same topic.

4.2.2.7.2 Other constraints relevant to `IS13250::a-type` properties.

If the value is non-empty (i.e., if the value is a t-topic):

1. each of the r-topics in the value of the `IS13250::a-roles{ }` property of the a-topic must have the same t-topic as the value of their `IS13250::r-type` properties.
2. the a-topic must reciprocally be a member of the set of a-topics that is the value of the `IS13250::t-assertions{ }` property of the t-topic.

Any topic that has an `IS13250::a-type` property is regarded as an a-topic (see [4.2.2.1.2](#)).

4.2.2.8 `IS13250::a-roles{ }` property

4.2.2.8.1 Constraints on values of `IS13250::a-roles{ }` properties.

The value is the set of r-topics whose subjects are the roles of the assertion of which the a-topic is the nexus. There must be two or more r-topics in the set.

4.2.2.8.2 Other constraints relevant to `IS13250::a-roles{ }` properties.

If the value of the `IS13250::a-type` property of the a-topic is a t-topic (i.e., is non-empty), then the value of the `IS13250::t-roles{ }` property of that t-topic must be the same as the value of the a-topic's `IS13250::a-roles{ }` property.

Each of the r-topics in the value:

1. must also be the value of the `IS13250::c-role` property of one of the c-topics in the value of the `IS13250:a-castings{ }` property.
2. must reciprocally include the a-topic in the value of its `IS13250::r-assertions{ }` property.

Any topic that has an `IS13250::a-roles{ }` property is regarded as an a-topic (see [4.2.2.1.2](#)).

4.2.3 Constraints relevant to TMM-defined properties of c-topics

4.2.3.1 `IS13250::c-sidp` property

4.2.3.1.1 Constraints on values of `IS13250::c-sidp` properties.

The value is a composite that comprehensively specifies the casting that is subject of the c-topic, in order to support the automatic determination of whether the subject is the same as, or different from, the subject of any other c-topic, in accordance with the Topic Maps Model's merging rule for c-topics.

The values of the components of the `IS13250::c-sidp` property (`IS13250::c-sidp.c-assertion`, `IS13250::c-sidp.c-role` and `IS13250::c-sidp.rolePlayer`) must meet the constraints specified for them.

4.2.3.1.2 Other constraints relevant to `IS13250::c-sidp` properties.

Any topic that has an `IS13250::c-sidp` property is regarded as a c-topic.

Every c-topic:

1. must have all of the TMM-defined properties whose names begin with `IS13250::c-`.
2. must not have any properties whose names begin with `IS13250::a-`, `IS13250::r-`, or `IS13250::t-`.
3. must not have any TMM-defined or TMA-defined SIDPs other than `IS13250::c-sidp`.

4.2.3.2 `IS13250::c-sidp.a` property component

4.2.3.2.1 Constraints on values of `IS13250::c-sidp.a` property components.

The value must be the same as the value of the `IS13250::c-assertion` property of the same c-topic.

4.2.3.3 `IS13250::c-sidp.r` property component

4.2.3.3.1 Constraints on values of `IS13250::c-sidp.r` property components.

The value must be the same as the value of the `IS13250::c-role` property of the same c-topic.

4.2.3.4 `IS13250::c-sidp.x` property component

4.2.3.4.1 Constraints on values of `IS13250::c-sidp.x` property components.

The value must be the same as the value of the `IS13250::c-rolePlayer` property of the same c-topic.

4.2.3.5 `IS13250::c-role` property

4.2.3.5.1 Constraints on values of `IS13250::c-role` properties.

The value is the r-topic whose subject is the role in which the role player (the subject of the x-topic, if any, that is the value of the `IS13250::c-rolePlayer` property) is being cast in the assertion (the subject of the a-topic that is the value of the `IS13250::c-assertion` property).

The value must be the same as the value of the `IS13250::c-sidp.r` property component of the same c-topic.

4.2.3.5.2 Other constraints relevant to `IS13250::c-role` properties.

The r-topic that is the value:

1. must be included in the value of the `IS13250::a-roles{ }` property of the a-topic that is the value of the `IS13250::c-assertion` property of the c-topic.
2. must be included in the value of the `IS13250::t-roles{ }` property of the t-topic, if any, that is the value of the `IS13250::a-type` property of the a-topic that is the value of the c-topic's `IS13250::c-assertion` property.
3. must reciprocally include the c-topic in the value of its `IS13250::r-castings{ }` property.

No two c-topics within the same assertion can have the same topic as the value of their respective `IS13250::c-role` properties.

Any topic that has an `IS13250::c-role` property is regarded as a c-topic (see [4.2.3.1.2](#)).

4.2.3.6 `IS13250::c-rolePlayer` property

4.2.3.6.1 Constraints on values of `IS13250::c-rolePlayer` properties.

The value is the x-topic, if any, whose subject is the role player that is being cast in the role (the subject of the r-topic that is the value of `IS13250::c-role` property) in the assertion (the subject of the a-topic that is the value of the `IS13250::c-assertion` property).

The value must be the same as the value of the `IS13250::c-sidp.x` property component of the same c-topic.

The value may be empty. If the value is empty, the role is not played.

4.2.3.6.2 Other constraints relevant to `IS13250::c-rolePlayer` properties.

The x-topic that is its value, if any, must reciprocally include the c-topic in the value of its `IS13250::x-castings{ }` property.

In any given assertion, at least one of the castings must have a role player. In other words, at least one of the c-topics must have a non-empty `IS13250::c-rolePlayer` property.

Any topic that has an `IS13250::c-rolePlayer` property is regarded as a c-topic (see [4.2.3.1.2](#)).

4.2.3.7 `IS13250::c-assertion` property

4.2.3.7.1 Constraints on values of `IS13250::c-assertion` properties.

The value is the a-topic whose subject is the assertion in which the role player (the subject of the x-topic that is the value of the `IS13250::c-rolePlayer` property) is being cast in the role (the subject of the topic that is the value of `IS13250::c-role` property).

The value must be the same as the value of the `IS13250::c-sidp.a` property component of the same c-topic.

4.2.3.7.2 Other constraints relevant to `IS13250::c-assertion` properties.

The a-topic that is the value:

1. must include the r-topic that is the value of the `IS13250::c-role` property of the c-topic in the value of its `IS13250::a-roles{ }` property.
2. must have the same value for its `IS13250::a-type` property as the value of the `IS13250::r-type` property of the r-topic that is the value of the `IS13250::c-role` property of the c-topic. Both values must either be empty, or be the same t-topic.
3. must reciprocally include the c-topic in the value of its `IS13250::a-castings{ }` property.

Any topic that has an `IS13250::c-assertion` property is regarded as a c-topic (see [4.2.3.1.2](#)).

4.2.3.8 `IS13250::c-otherCastings{ }` property

4.2.3.8.1 Constraints on values of `IS13250::c-otherCastings{ }` properties.

The value is the set of c-topics whose subjects are the other castings of the same assertion.

The value must be the same as the value of the `IS13250::a-castings{ }` property of the a-topic that is the value of the `IS13250::c-assertion` property of the same c-topic, minus the c-topic. (Therefore, the value must be at least one c-topic.)

4.2.3.8.2 Other constraints relevant to `IS13250::c-otherCastings{ }` properties.

Each of the c-topics in the value:

1. must have the same a-topic for the value of their `IS13250::c-assertion` properties as the a-topic that is the value of the c-topic's `IS13250::c-assertion` property.
2. must have a reciprocal `IS13250::otherCasting{ }` property which includes the c-topic in its value.

Any topic that has an `IS13250::c-otherCastings{ }` property is regarded as a c-topic (see [4.2.3.1.2](#)).

4.2.4 Constraints relevant to TMM-defined properties of r-topics

4.2.4.1 `IS13250::r-castings{ }` property

4.2.4.1.1 Constraints on values of `IS13250::r-castings{ }` properties.

The value is the set of all c-topics whose subjects are castings of the role that is the subject of the r-topic.

The value may be empty. If the value is empty, there is no instance of an assertion in which the subject of the r-topic is a role.

4.2.4.1.2 Other constraints relevant to `IS13250::r-castings{ }` properties.

A topic is considered an r-topic if it has an `IS13250::r-castings{ }` property, even if the value of the property is empty.

Every r-topic:

1. must have all of the TMM-defined properties whose names begin with `IS13250::r-`.
2. must not have any properties whose names begin with `IS13250::a-`, `IS13250::c-`, or `IS13250::t-`.
3. may have an empty value for either its `IS13250::r-castings{ }` property or its `IS13250::r-type{ }` property, but it must not have an empty value for both.

Each of the c-topics in the value:

1. must have a different value for its `IS13250::c-assertion` property than all of the other c-topics in the value. In other words, no two members of the set can be different castings of the same assertion.
2. must have, as the value of its `IS13250::c-assertion` property, an a-topic which is included in the value of the r-topic's `IS13250::r-assertions{ }` property. That a-topic's `IS13250::a-type` property must have the same value as the value of the r-topic's `IS13250::t-roles{ }` property. Both values must either be empty, or be the same t-topic.
3. must have a reciprocal `IS13250::c-role` property whose value is the r-topic.

4.2.4.2 `IS13250::r-assertions{ }` property

4.2.4.2.1 Constraints on values of `IS13250::r-assertions{ }` properties.

The value is the set of all a-topics whose subjects are assertions that have the role that is the subject of the r-topic as one of their roles.

The value may be empty. If the value is empty, there is no assertion in which the subject of the r-topic is a role.

4.2.4.2.2 Other constraints relevant to `IS13250::r-assertions{ }` properties.

A topic is considered an r-topic if it has an `IS13250::r-assertions{ }` property, even if the value of the property is empty (see [4.2.4.1.2](#)).

If the value is empty, the value of the `IS13250::r-castings{ }` property must also be empty.

If the value is non-empty, each of the a-topics in the value:

1. must include, in the value of its `IS13250::a-castings{ }` property, one of the c-topics that is included in the value of the `IS13250::r-castings{ }` property of the r-topic.
2. must have the same value for its `IS13250::a-type` property as the r-topic has for its `IS13250::r-type` property. (The values may both be the same t-topic, or they may both be empty.)
3. must have a reciprocal `IS13250::a-roles{ }` property whose value includes the r-topic.

4.2.4.3 `IS13250::r-type` property

4.2.4.3.1 Constraints on values of `IS13250::r-type` properties.

The value, if any, is the t-topic whose subject is the assertion type whose role set includes the role which is the subject of the r-topic.

The value may be empty. If the value is empty, there is no assertion type that includes the subject of the r-topic among its roles, and the r-topic cannot be a role in a typed assertion.

4.2.4.3.2 Other constraints relevant to `IS13250::r-type` properties.

If an r-topic's `IS13250::r-type` property is empty, then its `IS13250::r-otherATRoles{ }` property must also be empty.

If an r-topic has a non-empty `IS13250::r-type` property (i.e., if the value is a t-topic), then the t-topic:

1. must have a `IS13250::t-assertions{ }` property whose value is the same as the value of the r-topic's `IS13250::r-assertions{ }` property.
2. must have a reciprocal `IS13250::t-roles{ }` property whose value includes the r-topic.

A topic is considered an r-topic if it has an `IS13250::r-type` property, even if the value of the property is empty (see [4.2.4.1.2](#)).

4.2.4.4 `IS13250::r-otherATRoles{ }` property**4.2.4.4.1 Constraints on values of `IS13250::r-otherATRoles{ }` properties.**

The value, if any, is the set of r-topics whose subjects are the other roles of the same assertion type.

The value may be empty. If the value is empty, there is no assertion type that includes the r-topic among its roles, and the r-topic cannot be a role in a typed assertion.

If the value is non-empty, then the value must be the same as the value of the `IS13250::t-roles{ }` property of the t-topic that is the value of the `IS13250::r-type` property of the same r-topic, minus the r-topic.

4.2.4.4.2 Other constraints relevant to `IS13250::r-otherATRoles{ }` properties.

If an r-topic's `IS13250::r-otherATRole{ }` property is empty, then its `IS13250::r-type` property must also be empty.

If an r-topic's `IS13250::r-otherATRoles{ }` property is non-empty, then each of the r-topics in the value:

1. must have the same t-topic for their `IS13250::r-type` properties as the value of the r-topic's `IS13250::r-type` property.
2. must have an `IS13250::otherATRole{ }` property which reciprocally includes the r-topic in its value.

Any topic that has an `IS13250::r-otherATRoles{ }` property is regarded as an r-topic (see [4.2.4.1.2](#)).

4.2.5 Constraints relevant to TMM-defined properties of t-topics**4.2.5.1 `IS13250::t-assertions{ }` property****4.2.5.1.1 Constraints on values of `IS13250::t-assertions{ }` properties.**

The value is the set of all a-topics whose subjects are assertions that are instances of the assertion type that is the subject of the t-topic.

The value may be empty. If the value is empty, there are no instances of the assertion type.

4.2.5.1.2 Other constraints relevant to `IS13250::t-assertions{ }` properties.

A topic is considered a t-topic if it has an `IS13250::t-assertions{ }` property, even if the value of the property is empty.

Every t-topic:

1. must have all of the TMM-defined properties whose names begin with `IS13250::t-`.
2. must not have any properties whose names begin with `IS13250::a-`, `IS13250::c-`, or `IS13250::r-`.

If the value is non-empty, each of the a-topics in the value:

1. must have the same value for its `IS13250::a-roles{ }` property as the value of the t-topic's `IS13250::t-roles{ }` property.
2. must have a reciprocal `IS13250::a-type` property whose value is the t-topic.

4.2.5.2 `IS13250::t-roles{ }` property

4.2.5.2.1 Constraints on values of `IS13250::t-roles{ }` properties.

The value is the set of all r-topics whose subjects are the roles defined for the assertion type that is the subject of the t-topic. There must be two or more.

If the value of the t-topic's `IS13250::t-assertions{ }` property is a set of one or more a-topics (i.e., is non-empty), then the values of the `IS13250::a-roles{ }` properties of those a-topics must be the same as the value of the `IS13250::t-roles{ }` property.

4.2.5.2.2 Other constraints relevant to `IS13250::t-roles{ }` properties.

Each of the r-topics in the value of the `IS13250::t-roles{ }` property:

1. must have the same value for its `IS13250::r-assertions{ }` property as the value of the t-topic's `IS13250::t-assertions{ }` property.
2. must have a reciprocal `IS13250::r-type` property whose value is the t-topic.

A topic is considered a t-topic if it has an `IS13250::t-roles{ }` property, even if the value of the property is empty (see [4.2.5.1.2](#)).

4.2.6 Constraints relevant to TMM-defined properties of x-topics

4.2.6.1 `IS13250::x-castings{ }` property

4.2.6.1.1 Constraints on values of `IS13250::x-castings{ }` properties.

The value is the set of all c-topics that cast the subject of the x-topic in a role in some assertion.

The value may be empty. If the value is empty, the subject of the x-topic does not play any roles in any assertions, the topic is not an x-topic, and the `IS13250::x-castings{ }` property is ignored.

4.2.6.1.2 Other constraints relevant to `IS13250::x-castings{ }` properties.

Each of the c-topics in the value of the `IS13250::x-castings{ }` property must reciprocally have the x-topic as the value of its `IS13250::c-rolePlayer` property.

5 Situations and Property Values

A topic's "[situation](#)" is defined in terms of its status as an a-topic, c-topic, r-topic, t-topic, and/or x-topic in one or more assertions. (When a topic is none of the foregoing, it is called an "isolated topic"; such topics are not "situated". Their SIDP values must be [built-in](#).) Each TMM-defined OP value or [value component](#) of a topic that indicates that the topic is an a-, c-, r-, t-, or x-topic can be regarded as the beginning of a "path" of connections from that topic to another topic, either directly or via intervening topics. Each intervening topic has a TMM-defined OP value or [value](#)

[component](#) that is that next topic in the path.

"[Situation features](#)" are paths whose characteristics satisfy a definition (specified in a [TM Application Definition](#) or [Syntax Deserialization Definition](#)) that requires that a value or value component be "[conferred](#)" on a property of the situated topic.

Properties can also have values or value components that are not conferred on them on account of the [features of their situations](#). Instead, such values are required to be "[built-in](#)" by an applicable [TM Application Definition](#) or [Syntax Deserialization Definition](#).

TM Application Definitions and Syntax Deserialization Definitions define how the SIDP values of isolated topics, r-topics, t-topics, and x-topics are built into or conferred upon them. In all definitions that require values to be conferred, the value is conferred on a topic on account of a [situation feature](#) in which the topic plays a specific role in an instance of a specific assertion type.

Built-in property values or value components are "read-only". Values and value components must specify whether they are built-in or conferred, so that, during the merging process, built-in values will not automatically be superseded (i.e., erased or overwritten) by conferred values or [value components](#). The definition of a TM Application may or may not allow value components to be conferred on a property *in addition* to existing built-in value components. Conflicts that arise between definitions of built-in property value components and definitions that require value components to overwrite them must be reported by implementations.

6 TM Application Definitions

6.1 Components of TM Application Definitions

The definition of a [TM Application](#) must include all of the following:

6.1.1 TM Application Name

Every TM Application Definition must specify the name of the TM Application it defines. The name must be different from the name of any known TM Application. If the TM Application is evolving or configurable, each version, conformance level, or other configuration must be regarded as a distinct TM Application for purposes of naming.

All TM Application names that begin with "IS", irrespective of the cases of the letters, are reserved for International Standard TM Applications.

Note 14: Ideally, TM Applications should have universally unique names. That being a practical impossibility, designers of TM Applications are constrained to use names that do not conflict with any TM Applications known to them. The risk of name conflicts can be minimized by use of a URI within an internet domain that is controlled by the TM Application designer, or by registering the name in a TM Application namespace of a stable public organization such as OASIS, W3C, IDEAlliance, OCLC, or the Library of Congress.

6.1.2 Property Class Definitions

All property classes of topics must be defined. Every property instance of every topic must either be TMM-defined or defined by a single TM Application. Each property definition must indicate whether its value is for the purpose of distinguishing subjects, i.e., whether it is an SIDP or an OP. Every topic that has a TMA-defined property to which a value has been assigned (conferred or built-in) is said to be "governed" by that TM Application.

The names, value types and semantics of all properties must be specified in their definitions. If the value type is complex, the name of each member of each complex must be unique among the value components of the property.

Every TM Application must be defined in such a way that all topics governed by the TM Application will have a single SIDP instance that is governed by that TM Application. However, different topics may have instances of different SIDP classes; the number of SIDP classes defined by a single TM Application is not constrained. All SIDPs, and the values that are built into them and/or conferred upon them, must be defined in such a way that, under the merging rules defined by the TM Application, all topics that have the same subject will be merged, and topics that do not have the same subject will not be merged (see [9.5](#)).

6.1.3 Constraints on Property Values

If there are any constraints on the values of the properties of subjects, they must be stated in the TM Application Definition. Implementations of TM Applications must report all violations of such constraints (see [9.4](#)).

6.1.4 Assertion Types

The names and semantics of all assertion types must be defined.

Each assertion type definition must include a complete list of the roles (the r-topics) that must appear in all instances of the type. A minimum of two roles must be defined for each type. No two assertion types can include the same role.

For each role, any applicable validity constraints on players of the role should be specified.

For each role, the property value components that must be conferred on the players of that role must be defined. The only way in which TM Applications can require property values to be conferred upon (as opposed to being built into) topics is to define assertion types whose instances confer them on their role players.

The procedure whereby each conferred property value or value component must be calculated must be specified. The calculation may involve the properties of any topics that are included in any of the role player's [situation features](#).

Note 15: The definitions of the processing steps involved in calculating property values are not constrained by this International Standard. Such processing may, for example, involve resolving addresses that are values of properties of topics included in a [situation feature](#) of the role player, and using the addressed information in further processing steps. The defined process may or may not require human input or intervention.

6.1.5 Built-in Topics

For bootstrapping reasons, all TM Applications must define at least some topics to be present in all topic maps that contain topics that are governed by the TM Application. Such topics are called "built-in" topics, and they must be defined as having "built-in values" for their SIDPs.

Note 16: The ontological basis of a TM Application, how that ontological basis is bootstrapped, and how self-documenting (in terms of the topic map) the ontology is, are all in the realm of TM Application design. For example, a TM Application may be designed in such a way that all of its assertion types are represented by built-in topics. Alternatively, a TM Application may be designed in such a way that only enough "bootstrap" assertion types (with built-in SIDPs) are required to be present to allow external definitions of all other assertion types to be used to confer the SIDP values of such assertion type subjects upon the topics that represent them.

6.1.6 Merging Rules

For every SIDP class that it defines, a TM Application must define one or more rules for determining whether to merge the topic with any other topics that have instances of the same SIDP class, solely on the basis of the values of those SIDP instances. Every such rule must be designed to serve the [Subject Location Uniqueness Objective](#).

6.1.7 Merging Rules for Built-in Values

Sometimes, topics that have built-in values must be merged. The rules for merging all property values that can be built-in must be specified.

6.2 Other Constraints on TM Application Definitions

The names of assertion types, roles, and properties must be different from each other. Each must begin with the name of the TM Application, followed by a sequence of two colons (the field separator, "::"), followed by a string that makes the name unique among the names of all other assertion types, roles, and properties defined for the TM Application. Neither the name of the TM Application, nor the second field of the names of its assertion types, roles, or properties, may contain any period ("."), any pair of consecutive colons ("::"), left or right braces ("{" or "}") , or left or right brackets ("[" or "]"). The double-colon symbol is reserved for use as the separator between the name of the TM Application (the first field of each name) and the unique portion (i.e., the second field) of the name of each of the properties, assertion types, and roles that are governed by it. The period symbol is reserved as a separator between the names of the defined components of property values. The braces are reserved as set selection operators. The brackets are reserved as list selection operators.

TM Application Definitions must not replace or optionalize the assertion structure defined by this International Standard. All relationships between the subjects of topics must be represented by a-topics.

6.3 Included TM Application Definitions

TM Application Definitions can include other TM Applications in their entirety, by reference to their definitions. All of the definitions and constraints of such "included" TM Applications become, without exception, part of the "including" TM Application. The names of included properties, assertion and role types are not affected by inclusion by another TM Application, and they retain, as their first field, the name of the included TM Application. If the included TM Application itself includes other TM Applications, recursively, they, too, are included.

Note 17: A TM Application that includes disparate TM Applications may be designed to facilitate the merging of topic maps that are governed by such disparate TM Applications.

7 Requirements for Syntax Deserialization Definitions (SDDs)

Every instance of an interchangeable topic map must implicitly or explicitly declare the Syntax Deserialization Definition that is intended to govern its interpretation. If a topic map interchange syntax is known to be interpreted or is designed to be interpretable under more than one SDD, then instances of that syntax must explicitly declare the SDD that is intended to be applied to it.

Every SDD must declare the TM Application(s) to which the topic maps whose deserializations it governs will conform.

Every SDD must unambiguously define how each kind of construct found in instances of a specific syntax must, when encountered by any implementation of the SDD, deterministically result in the addition of topics (with specific built-in and/or conferred property values) to the topic map that the syntactic instance is intended to represent.

An SDD may define one or more "[topic demander](#)" conventions that allow topics to be addressed in terms of their corresponding syntactic constructs.

Note 18: The set of topic demander conventions may or may not support the comprehensive addressability of all the topics that the SDD requires to be added to the topic map when each kind of syntactic construct is encountered.

8 Requirements for Implementations

Every implementation that claims conformance to this International Standard must also explicitly claim conformance to at least one conforming TM Application Definition. For each syntax, if any, that the implementation deserializes, it must explicitly claim conformance to at least one SDD for that syntax.

In order to respect the SLUO, it is necessary for TM Applications and their implementations to distinguish between the subjects they regard as role players, and all others. If the TM Application Definition to which an implementation claims conformance allows a given subject to exist implicitly and [unreified](#) as a property value, but does not provide a way for it to be the subject of a topic that, via an instance of some [defined](#) assertion type, is involved in conferring that property value, then the implementation must not behave in any way that would lead one to think that the subject is, in fact, [reified](#) or reifiable under the TM Application Definition. Conversely, implementations of TM Applications must not refuse to honor the playing of any role by any subject, unless it would violate the constraints imposed by any TM Application to which the implementation claims conformance or this International Standard.

All implementations of all TM Applications must:

1. Allow a-topics, r-topics, t-topics, and c-topics to be role players.
2. Always merge a-topics and c-topics as specified in this International Standard.

9 Fully Merged Topic Maps

In a topic map, every topic represents a single subject, but some subjects may be represented by more than one topic. In a "fully merged" topic map, every subject is represented by a single topic. A process whereby a topic map automatically becomes a fully merged topic map is defined in this section. Conforming implementations may use any process that achieves the same result as the process defined in this section, but all implementations must achieve that

result.

9.1 Construct the Topic Map

The topic map must be constructed from one or more interchangeable topic maps, in conformance with an applicable Syntax Deserialization Definition, or by any other process or method that yields a topic map that conforms to one or more TM Applications. The resulting topic map must represent all [reified](#) subjects as topics, and all relationships between topics as assertions, each of which has an a-topic as its nexus, in conformance with this International Standard. Except for a-topics and c-topics, all of the topics in the topic map must have at least one TMA-defined SIDP.

9.2 Validate assertion instances for conformance to definitions

All assertions that are instances of assertion types must conform to the constraints imposed by the definitions of their types, as defined by their governing TM Applications. Any conflicts between the assertions and their definitions must be reported and remedied.

9.3 Confer Values on Properties

The values of all the properties of all topics must be adjusted to conform to the requirements imposed by the definitions of all applicable TM Applications. Any conflicts with built-in values must be remedied.

Conferred property values or value components must be overridden (changed or erased) when, after the topic map has been altered in a previous iteration of the merging process, conferred property values are recalculated, and a topic's [situation](#) no longer requires one of its properties to have the value that had been conferred upon it on account of its situation in the previous iteration.

Values cannot be conferred on the SIDPs of a-topics or c-topics on account of TMA-defined [situation features](#). When such a conflict arises, it must be reported and remedied.

9.4 Validate the values of the properties of topics

Any conflicts between the values of the properties and the constraints imposed on them by their governing TM Applications must be remedied.

9.5 Merge topics

All pairs of topics that both have had values assigned (including empty values) to SIDPs that have the same name must have the values of those SIDPs compared with each other in accordance with the applicable comparison algorithm defined by the applicable TM Application Definition or, in the case of a-topics and c-topics, defined by this International Standard. If the applicable comparison algorithm reveals that the two topics should be merged, they must be merged. No information, other than the values of instances of identical SIDP classes of the topics to be merged or left unmerged, can affect the merging process.

When two topics ("predecessor topics") governed by a TM Application are merged:

1. The resulting single topic ("result topic") serves as the union of the two [situations](#) of the two predecessor topics, and all TMM-defined property values are adjusted to reflect that unification.
2. The resulting single topic exhibits the union of the built-in property values, if any, of the two predecessor topics, in accordance with the rules for merging built-in property values defined by the TM Applications that govern the affected properties.
3. all of the conferred property value and value components of the result topic, and of all other topics whose [situation features](#) are changed as a result of the merger, are adjusted in such a way as to reflect their new post-merger [situation](#), in accordance with the definition(s) of the TM Application(s) that govern those properties.

When two a-topics have the same value for their `IS13250::a-sidp` properties, and their `IS13250::a-sidp.t` value components are non-empty, they must be merged; in all other cases, they are not merged. When two c-topics have the same value for their `IS13250::c-sidp` properties, they must be merged; in all other cases, they are not merged.

Note 19: Untyped assertions never merge.

9.6 Conditionally stop or repeat

If any topics were merged in the steps described in [9.5](#), then the steps described in [9.2](#), [9.3](#), [9.4](#) and [9.5](#) must be repeated. When an iteration of these steps results in no (further) merging, the topic map has been fully merged, and processing must stop.

10 Conformance

If a Topic Maps Application Definition or Syntax Deserialization Definition complies with all the provisions of this International Standard, then it is a conforming Topic Maps Application or Syntax Deserialization Definition.

If an implementation of a conforming Topic Maps Application and/or Syntax Deserialization Definition complies with all the provisions of this International Standard and with the constraints imposed by the Topic Maps Application(s) and/or Syntax Deserialization Definition(s) to which it claims conformance, it is a conforming implementation.

If the interchangeable form of a topic map implicitly or explicitly declares the Syntax Deserialization Definition (SDD) that is intended to govern its interpretation, and if it is unambiguously interpretable in terms of that SDD, then it is a conforming interchangeable topic map.

If a topic map conforms to the specifications of this International Standard and to the constraints imposed by all of its governing Topic Maps Applications, it is a conforming topic map.