

ISO/IEC JTC 1/SC 34

Date: 2005-01-10

ISO/IEC FCD 13250-3

ISO/IEC JTC 1/SC 34/WG 3

Secretariat: SCC

Topic Maps — Part 3: XML Syntax

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Copyright notice

This ISO document is a Draft International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO's member body in the country of the requester.

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

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents

Page

Foreword.....	iv
Introduction.....	v
1 Scope.....	1
2 Normative references.....	1
3 Terms and definitions.....	2
4 Syntax.....	2
4.1 General.....	2
4.2 Common declarations.....	2
4.3 The topicMap element.....	2
4.4 The topic element.....	3
4.5 The subjectIdentity element.....	3
4.6 The baseName element.....	4
4.7 The baseNameString element.....	4
4.8 The variant element.....	4
4.9 The variantName element.....	4
4.10 The parameters element.....	4
4.11 The scope element.....	4
4.12 The instanceOf element.....	5
4.13 The occurrence element.....	5
4.14 The resourceData element.....	5
4.15 The association element.....	5
4.16 The member element.....	6
4.17 The roleSpec element.....	6
4.18 The topicRef element.....	6
4.19 The subjectIndicatorRef element.....	6
4.20 The resourceRef element.....	7
4.21 The mergeMap element.....	7
5 Deserialization.....	8
5.1 General.....	8
5.2 Common processing rules.....	8
5.2.1 General.....	8
5.2.2 Creating a URI for an element.....	8
5.2.3 Creating URIs for reference elements.....	8

5.2.4	Creation of new information items.....	9
5.2.5	Processing external references.....	9
5.2.6	Merging topic maps.....	9
5.2.7	Canonicalizing embedded XML.....	9
5.3	The topicMap element.....	9
5.4	The topic element.....	10
5.5	The subjectIdentity element.....	10
5.6	The baseName element.....	10
5.7	The baseNameString element.....	10
5.8	The variant element.....	10
5.9	The variantName element.....	11
5.10	The parameters element.....	11
5.11	The scope element.....	11
5.12	The instanceOf element.....	11
5.13	The occurrence element.....	12
5.14	The association element.....	12
5.15	The member element.....	12
5.16	The topicRef element.....	12
5.17	The subjectIndicatorRef element.....	13
5.18	The resourceRef element.....	13
5.19	The mergeMap element.....	13
6	Conformance.....	13
	Annex A (normative) A RELAX-NG schema for XTM 1.1.....	15
	Annex B (informative) The XTM 1.1 DTD.....	17
	Annex C (informative) An XML Schema schema for XTM 1.1.....	20
	Annex D (informative) Serialization.....	25
	Annex E (informative) Differences with XTM 1.0.....	26
	Bibliography.....	27

Foreword

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

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

ISO/IEC 13250-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 34, Document Description and Processing Languages.

ISO/IEC 13250 consists of the following parts, under the general title *Topic Maps*:

- *Part 1: Overview and Basic Concepts*
- *Part 2: Data Model*
- *Part 3: XML Syntax*
- *Part 4: Canonicalization*

Introduction

XTM 1.1 is a syntax for the interchange of topic maps. The syntax is not designed to be extended or modified. Other XML syntaxes that represent topic map information may well define mappings to the XTM syntax, however. Ease of human authoring was not prioritized during the design of XTM, and consequently it is not recommended to edit the syntax directly.

This part of ISO/IEC 13250 should be read in conjunction with [ISO 13250-2] since the interpretation of the XTM syntax is defined through a mapping from the syntax to the data model there defined. Informative guidance on how to serialize instances of the data model to the XTM syntax is also provided.

XTM 1.1 is a revision of the XTM 1.0 syntax defined in [ISO/IEC 13250:2003]^[1] and [XTM1.0]. A description of what has been changed can be found in Annex E.

Topic Maps — Part 3: XML Syntax

1 Scope

This part of ISO/IEC 13250 defines an XML-based interchange syntax for topic maps, which can be used to interchange instances of the data model defined in [ISO 13250-2]. It also defines a mapping from the interchange syntax to the data model. The syntax is defined with a RELAX-NG schema, and more precision is provided through the mapping to the data model, which effectively also defines the interpretation of the syntax.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE Each of the following documents has a unique identifier that is used to cite the document in the text. The unique identifier consists of the part of the reference up to the first comma.

ISO 13250-2, *Topic Maps — Data Model*, available at <<http://www.isotopicmaps.org/sam/>>

W3C XML, *Extensible Markup Language (XML) 1.0 (Second Edition)*, W3C Recommendation, 6 October 2000, available at <<http://www.w3.org/TR/2000/REC-xml-20001006>>

W3C XML-Names, *Namespaces in XML*, W3C Recommendation, 14 January 1999, available at <<http://www.w3.org/TR/1999/REC-xml-names-19990114/>>

W3C XLink, *XML Linking Language (XLink) Version 1.0*, W3C Recommendation, 27 June 2001, available at <<http://www.w3.org/TR/2001/REC-xlink-20010627/>>

W3C XBase, *XML Base (XBase) Version 1.0*, W3C Recommendation, 27 June 2001, available at <<http://www.w3.org/TR/2001/REC-xmlbase-20010627/>>

W3C XPointer, *XPointer Framework Version 1.0*, W3C Recommendation, 25 March 2003, available at <<http://www.w3.org/TR/2003/REC-xptr-framework-20030325/>>

W3C XML-InfoSet, *XML Information Set (Second Edition)*, W3C Recommendation, 4 February 2004, available at <<http://www.w3.org/TR/2004/REC-xml-infoset-20040204>>

W3C Canonical XML, *Canonical XML Version 1.0*, W3C Recommendation, 15 March 2001, available at <<http://www.w3.org/TR/2001/REC-xml-c14n-20010315>>

ISO 19757-2, *Document Schema Definition Languages (DSDL) — Part 2: Grammar-based validation — RELAX NG*, available at <<http://www.relaxng.org>>

IETF RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax*, Internet Standards Track Specification, August 1998, available at <<http://www.ietf.org/rfc/rfc2396.txt>>

IETF RFC 2732, *Format for Literal IPv6 Addresses in URLs*, Internet Standards Track Specification, December 1999, available at <<http://www.ietf.org/rfc/rfc2732.txt>>

XTM1.0, *XML Topic Maps (XTM) 1.0 Specification*, Steve Pepper, Graham Moore, TopicMaps.Org, 2001, available at <<http://www.topicmaps.org/xtm/1.0/>>

3 Terms and definitions

For the purposes of this part of ISO/IEC 13250, the following terms and definitions apply.

3.1 data model

the data model defined in [ISO 13250-2]

3.2 deserialization

the process of building an instance of an implementation's internal representation of the data model from an instance of a topic map syntax

3.3 serialization

The process of exporting topic maps from an implementation's internal representation of the data model to an instance of a topic map syntax

3.4 XTM document

an XML document that conforms to this part of ISO/IEC 13250

4 Syntax

4.1 General

An *XTM document* is an XML document that conforms to this part of ISO/IEC 13250. This clause defines syntax of XTM documents, using prose and a RELAX-NG schema in compact syntax [ISO 19757-2]. The full schema can be found in Annex A, a DTD in Annex B, and a W3C XML Schema in Annex C.

The XTM syntax represents all references as simple XLinks conformant to the XLink specification [W3C XLink]. However, since the `xlink:type` attribute is not required XML documents can be valid XTM documents without being valid according to XLink. The inclusion of the `xlink:type` attribute on all XLink elements is encouraged for compatibility with XLink.

4.2 Common declarations

The following declarations are used throughout the schema for brevity.

```
default namespace = "http://www.topicmaps.org/xtm/1.0/"
namespace xlink = "http://www.w3.org/1999/xlink"

datatypes xsd = "http://www.w3.org/2001/XMLSchema-datatypes"

start = topicMap

topic-reference = topicRef | resourceRef | subjectIndicatorRef

id = attribute id { xsd:ID }

href = attribute xlink:href { xsd:anyURI }

type = attribute xlink:type { "simple" }

any-markup = (text | element * - xtm:* { attribute * { text }?, any-markup* })*
```

4.3 The topicMap element

The `topicMap` element type is the document element of all XTM documents. It acts as a container for the topic map, and can be used to reify it, but has no significance beyond that.

The topicMap element type is declared as follows:

```
topicMap = element topicMap { id?, xmlbase?, version,
    (topic | association | mergeMap)* }
```

```
xmlbase = attribute xml:base { xsd:anyURI }
```

```
version = attribute version { "1.1" }
```

The attributes have the following meanings:

- id**
A unique identifier for the topic map within the document. Used to refer to the topic map.
- version**
The version number of the XTM syntax to which the document conforms. For XTM 1.1 documents this must be "1.1".
- xml:base**
An attribute used to override the base URI of the document inside the topicMap element, as specified in [W3C XBase].

Implementations of XTM 1.1 are required to also support XTM 1.0 documents. It is assumed that any XTM document that does not have a version attribute on the topicMap element is an XTM 1.0 document. XTM 1.0 documents are mapped to the data model in the same way as XTM 1.1 documents, but follow the XTM 1.0 DTD rather than the RELAX-NG schema of XTM 1.1.

An XTM 1.0 document is considered valid provided:

- It conforms to [XTM1.0].
- It conforms to [W3C XML-Names].
- It can be deserialized according to the procedure defined in Clause 5 without causing any errors or violating any data model constraints.

4.4 The topic element

The topic element type is used to create topics, and acts as a container and point of reference for topic information. The child elements of the topic element provide some topic characteristic assignments, as well as other properties, while association roles played by the topic are specified outside the topic element.

The topic element type is declared as follows:

```
topic = element topic { id, instanceOf*, subjectIdentity?, (baseName | occurrence)* }
```

The id attribute provides a unique identifier for the topic, which is used to refer to it.

4.5 The subjectIdentity element

The subjectIdentity element type may in its child elements contain formal declarations of the subject of the topic defined by the parent element.

The subjectIdentity element type is declared as follows:

```
subjectIdentity =
    element subjectIdentity { id?, ( resourceRef | topicRef | subjectIndicatorRef )* }
```

The id attribute is ignored during deserialization.

4.6 The baseName element

The baseName element type is used to add topic names to the topic created by the parent topic element. The child elements of the baseName element provide the property values of the topic name item.

The baseName element type is declared as follows:

```
baseName = element baseName { id?, instanceOf?, scope?, baseNameString, variant* }
```

The id attribute provides a unique identifier for the topic name, which can be used to refer to it.

4.7 The baseNameString element

The baseNameString element type is used to provide the value of the base name.

The baseNameString element type is declared as follows:

```
baseNameString = element baseNameString { id?, text }
```

The id attribute is ignored during deserialization.

4.8 The variant element

The variant element type is used to create a variant name of a base name, and may also contain other variant names with scopes that are supersets of the scope of that of the containing variant name.

The variant element type is declared as follows:

```
variant = element variant { id?, parameters, variantName?, variant* }
```

The id attribute provides a unique identifier for the variant name, which is used to refer to it.

4.9 The variantName element

The variantName element type is used to contain the element that specifies the information resource that is the actual variant name.

The variantName element type is declared as follows:

```
variantName = element variantName { id?, (resourceRef | resourceData) }
```

The id attribute is ignored during deserialization.

4.10 The parameters element

The parameters element type is used to specify the scope of a variant name, in addition to the scope it inherits from its parent topic name or variant name.

The parameters element type is declared as follows:

```
parameters = element parameters { id?, topic-reference+ }
```

The id attribute is ignored during deserialization.

4.11 The scope element

The scope element type is used throughout XTM to indicate the scope of a topic characteristic assignment.

The scope element type is declared as follows:

```
scope = element scope { id?, topic-reference+ }
```

The id attribute is ignored during deserialization.

4.12 The instanceOf element

The instanceOf element type is used throughout XTM to indicate the type of the construct represented by its parent element. The type is always a topic, indicated by the child element.

The instanceOf element type is declared as follows:

```
instanceOf = element instanceOf { id?, topic-reference }
```

The id attribute is ignored during deserialization.

4.13 The occurrence element

The occurrence element type is used to assign an occurrence to the topic defined by the parent element.

The occurrence element type is declared as follows:

```
occurrence = element occurrence { id?,  
instanceOf?, scope?, ( resourceRef | resourceData ) }
```

The id attribute is used to refer to the occurrence.

4.14 The resourceData element

The resourceData element type is used to provide an information resource in the form of content contained within the XTM document. This information resource may be either a variant name or an occurrence, and it can have a datatype.

The resourceData element type is declared as follows:

```
datatype = attribute datatype { xsd:anyURI }  
resourceData = element resourceData { id?, datatype?, any-markup }
```

The attributes have the following meanings:

id

This attribute is ignored during deserialization.

datatype

Contains a URI identifying the datatype of the resource that is represented by the resourceData element.

4.15 The association element

The association element type is used to express associations between topics. The member child elements provide the association roles of the association.

The association element type is declared as follows:

```
association = element association { id?, instanceOf?, scope?, member+ }
```

The id attribute is used to refer to the association.

4.16 The member element

The member element type is used to add one or association roles of the same type to the association created by the association parent element.

The member element type is declared as follows:

```
member = element member { id?, roleSpec?, topic-reference* }
```

The id attribute is ignored when there is more than one topic reference child, but used to refer to the member element when there is one or zero such children.

4.17 The roleSpec element

The roleSpec element type is used to specify the association role type played by the association player contained in the member parent element.

The roleSpec element type is declared as follows:

```
roleSpec = element roleSpec { id?, topic-reference }
```

The id attribute is ignored during deserialization.

4.18 The topicRef element

The topicRef element type is used throughout XTM to refer to a topic, either within the same XML document or externally. The significance of the reference depends on the context; for detailed descriptions of the different cases, see Clause 5.

The topicRef element type is declared as follows:

```
topicRef = element topicRef { id?, href, type? }
```

The attributes have the following meanings:

id

This attribute is ignored during deserialization.

xlink:href

Contains the URI reference that is the topic reference. This URI reference must conform to the requirements of XLink and have a fragment identifier which must be what [W3C XPointer] calls a shorthand pointer (formerly barename).

xlink:type

This attribute, if included, declares the topicRef element to be a simple XLink link.

4.19 The subjectIndicatorRef element

The subjectIndicatorRef element type is used throughout XTM to refer to a subject indicator. The significance of the reference depends on the context; for detailed descriptions of the different cases, see Clause 5.

The subjectIndicatorRef element type is declared as follows:

```
subjectIndicatorRef = element subjectIndicatorRef { id?, href, type? }
```

The attributes have the following meanings:

id
This attribute is ignored during deserialization.

xlink:href
Contains the URI reference of the subject indicator being referred to. The URI reference must conform to the requirements of XLink. If this URI reference contains a fragment identifier the fragment identifier must be what [W3C XPointer] calls a shorthand pointer (formerly barename).

xlink:type
This attribute, if included, declares the subjectIndicatorRef element to be a simple XLink link.

4.20 The resourceRef element

The resourceRef element type is used throughout XTM to refer to an information resource. The significance of the reference depends on the context; for detailed descriptions of the different cases, see Clause 5.

The resourceRef element type is declared as follows:

```
resourceRef = element resourceRef { id?, href, type? }
```

The attributes have the following meanings:

id
This attribute is ignored during deserialization.

xlink:href
Contains the URI reference of the information resource being referred to. The URI reference must conform to the requirements of XLink. If this URI reference contains a fragment identifier the fragment identifier must be what [W3C XPointer] calls a shorthand pointer (formerly barename).

xlink:type
This attribute, if included, declares the resourceRef element to be a simple XLink link.

4.21 The mergeMap element

The mergeMap element type is used to refer to external XTM documents that are to be merged into the topic map that contains the mergeMap element. The child elements of the mergeMap element specify topics to be added to the scopes of all topic characteristic assignments in the topic map to be merged in.

The mergeMap element type is declared as follows:

```
mergeMap = element mergeMap { id?, href, type?, topic-reference* }
```

The attributes have the following meanings:

id
This attribute is ignored during deserialization.

xlink:href
This attribute contains the URI that refers to the XTM document to be merged into the current topic map. The URI reference must conform to the requirements of XLink. If it has a fragment identifier that identifier must be what [W3C XPointer] calls a shorthand pointer (formerly barename).

xlink:type

This attribute, if included, declares the mergeMap element to be a simple XLink link.

5 Deserialization

5.1 General

The process of exporting topic maps from an implementation's internal representation of the data model to an instance of a topic map syntax is known as *serialization*. The opposite process, *deserialization* is the process of building an instance of an implementation's internal representation of the data model from an instance of a topic map syntax. This part of ISO/IEC 13250 defines how to deserialize XTM documents in Clause 5, and serialization is implicitly defined.

This clause defines how instances of the XTM syntax are deserialized into instances of the data model. The *data model* is the data model defined in [ISO 13250-2].

The input to the deserialization process is:

- A document item as defined by [W3C XML-Infoset], representing an XTM document.
- An absolute URI. This is the URI from which the XTM document was retrieved, known as the document URI. This URI must always be provided, as it is necessary in order to assign the source locators of the information items created during deserialization. If the XML document was not read from any particular URI the application is responsible for providing a URI it considers suitable.

Deserialization is done by processing each element item in the document item in document order. For each element item encountered the operations specified in the clause about that element type are performed. An input element item matches a clause in this document when the [namespace name] property is set to "http://www.topicmaps.org/xtm/1.0/", and the [local name] matches the element type name given in that section's title.

NOTE This part of ISO/IEC 13250 requires an instance of the XML Information Set as input to the deserialization process, but in most cases the actual input will be an XML document. This part of ISO/IEC 13250 does not constrain how XML Information Set instances are built from XML documents, but assumes that in most cases this will be done by simply using an XML processor.

XML processors conformant to the XML Recommendation may produce different results given the same XML document, depending on whether they are validating or non-validating, and depending on which optional features they support. Reliance on any particular behaviour in the XML processors used by recipients is strongly discouraged.

5.2 Common processing rules

5.2.1 General

This section defines common processing rules used throughout this specification. These rules are referenced from the sections they apply to.

5.2.2 Creating a URI for an element

The URI of an element is computed by concatenating the document URI, a "#" character, and the value of the [normalized value] property of the attribute item in the [attributes] property of that element item whose [local name] property is "id".

5.2.3 Creating URIs for reference elements

Creation of a URI from an element information item in the XML Infoset representing a reference is done by:

- locating the attribute information item in the element's [attributes] property whose [namespace name] property is "http://www.w3.org/1999/xlink" and whose [local name] property is "href",

- the value of the attribute information item's [normalized value] property is unescaped by replacing %HH escape sequences with the characters they represent, and the resulting character sequence is decoded from UTF-8 to a sequence of abstract Unicode characters, and
- the resulting string is turned into an absolute URI by resolving it against the URI in the [base URI] property of the element information item.

Ed. Note.

Dependency on the xtm-same-doc-refs issue here.

5.2.4 Creation of new information items

Whenever a new information item is created, those of its properties which have set values are set to the empty set, while the other properties are initialized to null.

5.2.5 Processing external references

The URI to the external information resource is resolved and the resource is parsed with an XML processor according to [W3C XML] to produce an XML Information Set according to [W3C XML-Infoset]. It is an error if the resource is not a well-formed XML document.

The XML Information Set is then deserialized into a data model instance using the procedure in Clause 5 with the document item and the URI of the information resource as input. The data model instance is then returned as the result of processing.

5.2.6 Merging topic maps

A topic map item B loaded from an external reference is merged into another topic map item A currently being deserialized by:

- Adding all topic items in B's [topics] property to A's [topics] property.
- Adding all association items in B's [associations] property to A's [associations] property.

NOTE Adding topics and associations to A is may trigger further merges, as described in [ISO 13250-2].

5.2.7 Canonicalizing embedded XML

XTM documents may contain arbitrary markup inside `resourceData` elements, and this markup is represented in the data model as a string. This section describes the string representation of embedded markup.

A string representation is produced from the embedded markup by applying the canonicalization process described in [W3C Canonical XML]. The input to the canonicalization process is the set an XPath node set (as [W3C Canonical XML] requires this). The node set is produced as described below:

- Add XPath nodes for all element, attribute, and character information items that are descendants of the `resourceData` element.
- Remove all namespace nodes attached to these element nodes where there is not at least one element or attribute node in the set with this namespace URI and namespace prefix.

The second parameter to the canonicalization process is false (meaning that comments are not included).

NOTE The output of the [W3C Canonical XML] is defined as a UTF-8-encoded octet sequence, but the output of the process defined above should be the equivalent string.

5.3 The topicMap element

The `topicMap` element causes a topic map item to be created.

If the `topicMap` element has an `id` attribute a URI is created, as defined in 5.2.2, and added to the `[source locators]` property of the topic map item.

If the `topicMap` element has an `xml:base` attribute this does not affect the data model instance being built, except insofar as it modifies the input XML Information Set.

If the `topicMap` element has a `version` attribute its value must be 1.1, indicating an XTM 1.1 document. It is an error for the value to be anything else. If the attribute is not present the document is assumed to be an XTM 1.0 document, and processed accordingly.

5.4 The `topic` element

The `topic` element causes a topic item to be created and inserted into the `[topics]` property of the topic map item.

The `id` attribute causes a string to be created, as defined in 5.2.2, and added to the `[source locators]` property of the topic item.

5.5 The `subjectIdentity` element

The `subjectIdentity` element has no direct effect on the information set, but changes the interpretation of the child elements. The child elements are processed as follows:

- For every `resourceRef` child, a string is produced according to the procedure described in 5.2.3. That string is then added to the `[subject locators]` property of the topic item created by the parent topic element.
- For every `subjectIndicatorRef` child, a string is produced according to the procedure described in 5.2.3. That string is then added to the `[subject identifiers]` property of the topic item created by the parent topic element.
- For every `topicRef` child a topic item is produced according to the rules of 5.16. That topic item is then merged with the topic item created from the parent topic element according to the rules of [ISO 13250-2], 6.2.

If the `subjectIdentity` element has an `id` attribute that attribute is ignored.

5.6 The `baseName` element

The `baseName` element causes a topic name item to be created, and added to the `[topic names]` property of the topic item created by the parent topic element.

If the `baseName` element has an `id` attribute a string is created, as defined in 5.2.2, and added to the `[source locators]` property of the topic name item.

5.7 The `baseNameString` element

The information items in the `[children]` property of the `baseNameString` element are traversed, and for each character information item the Unicode character specified by the `[character code]` property is added to the `[value]` property of the topic name item created by the parent `baseName` element.

If the `baseNameString` element has an `id` attribute that attribute is ignored.

5.8 The `variant` element

The `variant` element causes a variant item to be created and added to the `[variants]` property of the topic name item created by its `baseName` ancestor.

If the `variant` element has an `id` attribute a string is created, as defined in 5.2.2, and added to the `[source locators]` property of the variant item.

The `[scope]` property is initialized to the value of the `[scope]` property of the variant or topic name item created by the parent element (which will be either a `baseName` element or a `variant` element).

NOTE It is here assumed that the scope of the parent element has already been processed, which it will have been, so long as the document is valid.

5.9 The variantName element

The variantName element has no direct effect on the information set being produced, but changes the interpretation of its child element.

If the child element is a resourceRef element a string is produced from it, following the procedure in 5.2.3, and set as the value of the [value] property of the new variant item. The string "http://www.w3.org/2001/XMLSchema#anyURI" is set as the value of the [datatype] property.

If the child element is a resourceData element the [datatype] property will be set to "http://www.w3.org/2001/XMLSchema#string" if the datatype property is not present, and to its value if it is present. If the [datatype] property is set to "http://www.w3.org/2001/XMLSchema#any" the procedure in 5.2.7 is followed. Otherwise the information items in the [children] property of the element item are traversed, and for each character information item the Unicode character specified by the [character code] property is added to the [value] property of the variant item created by the parent variant element. In this case it is an error for the resourceData element to have child elements.

If the variantName element has an id attribute that attribute is ignored.

5.10 The parameters element

The parameters element has no direct effect on the information set being produced, but changes the interpretation of its child elements. Each topicRef, resourceRef, and subjectIndicatorRef child element is processed according to the rules for that element type to produce a topic item. These topic items are gathered into a set that is assigned as the value of the [scope] property of the variant item produced by the parent element.

If the parameters element has an id attribute that attribute is ignored.

5.11 The scope element

The scope element has no direct effect on the information set being produced, but changes the interpretation of its child elements. Each topicRef, resourceRef, and subjectIndicatorRef child element is processed according to the rules for that element type to produce a topic item. These topic items are gathered into a set that is assigned as the value of the [scope] property of the information item produced by the parent element.

If the scope element has an id attribute that attribute is ignored.

5.12 The instanceOf element

The instanceOf element has no direct effect on the information set being produced, but changes the interpretation of its child elements. The exact interpretation depends on the parent element of the instanceOf element, however.

Regardless of what parent element the instanceOf element is found in, the child element produces a topic item. For topicRef elements the procedure in 5.16 is followed; for resourceRef elements the procedure in 5.18; and for subjectIndicatorRef elements the procedure in 5.17.

If the parent element is a baseName, occurrence, or association element, the produced topic item is set as the value of the [type] property of the information item produced by the parent element.

If the parent element is a topic element a new association item is created, with two association role items in its [association roles] property, and a topic item representing the type-instance association type (described in [ISO 13250-2], 7.2) in its [type] property. If no such topic item exists already, one is created, and the PSI added to its [subject identifiers] property.

The first association role item has its [type] property set to the topic item representing the type role in the same association (see the section referenced above), while the [player] property is set to the topic produced by the child element.

The second association role item has its [type] property set to the topic item representing the instance role in the same association (see the section referenced above), while the [player] property is set to the topic produced by the parent element (that is, the current topic).

If the instanceOf element has an id attribute that attribute is ignored.

5.13 The occurrence element

The occurrence element causes an occurrence item to be created and added to the [occurrences] property of the topic item created by the parent topic element.

If the occurrence element has an id attribute a string is created, as defined in 5.2.2, and added to the [source locators] property of the occurrence item.

If the occurrence element has a resourceRef child element a string is produced from it, following the procedure in 5.2.3, and set as the value of the [value] property of the new occurrence item. The string "http://www.w3.org/2001/XMLSchema#anyURI" is set as the value of the [datatype] property.

If the occurrence has a resourceData child element the [datatype] property will be set to "http://www.w3.org/2001/XMLSchema#string" if the datatype property is not present, and to its value if it is present. If the [datatype] property is set to "http://www.w3.org/2001/XMLSchema#any" the procedure in 5.2.7 is followed to produce the value of the [value] property. Otherwise the information items in the [children] property of the element item are traversed, and for each character information item the Unicode character specified by the [character code] property is added to the [value] property of the variant item created by the parent variant element. In this case it is an error for the resourceData element to have child elements.

5.14 The association element

The association element causes an association item to be created, and added to the [association] property of the topic map item.

If the association element has an id attribute a string is created, as defined in 5.2.2, and added to the [source locators] property of the association item.

5.15 The member element

The member element does not have any direct impact on the information set being created, but affects the processing of its descendant elements.

For each topicRef, resourceRef, and subjectIndicatorRef child of the member element a topic item is produced according to the rules for its element type. An association role item is created, and this topic item is then set as the value of its [players] property. The association role item is then added to the [roles] property of the association item.

If the member element has no topicRef, subjectIndicatorRef, or resourceRef children a new association role item is created anyway. A new topic item is created and set as the value of its [player] property, and the association role item is then added to the [roles] property of the association item. A unique source locator must be generated and added to the [source locators] property of the topic item.

If the member element has a roleSpec child element, which again has a topicRef, resourceRef, or subjectIndicatorRef child element, a topic item is produced according to the rules for its element type. That topic item is then set as the value of the [type] property of each association role item produced above.

If the member element has an id and no more than one topicRef, subjectIndicatorRef, resourceRef child, a string is created, as defined in 5.2.2, and added to the [source locators] property of the association role item.

5.16 The topicRef element

The topicRef element always produces a topic item, as described below. How the topic item is used depends on the context in which the topicRef element appears, and is described in the part of this document describing the processing of the topicRef element's parent element.

From the `topicRef` element a locator item is produced according to the rules in 5.2.3. If the locator item refers to an external resource it is processed according to the rules in 5.19 as though it were a `mergeMap` element with an empty added scope.

After processing the reference, if the data model has a topic item whose `[subject identifiers]` or `[source locators]` properties contain a string equal to the one produced above that topic item is the one produced by this `topicRef` element.

If no such topic item exists, a topic item is created, and the string added to its `[source locators]` property. That topic item is then the one produced by this `topicRef` element.

If the `topicRef` element has an `id` attribute that attribute is ignored.

5.17 The `subjectIndicatorRef` element

The `subjectIndicatorRef` element produces a topic item, as described below. How the topic item is used depends on the context in which the `subjectIndicatorRef` element appears, and is described in the part of this document describing the processing of the `subjectIndicatorRef` element's parent element.

From the `subjectIndicatorRef` element a string is produced according to the rules in 5.2.3. If the information set has a topic item whose `[subject identifiers]` or `[source locators]` properties contain a string equal to the one produced above that topic item is the one produced by this `subjectIndicatorRef` element.

If no such topic item exists, a topic item is created, and the string added to its `[subject identifiers]` property. That topic item is then the one produced by this `subjectIndicatorRef` element.

If the `subjectIndicatorRef` element has an `id` attribute that attribute is ignored.

5.18 The `resourceRef` element

The `resourceRef` element produces a topic item, as described below. How the topic item is used depends on the context in which the `resourceRef` element appears, and is described in the part of this document describing the processing of the `resourceRef` element's parent element.

From the `resourceRef` element a string is produced according to the rules in 5.2.3. If the information set has a topic item whose `[subject locators]` property contains a string equal to the one produced above that topic item is the one produced by this `resourceRef` element.

If no such topic item exists, a topic item is created, and the string added to its `[subject locators]` property. That topic item is then the one produced by this `resourceRef` element.

If the `resourceRef` element has an `id` attribute that attribute is ignored.

5.19 The `mergeMap` element

An absolute URI is produced from the `mergeMap` element's `xlink:href` attribute, following the procedure in 5.2.3. For each `topicRef`, `resourceRef`, and `subjectIndicatorRef` child element of the `mergeMap` element a topic item produced is produced as described in the section for that element type. The set of topic items thus produced is known as the added scope of this reference.

If the information resource referred to by the given URI has already been processed with an equal added scope nothing further is done. If it has not a data model instance is produced as described in 5.2.5. Each topic item in the added scope is then added to the `[scope]` property of every information item in the data model instance produced above. Finally, the data model instance is merged into the current data model instance, following the procedure described in 5.2.6.

If the `mergeMap` element has an `id` attribute that attribute is ignored, as is the value of its `xlink:type` attribute.

6 Conformance

To conform to this part of ISO/IEC 13250 an XTM document must:

- Be a well-formed XML document.
- Conform to [W3C XML-Names].
- Conform to the schema in Annex A.
- Be deserializable according to the procedure defined in Clause 5 without causing any errors or violating any data model constraints.

An XTM processor conforms to this specification provided that it meets all the requirements given below.

- The XTM processor must reject any input which is not a conforming XTM document.
- The XTM processor must produce a representation that is isomorphic to the data model instance created by the procedure given in Clause 5 for all XTM documents.

Annex A (normative)

A RELAX-NG schema for XTM 1.1

```
# =====
#
# XML Topic Maps 1.1
#
# This is the normative RELAX-NG schema for the XTM 1.1 syntax, as
# defined in ISO 13250-3.
#
# =====

# --- Common declarations

default namespace = "http://www.topicmaps.org/xtm/1.0/"
namespace xlink = "http://www.w3.org/1999/xlink"
namespace xtm = "http://www.topicmaps.org/xtm/1.0/"

datatypes xsd = "http://www.w3.org/2001/XMLSchema-datatypes"

start = topicMap

topic-reference = topicRef | resourceRef | subjectIndicatorRef

id = attribute id { xsd:ID }

href = attribute xlink:href { xsd:anyURI }

type = attribute xlink:type { "simple" }

any-markup = (text | element * - xtm:* { attribute * { text }?, any-markup* })*

# --- The schema

topicMap = element topicMap { id?, xmlbase?, version,
                             (topic | association | mergeMap)* }

xmlbase = attribute xml:base { xsd:anyURI }

version = attribute version { "1.1" }

topic = element topic { id, instanceOf*, subjectIdentity?, (baseName | occurrence)* }

subjectIdentity =
  element subjectIdentity { id?, ( resourceRef | topicRef | subjectIndicatorRef)* }

baseName = element baseName { id?, instanceOf?, scope?, baseNameString, variant* }

baseNameString = element baseNameString { id?, text }

variant = element variant { id?, parameters, variantName?, variant* }

variantName = element variantName { id?, (resourceRef | resourceData) }

parameters = element parameters { id?, topic-reference+ }

scope = element scope { id?, topic-reference+ }
```

```
instanceOf = element instanceOf { id?, topic-reference }
occurrence = element occurrence { id?,
  instanceOf?, scope?, ( resourceRef | resourceData ) }
datatype = attribute datatype { xsd:anyURI }
resourceData = element resourceData { id?, datatype?, any-markup }
association = element association { id?, instanceOf?, scope?, member+ }
member = element member { id?, roleSpec?, topic-reference* }
roleSpec = element roleSpec { id?, topic-reference }
topicRef = element topicRef { id?, href, type? }
subjectIndicatorRef = element subjectIndicatorRef { id?, href, type? }
resourceRef = element resourceRef { id?, href, type? }
mergeMap = element mergeMap { id?, href, type?, topic-reference* }
# --- End of schema
```

Annex B (informative)

The XTM 1.1 DTD

```

<!-- ..... -->
<!-- XML Topic Map DTD ..... -->

<!-- XML Topic Map (XTM) DTD, Version 1.1

This is XTM 1.1, an XML interchange syntax for ISO 13250 Topic
Maps, defined by ISO 13250-3.

Use this URI to identify the default XTM namespace:

    "http://www.topicMaps.org/xtm/1.0/"

Used to identify the XLink namespace:

    "http://www.w3.org/1999/xlink"

The formal public identifier for this DTD is:

    "ISO/IEC 13250-3:2005//DTD XML Topic Maps (XTM) 1.1//EN"

-->

<!-- topicMap: Topic Map document element ..... -->

<!ELEMENT topicMap
  ( topic | association | mergeMap )*
>
<!ATTLIST topicMap
  id          ID          #IMPLIED
  version     CDATA      #FIXED '1.1'
  xmlns       CDATA      #FIXED 'http://www.topicmaps.org/xtm/1.0/'
  xmlns:xlink CDATA      #FIXED 'http://www.w3.org/1999/xlink'
  xml:base    CDATA      #IMPLIED
>

<!-- topic: Topic element ..... -->

<!ELEMENT topic
  ( instanceOf*, subjectIdentity?, ( baseName | occurrence )* )
>
<!ATTLIST topic
  id          ID          #REQUIRED
>

<!-- instanceOf: Points to a Topic representing a type ..... -->

<!ELEMENT instanceOf ( topicRef | resourceRef | subjectIndicatorRef ) >
<!ATTLIST instanceOf
  id          ID          #IMPLIED
>

<!-- subjectIdentity: Subject represented by Topic ..... -->

<!ELEMENT subjectIdentity
  ( topicRef | resourceRef | subjectIndicatorRef )*
>

```

```

<!ATTLIST subjectIdentity
  id      ID      #IMPLIED
>

<!-- topicRef: Reference to a Topic element ..... -->

<!ELEMENT topicRef EMPTY >
<!ATTLIST topicRef
  id      ID      #IMPLIED
  xlink:type  NMTOKEN #FIXED 'simple'
  xlink:href  CDATA  #REQUIRED
>

<!-- subjectIndicatorRef: Reference to a Subject Indicator ..... -->

<!ELEMENT subjectIndicatorRef EMPTY >
<!ATTLIST subjectIndicatorRef
  id      ID      #IMPLIED
  xlink:type  NMTOKEN #FIXED 'simple'
  xlink:href  CDATA  #REQUIRED
>

<!-- baseName: Base Name of a Topic ..... -->

<!ELEMENT baseName ( instanceOf?, scope?, baseNameString, variant* ) >
<!ATTLIST baseName
  id      ID      #IMPLIED
>

<!-- baseNameString: Base Name String container ..... -->

<!ELEMENT baseNameString ( #PCDATA ) >
<!ATTLIST baseNameString
  id      ID      #IMPLIED
>

<!-- variant: Alternate forms of Base Name ..... -->

<!ELEMENT variant ( parameters, variantName?, variant* ) >
<!ATTLIST variant
  id      ID      #IMPLIED
>

<!-- variantName: Container for Variant Name ..... -->

<!ELEMENT variantName ( resourceRef | resourceData ) >
<!ATTLIST variantName
  id      ID      #IMPLIED
>

<!-- parameters: Processing context for Variant ..... -->

<!ELEMENT parameters ( topicRef | resourceRef | subjectIndicatorRef )+ >
<!ATTLIST parameters
  id      ID      #IMPLIED
>

<!-- occurrence: Resources regarded as an Occurrence ..... -->

<!ELEMENT occurrence
  ( instanceOf?, scope?, ( resourceRef | resourceData ) )
>
<!ATTLIST occurrence
  id      ID      #IMPLIED
>

```



```

<!-- resourceRef: Reference to a Resource ..... -->

<!ELEMENT resourceRef EMPTY >
<!ATTLIST resourceRef
  id          ID          #IMPLIED
  xlink:type  NMTOKEN    #FIXED 'simple'
  xlink:href  CDATA      #REQUIRED
>

<!-- resourceData: Container for Resource Data ..... -->

<!ELEMENT resourceData ( #PCDATA ) >
<!ATTLIST resourceData
  id          ID          #IMPLIED
  datatype    CDATA      #IMPLIED
>

<!-- association: Topic Association ..... -->

<!ELEMENT association
  ( instanceOf?, scope?, member+ )
>
<!ATTLIST association
  id          ID          #IMPLIED
>

<!-- member: Member in Topic Association ..... -->

<!ELEMENT member
  ( roleSpec?, ( topicRef | resourceRef | subjectIndicatorRef )* )
>
<!ATTLIST member
  id          ID          #IMPLIED
>

<!-- roleSpec: Points to a Topic serving as an Association Role .. -->

<!ELEMENT roleSpec ( topicRef | resourceRef | subjectIndicatorRef ) >
<!ATTLIST roleSpec
  id          ID          #IMPLIED
>

<!-- scope: Reference to Topic(s) that comprise the Scope ..... -->

<!ELEMENT scope ( topicRef | resourceRef | subjectIndicatorRef )+ >
<!ATTLIST scope
  id          ID          #IMPLIED
>

<!-- mergeMap: Merge with another Topic Map ..... -->

<!ELEMENT mergeMap ( topicRef | resourceRef | subjectIndicatorRef )* >
<!ATTLIST mergeMap
  id          ID          #IMPLIED
  xlink:type  NMTOKEN    #FIXED 'simple'
  xlink:href  CDATA      #REQUIRED
>

<!-- end of XML Topic Map (XTM) 1.1 DTD -->

```

Annex C (informative)

An XML Schema schema for XTM 1.1

The XML Schema consists of two parts: one for the XTM namespace and another for the XLink namespace. Below is the schema for the XTM namespace.

```

<xs:schema targetNamespace="http://www.topicMaps.org/xtm/1.0/"
  elementFormDefault="qualified"
  xmlns="http://www.topicMaps.org/xtm/1.0/"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <xs:annotation>
    <xs:appinfo>
      <dc:title>Draft W3C XML Schema for XTM 1.1</dc:title>
      <dc:date>November 2004</dc:date>
      <dc:owner>ISO/IEC JTC1 SC34</dc:owner>
      <dc:contributor>Max Voskob</dc:contributor>
      <dc:contributor>Lars Marius Garshol</dc:contributor>
      <dc:contributor>Ann Wrightson</dc:contributor>
    </xs:appinfo>
  </xs:annotation>
  <xs:annotation>
    <xs:documentation>
      Minimal XLink support provided in small local file for compactness.
      Consider interop issues with XLink proper, and schema location.
    </xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlink-xlink.xsd"/>
  <!-- topicMap ..... -->
  <xs:element name="topicMap">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="topic"/>
        <xs:element ref="association"/>
        <xs:element ref="mergeMap"/>
      </xs:choice>
      <xs:attribute name="id" type="xs:ID"/>
      <xs:attribute name="version" fixed="1.1"/>
    </xs:complexType>
  </xs:element>
  <!-- topic ..... -->
  <xs:element name="topic">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="instanceOf" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="subjectIdentity" minOccurs="0"/>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element ref="baseName"/>
          <xs:element ref="occurrence"/>
        </xs:choice>
      </xs:sequence>
      <xs:attribute name="id" type="xs:ID" use="required"/>
    </xs:complexType>
  </xs:element>
  <!-- instanceOf ..... -->
  <xs:element name="instanceOf">
    <xs:complexType>
      <xs:choice>

```

```

    <xs:element ref="topicRef"/>
    <xs:element ref="resourceRef"/>
    <xs:element ref="subjectIndicatorRef"/>
  </xs:choice>
  <xs:attribute name="id" type="xs:ID"/>
</xs:complexType>
</xs:element>
<!-- subjectIdentity ..... -->
<xs:element name="subjectIdentity">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="resourceRef"/>
      <xs:element ref="topicRef"/>
      <xs:element ref="subjectIndicatorRef"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<!-- topicRef ..... -->
<xs:element name="topicRef">
  <xs:complexType>
    <xs:attribute name="id" type="xs:ID"/>
    <xs:attribute ref="xlink:type"/>
    <xs:attribute ref="xlink:href" use="required"/>
  </xs:complexType>
</xs:element>
<!-- subjectIndicatorRef ..... -->
<xs:element name="subjectIndicatorRef">
  <xs:complexType>
    <xs:attribute name="id" type="xs:ID"/>
    <xs:attribute ref="xlink:type"/>
    <xs:attribute ref="xlink:href" use="required"/>
  </xs:complexType>
</xs:element>
<!-- baseName ..... -->
<xs:element name="baseName">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="instanceOf" minOccurs="0"/>
      <xs:element ref="scope" minOccurs="0"/>
      <xs:element ref="baseNameString"/>
      <xs:element ref="variant" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>
</xs:element>
<!-- baseNameString ..... -->
<xs:element name="baseNameString">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:ID"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<!-- variant ..... -->
<xs:element name="variant">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="parameters"/>
      <xs:element ref="variantName" minOccurs="0"/>
      <xs:element ref="variant" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>
</xs:element>

```

```

<!-- variantName ..... -->
<xs:element name="variantName">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="resourceRef"/>
      <xs:element ref="resourceData"/>
    </xs:choice>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>
</xs:element>
<!-- parameters ..... -->
<xs:element name="parameters">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="topicRef"/>
      <xs:element ref="resourceRef"/>
      <xs:element ref="subjectIndicatorRef"/>
    </xs:choice>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>
</xs:element>
<!-- occurrence ..... -->
<xs:element name="occurrence">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="instanceOf" minOccurs="0"/>
      <xs:element ref="scope" minOccurs="0"/>
      <xs:choice>
        <xs:element ref="resourceRef"/>
        <xs:element ref="resourceData"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>
</xs:element>
<!-- resourceRef ..... -->
<xs:element name="resourceRef">
  <xs:complexType>
    <xs:attribute name="id" type="xs:ID"/>
    <xs:attribute ref="xlink:type"/>
    <xs:attribute ref="xlink:href" use="required"/>
  </xs:complexType>
</xs:element>
<!-- resourceData ..... -->
<xs:element name="resourceData" type="any-markup"/>
<!-- any-markup ..... -->
<xs:complexType name="any-markup" mixed="true">
  <xs:complexContent mixed="true">
    <xs:restriction base="xs:anyType">
      <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="id" type="xs:ID"/>
    </xs:restriction>
  </xs:complexContent>
  <xs:attribute name="datatype"/>
</xs:complexType>
<!-- association ..... -->
<xs:element name="association">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="instanceOf" minOccurs="0"/>
      <xs:element ref="scope" minOccurs="0"/>
      <xs:element ref="member" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>

```

```

</xs:complexType>
</xs:element>
<!-- member ..... -->
<xs:element name="member">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="roleSpec" minOccurs="0"/>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="topicRef"/>
        <xs:element ref="resourceRef"/>
        <xs:element ref="subjectIndicatorRef"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>
</xs:element>
<!-- roleSpec ..... -->
<xs:element name="roleSpec">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="topicRef"/>
      <xs:element ref="resourceRef"/>
      <xs:element ref="subjectIndicatorRef"/>
    </xs:choice>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>
</xs:element>
<!-- scope ..... -->
<xs:element name="scope">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="topicRef"/>
      <xs:element ref="resourceRef"/>
      <xs:element ref="subjectIndicatorRef"/>
    </xs:choice>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>
</xs:element>
<!-- mergeMap ..... -->
<xs:element name="mergeMap">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="topicRef"/>
      <xs:element ref="resourceRef"/>
      <xs:element ref="subjectIndicatorRef"/>
    </xs:choice>
    <xs:attribute name="id" type="xs:ID"/>
    <xs:attribute ref="xlink:type"/>
    <xs:attribute ref="xlink:href" use="required"/>
  </xs:complexType>
</xs:element>
</xs:schema>

```

Below is the schema for the XLink namespace.

```

<xs:schema elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/1999/xlink"
  targetNamespace="http://www.w3.org/1999/xlink">

  <xs:annotation>
    <xs:documentation>Minimal XLink support as required by XTM
    1.1</xs:documentation>
  </xs:annotation>

```

```
<xs:attribute name="href" type="xs:anyURI" />  
<xs:attribute name="type" type="xs:NMTOKEN" />  
</xs:schema>
```

Annex D **(informative)**

Serialization

This section provides information on how to serialize Data Model instances using the XTM syntax. The main text of this specification already provides the constraints necessary to ensure interoperability, but as serialization is not entirely straightforward, this section provides additional guidance for implementors.

Generally, a serialization implementation should guarantee that for any data model instance the XTM serialization produced by the implementation should when deserialized to a new data model instance produce one that has the same canonicalization as the original data model instance, according to [ISO 13250-4][2].

Serialization is for the most part straightforward, though care must be taken to preserve any reification relationships in the original data model instance. Implementations should also be careful not to make any assumptions about the XML implementation used to read the produced XTM document or what information resources (like DTDs) are available to the recipient of the XTM document.

Annex E (informative)

Differences with XTM 1.0

This annex describes the differences between the syntax defined in this edition of ISO 13250 and that given in [ISO/IEC 13250:2003]^[1].

The differences are:

- The `instanceOf` element is now allowed inside the `baseName` element.
- The `resourceRef` element is now allowed inside the `parameters`, `instanceOf`, and `roleSpec` elements.
- The `topicMap` element now has a new `version` attribute.
- Multiple `resourceRef` elements are now allowed within the `subjectIdentity` element.

Bibliography

- [1] *Topic Maps*, 2003, available at http://www.y12.doe.gov/sgml/sc34/document/0322_files/iso13250-2nd-ed-v2.pdf
- [2] *Topic Maps — Canonicalization*, available at <http://www.isotopicmaps.org/sam/cxtm/>