ISO/IEC JTC 1/SC34 N0344

ISO/IEC JTC 1/SC34

Information Technology --

Document Description and Processing Languages

TITLE:	Reference Model for ISO 13250 Topic Maps (RM4TM)
SOURCE:	Steven R. Newcomb, Sam Hunting and Jan Algermissen
PROJECT:	Topic Map Models
PROJECT EDITORS:	Michel Biezunski, Martin Bryan, Steven R. Newcomb
STATUS:	Editor's Draft, Revision 1.0
ACTION:	For review and comment
DATE:	12 November 2002
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

12 November 2002

JTC 1/SC 34 N 344

Reference Model for ISO 13250 Topic Maps (RM4TM)

Version 1.0, 2002/11/12
(The current editors' working copy is available at http://www.isotopicmaps.org/rm4tm/.)

0 Introduction

This Reference Model for ISO 13250 Topic Maps (RM4TM) provides a framework for the definitions of Topic Map Applications (TM Applications). Diverse topic maps that conform to diverse TM Applications that are defined in keeping with this framework can be interpreted and amalgamated automatically by independently implemented systems, without losing information, and with predictable results.

Many of the key advantages of the Topic Maps paradigm derive from the achievement of its primary objective, the "Subject Location Uniqueness Objective", which is to make everything known about every subject in a topic space accessible from a single location within that space. The achievement of the Subject Location Uniqueness Objective means that the efficiency with which users can find information is maximized, not only because the subject's single location, once found, acts as a comprehensive catalog of the things that are known about it, but also because the subject's location can be found in terms of any of its relationships to other subjects.

This RM4TM facilitates the development of TM Applications and systems that can achieve the Subject Location Uniqueness Objective with respect to all subjects, including those that are only implicit in interchangeable topic map instances, as well as with respect to subjects that are relationships (and aspects of relationships) among other subjects.

Moreover, this RM4TM facilitates the development of TM Applications and implementations that can amalgamate the topic spaces represented by topic maps that conform to diverse Topic Maps Applications into a single resulting topic space in which each subject has a single location, there is no redundant information, and all of the information represented by the comprising topic maps is preserved.

This RM4TM provides definition requirements for user-defined Topic Map Applications that allow such definitions to serve as contracts between topic map creators, users, and system implementers, such that when the interchange or amalgamation of topic maps fails due to nonconformance to the definition of a Topic Maps Application, the nonconforming aspects of the topic maps or system implementations can be identified.

1 Scope

This RM4TM defines:

1. an abstract graph structure for the representation of relationships between subjects;
2. rules for defining Applications of the Topic Maps paradigm; and
3. rules for processing the information contained in topic maps.

Note 1: See Annex A for a brief informal overview of this RM4TM.

2 Glossary

Editor's Note 1: (The glossary hasn't been drafted yet.)

3 Topic map graphs

3.1 The common structural abstraction for topic maps

This RM4TM defines an abstract structure, called a "topic map graph", in terms of which all kinds of topic maps can be uniformly interpreted, regardless of their governing TM Applications, and regardless of the TM Application-defined interchange syntaxes in which they may be representable.

The "topic map graph" form of any given topic map represents all of the subjects that participate in the topic map explicitly, even if they were only implicitly represented in the interchangeable form of the given topic map.

The following subclauses name and define the rules and cases to which topic map graph components and entire topic map graphs must conform in order to be considered "well formed", and the additional rules to which topic map graphs must conform in order to be considered "fully merged". Topic map graphs that are under construction may or may not be well-formed, but only well-formed topic map graphs are eligible to become fully merged, in addition to being well-formed.

3.2 Topic map graphs consist of nodes and arcs.

A topic map graph consists of nodes and arcs. In a well-formed topic map graph, every arc is a typed, oriented connectedness of two nodes, and every node is one of the two endpoints of zero or more arcs.

Note 2: This RM4TM uses the neologism "connectedness" in order to avoid implying that TM Applications must be implemented in such a way that arcs are represented as a data structure. For example, The arc abstraction can be fully honored by the property values of the nodes that serve as its endpoints.

3.3 Arcs

Note 3: The reader's understanding of the remainder of this clause 3 is likely to be aided by referring to the informative "Assertion Diagrams" Annex B.

An "arc" in a topic map graph is a two-ended connectedness between nodes that satisfies all of the following criteria:

- it has two *different* nodes serving as its two endpoints, and
- it is one of the eight forms of connectedness enumerated in 3.3.3 between the nodes that serve as its two endpoints. (This necessarily means that it is one of the four arc types enumerated in 3.3.1.)

3.3.1 Four arc types

There are four arc types, named "AT", "AC", "CR", and "C*x*". The significance of each type of arc is different.

3.3.2 Names of arc types and arc endpoint types

The first letter of an arc type's name is the name of one of its endpoint types. The second letter of the arc type's name is the name of its other endpoint type. That is, an AT arc has two endpoints, one of endpoint type "A" and the other of endpoint type "T".

Note 4: In a well-formed topic map graph, only a-nodes serve as "A" endpoint types, only c-nodes serve as "C" endpoint types, only r-nodes serve as the "R" endpoint types, and only t-nodes serve as the "T" endpoint types. There is no such thing as an "x-node", because all kinds of nodes are eligible to serve as the *x* endpoints of C*x* arcs. The e*x*ceptional character of the *x* endpoints of C*x* arcs is the reason why "*x*" is the only endpoint type name that is always rendered in lower case.

3.3.3 Eight forms of connectedness are possible

In all instances of each type of arc, the significance of a node's service as one of the endpoints is different from the significance of a node's service as the other endpoint. Given two nodes, N1 and N2, there are eight possible forms of connectedness between them, since there are four types of arcs. They are enumerated in the following subclauses.

3.3.3.1 Form 1: "A" to "T"

The connectedness of N1 and N2 is an instance of an AT arc type in which N1 is the A endpoint, and N2 is the T endpoint.

3.3.3.2 Form 2: "T" to "A"

The connectedness of N1 and N2 is an instance of an AT arc type in which N1 is the T endpoint, and N2 is the A endpoint. (This is the reverse of Form 1.)

3.3.3.3 Form 3: "A" to "C"

The connectedness of N1 and N2 is an instance of an AC arc type in which N1 is the A endpoint, and N2 is the C endpoint.

3.3.3.4 Form 4: "C" to "A"

The connectedness of N1 and N2 is an instance of an AC arc type in which N1 is the C endpoint, and N2 is the A endpoint. (This is the reverse of Form 3.)

3.3.3.5 Form 5: "C" to "R"

The connectedness of N1 and N2 is an instance of a CR arc type in which N1 is the C endpoint, and N2 is the R endpoint.

3.3.3.6 Form 6: "R" to "C"

The connectedness of N1 and N2 is an instance of an CR arc type in which N1 is the R endpoint, and N2 is the C endpoint. (This is the reverse of Form 5.)

3.3.3.7 Form 7: "C" to "*x*"

The connectedness of N1 and N2 is an instance of a C*x* arc type in which N1 is the C endpoint, and N2 is the *x* endpoint.

3.3.3.8 Form 8: "*x*" to "C"

The connectedness of N1 and N2 is an instance of a C*x* arc type in which N1 is the *x* endpoint, and N2 is the C endpoint.

Note 5: The above list of Forms of Connectedness can be represented in tabular form as follows:

	N1	N2
1	A	T
2	T	A
3	A	C
4	C	A
5	C	R
6	R	C
7	C	*x*
8	*x*	C

Note 6: The above enumeration of the Forms of Connectedness serves two purposes in this RM4TM:

1. It establishes a name ("Form n", where n is an integer in the sequence 1..8) for each of the Forms of Connectedness that an arc can represent, as a convenience for use elsewhere in this document, and possibly in the definitions of TM Applications.

2. It establishes that the orientation of the connectedness represented by an arc is an essential aspect of the definition of "arc" in this RM4TM. For purposes of a TM Application’s definition of a "situation feature" (see 3.4.2), for example, it is insufficient merely to say that two nodes are connected by a certain type of arc. The specification of the arc must also include information as to which node serves as which endpoint type. In order to represent connectedness equivalent to the connectedness represented by an RM4TM arc in some "directed graph" paradigms, at least two directed graph arcs must be used, plus whatever additional machinery may be required to associate the two directed graph arcs in order to represent that both represent different directional aspects of the same connectedness. By contrast, RM4TM arcs are *non*directional, but *oriented.*

3.4 Nodes and subjects

3.4.1 One subject for each node

In topic map graphs, only nodes can represent subjects, and every node represents a single subject.

3.4.2 Situations and subjects

A node serves as one endpoint of zero or more arcs.

Note 7: A node that serves as the endpoints of no arcs at all is not well-formed unless it has at least one built-in SIDP value. (See 3.4.2.)

A node that is the endpoint of zero arcs is said to be "isolated." In a well-formed topic map graph, only "built-in" nodes (see Clause 4) can be isolated.

A node that is the endpoint of one or more arcs is said to be "situated." A node’s "situation" is its service as one of the endpoints of all of the "connected paths" through the graph to all other nodes accessible via such paths. (Given node n[0], a "connected path" is a finite alternating sequence n[0], arc[1], n[1], arc[2], n[3]... n[n] such that each arc[i] in the sequence connects node[i-1] and node[i].)

Except for the built-in values of the properties of built-in nodes, all of the values of the properties of nodes are determined by their situations. Thus, except for the built-in subjects of built-in nodes, the *subjects* of all nodes are entirely determined by their situations.

Except for the restrictions on the subjects of nodes that have special functions within assertion subgraphs (see 3.6.2.2), TM Applications are free to define "situation features" (features of the situations of nodes) and how those features, when they occur, affect the values of the properties of the nodes whose situations include those situation features. The values of all properties can be affected by such situation features, including both Subject Identity Discriminating Propertes (SIDPs) and Other Properties (OPs), in accordance with the specifications provided in the definition of the TM Application that defines the properties and the situation features (see 4.7.2.2).

Note 8: The situation of a node in a topic map graph is always and only as visible as the values of its properties make it. See Clause 4.

Note 9: The definition of a situation feature can include, but is not limited to, the situated node’s status as a role player in one or more assertions. The definition of a situation feature can also include the situated node’s status as another kind of assertion component node, such as an r-node component of one or more assertions (see 3.6.2.2).

3.5 Well-formed nodes

3.5.1 Six cases of well-formed nodes

A node that satisfies all the criteria in the subclauses of one of the six cases described in the following subclauses is well formed. A node that does not satisfy the criteria of one of the six cases is not well formed.

3.5.1.1 Well-formed node Case 1

3.5.1.1.1 Defining Characteristics of Case 1 nodes

3.5.1.1.1.1 The node serves as no endpoint of any arc.

3.5.1.1.1.2 The node has at least one built-in SIDP value (see Clause 4).

3.5.1.1.2 Node type name of Case 1 nodes

Case 1 nodes do not have a node type name.

3.5.1.1.3 Subjects of Case 1 nodes

The subjects of Case 1 nodes are not constrained by this RM4TM.

3.5.1.2 Well-formed node Case 2

3.5.1.2.1 Defining characteristics of Case 2 nodes

3.5.1.2.1.1 The node serves as one or more of the *x* endpoints of any number of well-formed C*x* arcs.

3.5.1.2.1.2 The node does not serve as any other endpoint type of any instance of any arc type.

3.5.1.2.1.3 The node either has at least one built-in SIDP value, or its situation as a role player causes at least one SIDP value to be conferred upon it.

3.5.1.2.2 Node type name of Case 2 nodes

Case 2 nodes do not have a node type name.

3.5.1.2.3 Subjects of Case 2 nodes

The subjects of Case 2 nodes are not constrained by this RM4TM.

3.5.1.3 Well-formed node Case 3 ("a-node")

3.5.1.3.1 Defining characteristics of Case 3 nodes

3.5.1.3.1.1 The node serves as zero or more of the *x* endpoints of any number of C*x* arcs.

3.5.1.3.1.2 The node serves as the A endpoint of two or more AC arcs.

3.5.1.3.1.3 The node may or may not serve as the A endpoint of one AT arc.

3.5.1.3.1.4 The node does not serve as any other endpoint of any instance of any arc type.

3.5.1.3.2 Node type name of Case 3 nodes

A Case 3 node is called an "a-node" (where "a" stands for "assertion").

3.5.1.3.3 Subjects of Case 3 nodes

The subject of an a-node is always the relationship that is specified via the assertion for which it serves as the unique nexus. The relationship is an instance of the type of relationship which is the subject of the node that serves as the T endpoint of the AT arc of which the a-node is the A endpoint, if any. If the a-node is not the A endpoint of an AT arc, the type of the relationship is unspecified.

3.5.1.4 Well-formed node Case 4 ("c-node")

3.5.1.4.1 Defining characteristics of Case 4 nodes

3.5.1.4.1.1 The node serves as zero or more of the *x* endpoints of any number of C*x* arcs.

3.5.1.4.1.2 The node serves as the C endpoint of a single AC arc.

3.5.1.4.1.3 The node serves as the C endpoint of a single CR arc.

3.5.1.4.1.4 The node may or may not serve as the C endpoint of a single C*x* arc.

3.5.1.4.1.5 The node does not serve as any other endpoint of any instance of any arc type.

3.5.1.4.2 Node type name of Case 4 nodes

A Case 4 node is called a "c-node" (where "c" stands for "casting").

Note 10: The term "casting" is consistent with the theatrical metaphor invoked by the term "role player". In an assertion, the role players are like the actors in a stage play. Each c-node represents the "casting" of an actor (a role player) in a specific role (a role type) in a specific stage production (a specific assertion), which may or may not be a production of a specific stage play (a specific assertion type). See 3.6.1.

3.5.1.4.3 Subjects of Case 4 nodes

3.5.1.4.3.1 Case 4 nodes with role players

If a c-node serves as the C endpoint of a C*x* arc, then its subject is the playing of a specific role type by a specific subject in a specific relationship.

3.5.1.4.3.2 Case 4 nodes without role players

If a c-node does not serve as the C endpoint of a C*x* arc, then its subject is the fact that a specific role type in a specific relationship is not played by any subject.

3.5.1.5 Well-formed node Case 5 ("r-node")

3.5.1.5.1 Defining characteristics of Case 5 nodes

3.5.1.5.1.1 The node serves as zero or more of the *x* endpoints of any number of C*x* arcs.

3.5.1.5.1.2 The node serves as the R endpoint of one or more CR arcs.

3.5.1.5.1.3 The node does not serve as any other endpoint of any instance of any arc type.

3.5.1.5.2 Node type name of Case 5 nodes

A Case 5 node is called an "r-node" (where "r" stands for "role type").

3.5.1.5.3 Subjects of Case 5 nodes

The subject of an r-node is a role type that can be played by subjects in relationships. The subjects of the c-nodes that serve as the C endpoints of the CR arcs whose R endpoints are the r-node are the role-player castings of role players that play the role type.

3.5.1.6 Well-formed node Case 6

3.5.1.6.1 Defining characteristics of Case 6 nodes ("t-node")

3.5.1.6.1.1 The node serves as zero or more of the *x* endpoints of any number of C*x* arcs.

3.5.1.6.1.2 The node serves as the T endpoint of one or more AT arcs.

3.5.1.6.1.3 The node does not serve as any other endpoint of any instance of any arc type.

3.5.1.6.2 Node type name of Case 6 nodes

A case 6 node is called a "t-node" (where "t" stands for assertion "type").

3.5.1.6.3 Subjects of Case 6 nodes

The subject of a t-node is a class of relationship, including the roles that can be played in instances of the class, and the values that are conferred on the properties of role players by virtue of their situations as players of specific roles in instances of the class. The subjects of all of the a-nodes that serve as the A endpoints of all of the AT arcs of which a t-node serves as the T endpoint are instances of the class of relationship that is the subject of the t-node.

Note 11: The above well-formedness requirements for nodes can be summarized in tabular form as follows:

Table 1: The Six Cases of Well-formed Nodes

Form of Connectedness (node N2) node N1	8 .... C *x*	7 .... *x* C	6 .... C R	5 .... R C	4 .... A C	3 .... C A	2 .... A T	1 .... T A	node type name (if any).	Subject constraint (if any). Subject is:	requires built-n SIDP value(s)?
N1 Case 1	0	0	0	0	0	0	0	0	(none)	(unconstrained)	yes
N1 Case 2	0	1+	0	0	0	0	0	0	(none)	(unconstrained)	no
N1 Case 3	0	0+	0	0	0	2+	0	1?	"a-node"	assertion	no
N1 Case 4	0	0+	1?	0	1	1	0	0	"c-node"	casting	no
N1 Case 5	0	0+	0	1+	0	0	0	0	"r-node"	role type	no
N1 Case 6	0	0+	0	0	0	0	1+	0	"t-node"	assertion type	no

Legend:

0 In order to conform to the well-formed node case described on this row, node N1 is not permitted to serve as the arc endpoint designated by this column.

0+ In order to conform to the well-formed node case described on this row, node N1 may serve as zero or more of the arc endpoints designated by this column.

1 In order to conform to the well-formed node case described on this row, node N1 *must* serve as exactly one of the arc endpoints designated by this column.

1? In order to conform to the well-formed node case described on this row, node N1 *may* serve as exactly one of the arc endpoints designated by this column.

1+ In order to conform to the well-formed node case described on this row, node N1 *must* serve as *at least one* of the arc endpoints designated by this column.

2+ In order to conform to the well-formed node case described on this row, node N1 *must* serve as *at least two* of the arc endpoints designated by this column.

3.6 Assertions

3.6.1 Introduction to assertions

Assertions are subgraphs of topic map graphs. In a well-formed topic map graph, every arc is a specific component of a single assertion, so well-formed topic map graphs consist entirely of assertions (except, possibly, for isolated "built-in" nodes).

Each assertion represents (asserts the existence of) a single strongly-typed relationship among the subjects that are its "role players". Each role player is a subject that plays a specific role in the relationship. The roles ("role types") themselves are subjects, and so is the type of relationship of which the relationship is an instance.

The design of assertions in this RM4TM enables diverse multiple topic map graphs to be amalgamated into a single topic map graph, such that:

- each of the original topic map graphs is a subgraph of the result, and
- each such subgraph is structurally identical to the corresponding original, even when one of them makes assertions about assertions in the other, about which the other made no assertions. Thus, the integrity of the original topic map graphs is maintained as subgraphs of the result.

Note 12: In order to maintain the integrity of merged topic maps, it is necessary to establish a common structure for all assertions. In this RM4TM, the decisions as to which aspects of the structure of assertions should be "reified" as nodes, and which aspects should remain "unreified" as arcs, were made by distinguishing between the aspects of assertions that are substantive with respect to the relationships that they assert (and that could conceivably, therefore, need to become role players in other assertions about those relationships), as opposed to the aspects of assertions that nobody would want to make other assertions about unless they were discussing the design of assertions in general. In the structure of assertions set forth in this RM4TM, the former aspects are represented by a-nodes and c-nodes, while the latter aspects are represented as the four types of arcs (the "eight forms of connectedness").

3.6.2 Inventory of the components of assertions

An assertion is a subgraph of a topic map graph that consists of certain arcs and the nodes that serve as their endpoints, constructed in conformance to the rules set forth in this clause. Every node, regardless of its node type, is eligible to be a role player (i.e., to serve as the *x* endpoint of a C*x* arc) in any number of assertions. Every arc is a component of a single assertion. The entire significance of every arc is its service as a unique component of a single assertion.

3.6.2.1 Inventory of the arcs in an assertion

The inventory of arcs that an assertion may have are defined in the subclauses that follow.

3.6.2.1.1 One or zero AT arcs

Note 13: The assertion type of an assertion may be specified or unspecified.

3.6.2.1.2 Two or more AC arcs

Note 14: In every assertion, there must be at least two role types, and therefore there must be at least two casting nodes.

3.6.2.1.3 Exactly as many RC arcs as there are AC arcs

Note 15: Every casting node must have a role type, as well as belong to a single assertion.

3.6.2.1.4 At least one C$_X$ arc

Note 16: Every assertion must have at least one role player.

3.6.2.2 Inventory of the nodes in an assertion

3.6.2.2.1 Nodes whose subjects are never dependent on their situation with respect to a given assertion:

3.6.2.2.1.1 Assertion type nodes (t-nodes; i.e., T endpoints of AT arcs)

3.6.2.2.1.2 Role type nodes (r-nodes; i.e., R endpoints of CR arcs)

3.6.2.2.2 Nodes whose subjects are always dependent on their situation with respect to a given assertion:

3.6.2.2.2.1 Assertion nodes (a-nodes; i.e., A endpoints of AT and AC arcs)

An assertion always includes a single well-formed a-node which serves as its unique nexus. The a-node’s subject is the relationship that the assertion represents.

3.6.2.2.2.2 Casting nodes (c-nodes; i.e., C endpoints of AC, CR, and C$_X$ arcs)

An assertion always includes at least two c-nodes. The subject of every c-node is that a specific role player (or that no role player at all) plays a specific role type in a specific assertion.

3.6.2.2.3 Nodes whose subjects may or may not be dependent on their situation with respect to a given assertion (role player nodes):

The governing TM Application defines situation features and their effects on the values of the SIDPs of role players. Except in cases where a subject (specified by a set of SIDP values) has been defined by the governing TM Application as being built into a node, a node’s subject depends entirely on the features of its situation (its "situation features" - see 3.4.2), on account of which the governing TM Application requires values to be conferred on the values of one or more of its SIDPs. Therefore, the situations of nodes as players of certain roles in instances of certain assertion types may or may not determine their subjects.

Note 17: For example, the subject of a node may be determined by its situation as a role player in a single assertion, even though it is also a role player in many others. For another example, the subject may be collectively determined by multiple assertions, perhaps by virtue of playing a role type or set of role types in a set of assertions, or perhaps by playing a role in an assertion in which another roleplayer’s subject is collectively determined.

3.6.2.3 What’s in and what’s not in an assertion

The assertion of which a given a-node is the unique nexus includes all of the nodes and arcs enumerated in the following subclauses, and it does not include any other nodes and arcs:

3.6.2.3.1 All of the AC arcs of which the given a-node serves as the A endpoint.

3.6.2.3.2 The well-formed c-nodes that serve as the C endpoints of the AC arcs identified in 3.6.2.3.1.

3.6.2.3.3 The RC arcs that have the c-nodes identified in 3.6.2.3.2 as their C endpoints.

3.6.2.3.4 The well-formed r-nodes that serve as the R endpoints of the RC arcs identified in 3.6.2.3.3.

3.6.2.3.5 The C*x* arcs that have the c-nodes identified in 3.6.2.3.2 as their C endpoints.

3.6.2.3.6 The well-formed nodes that serve as the *x* endpoints of the C*x* arcs identified in 3.6.2.3.5.

3.6.2.3.7 The AT arc, if any, of which the given a-node serves as the A end.

3.6.2.3.8 The well-formed t-node that serves as the T endpoint of the AT arc, if any, identified in 3.6.2.3.7.

3.6.3 Identity of assertions

Two assertions are always considered identical if they have the same assertion type, and the same role players (or the absences of role players) play the same roles. Two assertions are never considered identical, even if they have the same role players playing the same roles, if either or both of their assertion types are unspecified. This clause provides the operational definitions of these concepts.

The identity of the relationship instance that is the subject of an a-node is defined by that a-node’s situation as the nexus of an assertion subgraph. For all a-nodes, every TM Application is required to define a situation feature and a set of one or more SIDPs that unambiguously, comprehensively and exclusively reflects the combination of the following:

- unless the assertion’s type is unspecified, the t-node (whose subject is the type of relationship of which is the subject of the a-node is an instance) attached to the a-node by an AT arc in which the a-node serves as the A endpoint; and
- the set of role-player castings that are the subjects of the c-nodes that serve as the C endpoints of the AC arcs for which the a-node serves as the A endpoints,
    - including the role player node attached to each c-node by a C*x* arc in which the c-node serves as the C endpoint, or the lack thereof, and
    - including the r-node (whose subject is a role type) attached to each c-node by a CR arc in which the c-node serves as the C endpoint.

Note 18: One of the key features of this RM4TM is that the merging process does not need to understand the semantics of assertion types in order to merge identical assertions. If two assertions have the same type, regardless of what it is, and the same role players playing the same role types, regardless of what they are, they can be seen to be identical and automatically merged.

3.6.4 Assertion semantics

3.6.4.1 Semantics of assertion typing

3.6.4.1.1 When the assertion type is specified

A "typed" assertion is an assertion that specifies its assertion type (i.e., that has an AT arc and t-node). The semantics of a typed assertion are determined by the subject of its t-node, which is the assertion type of which the typed assertion is an instance. The subject of the t-node incorporates the semantics of all of the role types that can have role players in instances of the assertion type, all of which must be specified in the definition of the subject of the assertion type, either by reference or inclusion.

The semantics of a typed assertion may determine or affect the subjects of some or all of its role players, i.e., the existence of such an assertion may affect the values assigned to the SIDPs of its role players (see 4.7.2).

3.6.4.1.2 When the assertion type is not specified

An "untyped" assertion is an assertion that does not specify its assertion type (i.e., that has no AT arc). The semantics of an untyped assertion are determined by its role types, i.e., by the subjects of its r-nodes. The semantics of its role types may be such that the players of the role types have values conferred on their OPs (Other Properties -- see 4.4). However, the role types of untyped assertions must not be defined in such a way as to require values to be conferred upon the SIDPs of their players (see 5.2.5.3.2).

3.6.4.1.3 The subjects of assertion types and role types are never affected by their instances

The existence of a given assertion never implies anything about the subject which is the assertion type (if any) of which the assertion is an instance, or about the subjects that are the assertion’s role types. No values can be conferred upon the SIDPs of assertion types or role types by virtue of their situations, respectively, as the T endpoints of AT arcs, or as the R endpoints of CR arcs.

Note 19: Like all other nodes, the t-node and r-nodes that represent the subjects that are an assertion’s type and role types, respectively, may have their subjects (i.e., the values of their SIDPs) built into them, or their subjects may be conferred upon them by virtue of their situations as role players in other assertions.

Note 20: TM Applications may confer values on the OPs of t-nodes and r-nodes by virtue of their situations as t-nodes and r-nodes.

3.6.4.2 Semantics of role playing

3.6.4.2.1 No multiple role players of a single role type

In any given assertion, each role type is either played by a single subject, represented by a single node, or the role type is "unplayed", i.e., the role type has no role player. Multiple subjects cannot play the same role in the same assertion.

Note 21: However, the subject of a role player can be a group of subjects, if the governing TM Application defines the assertion types required to allow the subjects of nodes to be groups of subjects.

No grouping semantics of any kind are defined by this RM4TM. This RM4TM requires all groups to be explicitly represented as nodes. Any other approach would open the possibility for knowledge about a group to fail to be connected to the single node whose subject is the group, and that would be contrary to the Subject Location Uniqueness Objective.

3.6.4.2.2 Semantics of nodes' situations as role players

A node's situation as a role player in any given assertion indicates that the subject represented by that node participates in the relationship that is the subject of the assertion, as represented by the assertion's a-node. In an asserted relationship, each role player plays a distinct role; the nature of each role is the subject (called a "role type") of one of the assertion's r-nodes. The relationship itself is an instance of the kind of relationship that is the subject of the assertion's t-node, if any. If the assertion has no t-node, the subject of which the relationship is an instance is not specified.

3.6.4.2.3 All role types are always represented in any assertion of a given type

In the topic map graph, the representation of every assertion always includes the representation of all of the role types defined by its assertion type's definition, regardless of whether they are played or unplayed. (If the assertion type is unspecified, then the set of role types that the assertion specifies is assumed to be comprehensive for that assertion.)

3.7 Well-formedness constraints on Assertions

An assertion that does not conform to all of the following rules is not well-formed:

3.7.1 No two role types the same; each has zero or one role player

No two c-nodes that participate in the assertion are connected to the same r-node via the CR arcs for which the c-nodes serve as the C endpoints.

The role types that participate in any given assertion instance must always constitute a set, i.e., within any single assertion, no two role types can be the same. Each role type has a maximum of one role player.

Note 22: If the governing Application defines assertion types that allow nodes to have subjects that are groups of subjects, such a group of subjects can be a role player. Still, even in such cases, there is still only one role player: the group.

3.7.2 There must be at least one role player

The set of arcs that are members of the set of arcs that specify the assertion must include at least one C*x* arc.

3.8 Well-formedness constraints on topic map graphs

A topic map graph that conforms to the criteria specified in both of the following clauses is well-formed. A topic map graph that does not satisfy either or both criteria is not well-formed.

3.8.1 There is at least one node.

3.8.2 There are no arcs that do not participate in a single well-formed assertion.

3.9 Well-formed and fully merged topic map graphs

When a topic map takes the form of a topic map graph, all of the subjects that participate in the topic map are represented as nodes.

In a well-formed topic map graph, every node represents a single subject, but some subjects may be represented by more than one node. In a fully merged topic map graph, every subject is represented by a single node.

A well-formed topic map graph may or may not be fully merged, but a fully merged topic map graph is always well-formed.

A topic map graph that does not meet this RM4TM’s criteria for well-formedness is not eligible to undergo the merging process.

Note 23: The process whereby well-formed topic map graphs are converted into fully merged topic map graphs is defined in Clause 6.

4 Properties of nodes

4.1 Only a common framework for properties; no common properties

This RM4TM defines a framework within which each TM Application defines all of the properties of the nodes that it governs. The framework is designed to constrain the definitions of TM Applications in such a way that they can be implemented independently, with each implementation able to demonstrate the conformance of its behavior to the definition of the TM Application, and, therefore, with the behavior of all other conforming implementations.

Note 24: This RM4TM defines no properties of nodes. It does, however, impose certain constraints on the definitions of such properties within the definitions of TM Applications.

4.2 Every property is governed by a single TM Application

All of the properties of nodes, their value types, and the requirements for assigning values to them are defined by TM Applications. Every property defined by a TM Application, and every node that exhibits values for any of the properties defined by that TM Application, is said to be "governed" by that TM Application. Every node must be governed by one or more TM Applications. Every property is governed by a single TM Application.

4.3 Subject identity discrimination properties ("SIDPs")

4.3.1 Identical subjects must be recognizably identical

The fact that two nodes have the same subject must be detectable in order to trigger the merging operations that transform a well-formed topic map graph into a fully merged one. Therefore, at least one property of every node must be defined by its governing TM Application for the express purpose of allowing the subject of the node to be distinguishable from all other subjects, and in order to allow the subjects of nodes, when they are identical, to be recognizable as identical by the topic map graph merging process. Such properties are called "Subject Identity Discrimination Properties" (SIDPs). The values of SIDPs, and no other data of any kind, are used in TM Application-defined calculations to determine whether any two nodes should be merged.

4.3.2 Subject identity is the values of SIDPs

All merging rules defined by a TM Application must serve the Subject Location Uniqueness Objective, and all must be expressed entirely in terms of the values of the SIDPs defined by that TM Application. TM Applications must define sufficient SIDPs, and constrain the calculations and assignments of their values, in sufficient detail to support all of the merging rules defined by the TM Application.

4.3.3 The merging of nodes

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

1. the resulting single node ("result node") serves as the union of the two sets of arc endpoints of the two predecessor nodes,

2. the resulting single node exhibits the union of the built-in property values, if any, of the two predecessor nodes, and

3. all of the property values of the result node, and of all other nodes whose situation features are changed as a result of the merger, are adjusted in such a way as to reflect their new situations, in accordance with the definition(s) of the TM Application(s) that

govern the properties.

Note 25: Nodes never merge for any reason other than the fact that they are regarded as having the same subject; all merging operations must serve the Subject Location Uniqueness Objective. However, TM Applications may require the application of any number of rules for determining whether two nodes have the same subject. Such merging rules may be based on diverse combinations of subject property values, each of which may be based on a complex situation feature definition, possibly involving intermediary assertions and nodes through which the situated node is connected to many other nodes.

4.3.4 RM4TM constrains the SIDPs and SIDP values of a-nodes and c-nodes

The subjects of a-nodes and c-nodes are comprehensively and exclusively defined by this RM4TM in terms of their situations in the assertions of which they are components. The properties and value-assignment rules of TM Applications are not permitted to override, obscure, add to, or fail to expose these subjects.

4.4 Other properties ("OPs")

TM Applications may also define properties whose values are not used for subject discrimination purposes; such properties are called "OPs" (other properties). TM Applications define the purposes of OPs, and the processes by which their values are calculated and assigned.

4.5 Names of properties of nodes

Each property has a name that is unique, within the TM Application, among all the names of the properties, assertion types, and role types defined by the TM Application. In a topic map graph, however, property names may be defined by multiple TM Applications, so different TM Applications may define the same property name. Therefore, each property name consists of two fields, separated by the field separator symbol defined in 4.5. The first field is the name of the TM Application itself, and the second field is the property name which is unique within the TM Application.

Editor’s Note 2: TO DO: Select a field separator symbol, so everybody knows what *not* to use in the name of a TM Application, property, assertion type, or role type. It can’t be a colon (":") if we expect people to use IETF scheme names in their TM Application-name URIs, such as "http:".

4.6 Values of properties of nodes

The values of properties of nodes, the types of their values, and the methods whereby their values are calculated and assigned, are all defined by their governing TM Applications.

4.7 Assignment of values of properties of nodes

The values of the properties of nodes are assigned in two ways. They are either:

1. "built-in" or
2. "conferred".

4.7.1 Built-in values of properties of nodes

For bootstrapping reasons, TM Applications must define at least some nodes to be present in all topic map graphs that contain nodes that are governed by the TM Application, regardless of whether they appear explicitly in any interchangeable topic map governed by that TM Application. Such nodes are called "built-in" nodes, and they must be defined as having "built-in values" for at least one of their SIDPs.

A node's built-in property values cannot be overridden by virtue of its situation in the topic map graph. It is a Reportable TM Processing Error if a built-in node's situation requires any of its properties that have built-in values to have values conferred upon them that are different than their built-in values.

Note 26: Values can be conferred on properties of built-in nodes that do not have built-in values.

Note 27: The determination of 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 nodes. 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 nodes that represent them.

4.7.2 Conferred values of properties of nodes

The properties of nodes can have values that are conferred upon them by their nodes' situations in the topic map graph. These values are called "conferred" values.

4.7.2.1 Overview of requirements governing definitions of conferred property values

With respect to the values conferred on the properties of nodes, TM Applications must define:

1. the situation features of nodes that call for values to be conferred upon the properties of such nodes,
2. the properties of such nodes to which the values are assigned,
3. the types of the property values, and
4. how the values are calculated.

Note 28: The definitions of the processing steps involved in calculating property values are not constrained by this RM4TM. Such processing may, for example, involve resolving addresses and using whatever information is addressed in further processing steps.

4.7.2.2 Situation features that TM Applications define as requiring values to be conferred on the properties of nodes

For all purposes of defining situation features that require values to be conferred on the properties of nodes, such situation features may be described in terms of whole assertions, or in terms of specific nodes and arcs, or both. In any case, however, for a given node, a situation feature is always fundamentally describable as the given node’s service as the endpoints of some set of paths whose characteristics are defined by the TM Application as constituting a situation feature that requires values to be conferred.

When a node’s service as the *x* endpoint of one or more C*x* arcs (i.e., when a node’s situation as a role player) is an aspect of a TM Application-defined situation feature that requires values to be assigned to one or more of its properties, the definitions of such situation features, the properties to which the values are assigned, the types of the values, and how the values are calculated, must all be defined as part of, or at least with respect to, the definition of the type of assertion of which the assertion that has the node as a role player is an instance.

Note 29: For example, if the TM Application defines an assertion type for the purpose of expressing set memberships, in which one role is played by the node whose subject is the set, and the other role is played by a node whose subject is a member of the set, then the value of the corresponding property of the node can be a node set which is the set of all the nodes whose subjects are members of the set.

Note 30: Not all situation features that require property values to be conferred are situations in which the conferred-upon node is a role player. Some situation features are within a single assertion subgraph. For example, all TM Applications must define a property for all the a-nodes they govern, whose value is the assertion type of the a-node; this property value is conferred upon it on account of its service as the A endpoint of an AT arc (see 4.3.4).

4.7.2.3 SIDP values cannot be conferred on a-nodes or c-nodes on account of their

situations as role players.

The SIDP values that reflect the subjects of a-nodes and c-nodes, and that, therefore, determine whether they should be merged, can only be conferred upon them by virtue of their service as the A and C endpoints of arcs. This RM4TM defines the merging rules for assertions (see 5.2.8.2), and conforming TM Applications cannot violate these rules. Therefore, TM Applications cannot require the values of the subject identity discrimination properties (SIDPs) of a-nodes or c-nodes to be conferred upon them on the basis of their situations as role players (i.e. on the basis of their service as the *x* endpoints of C*x* arcs).

4.7.2.4 SIDP values cannot be conferred on either r-nodes or t-nodes on account of their situations as R or T endpoints of CR or AT arcs, respectively.

The SIDP values that reflect the subjects of r-nodes and t-nodes are not, and cannot be, conferred upon them by virtue of their service as the R endpoints of any CR arcs, or the T endpoints of AT arcs, respectively. SIDP values can only be conferred upon r-nodes and t-nodes by virtue of their situations as role players (i.e., as the *x* endpoints of C*x* arcs. (Alternatively, their SIDP values can be built-in.)

4.8 Internal consistency of the values of a node’s SIDPs

TM Applications must define consistency rules regarding the combinations of values that any given node’s SIDPs can exhibit in order for that node to be regarded as exhibiting a valid combination of SIDP values. Merging processes must be implemented in such a way as to detect and report (as Reportable TM Processing Errors) conditions that violate these consistency rules.

Note 31: For example, if one of a node’s SIDP values indicates that the node’s subject is a name, and another SIDP value indicates that the node’s subject is a set of subjects, the definition of the TM Application can require such a node to be regarded as exhibiting an invalid combination of SIDP values. By stating such a constraint, the TM Application’s definition can reflect its designers’ conviction that there can never be a single subject that is both a name and a set.

5 Definitions of TM Applications

5.1 Introduction

This RM4TM constrains the definitions of "Topic Maps Applications (TM Applications)", establishing the criteria that such definitions must meet in order to facilitate the achievement of the Subject Location Uniqueness Objective, and to assure that topic maps can be interchanged, understood, and amalgamated predictably, regardless of their governing TM Applications, and regardless of the combinations of TM Applications that may govern the subjects represented by any single topic map graph that may result from amalgamating multiple topic maps.

5.1.1 Any participating subjects

This RM4TM does not constrain the nature or properties of subjects that can participate in topic map graphs.

5.1.2 Most constraints are imposed by TM Applications

This RM4TM imposes minimal constraints on the definitions of "Topic Maps Applications (TM Applications)," so that the definition of each TM Application establishes a context within which the nature of the topic map information being represented under its governance is well-defined.

5.1.3 Purpose of TM Application definition requirements

This RM4TM does not define any specific TM Applications, nor does it define any aspects of any specific TM Applications. Instead, it imposes constraints on the definitions of conforming TM Applications. The purpose of these constraints is to require TM Applications to be defined in sufficient detail, and with sufficient rigor, so that:

5.1.3.1 conforming implementations and conforming topic maps can be created by diverse and independent creators and creative processes,

5.1.3.2 given any conforming topic map created by any conforming implementation, the interpretation of that topic map by any other conforming implementation will be verifiably consistent with the TM Application, and

5.1.3.3 the effort and expense involved in amalgamating the knowledge represented by topic maps that conform to single and multiple TM Applications can be minimized, while the consistency of the knowledge represented by the resulting amalgamated topic maps can be maximized, without information loss, and with the greatest possible achievement of the Subject Location Uniqueness Objective by automatic means.

5.1.4 Overview of required TM Application definition components

The definition of a conforming TM Application must include all of the following:

1. A name that is different from the name of any other conforming TM Application. (See 5.2.1.)

2. A set of definitions of the properties of nodes and their value types, specifying which property values are intended to be used for purposes of deciding whether nodes have identical subjects (i.e., specifying which are SIDPs, and which are OPs). (See 5.2.2.)

3. The validity constraints on the values of the properties of nodes. (See 5.2.3.)

4. A set of situation features other than service as the *x* endpoints of C*x* arcs, and the property values that must be conferred on the nodes so situated. (The purpose of these property values is to enable arc traversals within assertions. Not all intra-assertion arc traversals are required to be enabled. See 5.2.4.)

5. A set of assertion types, the role types of each assertion type, the validation constraints on their instances, and the property values that must be conferred upon the role players of their instances. (See 5.2.5.)

6. Rules for determining whether the values of any given node’s subject identity discrimination properties (SIDPs) are consistent with each other. (See 5.2.6.)

7. A set of built-in nodes, with built-in property values, that must appear in every topic map graph that conforms to the TM Application. (See 5.2.7.)

8. The rules for merging nodes on the basis of their subject identity discrimination properties (SIDPs). (See 5.2.8.)

9. The rules for combining the built-in values of the properties of built-in nodes during merging, if the designers of the TM Application anticipate the need for such combination. (See 5.2.9.)

10. If the TM Application defines one or more interchange syntaxes, the procedures for constructing topic map graphs from instances of each syntax ("Syntax Processing Models"), and "node demander" rules that allow topic map graph nodes to be indirectly addressed by addressing their corresponding syntactic constructs. (See 5.2.10.)

5.2 Constraints on definitions of aspects of TM Applications

The following subclauses specify the detailed constraints governing each of the required aspects of the definitions of TM Applications.

5.2.1 Definition of TM Application name

The name of the TM Application must be specified. Care should be taken to select a name that is unlikely to be used as the name of any other TM Application, including other versions and/or conformance levels of an evolving or configurable TM Application. (Each version, conformance level, or other configuration must be regarded as a distinct TM Application for purposes of naming.) This name must be used as the first field of all of the property names that it defines. The name must not include the "name field separator" symbol shared by all TM Applications whose definitions conform to this RM4TM. (See 4.5.)

Non-ISO-standard TM Applications are not permitted to use names that begin with "IS", irrespective of the cases of the letters, in the first field.

Note 32: One way to minimize the risk of ambiguity that might result from coincidental use of identical names for TM Applications created by different TM Application designers is for designers to use, as their TM Application names, URIs that address the internet domain names that the designers themselves control, or that are registered names within controlled TM Application namespaces within the internet domains of such standards organizations as OASIS, the World Wide Web Consortium, IDEAlliance, or such library service organizations as the Online Computer Library Center (OCLC), the Library of Congress, etc.

5.2.2 Definition of properties and property values

All properties of nodes should be explicitly defined. All properties whose values are used to determine whether two nodes have the same subject (i.e., all SIDPs) *must* be explicitly defined.

Each property definition must specify all of the aspects described in the following subclauses:

5.2.2.1 Property name

The property definition must specify a name that is unique among the names of all the properties, assertion types, and role types defined by the TM Application. The name must not include the "name field separator" symbol (see 4.5).

5.2.2.2 Property value type

The property definition must specify the type of value of which the value must be an instance, if the property exhibits a value.

Note 33: Property value types are not constrained by this RM4TM. They can be simple and/or complex. They can be data and/or nodes.

5.2.2.3 Constraints on property values

The property definition may specify validity constraints on the value of the property. During the process of converting a well-formed topic map graph into a fully merged one, implementations of the TM Application must validate all SIDP values for conformance to all of the validity constraints defined for them. (See 6.4.)

5.2.2.4 Subject identity discrimination properties (SIDPs)

The property definition must indicate whether the property being defined is a subject identity discrimination property (SIDP).

5.2.2.5 Semantics of the property

Each property definition should include an explanation of the significance of the property and its values, including an explicit indication, where appropriate, of the significance of the condition in which no value is exhibited. If the property is a subject identity discrimination property (SIDP), such an explanation *must* be provided.

5.2.3 Definitions of validity constraints on the values of properties

If, in order to be considered valid, a property value must conform to certain constraints, the TM Application should define such constraints for each such property, wherever possible.

5.2.4 Definition of assignment of property values conferred on account of arc endpoint service other than service as the x endpoints of Cx arcs

All TM Applications are required to define subject identity discrimination properties (SIDPs) for a-nodes and c-nodes, and rules for conferring values upon them, such that all a-nodes and c-nodes will exhibit values for those properties that will support the merging of assertions in conformance with the assertion merging rules specified in 5.2.8.2.

Note 34: This RM4TM does not require TM Applications to define properties whose values reflect the internal structure of assertions comprehensively.

Note 35: See Annex C for an informative example of a set of property definitions that reflect the internal structure of assertions.

5.2.5 Definitions of assertion types

The definition of each assertion type defined by a TM Application must include all of the aspects specified in the following subclauses.

5.2.5.1 Definitions of names of assertion types

For each assertion type, a name that is unique among all the names of assertion types, role types, and properties defined by the TM Application must be specified. The names of assertion types have two fields, in the same manner as property names, with the name of the TM Application in the first field, and the name of the assertion type in the second field. The name must not include the "name field separator" symbol defined in 4.5.

5.2.5.2 Definition of the semantics of the assertion type

The semantics of each assertion type must be explained.

5.2.5.3 Definitions of role types

A set of role types must be specified, each member of which will always be represented in all instances of the assertion type in the topic map graph, regardless of whether they have role players.

This RM4TM does not prohibit multiple assertion types from incorporating the identical role type(s).

Note 36: The designs of TM Applications may be inherently more robust if all of the role types defined as components of their assertions types are regarded as unique subjects, even when they share the same names. For example, the father-daughter relationship type and the father-son relationship type may, in some cultures, be different in character, and the role of fatherhood may therefore also turn out to be different. If a TM Application defines both the father-daughter and father-son relationship types in such a way as to regard the role type of "father" as the same subject in both relationship types, then no distinction can ever be made between the two kinds of fatherhood, other than by defining a new TM Application.

Each role type definition includes all of the aspects specified in the following subclauses.

5.2.5.3.1 Definitions of names of role types

For each role type, a name which is unique among all the names of assertion types, role types, and properties defined by the TM Application must be specified. The names of role types have two fields, in the same manner as property names, with the name of the TM Application in the first field, and the name of the role type in the second field. The name must not include the "name field separator" symbol defined in 4.5.

5.2.5.3.2 Definitions of property values conferred on role players of assertion instances

If, in instances of the assertion type being defined, role players of the role being defined are required to have property values conferred upon them, the procedure required to calculate such values should be defined. It *must* be defined for subject identity discrimination properties (SIDPs).

TM Applications must not allow values to be conferred on the SIDPs of any of the role players of assertions whose assertion types are unspecified.

5.2.5.3.3 Definition of semantics of role type

The semantics of each role type must be explained.

5.2.6 Definition of consistency of the values of SIDPs of a node

The rules for detecting conditions in which the subject identity discrimination properties (SIDPs) of a node have conflicting values must be defined.

5.2.7 Definitions of built-in nodes and their built-in property values

Some of the subjects defined by a Topic Maps Application - at least enough to bootstrap at least some of its assertion types and role types into existence - must be represented by "built-in" nodes that are logically present in all topic map graphs at the moment that they begin to be constructed.

These built-in nodes and their built-in subject identity discrimination property values must be defined.

If there are any built-in assertions, the built-in property values that correspond to their arcs must be defined, and their built-in a-nodes and c-nodes must be provided with built-in values for their subject identity discrimination properties (SIDPs) such that the merging of the built-in assertions in conformance with the assertion merging rules specified in 5.2.8.2 will occur. The definitions of the properties that have built-in values in the built-in nodes defined by the TM Application must be such that, when topic map graphs governed by the TM Application are constructed, any assertions that are implicit in the built-in property values will be unambiguously recognized, so that they can be represented explicitly in the graph.

Note 37: Whenever two or more topic maps that are governed by the same TM Application are merged, all of their built-in nodes necessarily must merge.

5.2.8 Definition of merging rules

5.2.8.1 Node merging is based only on SIDP values

TM Applications must define node merging rules that determine whether any two nodes must be merged, and these rules must operate solely on the basis of the values of subject identity discrimination properties (SIDPs).

5.2.8.2 Merging rules for assertions

5.2.8.2.1 Definition of subject identity of a-nodes

In all conforming TM Applications, two assertions are merged to become a single assertion when their respective a-nodes are deemed to represent the same subject. All TM Applications are required to define merging rules that apply uniformly to all assertions, such that they will always be merged during the process of converting a well-formed topic map graph into a fully merged topic map graph under the conditions described in the following subclauses, and such that they will be automatically merged under no other conditions and on no other basis:

5.2.8.2.1.1 Both assertions specify the same assertion type.

Note 38: If neither assertion specifies its assertion type, it cannot be assumed that the lack of an assertion type itself constitutes a specific assertion type which is the same for both.

5.2.8.2.1.2 Both assertions have the same role player, or both have no role player, for each of the same role types.

5.2.8.2.2 Merging process for assertions

When two assertions are merged, the two a-nodes become a single a-node, and each pair of c-nodes that are connected to the same r-node and a-node become a single c-node. (Nodes are merged as described in 4.3.3.)

5.2.8.3 The human factor in merging

The merging rules defined by TM Applications are intended be exploited by creators of topic maps, so that the topic maps they create can incorporate other topic maps by reference, and so that when such references are resolved, the resulting merged topic map graph will be identical to the one that the creator intended.

In all cases, and regardless of their governing Application(s), when two nodes represent the same subject, they must be merged. In other words, the Subject Location Uniqueness Objective always applies. It is the responsibility of the creator of every topic map to see to it that all such mergers will occur when the topic map is processed in conformance with the rules defined by its governing TM Applications.

Topic map creators must accept responsibility for the fully merged topic map graphs represented by the interchangeable topic maps that they create, even when their interchangeable topic maps incorporate topic maps that were created by others. When interchangeable topic maps incorporate other topic maps by reference, they must also contain (or incorporate by reference) subjects and assertions that cause the merging process to yield a satisfactory result in which no two nodes have the same subject, even when, in the absence of any special arrangements made by the creator of the topic map, no governing TM Application would cause the two nodes to merge. It is the responsibility of topic map creators to make such special arrangements, by adding assertions that will cause the nodes that must be merged to have SIDP values that will be recognized as requiring their merger. (See 7.4.)

Note 39: Such special arrangements may involve indirectly addressing the nodes of the topic map graph represented by the interchangeable forms of the topic maps that are incorporated by reference, by addressing the syntactic "node demanders" of the nodes that must be merged. See 5.2.10.3.

5.2.9 Definitions of rules for merging property values when merging nodes

5.2.9.1 Merging built-in property values

The Subject Location Uniqueness Objective may demand that built-in nodes be merged, but the effect of merging their built-in values cannot be determined by the situation features of the node that results from their merger. Therefore, TM Applications must define rules for combining the built-in values of built-in nodes.

5.2.9.2 Merging conferred property values

In order to optimize the merging process, TM Applications may also define procedures for combining the conferred property values of two nodes in the conferred property values of the single node that results from merging them. All such rules must be such that the result of applying these procedures is indistinguishable from the result of recalculating the merged node’s conferred property values on the basis of its new situation.

Note 40: In any case, whenever two nodes are merged, the situations of other nodes may also be affected, necessitating recalculation of their property values, as well.

5.2.10 Definitions of interchange syntaxes

The definition of a Topic Maps Application may or may not define one or more syntaxes for the interchange of the topic maps it governs. The constraints on the definitions of such syntaxes are specified in the following subclauses.

5.2.10.1 Syntactic rigor

The syntax itself must be defined in such a way that instances of it can be validated for conformance with its syntactic rules before any attempt is made to render it as a topic map graph.

5.2.10.2 Syntax Processing Models

A "Syntax Processing Model" must be defined that specifies, in terms of the definition of each such syntax, how the information represented by instances of the syntax must be comprehensively represented as topic map graphs.

Note 41: In other words, a Syntax Processing Model specifies how to construct topic map graphs from instances of the syntax, without omitting any information represented in the instances.

5.2.10.3 Facilities for indirect node addressing via syntactic constructs

5.2.10.3.1 Node demanders

A list of syntactic constructs ("node demanders") whose instances can be unambiguously addressed within the instances of the syntax must be provided. Each such node demander must be defined as being associated with a specific node whose existence in the topic map graph that the instance represents can reasonably be regarded as being "demanded" by the existence of the demander.

The list of node demanders may or may not provide a facility for comprehensively addressing every node in the topic map graph constructed from a syntactic instance.

5.2.10.3.2 "Same subject as demanded node" assertion type

Each TM Application that defines one or more Syntax Processing Models must also define at least one assertion type of which one of the role types can be played by a node demander, that confers one or more SIDP values on the player of another of its role types such that its subject will be recognized by the merging process as being the same as the subject of the node whose existence is demanded by the node demander.

Note 42: The "node demander" facilities defined for the interchange syntaxes of TM Applications allow interchangeable topic maps to refer to each other in ways that guarantee the merging of nodes that are separately demanded by each of them.

5.2.11 Borrowed definitions

TM Applications can include, as portions of themselves, other TM Applications, by reference, but only in their entirety. The names of borrowed properties, assertion types and role types are not affected by being borrowed; each remains as defined in the definition of its TM Application of origin.

6 Constructing fully-merged topic map graphs from well-formed topic map graphs

This RM4TM is designed to allow all well-formed topic map graphs, regardless of their governing TM Application(s), to be processed in essentially the same way, in order to achieve the result of a fully-merged topic map graph. The process is designed to allow modular implementation of systems for processing topic maps that are governed by multiple TM Applications.

Conforming implementations of tools that build fully-merged topic map graphs are free to construct fully merged topic map graphs from well-formed topic map graphs in any way that, in any instance, results in a graph that is indistinguishable from the graph that would theoretically result by applying the process described in the following subclauses. The subclauses (and the paragraphs within them) appear in the order in which the steps must be performed (at least theoretically, for purposes of this RM4TM's definition of the merging process in terms of its required results).

6.1 Construct the topic map graph

The first step is to construct a well-formed topic map graph. The process of constructing well-formed topic map graphs is only partly constrained by this RM4TM.

6.1.1 Endow the graph with built-in nodes

When constructing a new topic map graph, it must first be endowed with all of the built-in nodes and arcs defined by the TM Application(s) that govern the graph.

Note 43: Built-in arcs are implicitly represented by the built-in property values that correspond to them. See 5.2.7.

6.1.2 Interpret interchangeable topic map as topic map graph

If the graph is being constructed from an instance of an interchange syntax, the Syntax Processing Model defined by the governing TM Application must be applied to the instance, with the output being added to the well-formed topic map graph that is under construction.

6.1.3 Add nodes and assertions

This RM4TM does not constrain any other aspects of the original construction of a well-formed topic map graph.

Note 44: The well-formed topic map graph can be interactively constructed, or constructed from sources that are not instances of interchange syntaxes of TM Applications, or in any other way.

Note 45: Any notation or schema for any kind of information can have a TM Application built around it, so that, in effect, it becomes a topic map interchange syntax.

6.2 Validate assertion instances for conformance to definitions

All of the assertions must be validated for conformance to the definitions of their assertion types specified by their governing TM Applications. (See 5.2.5.)

6.3 Assign values to properties of nodes

All of the nodes that appear in situations that have situation features that are defined by any of the governing TM Applications as demanding that values be conferred upon their SIDPs must be discovered, and the appropriate values must be calculated and assigned to the designated SIDPs, as specified by the definition of the TM Application.

6.4 Validate the values of the SIDPs of nodes

Each SIDP value of each node must be examined individually, to see whether it conforms to the constraints defined for it by the definition of its governing TM Application. Any values that are not of the defined type (see 5.2.2.2), or that do not conform to other constraints defined for them by the governing TM Application (see 5.2.2.3), must be detected and reported as Reportable Topic Map Processing errors.

For each node, and for each TM Application that governs it, all of the property values governed by that TM Application, including properties defined in "borrowed" TM Applications, must be examined for consistency with each other, as such consistency is defined by the governing TM Application (see 5.2.6). If there are any inconsistencies among the values of its SIDPs, they must be reported as Reportable Topic Map Processing Errors.

If any errors are reported, the conditions that required the report must be changed in such a way as to rectify the problem, and the merging process must (at least theoretically, for purposes of this RM4TM's definition of the merging process in terms of its required results) be restarted at the step described in 6.2.

6.5 Merge nodes according to the defined merging rules

The values of the subject identity discrimination properties (SIDPs) of each pair of nodes must be compared, and the merging rules defined by each of the governing TM Applications must be used to determine whether the two nodes should be merged. When a rule indicates that the nodes should be merged, they must be merged in accordance with 4.3.3.

Assertions that represent the same relationships must always be merged in accordance with 5.2.8.2.

6.6 Conditionally stop or repeat

If any nodes were merged in the steps described in 6.5, then the steps described in 6.3, 6.4, and 6.5 must be repeated. When this same sequence of steps has been repeated and no merging occurs in the step described in 6.5, the topic map graph has been fully merged, and processing must stop.

7 Conformance

7.1 Conforming TM Applications

Topic Maps Applications must not claim conformance to this RM4TM if their designs are inconsistent, in any way, with the constraints imposed by this RM4TM on the designs of conforming Topic Maps Applications.

Each TM Application must have a conforming Topic Map Application Definition (see 7.2).

7.2 Conforming TM Application definitions

Each conforming Topic Map Application Definition must include comprehensive and explicit definitions of all of the components of Topic Maps Applications, as specified by this RM4TM.

Note 46: If the design (ontology) of a TM Application permits the subjects of nodes to be conferred upon them by assertions that connect these nodes to pieces of addressable information that are regarded as their "subject indicators" (the Standard Application is an example of such a TM Application), then it seems only natural to make the components of the TM Application’s design document(s) that define the TM Application’s assertion types and role types conveniently addressable, and to make the addresses of these components the built-in values of the appropriate SIDPs of some of the built-in nodes defined by the TM Application. In this way, the topic maps governed by the TM Application can be authoritatively self-documenting with respect to their assertion types and role types.

7.3 Conforming implementations of TM Applications

The behaviors of conforming implementations must be consistent with all of the behavioral constraints imposed on them by this RM4TM and by the TM Application definitions they claim to implement.

Implementations must report Reportable Topic Map Processing Errors when they encounter assertion types, role types, or properties that are not defined by their governing TM Applications, or for which they cannot perform the property value calculations, and when they cannot apply the property value calculations or merging rules required by those definitions.

7.4 Conforming interchangeable topic maps

Conforming interchangeable topic maps conform in all respects to the syntactic and semantic constraints imposed by the definitions of the TM Applications that govern them.

When interpreted in accordance with their governing TM Applications, conforming topic maps yield topic map graphs in which all subjects are represented as nodes, in which no node is treated as having, or apparently has, more or less than a single subject, and in which the Subject Location Uniqueness Objective is honored, i.e., in which no two nodes represent the same subject.

Annex A Brief informal overview (informative)

A.1 The structure of topic spaces: topic map graphs

Every topic map defines a multidimensional "topic space" -- a space in which the only locations are topics, and in which the distances between topics are measurable in terms of the number of intervening topics which must be visited in order to get from one topic to another, and the kinds of relationships that define the path from one topic to another, if any, through the intervening topics, if any.

This RM4TM describes the abstract structure of topic spaces, which it calls "topic map graphs". It allows Topic Map Applications to be described in terms of this abstract structure. All topic maps, regardless of the diversity of their ontologies, interchange syntaxes, subject discrimination rules, implementation interfaces, etc., can be understood in terms of this common abstraction.

A.2 One subject per node; one node per subject

In all topic maps, every topic represents a single subject. In the topic space represented by a topic map, every location (in Greek, every *topos*) represents exactly one subject; this is the case in the "well-formed topic map graph" abstraction defined by this RM4TM. In a "fully merged topic map graph," the Subject Location Uniqueness Objective has been achieved; every subject has a single location. This RM4TM specifies the process whereby a fully merged topic map graph is constructed from well-formed topic map graph.

Well-formed topic map graphs consist of subgraphs, called "assertions," that represent relationships between subjects. (See Annex B for a very brief introduction to assertions.)

A.3 All subjects are represented by nodes

Even though every interchangeable topic map is a map of a topic space, there is a key difference between an interchangeable topic map and the topic map graph that it represents: in a topic map graph, every subject, in order to exist in the topic space, must be represented as a node. By contrast, in an interchangeable topic map, some subjects are not explicitly represented by syntactic constructs. Instead, these subjects are present only by virtue of the implicit semantics that are built into the syntax, as defined by the Topic Map Application that governs that syntax.

In order to eliminate ambiguity as to the contents of the topic spaces they represent, this RM4TM requires the definitions of conforming Topic Map Applications to define "Syntax Processing Models" for their topic map interchange syntaxes. A Syntax Processing Model for a topic map interchange syntax constrains the construction of topic map graphs such that all subjects that participate, implicitly or explicitly, in instances of that syntax are explicitly represented in the topic map graph by nodes.

A.4 Nodes have properties

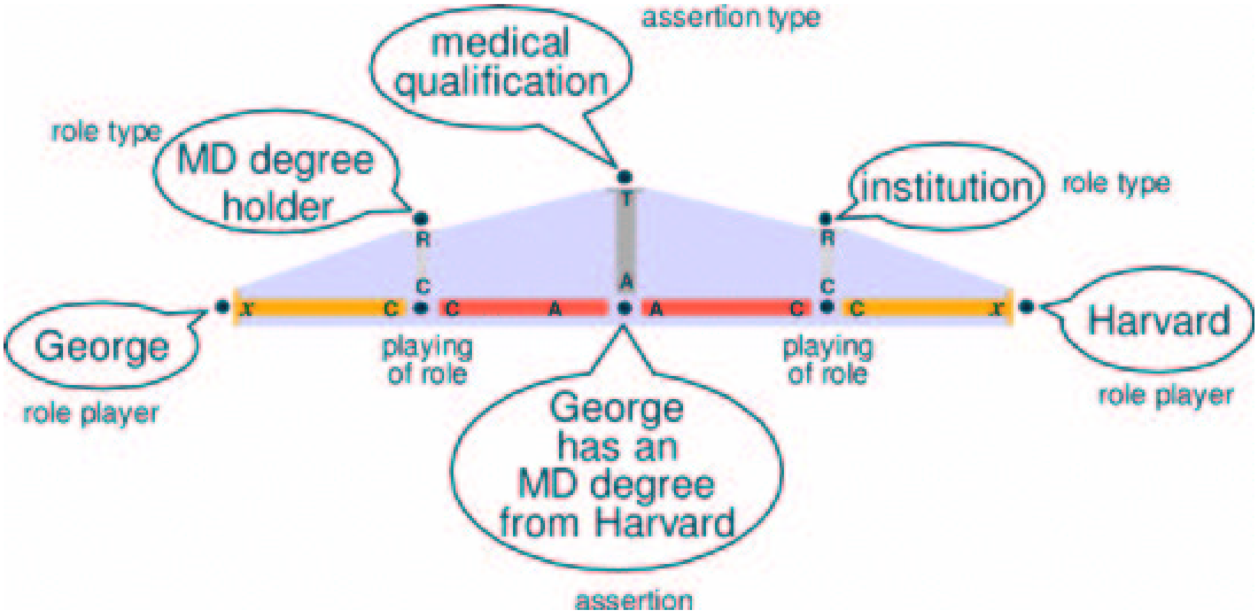
The subjects (and all other characteristics) of nodes are expressed by the values of their properties. The properties, their value types, and the rules for conferring values on the properties are all defined by TM Applications. The rules for conferring property values are expressed in terms of the relationships in which the node participates in the graph.

The values of the properties of nodes are used to determine whether they represent the same subjects. The rules for comparing property values, in order to make this determination, are defined by TM Applications. These rules are applied when a fully merged topic map graph is constructed from well-formed topic map graph. Thus, there is a sense in which the property values are determined by the graph structure, and a different sense in which the graph structure is determined by the property values; the merging process iteratively applies the two senses in sequence until no further merging occurs.

Annex B Assertion diagrams (informative)

Figure 1:

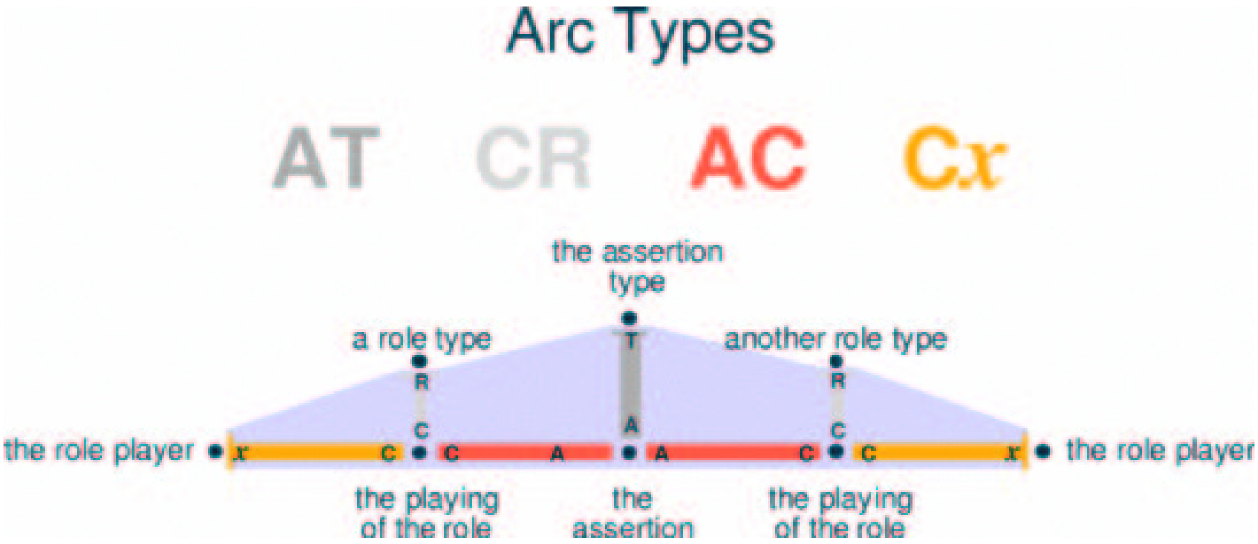
This diagram shows an instance of an assertion. Each of the eight participating subjects is shown as a black dot, and each arc is shown as a colored stripe, each end of which is labeled with an endpoint type name. For example, on the left, a C*x* arc appears with its *x* endpoint on the left end, and its C endpoint on the right end. The subject of this assertion is the idea that George (the "role player" on the left) has an MD degree from Harvard (the "role player" on the right). It is a relationship between George and Harvard in which Harvard plays the role of a degree-conferring institution (the "institution" role type), and George plays the role of the person upon whom the degree is conferred (the "MD degree holder" role type). The assertion is an instance of a "medical qualification" assertion type.



In addition to the six different subjects already discussed, there are still two more, each of which is shown as a black dot where the C endpoints of three different arcs converge; these are called "casting" nodes. The subject of the left-hand casting node is the fact that George plays the "MD degree holder" role in this particular assertion. The subject of the right-hand casting node is the fact that Harvard plays the "institution" role in this particular assertion. Every assertion asserts a relationship among its role players, which are always and only found at the *x* endpoints of C*x* arcs. Every node (here, every black dot) can play any number of roles in any number of assertions. In the very small, single-assertion topic map graph depicted here, there are only two role players (George and Harvard), and each of them plays only one role in one assertion.

Figure 2:

This diagram shows the structure of all assertions that have a specified assertion type, two role types, and two role players. The structure of a 2-role, 2-role-player assertion with an *un*specified assertion type is the same, except that the AT arc and the t-node are not present. The structure of a 2-role, 1-role-player assertion is the same except that one of the C*x* arcs, and the node at its *x* endpoint, are not present. Assertions that have more than two role types have the same structure, except that for each additional role type, there is an additional AC arc, an additional c-node, an additional CR arc, an additional r-node, and possibly an additional C*x* arc with a role player node serving as its *x* endpoint.



Annex C Sample properties that reflect assertion structure (informative)

The following list of property definitions is intended to illustrate how the internal structure of assertions could be reflected in a set of property definitions within the definition of a TM Application.

Editor's Note 3: Consider: should there be a DTD for TM Application Definitions? If so, should it be normative or informative?

Editor's Note 4: Consider: How often will TM Applications borrow the definitions provided here (or definitions like them)? If we anticipate that they are going to be borrowed, should we present these definitions as a normative TM Application? Should the SAM define them as a separate TM Application module so that they can be borrowed by TM Applications that don't want to borrow the entire SAM? If the SAM defines them (or something like them), should they appear in the RM at all, even informatively?

On the other hand, maybe the SAM won't include such a comprehensive set of properties for reflecting the structure of assertions, with full traversibility of all the arc types. In that case, does it make more sense for these definitions to appear in the RM, as they do here?

```
* Properties for which only a-nodes can exhibit
  values:

    Name: roleCastings
    Value type: node set
    Constraints on values: Only a-nodes exhibit values for this
      property, and all a-nodes must exhibit a value for this property.
      The value must be a set of c-nodes.
    SIDP or OP?: SIDP
    Semantics: The value is the node set which is the set of c-nodes
      that serve as the C endpoints of the set of AC arcs of which the
      a-node serves as the A endpoint.

    Name: assertionType
    Value type: node
    Constraints on values: Only a-nodes exhibit values for this
      property.  The value must be a t-node.
    SIDP or OP?: SIDP
    Semantics: The value is the node, if any, that serves as the T
      endpoint of the AT arc of which the a-node serves as the A
      endpoint.  If no value is exhibited, the assertion type of the
      assertion of which the a-node serves as the nexus is unspecified.

    Name: roleTypes
    Value type: node set
    Constraints on values: Only a-nodes exhibit values for this
      property.  The value must be a set of r-nodes.
    SIDP or OP?: OP
    Semantics: The value is the node set which is the set of r-nodes
      that serve as the R endpoints of the set of RC arcs of which the
      set of c-nodes serve as the C endpoints, which set of c-nodes
      serve as the C endpoints of the set of AC arcs of which the
      a-node serves as the A endpoint.

    Name: players
    Value type: node set

    Constraints on values: Only a-nodes exhibit values for this
      property.  (There are no other constraints; any nodes can be
      members of the node set.)
    SIDP or OP?: OP
    Semantics: The value is the node set which is the set of nodes that
      serve as the x endpoints of the set of Cx arcs of which the set
      of c-nodes serve as the C endpoints, which set of c-nodes serve
      as the C endpoints of the set of AC arcs of which the a-node
      serves as the A endpoint.

* Properties for which only c-nodes can exhibit
  values:

    Name: rolePlayer
```

Value type: node

Constraints on values: Only c-nodes exhibit values for this property. There are no other constraints; any node can be the value.
SIDP or OP?: SIDP
Semantics: This property may or may not exhibit a value. If it does, the value is the node, if any, that serves as the x endpoint of the Cx arc of which the c-node serves as the C endpoint.

Name: roleType
Value type: node
Constraints on values: Only c-nodes exhibit values for this property, and all c-nodes must exhibit a value for this property. The value must be an r-node.
SIDP or OP?: SIDP
Semantics: The value is the node that serves as the R endpoint of the CR arc of which the c-node serves as the C endpoint.

Name: assertion
Value type: node
Constraints on values: Only c-nodes exhibit values for this property, and all c-nodes must exhibit a value for this property. The value must be an a-node.
SIDP or OP?: SIDP
Semantics: The value is the node that serves as the A endpoint of the AC arc of which the c-node serves as the C endpoint.

* Properties for which only r-nodes can exhibit values:

Name: castingsOfRole
Value type: node set

Constraints on values: Only r-nodes exhibit values for this property. All members of the node set must be c-nodes.
SIDP or OP?: OP
Semantics: The value is the node set which is the set of c-nodes that serve as the C endpoints of the set of CR arcs of which the r-node serves as the R endpoint.

* Properties for which only t-nodes can exhibit values:

Name: assertionsOfType
Value type: node set

Constraints on values: Only t-nodes exhibit values for this property, and all t-nodes must (by definition) exhibit a value for this property. All members of the node set must be a-nodes.
SIDP or OP?: OP
Semantics: The value is the node set which is the set of a-nodes that serve as the A endpoints of the set of AT arcs of which the t-node serves as the T endpoint.

* Properties for which all kinds of nodes (including but not limited to a-nodes, c-nodes, r-nodes, and t-nodes) can exhibit values:

Name: rolePlayings
Value type: node set
Constraints on values: All nodes in the set must be c-nodes.
SIDP or OP?: OP
Semantics: The node set whose members are the c-nodes at the C endpoints of the Cx arcs whose x endpoints are the node. If no

value is exhibited, then the node plays no roles in any assertions.

* Properties for which only a-nodes, c-nodes, r-nodes, and t-nodes can exhibit values:

Name: nodeType
Value type: enumeration
Constraints on values: Value must be one of "assertion", "casting", "roleType", or "assertionType"
SIDP or OP?: SIDP
Semantics: Exhibits a corresponding value ("assertion", "casting", "roleType", or "assertionType") when the node is an a-node, c-node, r-node or t-node. When it exhibits no value, the node is neither an a-node, nor a c-node, nor an r-node, nor a t-node.