Schema documentation for DEMML_0.1.xsd

13 december 2010

Table of Contents

Namespace: "http://www.demml.org/schemas/alpha/0.1/demml"

Schemas

Main schema DEMML_0.1.xsd

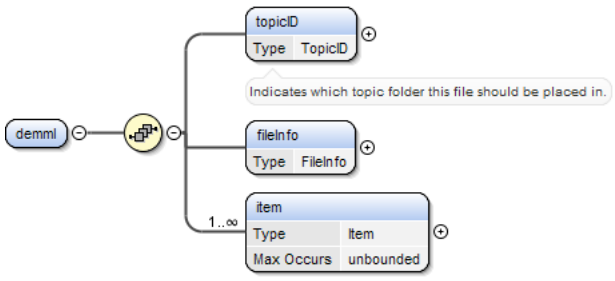
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	This is the first alpha version of the DEMML(tm) schema for content (version 0.1). This schema is far from complete and the structure may need to be modified considerably in order to incorporate all the features of the DEMML(tm). While I plan to make future versions of this schema a semi-open standard in the future, the entire schema is currently copyright (c) 2010 by Grant Sheridan Robertson. It is currently not permitted to use this schema - or any portions thereof - in any manner without written permission. However, if you write, I will probably give permision. I simply want to keep track of what is being done with DEMML(tm) so I can keep a handle on it till it is ready for release. Thank you for your cooperation. To learn more about this DEMML(tm) go to www.demml.org. To learn more about the schema itself (past, present, & future) as well as a general description of how the DEMML system is organized see www.demml.org/standard/.
Properties	attribute form default: unqualified element form default: qualified version: 0.1
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Included schema DEMMLcommon.xsd

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	This schema file contains types which are common to many of the different schemas used by the DEMML system. It has no target namespace so, when it is included in another schema document, it will take on the namespace of the including document. This is the first alpha version of the DEMML(tm) schema for content (version 0.1). This schema is far from complete and the structure may need to be modified considerably in order to incorporate all the features of the DEMML(tm). While I plan to make future versions of this schema a semi-open standard in the future, the entire schema is currently copyright (c) 2010 by Grant Sheridan Robertson. It is currently not permitted to use this schema - or any portions thereof - in any manner without written permission. However, if you write, I will probably give permision. I simply want to keep track of what is being done with DEMML(tm) so I can keep a handle on it till it is ready for release. To learn more about this DEMML(tm) go to www.demml.org. To learn more about the schema itself (past, present, & future) as well as a general description of how the DEMML system is organized see www.demml.org/standard/.
Properties	attribute form default: unqualified element form default: qualified
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

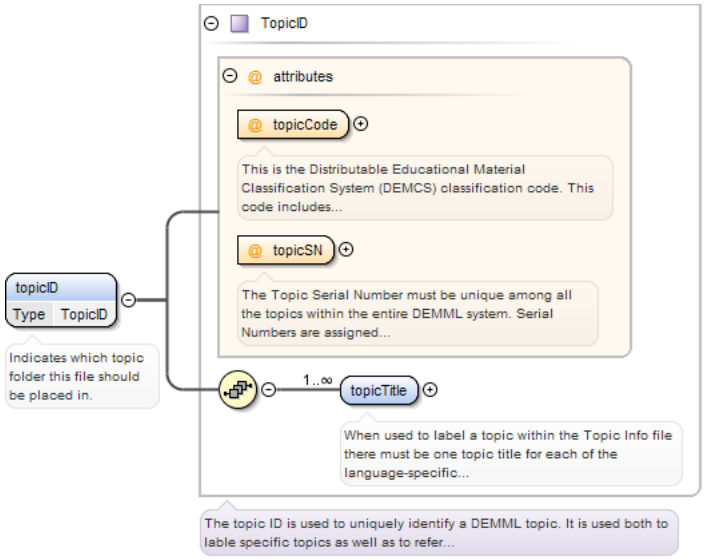
Elements

Element demml

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Properties	content: complex

Model	topicID , fileInfo , item+
Children	fileInfo, item, topicID
Instance	<demml> <topicID topicCode="" topicSN="">{1,1}</topicID> <fileInfo>{1,1}</fileInfo> <item>{1,unbounded}</item> </demml>
Source	<xs:element name="demml"> <xs:complexType> <xs:sequence> <xs:element name="topicID" type="TopicID"> <xs:annotation> <xs:documentation>Indicates which topic folder this file should be placed in.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="fileInfo" type="FileInfo"/> <xs:element name="item" type="Item" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element demml / topicID

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Indicates which topic folder this file should be placed in.
Diagram	
Type	TopicID
Properties	content: complex
Model	topicTitle+
Children	topicTitle
Instance	<topicID topicCode="" topicSN=""> <topicTitle xml:lang="">{1,unbounded}</topicTitle> </topicID>

Attributes

QName	Type	Fixed	Default	Use
topicCode	xs:token			required
	This is the Distributable Educational Material Classification System (DEMCS) classification code. This code includes the full path to the topic within the DEMCS. See http://demml.org/standard/classification/for more information. Do not include the demml:// protocol prefix in the value for this attribute.			
topicSN	SerialNumber			required
	The Topic Serial Number must be unique among all the topics within the entire DEMML system. Serial Numbers are assigned sequentially as each			

	QName	Type	Fixed	Default	Use
		topic is added the DEMCS. In case some topics need to be rearranged at some time in the future (hopefully very rarely) the serial numbers can be used by software to locate the correct topics and update their locations within individual users' databases of DEMML content.			

Source

```
<xs:element name="topicID" type="TopicID">
  <xs:annotation>
    <xs:documentation>Indicates which topic folder this file should be placed in.</
xs:documentation>
  </xs:annotation>
</xs:element>
```

Schema location: file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element TopicID / topicTitle

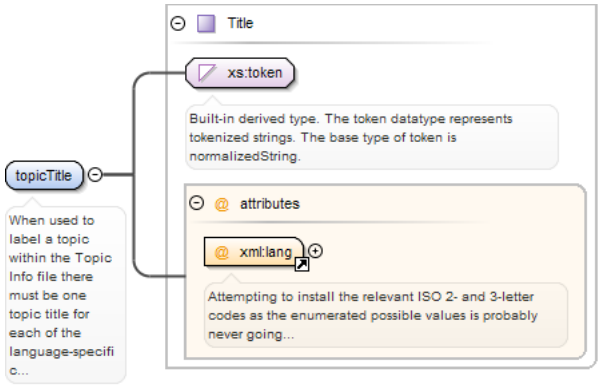
Namespace: http://www.demml.org/schemas/alpha/0.1/demml

Annotations:

When used to label a topic within the Topic Info file there must be one topic title for each of the language-specific stem folders in the topic folder. When used to indicate the topic in which a DEMML content file belongs, the topicID must include a descriptive title in the same language as the topic stem-folder in which the file belongs or resides. However, one may provide optional additional titles in other languages if desired.

Remember, the title is purely for human consumption. All software must use only the topicCode and topicSN to refer to and index the topics.

Diagram



Type: Title

Properties

content:	complex
maxOccurs:	unbounded

Attributes

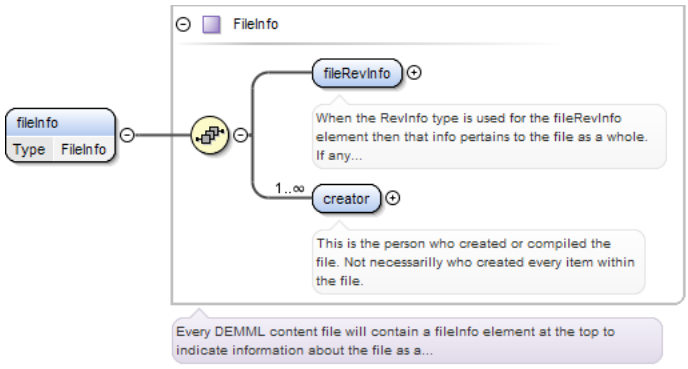
QName	Type	Fixed	Default	Use
xml:lang	union of(xs:language, restriction of xs:string)			required
	Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.			

Source

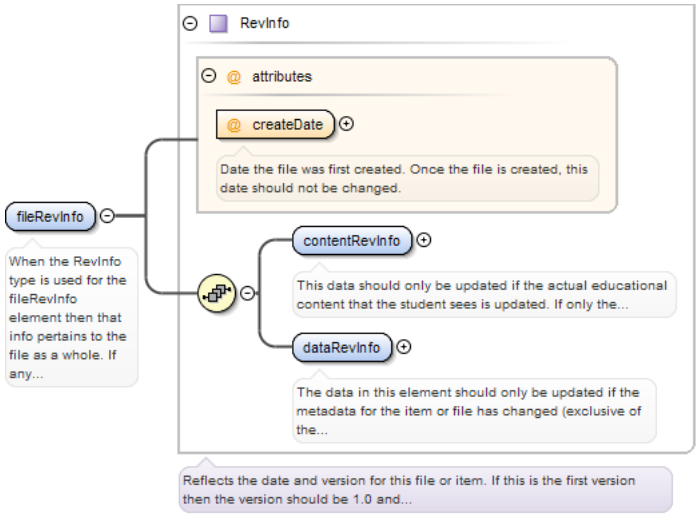
```
<xs:element name="topicTitle" type="Title" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>When used to label a topic within the Topic Info file there must
 be one topic title for each of the language-specific stem folders in the topic folder.
 When used to indicate the topic in which a DEMML content file belongs, the topicID must
 include a descriptive title in the same language as the topic stem-folder in which the
 file belongs or resides. However, one may provide optional additional titles in other
 languages if desired.</xs:documentation>
    <xs:documentation>Remember, the title is purely for human consumption. All
 software must use only the topicCode and topicSN to refer to and index the topics.</
xs:documentation>
  </xs:annotation>
</xs:element>
```

Schema location: http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element demml / fileInfo

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	FileInfo
Properties	content: complex
Model	fileRevInfo , creator+
Children	creator, fileRevInfo
Instance	<fileInfo> <fileRevInfo createDate="">{1,1}</fileRevInfo> <creator>{1,unbounded}</creator> </fileInfo>
Source	<xs:element name="fileInfo" type="FileInfo"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element FileInfo / fileRevInfo

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	When the RevInfo type is used for the fileRevInfo element then that info pertains to the file as a whole. If any content for any of the items within the file is updated then the fileRefInfo/contentRevInfo element should be updated. Similarly if any of the metadata for any of the items within the file or if the file's metadata (exclusive of the fileRefInfo/contentRevInfo element) is modified, then the fileRefInfo/dataRevInfo element should be updated.
Diagram	
Type	RevInfo
Properties	content: complex
Model	contentRevInfo , dataRevInfo
Children	contentRevInfo, dataRevInfo
Instance	<fileRevInfo createDate=""> <contentRevInfo contentModDate="" contentVersion="">{1,1}</contentRevInfo> <dataRevInfo dataModDate="" dataVersion="">{1,1}</dataRevInfo>

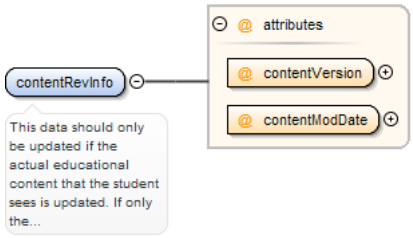
	</fileRevInfo>

Attributes	QName	Type	Fixed	Default	Use
	createDate	xs:date			required
		Date the file was first created. Once the file is created, this date should not be changed.			

```
Source
<xs:element name="fileRevInfo" type="RevInfo">
  <xs:annotation>
    <xs:documentation>When the RevInfo type is used for the fileRevInfo element then that info pertains to the file as a whole. If any content for any of the items within the file is updated then the fileRefInfo/contentRevInfo element should be updated. Similarly if any of the metadata for any of the items within the file or if the file's metadata (exclusive of the fileRefInfo/contentRevInfo element) is modified, then the fileRefInfo/dataRevInfo element should be updated.</xs:documentation>
  </xs:annotation>
</xs:element>
```

Schema location: http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element RevInfo / contentRevInfo

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	This data should only be updated if the actual educational content that the student sees is updated. If only the metadata changes then do not update this element.
Diagram	

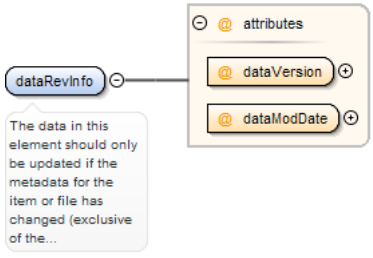
Properties	content: complex

Attributes	QName	Type	Fixed	Default	Use
	contentModDate	xs:date			required
	contentVersion	xs:token			required

```
Source
<xs:element name="contentRevInfo">
  <xs:annotation>
    <xs:documentation>This data should only be updated if the actual educational content that the student sees is updated. If only the metadata changes then do not update this element.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="contentVersion" type="xs:token" use="required"/>
    <xs:attribute name="contentModDate" type="xs:date" use="required"/>
  </xs:complexType>
</xs:element>
```

Schema location: http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element RevInfo / dataRevInfo

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	The data in this element should only be updated if the metadata for the item or file has changed (exclusive of the contentRevInfo element). If only the content has been changed then do not update this element.
Diagram	

Properties	content: complex

Attributes	QName	Type	Fixed	Default	Use
	dataModDate	xs:date			required
	dataVersion	xs:token			required

Source

```
<xs:element name="dataRevInfo">
  <xs:annotation>
    <xs:documentation>The data in this element should only be updated if the metadata
 for the item or file has changed (exclusive of the contentRevInfo element). If only the
 content has been changed then do not update this element.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="dataVersion" type="xs:token" use="required"/>
    <xs:attribute name="dataModDate" type="xs:date" use="required"/>
  </xs:complexType>
</xs:element>
```

Schema location: http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

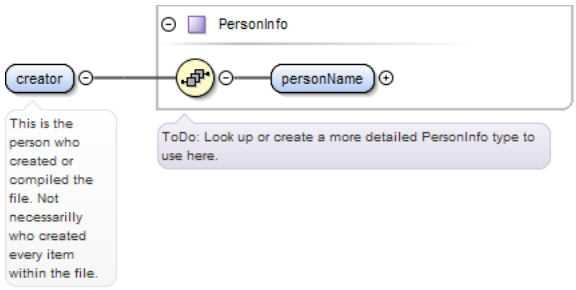
Element `FileInfo / creator`

Namespace: http://www.demml.org/schemas/alpha/0.1/demml

Annotations

```
This is the person who created or compiled the file. Not necessarilly who created every
 item within the file.
```

Diagram



Type: PersonInfo

Properties

content:	complex
maxOccurs:	unbounded

Model: personName

Children: personName

Instance

```
<creator>
  <personName>{1,1}</personName>
</creator>
```

Source

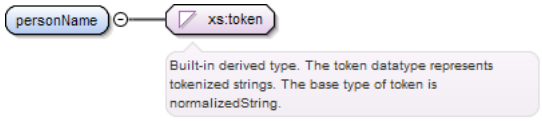
```
<xs:element name="creator" maxOccurs="unbounded" type="PersonInfo">
  <xs:annotation>
    <xs:documentation>This is the person who created or compiled the file. Not
 necessarilly who created every item within the file.</xs:documentation>
  </xs:annotation>
</xs:element>
```

Schema location: http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element `PersonInfo / personName`

Namespace: http://www.demml.org/schemas/alpha/0.1/demml

Diagram



Type: xs:token

Properties

content:	simple

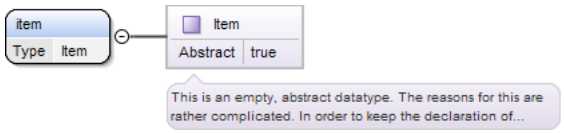
Source

```
<xs:element name="personName" type="xs:token"/>
```

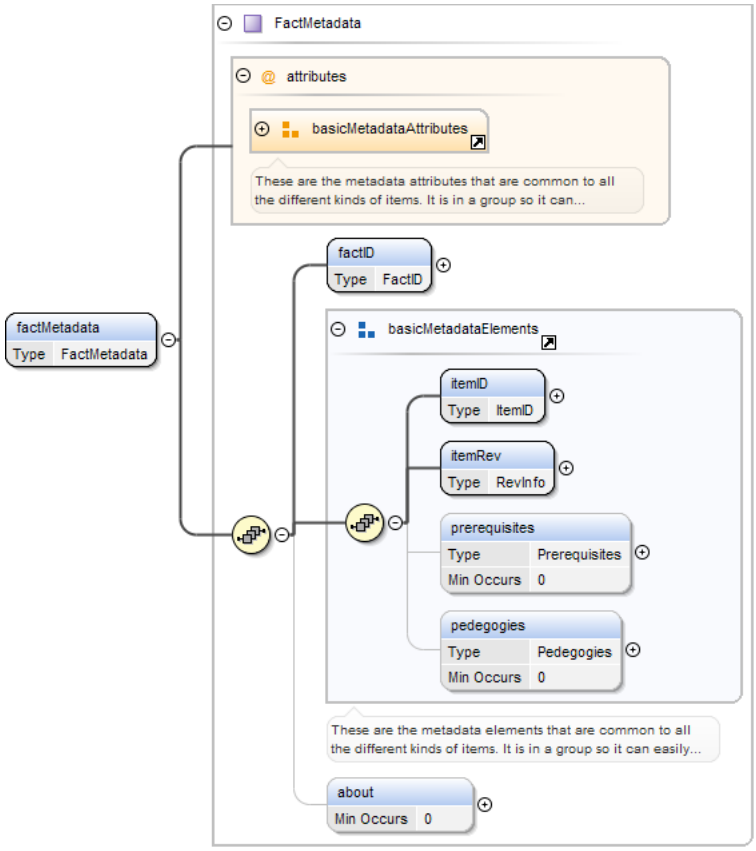
Schema location: http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element `demml / item`

Namespace: http://www.demml.org/schemas/alpha/0.1/demml

Diagram	
Type	Item
Properties	content: complex maxOccurs: unbounded
Source	`<xs:element name="item" type="Item" maxOccurs="unbounded"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

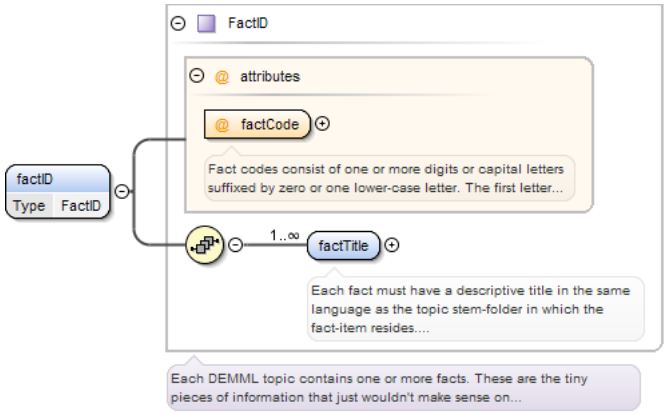
Element `FactItem / factMetadata`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	FactMetadata
Properties	content: complex
Model	factID , itemID , itemRev , prerequisites{0,1} , pedegogies{0,1} , about{0,1}
Children	about, factID, itemID, itemRev, pedegogies, prerequisites

Instance	`<factMetadata aETM="" difficulty="">`
	`  <factID factCode="">{1,1}</factID>`
	`  <about>{0,1}</about>`
	`</factMetadata>`

Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **aETM** / AETM / / / optional **difficulty** / Percent / / / optional
Source	`<xs:element name="factMetadata" type="FactMetadata"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `FactMetadata / factID`

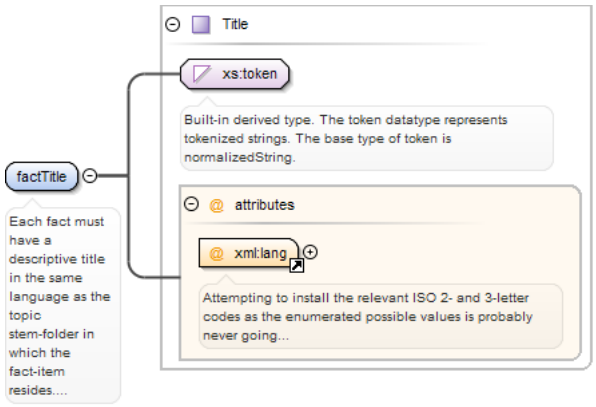
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	FactID
Properties	content: complex
Model	factTitle+
Children	factTitle

Instance	`<factID factCode="">`
	`  <factTitle xml:lang="">{1,unbounded}</factTitle>`
	`</factID>`

Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **factCode** | restriction of xs:string | | | required Fact codes consist of one or more digits or capital letters suffixed by zero or one lower-case letter. The first letter is always a capital 'F' to make them easier to recognize when written down. In order to allow additional facts to be inserted into an ordered list of preexisting facts, fact codes follow the same rules and pattern as the DEMCS coding system (see http://demml.org/standard/classification/4-assign_codes.htm). In some cases one level of sub-facts will be appropriate. In which case the sub-facts will have the code of the parent fact followed directly by one lower-case letter. For instance, N7b or 3La. If at any time one feels it necessary to create a third level of facts then serious consideration should be given to creating sub-topics instead, putting each of the primary parent facts in its own topic. Sometimes the order of facts will not matter. But for some topics where the order the facts are listed helps provide meaning to these facts then make sure the facts are numbered in an appropriate order. Fact F0 is the statement of the main point of the topic.
Source	`<xs:element name="factID" type="FactID"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `FactID / factTitle`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Each fact must have a descriptive title in the same language as the topic stem-folder in which the fact-item resides. However, one may provide optional additional titles in other languages if desired. Because each fact is contained in a separate item, the Item Title for the item containing the fact shuld be the same as the Fact Title.

Diagram	
Type	Title
Properties	content: complex maxOccurs: unbounded

Attributes

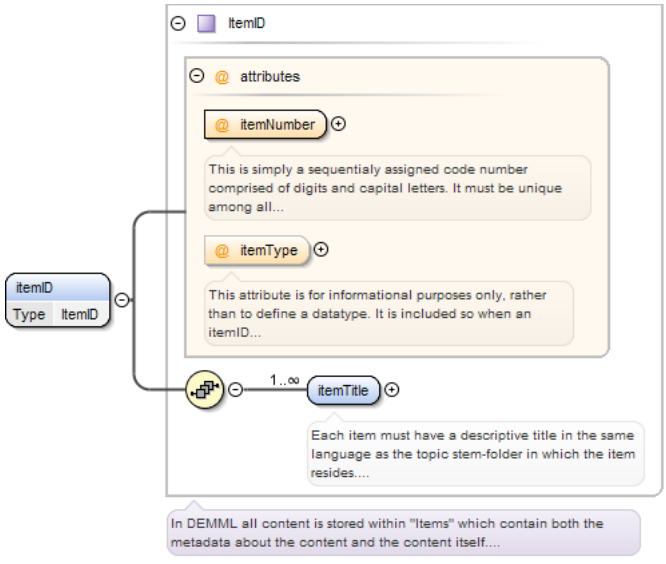
QName	Type	Fixed	Default	Use
xml:lang	union of(xs:language, restriction of xs:string)			required
	Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.			

Source

```
<xs:element name="factTitle" type="Title" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Each fact must have a descriptive title in the same language as
 the topic stem-folder in which the fact-item resides. However, one may provide optional
 additional titles in other languages if desired.</xs:documentation>
    <xs:documentation>Because each fact is contained in a separate item, the Item Title
 for the item containing the fact shuld be the same as the Fact Title.</xs:documentation>
  </xs:annotation>
</xs:element>
```

Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element basicMetadataElements / itemID

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	

Type	ItemID
Properties	content: complex
Model	itemTitle+
Children	itemTitle

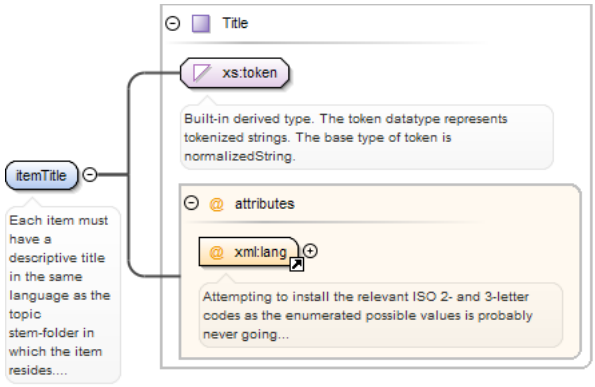
Instance	`<itemID itemNumber="" itemType="">`
	`  <itemTitle xml:lang="">{1,unbounded}</itemTitle>`
	`</itemID>`

Attributes

QName	Type	Fixed	Default	Use
itemNumber	SerialNumber			required
	This is simply a sequentialy assigned code number comprised of digits and capital letters. It must be unique among all the other items in the same topic folder.			
itemType	restriction of xs:string			optional
	This attribute is for informational purposes only, rather than to define a datatype. It is included so when an itemID is used in a reference, software or human readers can tell what type of item is being referenced without retrieving that actual item.			

Source	`<xs:element name="itemID" type="ItemID"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element ItemID / itemTitle

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Each item must have a descriptive title in the same language as the topic stem-folder in which the item resides. However, one may provide optional additional titles in other languages if desired. If the item is a question or problem then the title should not give away the answer, just in case some software chooses to display the titles of each item.
Diagram	
Type	Title
Properties	content: complex maxOccurs: unbounded

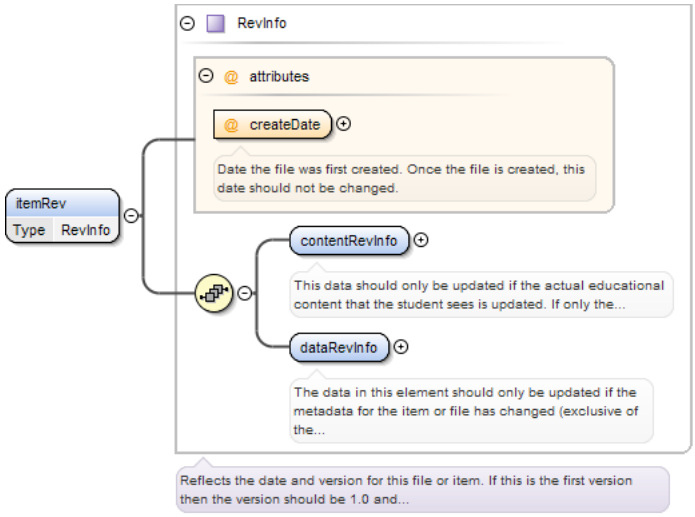
Attributes

QName	Type	Fixed	Default	Use
xml:lang	union of(xs:language, restriction of xs:string)			required
	Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.			

Source	`<xs:element name="itemTitle" type="Title" maxOccurs="unbounded">` `  <xs:annotation>`

	<xs:documentation>Each item must have a descriptive title in the same language as the topic stem-folder in which the item resides. However, one may provide optional additional titles in other languages if desired.</xs:documentation> <xs:documentation>If the item is a question or problem then the title should not give away the answer, just in case some software chooses to display the titles of each item.</xs:documentation> </xs:annotation> </xs:element>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element basicMetadataElements / itemRev

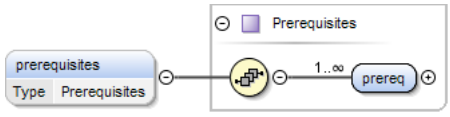
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	RevInfo
Properties	content: complex
Model	contentRevInfo , dataRevInfo
Children	contentRevInfo, dataRevInfo
Instance	<itemRev createDate=""> <contentRevInfo contentModDate="" contentVersion="">{1,1}</contentRevInfo> <dataRevInfo dataModDate="" dataVersion="">{1,1}</dataRevInfo> </itemRev>

Attributes

QName	Type	Fixed	Default	Use
createDate	xs:date			required
	Date the file was first created. Once the file is created, this date should not be changed.			

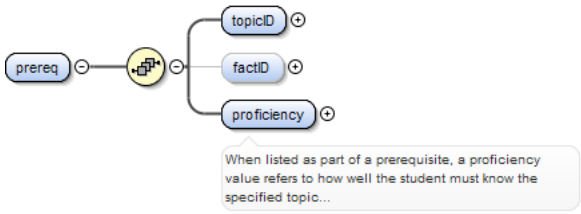
Source	<xs:element name="itemRev" type="RevInfo"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element basicMetadataElements / prerequisites

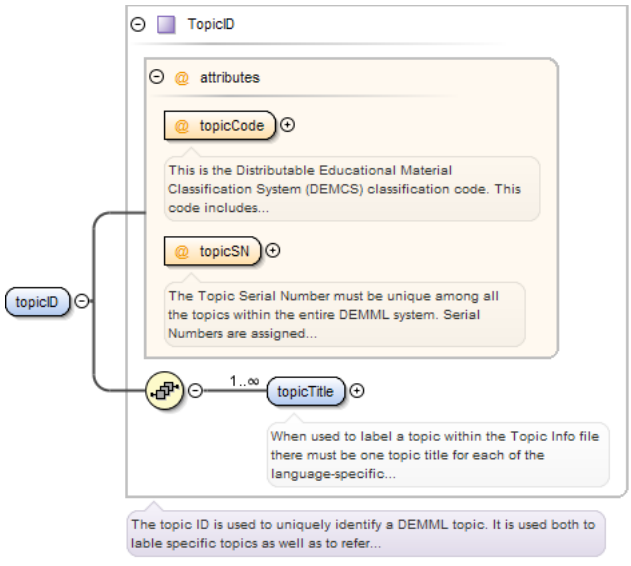
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	Prerequisites
Properties	content: complex minOccurs: 0
Model	prereq+
Children	prereq
Instance	<prerequisites> <prereq>{1,unbounded}</prereq>

	</prerequisites>
Source	<xs:element name="prerequisites" type="Prerequisites" minOccurs="0"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element Prerequisites / prereq

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Properties	content: complex maxOccurs: unbounded
Model	topicID , factID{0,1} , proficiency
Children	factID, proficiency, topicID
Instance	<prereq> <topicID topicCode="" topicSN="">{1,1}</topicID> <factID factCode="">{0,1}</factID> <proficiency aETM="" percent="">{1,1}</proficiency> </prereq>
Source	<xs:element name="prereq" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="topicID" type="TopicID"/> <xs:element name="factID" type="FactID" minOccurs="0"/> <xs:element name="proficiency" type="Proficiency"> <xs:annotation> <xs:documentation>When listed as part of a prerequisite, a proficiency value refers to how well the student must know the specified topic or fact within a topic. In addition, the estimated time to master (etm) attribute should be eiter set to "PT0S" for zero seconds or left out.</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType> </xs:element>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element prereq / Prerequisites / topicID

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	

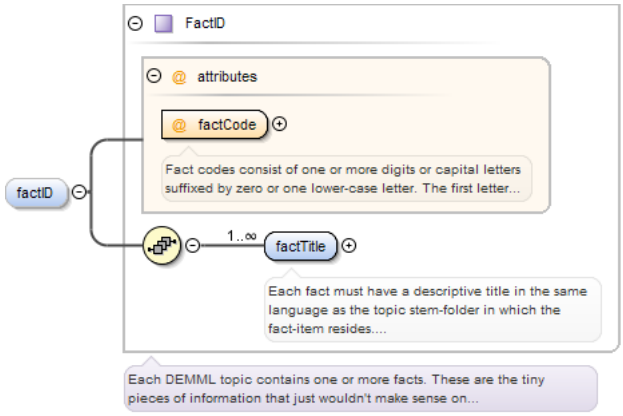
Type	TopicID
Properties	content: complex
Model	topicTitle+
Children	topicTitle
Instance	<topicID topicCode="" topicSN=""><topicTitle xml:lang="">{1,unbounded}</topicTitle></topicID>

Attributes

QName	Type	Fixed	Default	Use
topicCode	xs:token			required
	This is the Distributable Educational Material Classification System (DEMCS) classification code. This code includes the full path to the topic within the DEMCS. See http://demml.org/standard/classification/for more information. Do not include the demml:// protocol prefix in the value for this attribute.			
topicSN	SerialNumber			required
	The Topic Serial Number must be unique among all the topics within the entire DEMML system. Serial Numbers are assigned sequentially as each topic is added the DEMCS. In case some topics need to be rearranged at some time in the future (hopefully very rarely) the serial numbers can be used by software to locate the correct topics and update their locations within individual users' databases of DEMML content.			

Source	<xs:element name="topicID" type="TopicID"/>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element prereq / Prerequisites / factID

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	

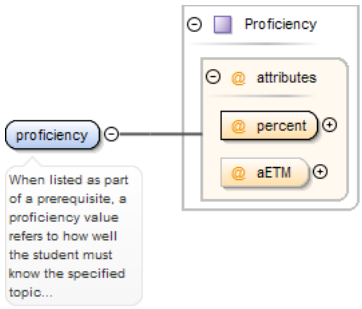
Type	FactID
Properties	content: complex minOccurs: 0
Model	factTitle+
Children	factTitle
Instance	<factID factCode=""><factTitle xml:lang="">{1,unbounded}</factTitle></factID>

Attributes

QName	Type	Fixed	Default	Use
factCode	restriction of xs:string			required
	Fact codes consist of one or more digits or capital letters suffixed by zero or one lower-case letter. The first letter is always a capital 'F' to make them easier to recognize when written down. In order to allow additional facts to be inserted into an ordered list of preexisting facts, fact codes follow the same rules and pattern as the DEMCS coding system (see http://demml.org/standard/classification/4-assign_codes.htm). In some cases one level of sub-facts will be appropriate. In which case the sub-facts will have the code of the			

	QName	Type	Fixed	Default	Use
		parent fact followed directly by one lower-case letter. For instance, N7b or 3La. If at any time one feels it necessary to create a third level of facts then serious consideration should be given to creating sub-topics instead, putting each of the primary parent facts in its own topic. Sometimes the order of facts will not matter. But for some topics where the order the facts are listed helps provide meaning to these facts then make sure the facts are numbered in an appropriate order. Fact F0 is the statement of the main point of the topic.			
Source	<xs:element name="factID" type="FactID" minOccurs="0"/>				
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd				

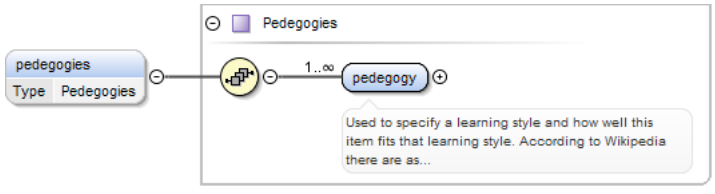
Element prereq / Prerequisites / proficiency

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	When listed as part of a prerequisite, a proficiency value refers to how well the student must know the specified topic or fact within a topic. In addition, the estimated time to master (etm) attribute should be eiter set to "PT0S" for zero seconds or left out.
Diagram	
Type	Proficiency
Properties	content: complex

Attributes	QName	Type	Fixed	Default	Use
	aETM	AETM			optional
	percent	PercentPlus			required

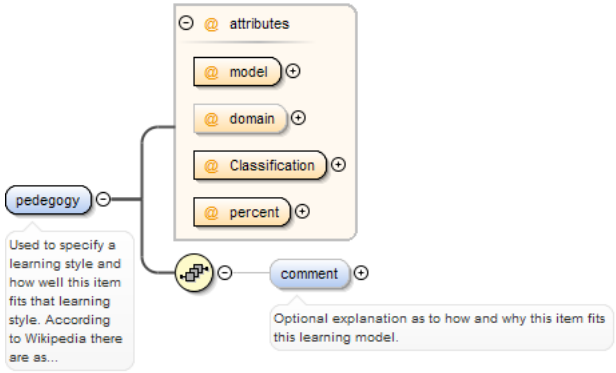
Source	<xs:element name="proficiency" type="Proficiency"> <xs:annotation> <xs:documentation>When listed as part of a prerequisite, a proficiency value refers to how well the student must know the specified topic or fact within a topic. In addition, the estimated time to master (etm) attribute should be eiter set to "PT0S" for zero seconds or left out.</xs:documentation> </xs:annotation> </xs:element>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element basicMetadataElements / pedegogies

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	Pedegogies
Properties	content: complex minOccurs: 0
Model	pedegogy+
Children	pedegogy
Instance	<pedegogies>

	<pedegogy Classification="" domain="" model="" percent="">{1,unbounded}</pedegogy> </pedegogies>
Source	<xs:element name="pedegogies" type="Pedegogies" minOccurs="0"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element Pedegogies / pedegogy

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Used to specify a learning style and how well this item fits that learning style. According to Wikipedia there are as many as 71 different "learning style" models, none of which has been proven valid in any way (Pashler, H.; McDaniel, M.; Rohrer, D.; Bjork, R. (2009). "Learning styles: Concepts and evidence". Psychological Science in the Public Interest 9: 105–119. cited in: http://en.wikipedia.org/wiki/Learning_styles#cite_ref-pashler_1-2). However, this element is made available for those who either subscribe to the Learning Style theories or simply want to provide more information about their content. At the very least, software can use these tags to ensure that students get a good mix of content based on different learning styles for any particular topic or fact.
Diagram	
Properties	content: complex maxOccurs: unbounded
Model	comment{0,1}
Children	comment
Instance	<pedegogy Classification="" domain="" model="" percent=""> <comment>{0,1}</comment> </pedegogy>

Attributes

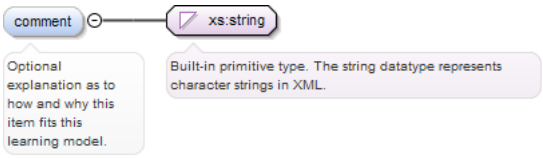
QName	Type	Fixed	Default	Use
Classification	TokenURI			required
domain	TokenURI			optional
model	TokenURI			required
percent	Percent			required

Source

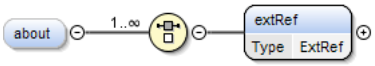
```
<xs:element name="pedegogy" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Used to specify a learning style and how well this item fits that
 learning style. According to Wikipedia there are as many as 71 different "learning
 style" models, none of which has been proven valid in any way (Pashler, H.; McDaniel,
 M.; Rohrer, D.; Bjork, R. (2009). "Learning styles: Concepts and evidence". Psychological
 Science in the Public Interest 9: 105–119. cited in: http://en.wikipedia.org/wiki/
Learning_styles#cite_ref-pashler_1-2). However, this element is made available for those
 who either subscribe to the Learning Style theories or simply want to provide more
 information about their content. At the very least, software can use these tags to
 ensure that students get a good mix of content based on different learning styles for any
 particular topic or fact.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="comment" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Optional explanation as to how and why this item fits this
 learning model.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="model" type="TokenURI" use="required"/>
    <xs:attribute name="domain" type="TokenURI"/>
    <xs:attribute name="Classification" type="TokenURI" use="required"/>
```

	<xs:attribute name="percent" type="Percent" use="required"/> </xs:complexType> </xs:element>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

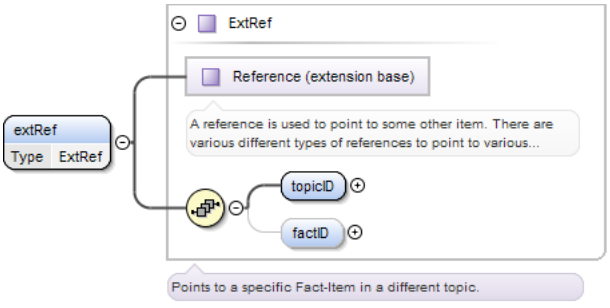
Element pedegogy / Pedegogies / comment

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Optional explanation as to how and why this item fits this learning model.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0
Source	<xs:element name="comment" type="xs:string" minOccurs="0"> <xs:annotation> <xs:documentation>Optional explanation as to how and why this item fits this learning model.</xs:documentation> </xs:annotation> </xs:element>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element FactMetadata / about

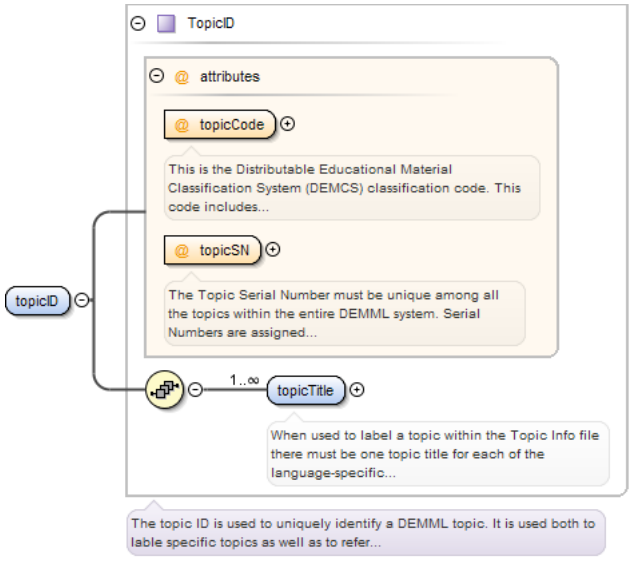
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Properties	content: complex minOccurs: 0
Model	extRef
Children	extRef
Instance	<about> <extRef>{1,1}</extRef> </about>
Source	<xs:element name="about" minOccurs="0"> <xs:complexType> <xs:choice maxOccurs="unbounded"> <xs:element name="extRef" type="ExtRef"/> </xs:choice> </xs:complexType> </xs:element>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element about / FactMetadata / extRef

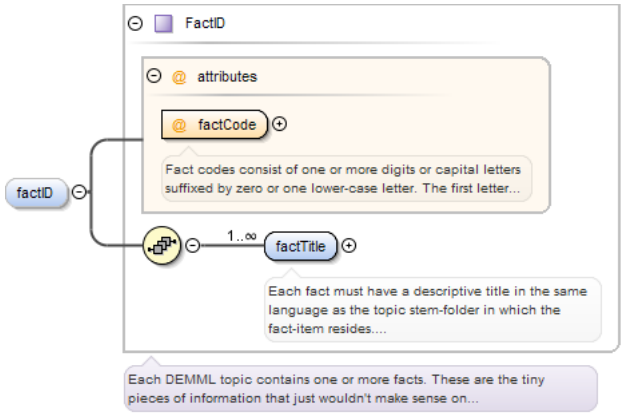
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	

Type	ExtRef
Type hierarchy	• Reference • ExtRef
Properties	content: complex
Model	topicID , factID{0,1}
Children	factID, topicID
Instance	<extRef> <topicID topicCode="" topicSN="">{1,1}</topicID> <factID factCode="">{0,1}</factID> </extRef>
Source	<xs:element name="extRef" type="ExtRef"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element ExtRef / topicID

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	TopicID
Properties	content: complex
Model	topicTitle+
Children	topicTitle
Instance	<topicID topicCode="" topicSN=""> <topicTitle xml:lang="">{1,unbounded}</topicTitle> </topicID>
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **topicCode** | xs:token | | | required This is the Distributable Educational Material Classification System (DEMCS) classification code. This code includes the full path to the topic within the DEMCS. See http://demml.org/standard/classification/for more information. Do not include the demml:// protocol prefix in the value for this attribute. **topicSN** | SerialNumber | | | required The Topic Serial Number must be unique among all the topics within the entire DEMML system. Serial Numbers are assigned sequentially as each topic is added the DEMCS. In case some topics need to be rearranged at some time in the future (hopefully very rarely) the serial numbers can be used by software to locate the correct topics and update their locations within individual users' databases of DEMML content.
Source	<xs:element name="topicID" type="TopicID"/>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

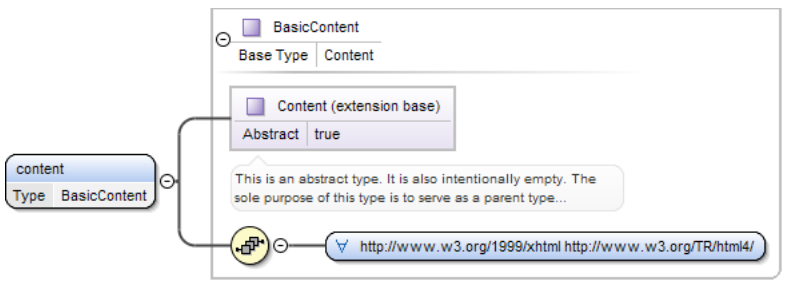
Element `ExtRef / factID`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	FactID
Properties	content: `complex` minOccurs: `0`
Model	factTitle+
Children	factTitle

`Instance`	`<factID factCode="">`
	`  <factTitle xml:lang="">{1,unbounded}</factTitle>`
	`</factID>`

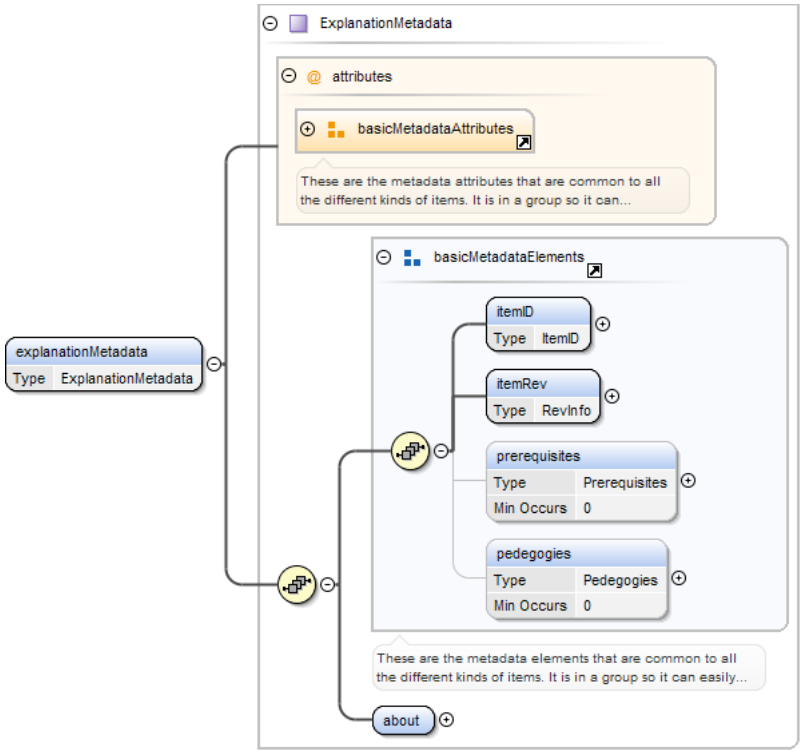
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **factCode** | restriction of xs:string | | | required `Fact codes consist of one or more digits or capital letters suffixed by zero or one lower-case letter. The first letter is always a capital 'F' to make them easier to recognize when written down. In order to allow additional facts to be inserted into an ordered list of preexisting facts, fact codes follow the same rules and pattern as the DEMCS coding system (see http://demml.org/standard/classification/4-assign_codes.htm). In some cases one level of sub-facts will be appropriate. In which case the sub-facts will have the code of the parent fact followed directly by one lower-case letter. For instance, N7b or 3La. If at any time one feels it necessary to create a third level of facts then serious consideration should be given to creating sub-topics instead, putting each of the primary parent facts in its own topic. Sometimes the order of facts will not matter. But for some topics where the order the facts are listed helps provide meaning to these facts then make sure the facts are numbered in an appropriate order.` `Fact F0 is the statement of the main point of the topic.`
`Source`	`<xs:element name="factID" type="FactID" minOccurs="0"/>`
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element `FactItem / content`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	BasicContent

Type hierarchy	• Content • BasicContent
Properties	content: complex
Model	ANY element from namespace(s) 'http://www.w3.org/1999/xhtml http://www.w3.org/TR/html4/'
Source	`<xs:element name="content" type="BasicContent"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `ExplanationItem / explanationMetadata`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	ExplanationMetadata
Properties	content: complex
Model	itemID , itemRev , prerequisites{0,1} , pedegogies{0,1} , about
Children	about, itemID, itemRev, pedegogies, prerequisites

Instance	`<explanationMetadata aETM="" difficulty="">`
	`  <itemID itemNumber="" itemType="">{1,1}</itemID>`
	`  <itemRev createDate="">{1,1}</itemRev>`
	`  <prerequisites>{0,1}</prerequisites>`
	`  <pedegogies>{0,1}</pedegogies>`
	`  <about>{1,1}</about>`
	`</explanationMetadata>`

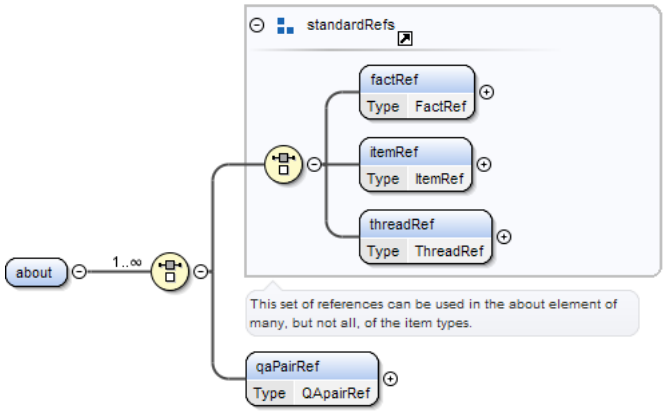
Attributes

QName	Type	Fixed	Default	Use
aETM	AETM			optional
difficulty	Percent			optional

Source	`<xs:element name="explanationMetadata" type="ExplanationMetadata"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `ExplanationMetadata / about`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml

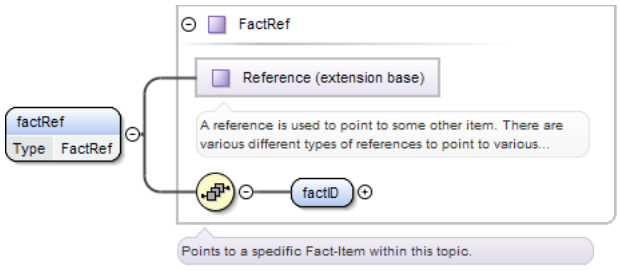
Diagram	
Properties	content: `complex`
Model	factRef | itemRef | threadRef | qaPairRef
Children	factRef, itemRef, qaPairRef, threadRef

`Instance`	`<about>`
	`  <factRef>{1,1}</factRef>`
	`  <itemRef>{1,1}</itemRef>`
	`  <threadRef>{1,1}</threadRef>`
	`  <qaPairRef>{1,1}</qaPairRef>`
	`</about>`

`Source`	`<xs:element name="about">`
	`  <xs:complexType>`
	`  <xs:choice maxOccurs="unbounded">`
	`  <xs:group ref="standardRefs"/>`
	`  <xs:element name="qaPairRef" type="QApairRef"/>`
	`  </xs:choice>`
	`  </xs:complexType>`
	`</xs:element>`

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `standardRefs / factRef`

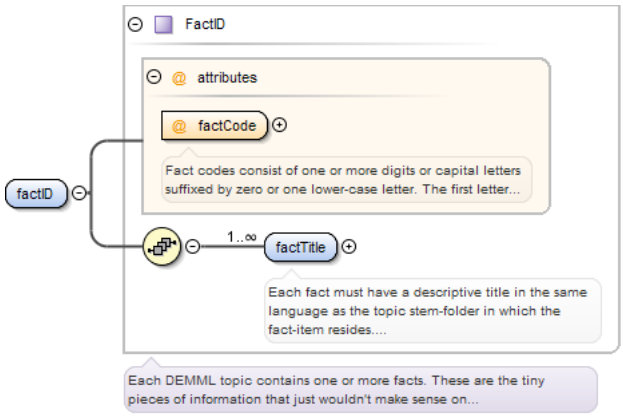
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	FactRef
Type hierarchy	• Reference • FactRef
Properties	content: `complex`
Model	factID
Children	factID

`Instance`	`<factRef>`
	`  <factID factCode="">{1,1}</factID>`
	`</factRef>`

`Source`	`<xs:element name="factRef" type="FactRef"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `FactRef / factID`

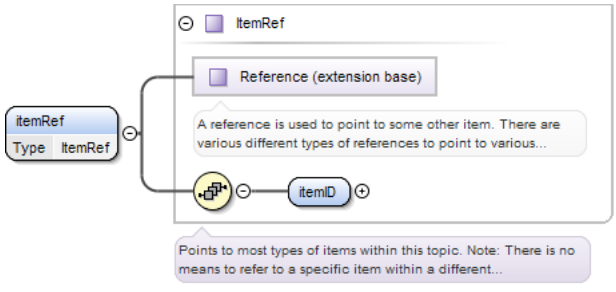
Namespace	http://www.demml.org/schemas/alpha/0.1/demml

Diagram	
Type	FactID
Properties	content: `complex`
Model	factTitle+
Children	factTitle

`Instance`	`<factID factCode="">`
	`  <factTitle xml:lang="">{1,unbounded}</factTitle>`
	`</factID>`

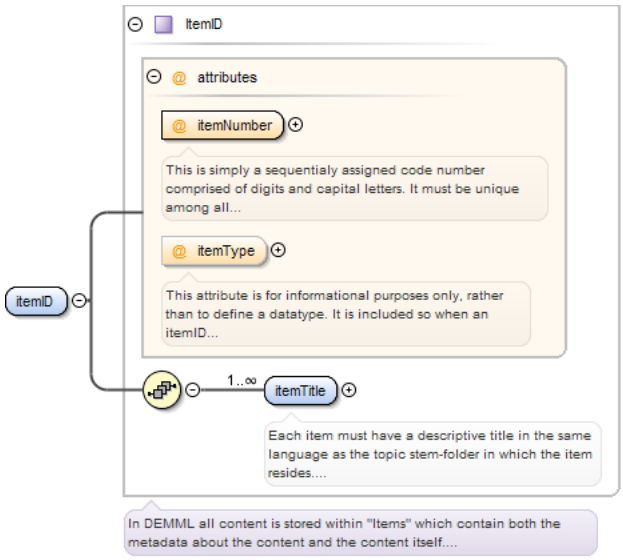
Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **factCode** / restriction of xs:string / / / required `Fact codes consist of one or more digits or capital letters suffixed by zero or one lower-case letter. The first letter is always a capital 'F' to make them easier to recognize when written down. In order to allow additional facts to be inserted into an ordered list of preexisting facts, fact codes follow the same rules and pattern as the DEMCS coding system (see http://demml.org/standard/classification/4-assign_codes.htm). In some cases one level of sub-facts will be appropriate. In which case the sub-facts will have the code of the parent fact followed directly by one lower-case letter. For instance, N7b or 3La. If at any time one feels it necessary to create a third level of facts then serious consideration should be given to creating sub-topics instead, putting each of the primary parent facts in its own topic. Sometimes the order of facts will not matter. But for some topics where the order the facts are listed helps provide meaning to these facts then make sure the facts are numbered in an appropriate order.` `Fact F0 is the statement of the main point of the topic.`
`Source`	`<xs:element name="factID" type="FactID"/>`
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element `standardRefs / itemRef`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	ItemRef
Type hierarchy	• Reference • ItemRef
Properties	content: `complex`
Model	itemID

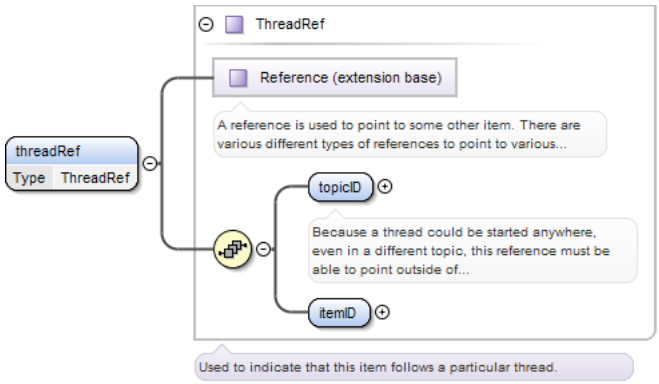
Children	itemID
Instance	<itemRef> <itemID itemNumber="" itemType="">{1,1}</itemID> </itemRef>
Source	<xs:element name="itemRef" type="ItemRef"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element ItemRef / itemID

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	ItemID
Properties	content: complex
Model	itemTitle+
Children	itemTitle
Instance	<itemID itemNumber="" itemType=""> <itemTitle xml:lang="">{1,unbounded}</itemTitle> </itemID>
Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **itemNumber** / SerialNumber / / / required This is simply a sequentialy assigned code number comprised of digits and capital letters. It must be unique among all the other items in the same topic folder. **itemType** / restriction of xs:string / / / optional This attribute is for informational purposes only, rather than to define a datatype. It is included so when an itemID is used in a reference, software or human readers can tell what type of item is being referenced without retrieving that actual item.
Source	<xs:element name="itemID" type="ItemID"/>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element standardRefs / threadRef

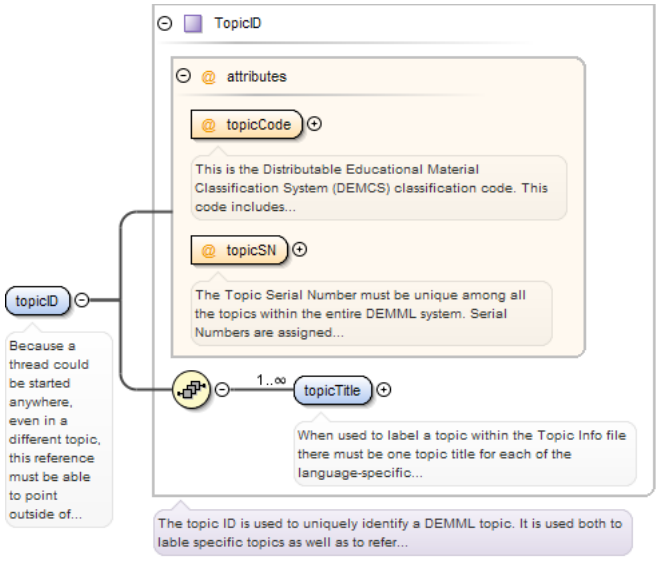
Namespace	http://www.demml.org/schemas/alpha/0.1/demml

Diagram	
Type	ThreadRef
Type hierarchy	• Reference • ThreadRef
Properties	content: complex
Model	topicID , itemID
Children	itemID, topicID

Instance	`<threadRef>`
	`  <topicID topicCode="" topicSN="">{1,1}</topicID>`
	`  <itemID itemNumber="" itemType="">{1,1}</itemID>`
	`</threadRef>`

Source	`<xs:element name="threadRef" type="ThreadRef"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

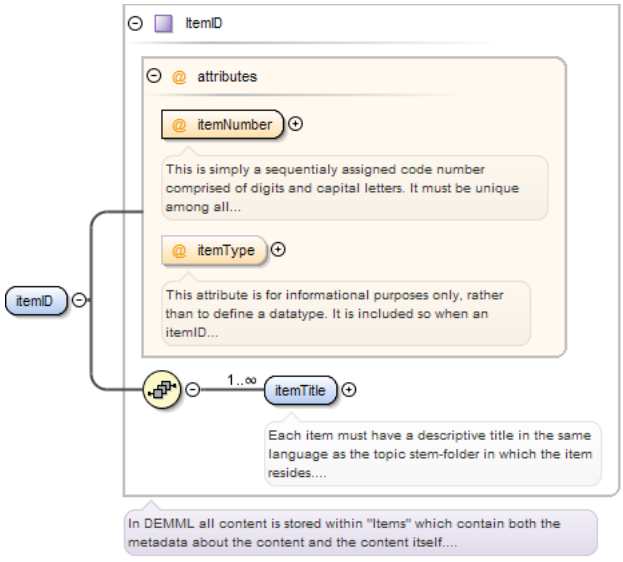
Element ThreadRef / topicID

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Because a thread could be started anywhere, even in a different topic, this reference must be able to point outside of the topic where it is used.
Diagram	
Type	TopicID
Properties	content: complex
Model	topicTitle+
Children	topicTitle

Instance	`<topicID topicCode="" topicSN="">`
	`  <topicTitle xml:lang="">{1,unbounded}</topicTitle>`
	`</topicID>`

Attributes	**QName**	**Type**	**Fixed**	**Default**	**Use**
	topicCode	xs:token			required
		This is the Distributable Educational Material Classification System (DEMCS) classification code. This code includes the full path to the topic within the DEMCS. See http://demml.org/standard/classification/for more information. Do not include the demml:// protocol prefix in the value for this attribute.			
	topicSN	SerialNumber			required
		The Topic Serial Number must be unique among all the topics within the entire DEMML system. Serial Numbers are assigned sequentially as each topic is added the DEMCS. In case some topics need to be rearranged at some time in the future (hopefully very rarely) the serial numbers can be used by software to locate the correct topics and update their locations within individual users' databases of DEMML content.			
Source	<xs:element name="topicID" type="TopicID"><xs:annotation><xs:documentation>Because a thread could be started anywhere, even in a different topic, this reference must be able to point outside of the topic where it is used.</xs:documentation></xs:annotation></xs:element>				
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd				

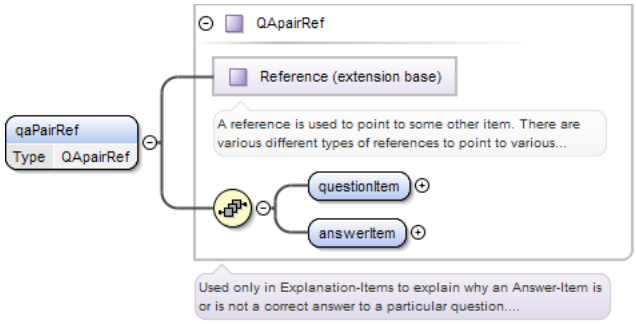
Element ThreadRef / itemID

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	ItemID
Properties	content: complex
Model	itemTitle+
Children	itemTitle
Instance	<itemID itemNumber="" itemType=""><itemTitle xml:lang="">{1,unbounded}</itemTitle></itemID>

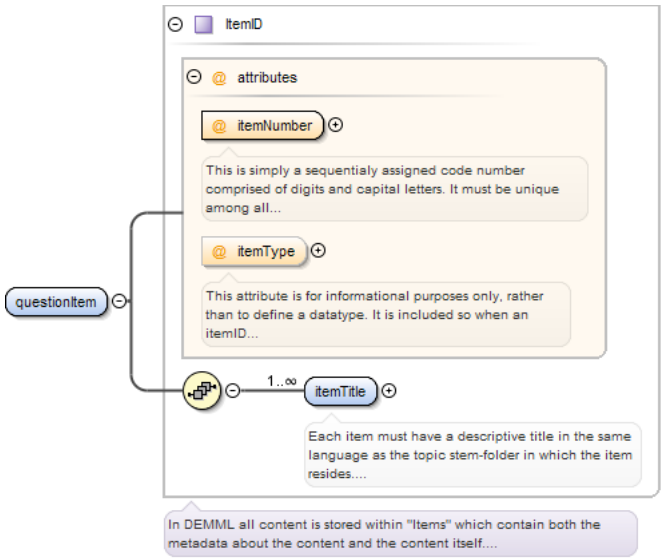
Attributes	**QName**	**Type**	**Fixed**	**Default**	**Use**
	itemNumber	SerialNumber			required
		This is simply a sequentialy assigned code number comprised of digits and capital letters. It must be unique among all the other items in the same topic folder.			
	itemType	restriction of xs:string			optional
		This attribute is for informational purposes only, rather than to define a datatype. It is included so when an itemID is used in a			

	QName	Type	Fixed	Default	Use
		reference, software or human readers can tell what type of item is being referenced without retrieving that actual item.			
Source	<xs:element name="itemID" type="ItemID"/>				
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd				

Element about / ExplanationMetadata / qaPairRef

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	QApairRef
Type hierarchy	• Reference • QApairRef
Properties	content: complex
Model	questionItem , answerItem
Children	answerItem, questionItem
Instance	<qaPairRef> <questionItem itemNumber="" itemType="">{1,1}</questionItem> <answerItem itemNumber="" itemType="">{1,1}</answerItem> </qaPairRef>
Source	<xs:element name="qaPairRef" type="QApairRef"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element QApairRef / questionItem

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	ItemID
Properties	content: complex

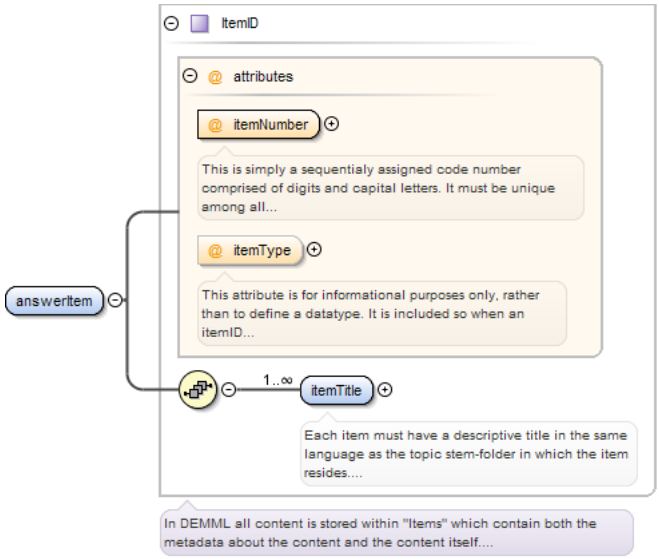
Model	itemTitle+
Children	itemTitle
Instance	<questionItem itemNumber="" itemType=""> <itemTitle xml:lang="">{1,unbounded}</itemTitle> </questionItem>

Attributes

QName	Type	Fixed	Default	Use
itemNumber	SerialNumber			required
	This is simply a sequentialy assigned code number comprised of digits and capital letters. It must be unique among all the other items in the same topic folder.			
itemType	restriction of xs:string			optional
	This attribute is for informational purposes only, rather than to define a datatype. It is included so when an itemID is used in a reference, software or human readers can tell what type of item is being referenced without retrieving that actual item.			

Source	<xs:element name="questionItem" type="ItemID"/>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element QApairRef / answerItem

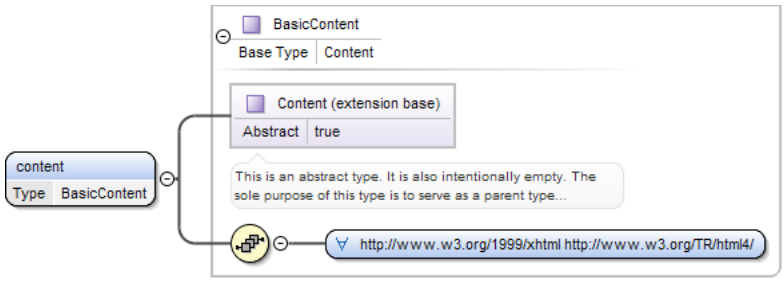
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	ItemID
Properties	content: complex
Model	itemTitle+
Children	itemTitle
Instance	<answerItem itemNumber="" itemType=""> <itemTitle xml:lang="">{1,unbounded}</itemTitle> </answerItem>

Attributes

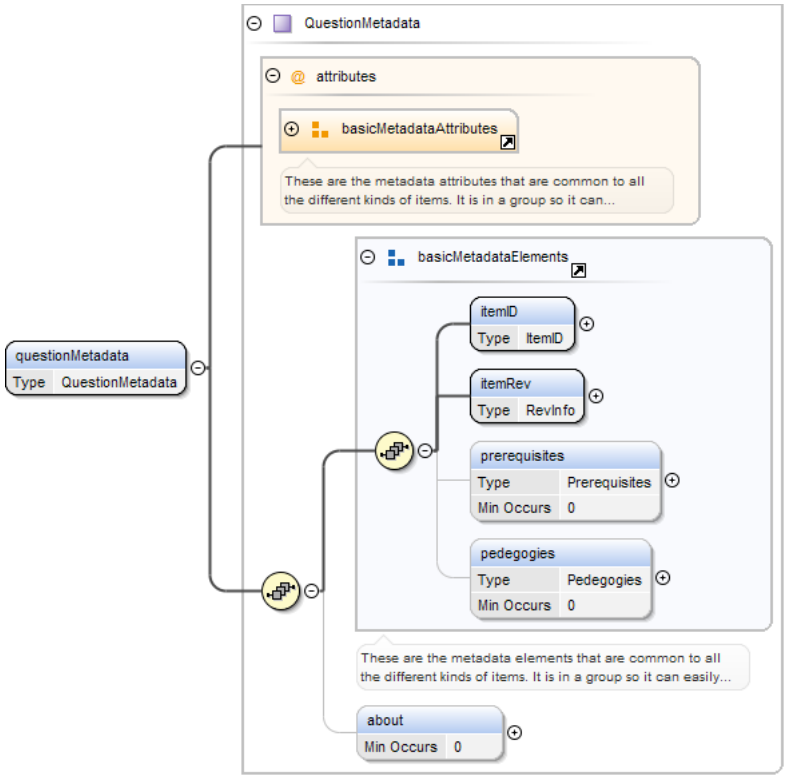
QName	Type	Fixed	Default	Use
itemNumber	SerialNumber			required
	This is simply a sequentialy assigned code number comprised of digits and capital letters. It must be unique among all the other items in the same topic folder.			
itemType	restriction of xs:string			optional
	This attribute is for informational purposes only, rather than to define a datatype. It is included so when an itemID is used in a reference, software or human readers can tell what type of item is being referenced without retrieving that actual item.			

Source	<xs:element name="answerItem" type="ItemID"/>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element `ExplanationItem / content`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	BasicContent
Type hierarchy	• Content • BasicContent
Properties	content: `complex`
Model	ANY element from namespace(s) 'http://www.w3.org/1999/xhtml http://www.w3.org/TR/html4/'
`Source`	`<xs:element name="content" type="BasicContent"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

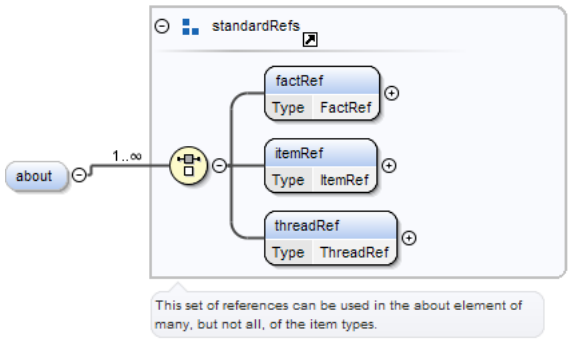
Element `QuestionItem / questionMetadata`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	QuestionMetadata
Properties	content: `complex`
Model	itemID , itemRev , prerequisites{0,1} , pedegogies{0,1} , about{0,1}
Children	about, itemID, itemRev, pedegogies, prerequisites

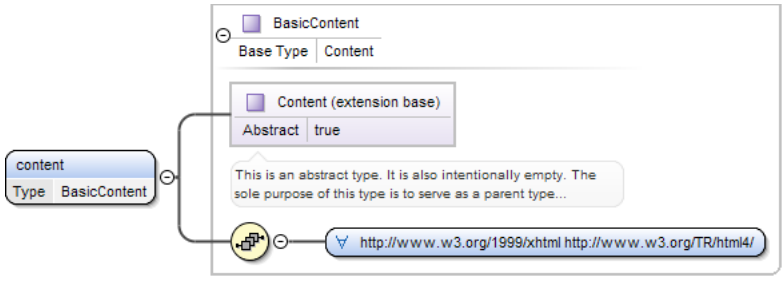
`Instance`	`<questionMetadata aETM="" difficulty="">`
	`  <itemID itemNumber="" itemType="">{1,1}</itemID>`
	`  <itemRev createDate="">{1,1}</itemRev>`
	`  <prerequisites>{0,1}</prerequisites>`

	<pedegogies>{0,1}</pedegogies> <about>{0,1}</about> </questionMetadata>
Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **aETM** / AETM / / / optional **difficulty** / Percent / / / optional
Source	<xs:element name="questionMetadata" type="QuestionMetadata"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element QuestionMetadata / about

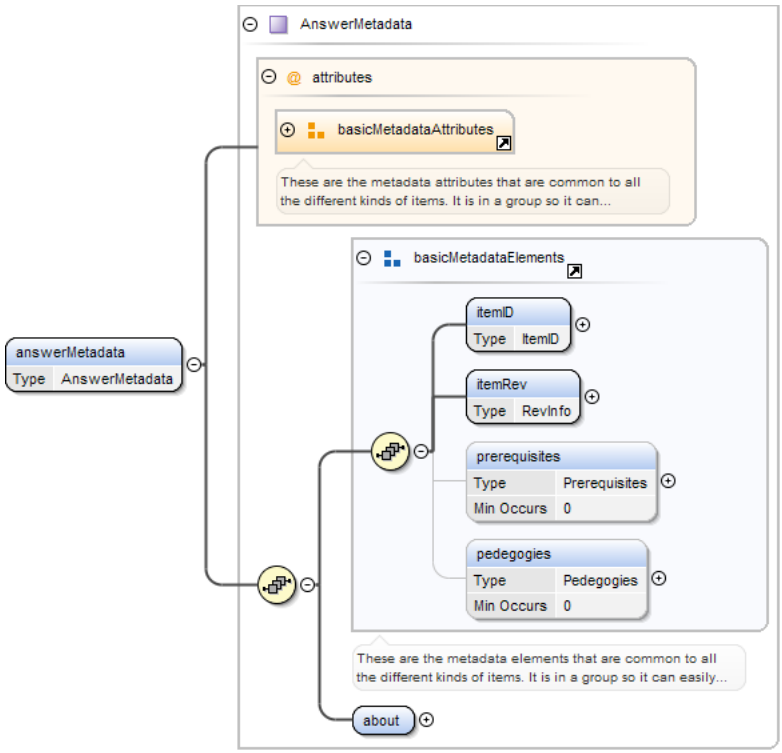
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Properties	content: complex minOccurs: 0
Model	factRef | itemRef | threadRef
Children	factRef, itemRef, threadRef
Instance	<about> <factRef>{1,1}</factRef> <itemRef>{1,1}</itemRef> <threadRef>{1,1}</threadRef> </about>
Source	<xs:element name="about" minOccurs="0"> <xs:complexType> <xs:group ref="standardRefs" maxOccurs="unbounded"/> </xs:complexType> </xs:element>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element QuestionItem / content

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	BasicContent
Type hierarchy	• Content • BasicContent
Properties	content: complex
Model	ANY element from namespace(s) 'http://www.w3.org/1999/xhtml http://www.w3.org/TR/html4/'
Source	<xs:element name="content" type="BasicContent"/>

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `AnswerItem / answerMetadata`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	AnswerMetadata
Properties	content: `complex`
Model	itemID , itemRev , prerequisites{0,1} , pedegogies{0,1} , about
Children	about, itemID, itemRev, pedegogies, prerequisites

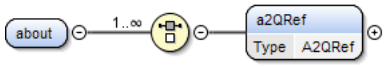
`Instance`	`<answerMetadata aETM="" difficulty="">`
	`  <itemID itemNumber="" itemType="">{1,1}</itemID>`
	`  <itemRev createDate="">{1,1}</itemRev>`
	`  <prerequisites>{0,1}</prerequisites>`
	`  <pedegogies>{0,1}</pedegogies>`
	`  <about>{1,1}</about>`
	`</answerMetadata>`

Attributes

QName	Type	Fixed	Default	Use
aETM	AETM			optional
difficulty	Percent			optional

`Source`	`<xs:element name="answerMetadata" type="AnswerMetadata"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `AnswerMetadata / about`

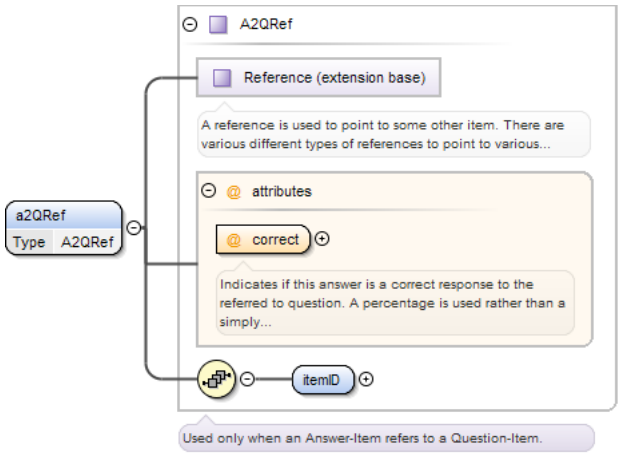
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Properties	content: `complex`
Model	a2QRef
Children	a2QRef
`Instance`	`<about>` `  <a2QRef correct="">{1,1}</a2QRef>`

	`</about>`

`Source`	`<xs:element name="about">`
	`  <xs:complexType>`
	`  <xs:choice maxOccurs="unbounded">`
	`  <xs:element name="a2QRef" type="A2QRef"/>`
	`  </xs:choice>`
	`  </xs:complexType>`
	`</xs:element>`

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `about / AnswerMetadata / a2QRef`

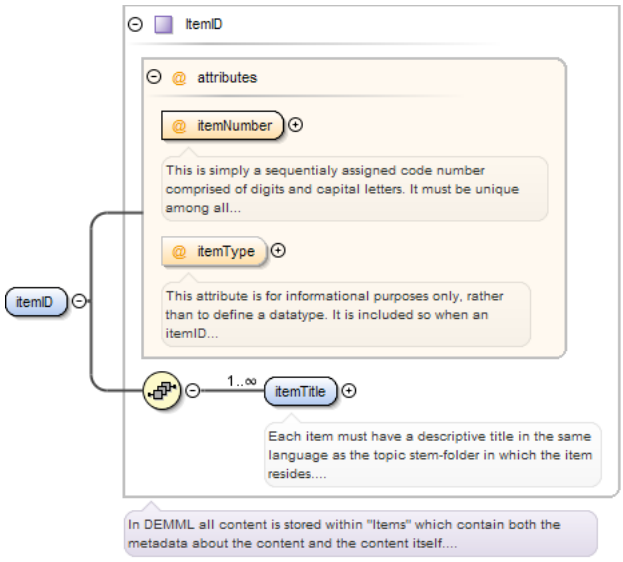
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	A2QRef
Type hierarchy	• Reference • A2QRef
Properties	content: `complex`
Model	itemID
Children	itemID

`Instance`	`<a2QRef correct="">`
	`  <itemID itemNumber="" itemType="">{1,1}</itemID>`
	`</a2QRef>`

Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **correct** / Percent / / / required `Indicates if this answer is a correct response to the referred to question. A percentage is used rather than a simply boolean because some answers are worth partial credit. Note: it is not necessary to create a reference from every Answer-Item to every Question-Item indicating the degree of correctness. Instead, only create these references when the answer is a plausible answer to the referred to question or is a good distractor for that question. This way software can automatically create quizes that are different each time.`
`Source`	`<xs:element name="a2QRef" type="A2QRef"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `A2QRef / itemID`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml

Diagram	
Type	ItemID
Properties	content: `complex`
Model	itemTitle+
Children	itemTitle

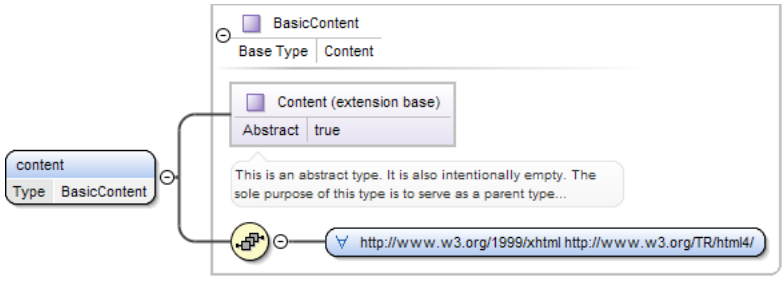
`Instance`	`<itemID itemNumber="" itemType="">`
	`  <itemTitle xml:lang="">{1,unbounded}</itemTitle>`
	`</itemID>`

Attributes

QName	Type	Fixed	Default	Use
itemNumber	SerialNumber			required
	`This is simply a sequentialy assigned code number comprised of digits and capital letters. It must be unique among all the other items in the same topic folder.`			
itemType	restriction of xs:string			optional
	`This attribute is for informational purposes only, rather than to define a datatype. It is included so when an itemID is used in a reference, software or human readers can tell what type of item is being referenced without retrieving that actual item.`			

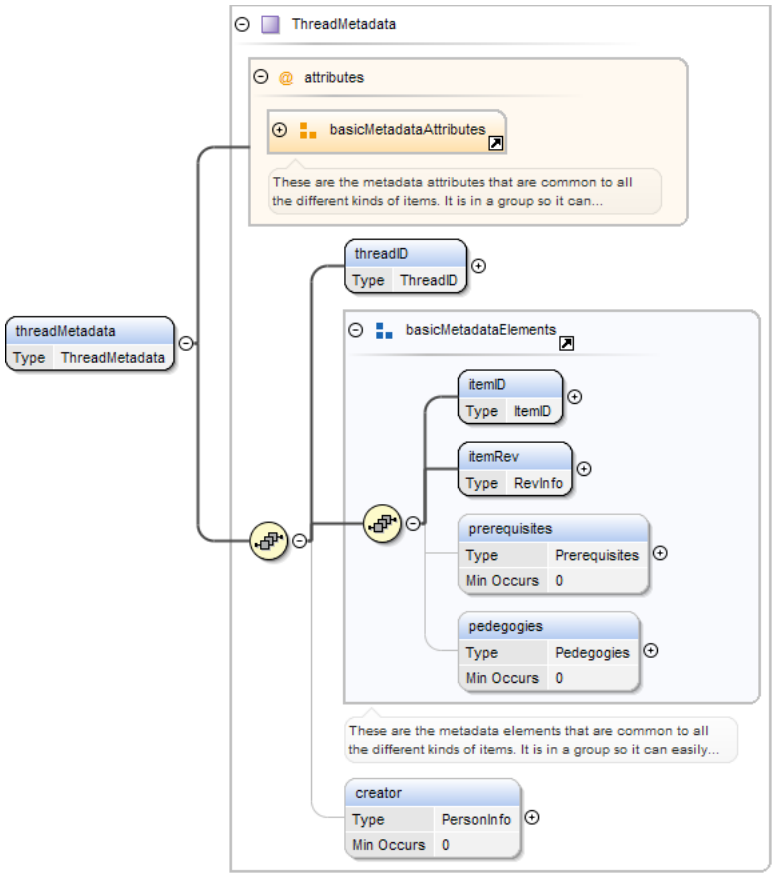
`Source`	`<xs:element name="itemID" type="ItemID"/>`
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Element `AnswerItem / content`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	BasicContent
Type hierarchy	• Content • BasicContent
Properties	content: `complex`
Model	ANY element from namespace(s) 'http://www.w3.org/1999/xhtml http://www.w3.org/TR/html4/'
`Source`	`<xs:element name="content" type="BasicContent"/>`

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `ThreadItem / threadMetadata`

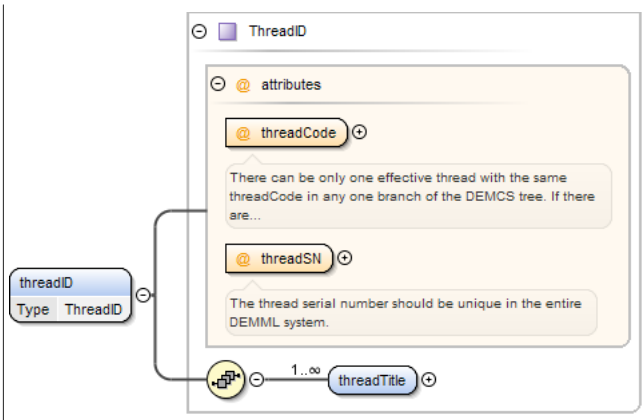
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	ThreadMetadata
Properties	content: `complex`
Model	threadID , itemID , itemRev , prerequisites{0,1} , pedegogies{0,1} , creator{0,1}
Children	creator, itemID, itemRev, pedegogies, prerequisites, threadID

`Instance`	`<threadMetadata aETM="" difficulty="">`
	`  <threadID threadCode="" threadSN="">{1,1}</threadID>`
	`  <creator>{0,1}</creator>`
	`</threadMetadata>`

Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **aETM** / AETM / / / optional **difficulty** / Percent / / / optional
`Source`	`<xs:element name="threadMetadata" type="ThreadMetadata"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `ThreadMetadata / threadID`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml

Diagram	
Type	ThreadID
Properties	content: complex
Model	threadTitle+
Children	threadTitle

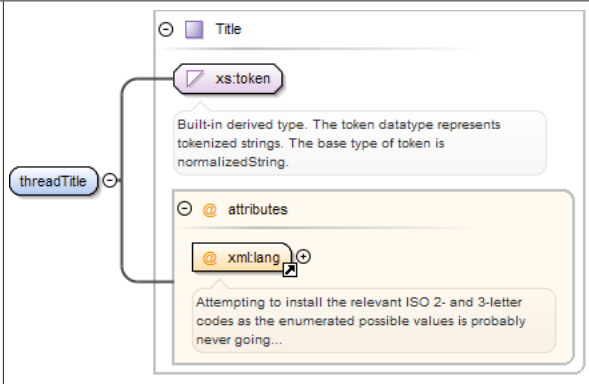
Instance	`<threadID threadCode="" threadSN="">`
	`  <threadTitle xml:lang="">{1,unbounded}</threadTitle>`
	`</threadID>`

Attributes

QName	Type	Fixed	Default	Use
threadCode	xs:token			required
	There can be only one effective thread with the same threadCode in any one branch of the DEMCS tree. If there are threads with the same threadCode in more than one level of the same branch of the tree, the thread in the branch farthest from the root of the tree takes precedence. This allows a type of inheritance in that a thread can be more general closer to the root of the tree but then be modified to be more specific to the topics in the branches farther out from the root. There is no formal mechanism for controling inheritance. Whichever thread is referenced is the thread that will be applied to the referencing item.			
threadSN	SerialNumber			required
	The thread serial number should be unique in the entire DEMML system.			

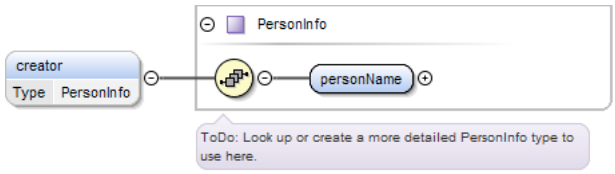
Source	`<xs:element name="threadID" type="ThreadID"/>`
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element ThreadID / threadTitle

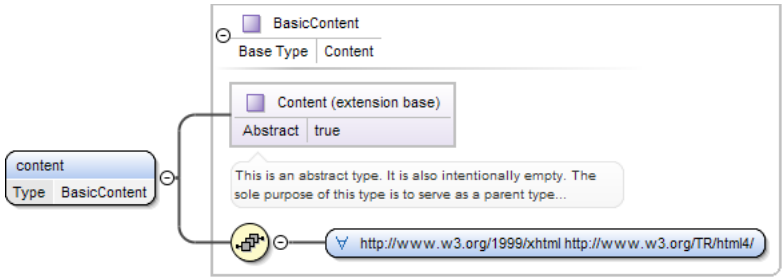
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	Title
Properties	content: complex maxOccurs: unbounded

Attributes	**QName**	**Type**	**Fixed**	**Default**	**Use**
	xml:lang	union of(xs:language, restriction of xs:string)			required
	Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.				
Source	<xs:element name="threadTitle" type="Title" maxOccurs="unbounded"/>				
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd				

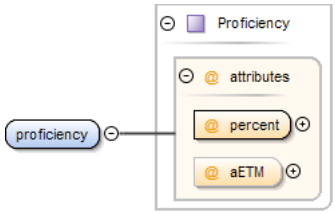
Element ThreadMetadata / creator

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	PersonInfo
Properties	content: complex minOccurs: 0
Model	personName
Children	personName
Instance	<creator> <personName>{1,1}</personName> </creator>
Source	<xs:element name="creator" minOccurs="0" type="PersonInfo"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element ThreadItem / content

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	BasicContent
Type hierarchy	• Content • BasicContent
Properties	content: complex
Model	ANY element from namespace(s) 'http://www.w3.org/1999/xhtml http://www.w3.org/TR/html4/'
Source	<xs:element name="content" type="BasicContent"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Element `Proficiencies / proficiency`

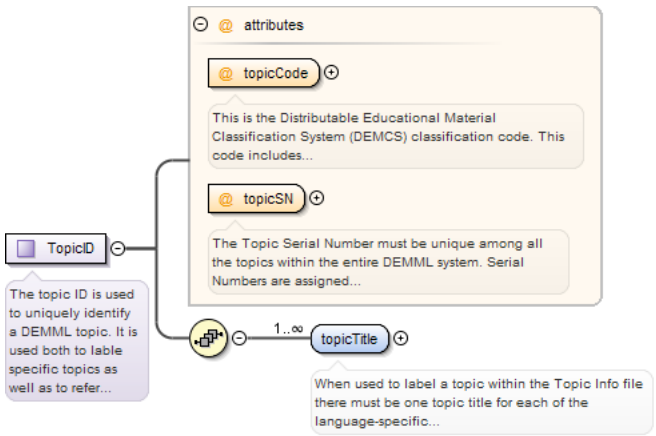
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	Proficiency
Properties	content: `complex` maxOccurs: `unbounded`

Attributes	QName	Type	Fixed	Default	Use
	aETM	AETM			optional
	percent	PercentPlus			required

`Source`	`<xs:element name="proficiency" maxOccurs="unbounded" type="Proficiency"/>`
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Types

Complex Type `TopicID`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	`The topic ID is used to uniquely identify a DEMML topic. It is used both to lable specific topics as well as to refer to them.`
Diagram	
Used by	Elements ExtRef/topicID, Prerequisites/prereq/topicID, ThreadRef/topicID, demml/topicID
Model	topicTitle+
Children	topicTitle

Attributes	QName	Type	Fixed	Default	Use
	topicCode	xs:token			required
		`<p>This is the Distributable Educational Material Classification System (DEMCS) classification code. This code includes the full path to the topic within the DEMCS. See <a href="http://demml.org/standard/classification/">http://demml.org/standard/classification/</a>for more information.</p>` `<p>Do not include the demml:// protocol prefix in the value for this attribute.</p>`			
	topicSN	SerialNumber			required
		`The Topic Serial Number must be unique among all the topics within the entire DEMML system. Serial Numbers are assigned sequentially as each topic is added the DEMCS. In case some topics need to be rearranged`			

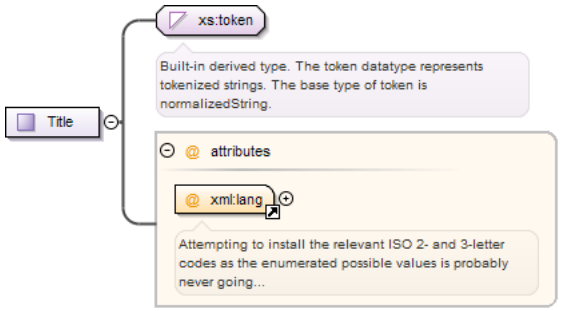
	QName	Type	Fixed	Default	Use
		at some time in the future (hopefully very rarely) the serial numbers can be used by software to locate the correct topics and update their locations within individual users' databases of DEMML content.			

Source

```
<xs:complexType name="TopicID">
  <xs:annotation>
    <xs:documentation>The topic ID is used to uniquely identify a DEMML topic. It is used
 both to lable specific topics as well as to refer to them.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="topicTitle" type="Title" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>When used to label a topic within the Topic Info file there must
 be one topic title for each of the language-specific stem folders in the topic folder.
 When used to indicate the topic in which a DEMML content file belongs, the topicID must
 include a descriptive title in the same language as the topic stem-folder in which the
 file belongs or resides. However, one may provide optional additional titles in other
 languages if desired.</xs:documentation>
        <xs:documentation>Remember, the title is purely for human consumption. All
 software must use only the topicCode and topicSN to refer to and index the topics.</
xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="topicCode" type="xs:token" use="required">
    <xs:annotation>
      <xs:documentation>
        <p>This is the Distributable Educational Material Classification System (DEMCS)
 classification code. This code includes the full path to the topic within the DEMCS. See
          <a href="http://demml.org/standard/classification/">http://demml.org/standard/
classification/</a>for more information.</p>
        <p>Do not include the demml:// protocol prefix in the value for this attribute.</
p>
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="topicSN" type="SerialNumber" use="required">
    <xs:annotation>
      <xs:documentation>The Topic Serial Number must be unique among all the topics
 within the entire DEMML system. Serial Numbers are assigned sequentially as each topic
 is added the DEMCS. In case some topics need to be rearranged at some time in the future
 (hopefully very rarely) the serial numbers can be used by software to locate the correct
 topics and update their locations within individual users' databases of DEMML content.</
xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `Title`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	extension of xs:token
Used by	Elements FactID/factTitle, ItemID/itemTitle, ThreadID/threadTitle, TopicID/topicTitle

Attributes

QName	Type	Fixed	Default	Use
xml:lang	union of(xs:language, restriction of xs:string)			required
	Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See			

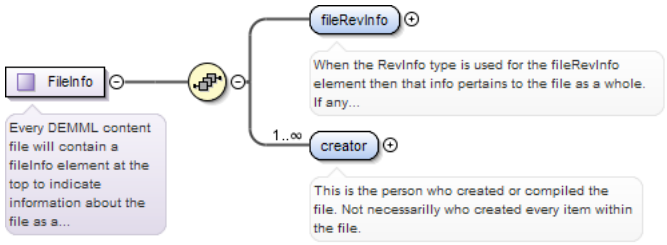
	QName	Type	Fixed	Default	Use
		RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.			

Source

```
<xs:complexType name="Title">
  <xs:simpleContent>
    <xs:extension base="xs:token">
      <xs:attribute use="required" ref="xml:lang"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Schema location: http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `FileInfo`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Every DEMML content file will contain a fileInfo element at the top to indicate information about the file as a whole. Any elements declared using this type should be called 'fileInfo'
Diagram	
Used by	Element demml/fileInfo
Model	fileRevInfo , creator+
Children	creator, fileRevInfo

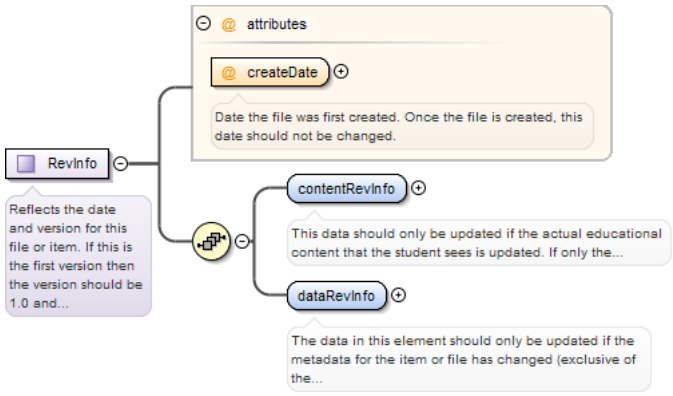
Source

```
<xs:complexType name="FileInfo">
  <xs:annotation>
    <xs:documentation>Every DEMML content file will contain a fileInfo element at the top
 to indicate information about the file as a whole.</xs:documentation>
    <xs:documentation>Any elements declared using this type should be called 'fileInfo'</
xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="fileRevInfo" type="RevInfo">
      <xs:annotation>
        <xs:documentation>When the RevInfo type is used for the fileRevInfo element then
 that info pertains to the file as a whole. If any content for any of the items within the
 file is updated then the fileRefInfo/contentRevInfo element should be updated. Similarly
 if any of the metadata for any of the items within the file or if the file's metadata
 (exclusive of the fileRefInfo/contentRevInfo element) is modified, then the fileRefInfo/
dataRevInfo element should be updated.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="creator" maxOccurs="unbounded" type="PersonInfo">
      <xs:annotation>
        <xs:documentation>This is the person who created or compiled the file. Not
 necessarilly who created every item within the file.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Schema location: http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `RevInfo`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Reflects the date and version for this file or item. If this is the first version then the version should be 1.0 and the dates should match the createDate attribute of the fileRevInfo element.

Diagram	
Used by	Elements FileInfo/fileRevInfo, basicMetadataElements/itemRev
Model	contentRevInfo , dataRevInfo
Children	contentRevInfo, dataRevInfo

Attributes

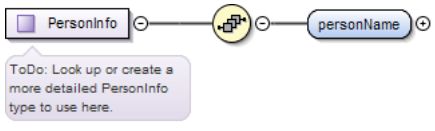
QName	Type	Fixed	Default	Use
createDate	xs:date			required
	Date the file was first created. Once the file is created, this date should not be changed.			

Source

```
<xs:complexType name="RevInfo">
  <xs:annotation>
    <xs:documentation>Reflects the date and version for this file or item. If this is the
 first version then the version should be 1.0 and the dates should match the createDate
 attribute of the fileRevInfo element.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="contentRevInfo">
      <xs:annotation>
        <xs:documentation>This data should only be updated if the actual educational
 content that the student sees is updated. If only the metadata changes then do not update
 this element.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:attribute name="contentVersion" type="xs:token" use="required"/>
        <xs:attribute name="contentModDate" type="xs:date" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="dataRevInfo">
      <xs:annotation>
        <xs:documentation>The data in this element should only be updated if the metadata
 for the item or file has changed (exclusive of the contentRevInfo element). If only the
 content has been changed then do not update this element.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:attribute name="dataVersion" type="xs:token" use="required"/>
        <xs:attribute name="dataModDate" type="xs:date" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="createDate" type="xs:date" use="required">
    <xs:annotation>
      <xs:documentation>Date the file was first created. Once the file is created, this
 date should not be changed.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

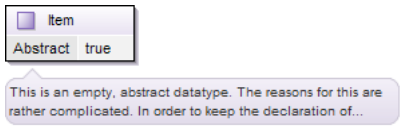
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `PersonInfo`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	ToDo: Look up or create a more detailed PersonInfo type to use here.
Diagram	

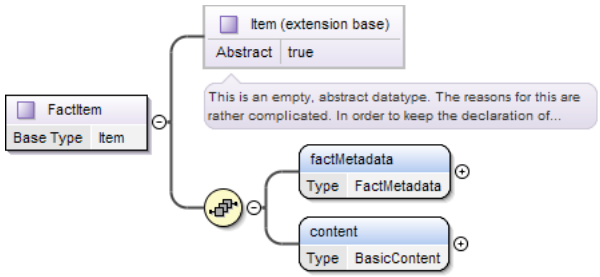
Used by	Elements FileInfo/creator, ThreadMetadata/creator
Model	personName
Children	personName
Source	<xs:complexType name="PersonInfo"> <xs:annotation> <xs:documentation>ToDo: Look up or create a more detailed PersonInfo type to use here.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="personName" type="xs:token"/> </xs:sequence> </xs:complexType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type Item

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	This is an empty, abstract datatype. The reasons for this are rather complicated. In order to keep the declaration of the demml element simple, even though there will be many different types of items, it was necessary for demml/item element to use an abstract datatype allowing type substitution of the child datatypes in the place of this element. In order to keep the structure of the item elements consistent it is best to keep the metadata on top and the content last. Therefore, it is also best to keep the metadata and content in their own separate elements, allowing each of their datatypes to also be abstracted and extended. While itemMetadata and content elements could have been placed here to then reference those abstract datatypes, allowing their replacement with concrete datatypes, this would not have ensured that the correct sets of metadata were always paired with the correct type of content. It is also the case that not all items will actually contain content. Others may have more complicated content than can be expressed by a simple element with some HTML in it. Therefore, it is necessary to declare each different item type as an extension of this type with a metadata element and content element designed just for that item type.
Diagram	
Properties	abstract: true
Used by	Element demml/item Complex Types AnswerItem, ExplanationItem, FactItem, QuestionItem, ThreadItem
Source	<xs:complexType name="Item" abstract="true"> <xs:annotation> <xs:documentation>This is an empty, abstract datatype. The reasons for this are rather complicated. In order to keep the declaration of the demml element simple, even though there will be many different types of items, it was necessary for demml/item element to use an abstract datatype allowing type substitution of the child datatypes in the place of this element. In order to keep the structure of the item elements consistent it is best to keep the metadata on top and the content last. Therefore, it is also best to keep the metadata and content in their own separate elements, allowing each of their datatypes to also be abstracted and extended. While itemMetadata and content elements could have been placed here to then reference those abstract datatypes, allowing their replacement with concrete datatypes, this would not have ensured that the correct sets of metadata were always paired with the correct type of content. It is also the case that not all items will actually contain content. Others may have more complicated content than can be expressed by a simple element with some HTML in it. Therefore, it is necessary to declare each different item type as an extension of this type with a metadata element and content element designed just for that item type.</xs:documentation> </xs:annotation> </xs:complexType>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Complex Type FactItem

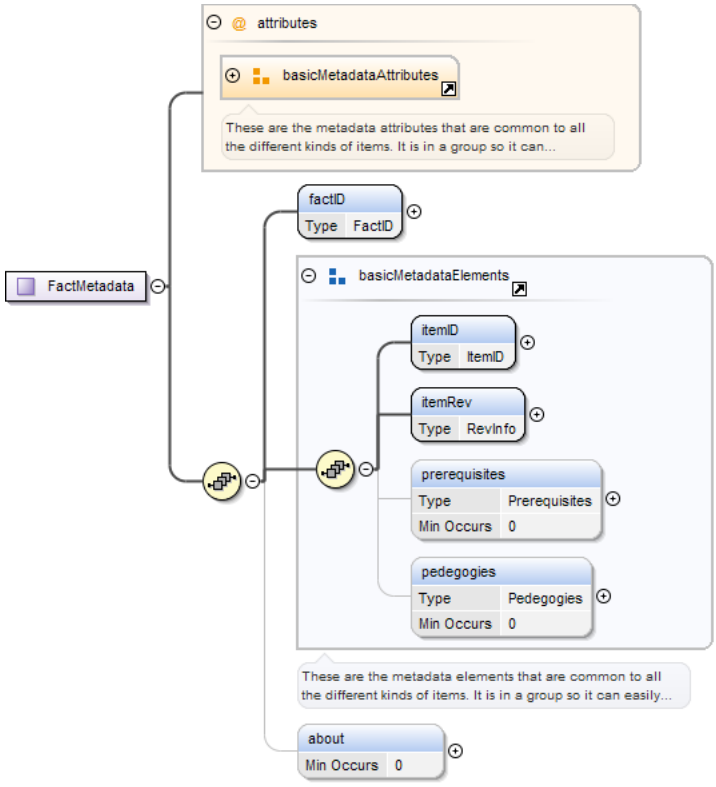
Namespace	http://www.demml.org/schemas/alpha/0.1/demml

Diagram	
Type	extension of Item
Type hierarchy	• Item • FactItem
Model	factMetadata , content
Children	content, factMetadata

Source	`<xs:complexType name="FactItem">`
	`  <xs:complexContent>`
	`  <xs:extension base="Item">`
	`  <xs:sequence>`
	`  <xs:element name="factMetadata" type="FactMetadata"/>`
	`  <xs:element name="content" type="BasicContent"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Complex Type `FactMetadata`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Used by	Element FactItem/factMetadata
Model	factID , itemID , itemRev , prerequisites{0,1} , pedegogies{0,1} , about{0,1}
Children	about, factID, itemID, itemRev, pedegogies, prerequisites

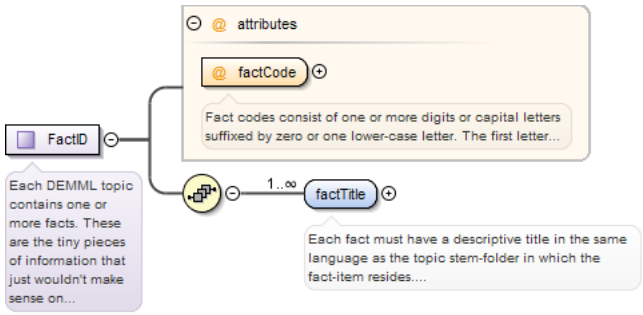
Attributes	**QName**	**Type**	**Fixed**	**Default**	**Use**
	aETM	AETM			optional
	difficulty	Percent			optional

Source

```
<xs:complexType name="FactMetadata">
  <xs:sequence>
    <xs:element name="factID" type="FactID"/>
    <xs:group ref="basicMetadataElements"/>
    <xs:element name="about" minOccurs="0">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="extRef" type="ExtRef"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="basicMetadataAttributes"/>
</xs:complexType>
```

Schema location: file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Complex Type `FactID`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Each DEMML topic contains one or more facts. These are the tiny pieces of information that just wouldn't make sense on their own. A fact is one of the many special types of items. For each fact there could be more than one way to state that fact. All the different ways to state the same fact will each be in their own individual items but will all have the same fact code. This way software can allow students or teachers to choose the way of stating that fact that makes the most sense to them. Any elements created from this type should have the name 'factID'
Diagram	
Used by	Elements: ExtRef/factID, FactMetadata/factID, FactRef/factID, Prerequisites/prereq/factID	
Model	factTitle+	
Children	factTitle	

Attributes	QName	Type	Fixed	Default	Use
	factCode	restriction of xs:string			required

Fact codes consist of one or more digits or capital letters suffixed by zero or one lower-case letter. The first letter is always a capital 'F' to make them easier to recognize when written down. In order to allow additional facts to be inserted into an ordered list of preexisting facts, fact codes follow the same rules and pattern as the DEMCS coding system (see http://demml.org/standard/classification/4-assign_codes.htm). In some cases one level of sub-facts will be appropriate. In which case the sub-facts will have the code of the parent fact followed directly by one lower-case letter. For instance, N7b or 3La. If at any time one feels it necessary to create a third level of facts then serious consideration should be given to creating sub-topics instead, putting each of the primary parent facts in its own topic. Sometimes the order of facts will not matter. But for some topics where the order the facts are listed helps provide meaning to these facts then make sure the facts are numbered in an appropriate order.

Fact F0 is the statement of the main point of the topic.

Source

```
<xs:complexType name="FactID">
  <xs:annotation>
    <xs:documentation>Each DEMML topic contains one or more facts. These are the tiny
 pieces of information that just wouldn't make sense on their own. A fact is one of the
 many special types of items. For each fact there could be more than one way to state that
```

	fact. All the different ways to state the same fact will each be in their own individual items but will all have the same fact code. This way software can allow students or teachers to choose the way of stating that fact that makes the most sense to them.</xs:documentation> <xs:documentation>Any elements created from this type should have the name 'factID'</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="factTitle" type="Title" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>Each fact must have a descriptive title in the same language as the topic stem-folder in which the fact-item resides. However, one may provide optional additional titles in other languages if desired.</xs:documentation> <xs:documentation>Because each fact is contained in a separate item, the Item Title for the item containing the fact shuld be the same as the Fact Title.</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> <xs:attribute name="factCode" use="required"> <xs:annotation> <xs:documentation>Fact codes consist of one or more digits or capital letters suffixed by zero or one lower-case letter. The first letter is always a capital 'F' to make them easier to recognize when written down. In order to allow additional facts to be inserted into an ordered list of preexisting facts, fact codes follow the same rules and pattern as the DEMCS coding system (see http://demml.org/standard/classification/4-assign_codes.htm). In some cases one level of sub-facts will be appropriate. In which case the sub-facts will have the code of the parent fact followed directly by one lower-case letter. For instance, N7b or 3La. If at any time one feels it necessary to create a third level of facts then serious consideration should be given to creating sub-topics instead, putting each of the primary parent facts in its own topic. Sometimes the order of facts will not matter. But for some topics where the order the facts are listed helps provide meaning to these facts then make sure the facts are numbered in an appropriate order.</xs:documentation> <xs:documentation>Fact F0 is the statement of the main point of the topic.</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:pattern value="F[0-9A-Z]+[a-z]*"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

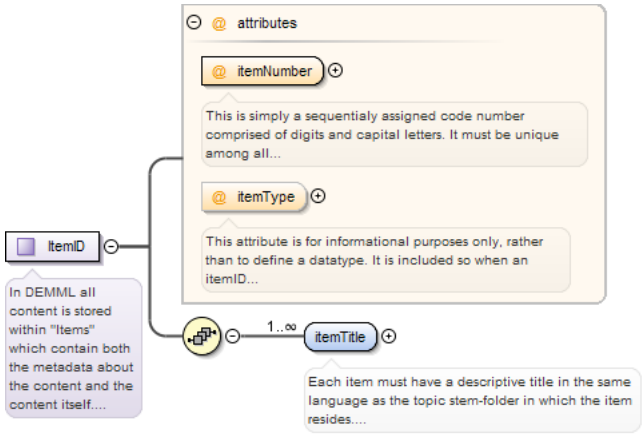
Complex Type `ItemID`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	In DEMML all content is stored within "Items" which contain both the metadata about the content and the content itself. All items about a particular topic are stored in the appropriate language stem folder under the topic folder. However, it is possible for an item to be associated with more than one topic. In that case the item will be stored under the most closely associated topic (or the topic closest to the root of the tree, or just choose one at random). More than one item can be stored in a particular DEMML content file but all items in a file obviously must be associated with the same topic in order to meet the above requirement. DEMML items can be of various types. Most important are the Fact-Items which state one of the facts that must be learned about the topic. Fact-Items should be concise and to the point. Explanation-Items are items which explain something within the topic. This is where most of the learning takes place. An Explanation-Item can explain a fact or even why a particular answer is right or wrong. There are Question-Items and Answer-Items as well as Problem-Items and Solution-Items. There can be questions about one or more facts or even about one particular explanation. Explanations can be about almost any other kind of item. However, there should never be questions about explanations which are already about some other question or answer. One has to draw the line somewhere. While most XML developers have become accustomed to mirroring the hierarchical structure of their data within the XML document itself, DEMML takes a different tack for the structure of items within a topic. One of the primary goals of DEMML is to allow easy distribution of the content in the form of relatively small files. This way a student does not need to download (or copy) and store much more than they really need and the transfer will be as efficient as possible. In addition, there is the possibility of there being hundreds - or even thousands - of items within a topic, especially when you consider all the different problems, solutions, and explanations for those solutions for a single mathematical concept. Therefore, it is not reasonable to attempt to store a topic's entire structure of items within one XML document. All this means the structure of the relationships between DEMML items is a limited sort of directed graph with the references pointing backwards. Rather than have each fact contain a reference to every question about that fact, as many software developers are accustomed to seeing, it is far more efficient to have each question contain references to the few items that question is about. This way the Fact-Items do not need to be

updated every time a new Question-Item is published. All a user has to do is copy the file with the Question-Item into the appropriate folder on their computer (which can be done automatically by some software) and the software can read the file and update its internal database.</p>

Any time an element is declared using this type it should be called 'itemID'

Diagram



Used by

Elements A2QRef/itemID, ItemRef/itemID, QApairRef/answerItem, QApairRef/questionItem, ThreadRef/itemID, basicMetadataElements/itemID

Model itemTitle+

Children itemTitle

Attributes

QName	Type	Fixed	Default	Use
itemNumber	SerialNumber			required
	This is simply a sequentialy assigned code number comprised of digits and capital letters. It must be unique among all the other items in the same topic folder.			
itemType	restriction of xs:string			optional
	This attribute is for informational purposes only, rather than to define a datatype. It is included so when an itemID is used in a reference, software or human readers can tell what type of item is being referenced without retrieving that actual item.			

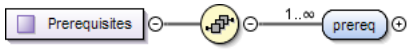
Source

```
<xs:complexType name="ItemID">
  <xs:annotation>
    <xs:documentation>
      <p>In DEMML all content is stored within "Items" which contain both the metadata
 about the content and the content itself. All items about a particular topic are stored
 in the appropriate language stem folder under the topic folder. However, it is possible
 for an item to be associated with more than one topic. In that case the item will be
 stored under the most closely associated topic (or the topic closest to the root of the
 tree, or just choose one at random). More than one item can be stored in a particular
 DEMML content file but all items in a file obviously must be associated with the same
 topic in order to meet the above requirement.</p>
      <p>DEMML items can be of various types. Most important are the Fact-Items which
 state one of the facts that must be learned about the topic. Fact-Items should be concise
 and to the point. Explanation-Items are items which explain something within the topic.
 This is where most of the learning takes place. An Explanation-Item can explain a fact
 or even why a particular answer is right or wrong. There are Question-Items and Answer-
Items as well as Problem-Items and Solution-Items. There can be questions about one or
 more facts or even about one particular explanation. Explanations can be about almost any
 other kind of item. However, there should never be questions about explanations which are
 already about some other question or answer. One has to draw the line somewhere.</p>
      <p>While most XML developers have become accustomed to mirroring the hierarchical
 structure of their data within the XML document itself, DEMML takes a different tack for
 the structure of items within a topic. One of the primary goals of DEMML is to allow easy
 distribution of the content in the form of relatively small files. This way a student
 does not need to download (or copy) and store much more than they really need and the
 transfer will be as efficient as possible. In addition, there is the possibility of
 there being hundreds - or even thousands - of items within a topic, especially when you
 consider all the different problems, solutions, and explanations for those solutions
 for a single mathematical concept. Therefore, it is not reasonable to attempt to store a
 topic's entire structure of items within one XML document.</p>
      <p>All this means the structure of the relationships between DEMML items is a
 limited sort of directed graph with the references pointing backwards. Rather than
 have each fact contain a reference to every question about that fact, as many software
 developers are accustomed to seeing, it is far more efficient to have each question
 contain references to the few items that question is about. This way the Fact-Items do
```

```
not need to be updated every time a new Question-Item is published. All a user has to
do is copy the file with the Question-Item into the appropriate folder on their computer
(which can be done automatically by some software) and the software can read the file and
update its internal database.</p>
   </xs:documentation>
   <xs:documentation>Any time an element is declared using this type it should be called
               'itemID'</xs:documentation>
 </xs:annotation>
 <xs:sequence>
   <xs:element name="itemTitle" type="Title" maxOccurs="unbounded">
     <xs:annotation>
       <xs:documentation>Each item must have a descriptive title in the same language
as the topic stem-folder in which the item resides. However, one may provide optional
additional titles in other languages if desired.</xs:documentation>
       <xs:documentation>If the item is a question or problem then the title should not
give away the answer, just in case some software chooses to display the titles of each
item.</xs:documentation>
     </xs:annotation>
   </xs:element>
 </xs:sequence>
 <xs:attribute name="itemNumber" type="SerialNumber" use="required">
   <xs:annotation>
     <xs:documentation>This is simply a sequentialy assigned code number comprised of
digits and capital letters. It must be unique among all the other items in the same topic
folder.</xs:documentation>
   </xs:annotation>
 </xs:attribute>
 <xs:attribute name="itemType">
   <xs:annotation>
     <xs:documentation>This attribute is for informational purposes only, rather than
to define a datatype. It is included so when an itemID is used in a reference, software
or human readers can tell what type of item is being referenced without retrieving that
actual item.</xs:documentation>
   </xs:annotation>
   <xs:simpleType>
     <xs:restriction base="xs:string">
       <xs:enumeration value="fact"/>
       <xs:enumeration value="explanation"/>
       <xs:enumeration value="question"/>
       <xs:enumeration value="answer"/>
       <xs:enumeration value="thread"/>
       <!--<xs:enumeration value="problem"/>-->
       <!--<xs:enumeration value="solution"/>-->
       <!--<xs:enumeration value="definition"/>-->
       <!--<xs:enumeration value="exercise"/>-->
       <!--<xs:enumeration value="exerciseExternal"/>-->
       <!--<xs:enumeration value="lab"/>-->
       <!--<xs:enumeration value="project"/>-->
       <!--<xs:enumeration value="identification"/>-->
       <!--<xs:enumeration value="demmlLink"/>-->
       <!--<xs:enumeration value="externalLink"/>-->
     </xs:restriction>
   </xs:simpleType>
 </xs:attribute>
</xs:complexType>
```

Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `Prerequisites`

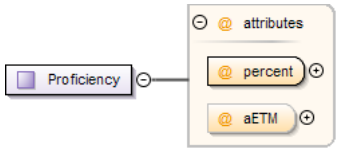
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Used by	Element basicMetadataElements/prerequisites
Model	prereq+
Children	prereq
Source	

```
<xs:complexType name="Prerequisites">
  <xs:sequence>
    <xs:element name="prereq" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="topicID" type="TopicID"/>
          <xs:element name="factID" type="FactID" minOccurs="0"/>
          <xs:element name="proficiency" type="Proficiency">
            <xs:annotation>
              <xs:documentation>When listed as part of a prerequisite, a proficiency value
 refers to how well the student must know the specified topic or fact within a topic. In
```

```
addition, the estimated time to master (etm) attribute should be eiter set to "PT0S" for
zero seconds or left out.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
 </xs:sequence>
</xs:complexType>
```

Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `Proficiency`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Used by	Elements Prerequisites/prereq/proficiency, Proficiencies/proficiency

Attributes

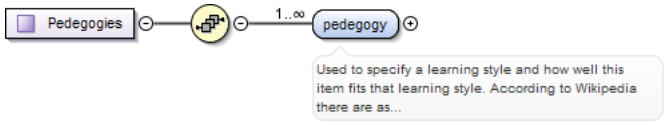
QName	Type	Fixed	Default	Use
aETM	AETM			optional
percent	PercentPlus			required

Source

```
<xs:complexType name="Proficiency">
  <xs:attribute name="percent" type="PercentPlus" use="required"/>
  <xs:attribute name="aETM" type="AETM"/>
</xs:complexType>
```

Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `Pedegogies`

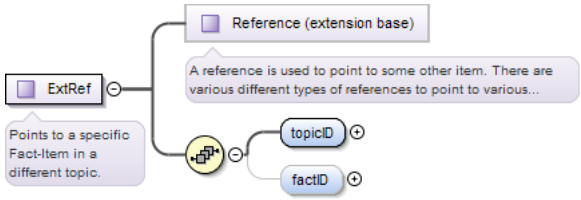
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Used by	Element basicMetadataElements/pedegogies
Model	pedegogy+
Children	pedegogy

Source

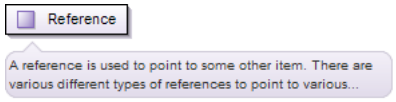
```
<xs:complexType name="Pedegogies">
  <xs:sequence>
    <xs:element name="pedegogy" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Used to specify a learning style and how well this item fits
 that learning style. According to Wikipedia there are as many as 71 different "learning
 style" models, none of which has been proven valid in any way (Pashler, H.; McDaniel, M.;
 Rohrer, D.; Bjork, R. (2009). "Learning styles: Concepts and evidence". Psychological
 Science in the Public Interest 9: 105–119. cited in: http://en.wikipedia.org/wiki/
Learning_styles#cite_ref-pashler_1-2). However, this element is made available for those
 who either subscribe to the Learning Style theories or simply want to provide more
 information about their content. At the very least, software can use these tags to
 ensure that students get a good mix of content based on different learning styles for any
 particular topic or fact.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="comment" type="xs:string" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Optional explanation as to how and why this item fits this
 learning model.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="model" type="TokenURI" use="required"/>
        <xs:attribute name="domain" type="TokenURI"/>
        <xs:attribute name="Classification" type="TokenURI" use="required"/>
        <xs:attribute name="percent" type="Percent" use="required"/>
      </xs:complexType>
```

	</xs:element> </xs:sequence> </xs:complexType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `ExtRef`

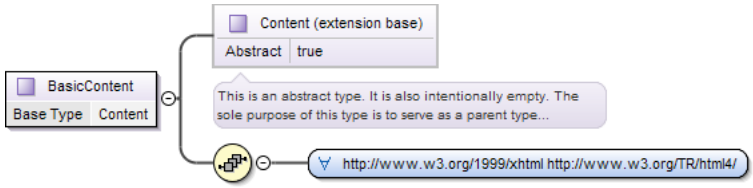
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Points to a specific Fact-Item in a different topic.
Diagram	
Type	extension of Reference
Type hierarchy	• Reference • ExtRef
Used by	Element FactMetadata/about/extRef
Model	topicID , factID{0,1}
Children	factID, topicID
Source	<xs:complexType name="ExtRef"> <xs:annotation> <xs:documentation>Points to a specific Fact-Item in a different topic.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="Reference"> <xs:sequence> <xs:element name="topicID" type="TopicID"/> <xs:element name="factID" type="FactID" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `Reference`

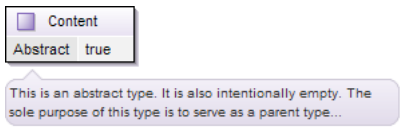
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	A reference is used to point to some other item. There are various different types of references to point to various different categories of items. This datatype is an empty, abstract datatype. It exists solely to indicate that all the different types of references are of one family. Though it would have been possible to create one reference datatype which optionally allowed all the different types of identifiers, that method would have resulted in DEMML content documents that were not as easy to read. It would also have allowed invalid combinations of the various identifiers. This method enforces a specific set of combinations of identifiers. It also forces content creators to use the xsi:type= attribute in their reference elements to indicate the actual datatype being used, which allows human readers to easily see what that reference is pointing to. Finally, it allows developers to make use of inheritance and polymorphism, specifying this datatype in an object which then allows all the child datatypes.
Diagram	
Properties	abstract: true
Used by	Complex Types A2QRef, ExtRef, FactRef, ItemRef, QApairRef, ThreadRef
Source	<xs:complexType name="Reference" abstract="true"> <xs:annotation> <xs:documentation>A reference is used to point to some other item. There are various different types of references to point to various different categories of items.</xs:documentation>

	<xs:documentation>This datatype is an empty, abstract datatype. It exists solely to indicate that all the different types of references are of one family. Though it would have been possible to create one reference datatype which optionally allowed all the different types of identifiers, that method would have resulted in DEMML content documents that were not as easy to read. It would also have allowed invalid combinations of the various identifiers. This method enforces a specific set of combinations of identifiers. It also forces content creators to use the xsi:type= attribute in their reference elements to indicate the actual datatype being used, which allows human readers to easily see what that reference is pointing to. Finally, it allows developers to make use of inheritance and polymorphism, specifying this datatype in an object which then allows all the child datatypes.</xs:documentation> </xs:annotation> </xs:complexType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `BasicContent`

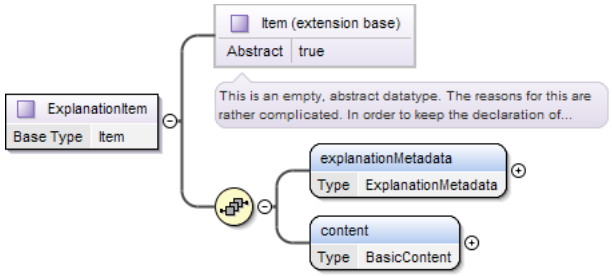
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	extension of Content
Type hierarchy	• Content • BasicContent
Used by	Elements AnswerItem/content, ExplanationItem/content, FactItem/content, QuestionItem/content, ThreadItem/content
Model	ANY element from namespace(s) 'http://www.w3.org/1999/xhtml http://www.w3.org/TR/html4/'
Source	<xs:complexType name="BasicContent"> <xs:complexContent> <xs:extension base="Content"> <xs:sequence> <xs:any namespace="http://www.w3.org/1999/xhtml http://www.w3.org/TR/html4/" processContents="skip"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Complex Type `Content`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	This is an abstract type. It is also intentionally empty. The sole purpose of this type is to serve as a parent type for other types of content elements. This allows the demml/item element to remain relatively simple. It also allows software developers to create a parent or interface class from which child classes can be extended or implemented. This way any of these content types can be stored in the same container if necessary.
Diagram	
Properties	abstract: true
Used by	Complex Type BasicContent
Source	<xs:complexType name="Content" abstract="true"> <xs:annotation> <xs:documentation>This is an abstract type. It is also intentionally empty. The sole purpose of this type is to serve as a parent type for other types of content elements. This allows the demml/item element to remain relatively simple. It also allows software developers to create a parent or interface class from which child classes can be extended or implemented. This way any of these content types can be stored in the same container if necessary.</xs:documentation> </xs:annotation> </xs:complexType>

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

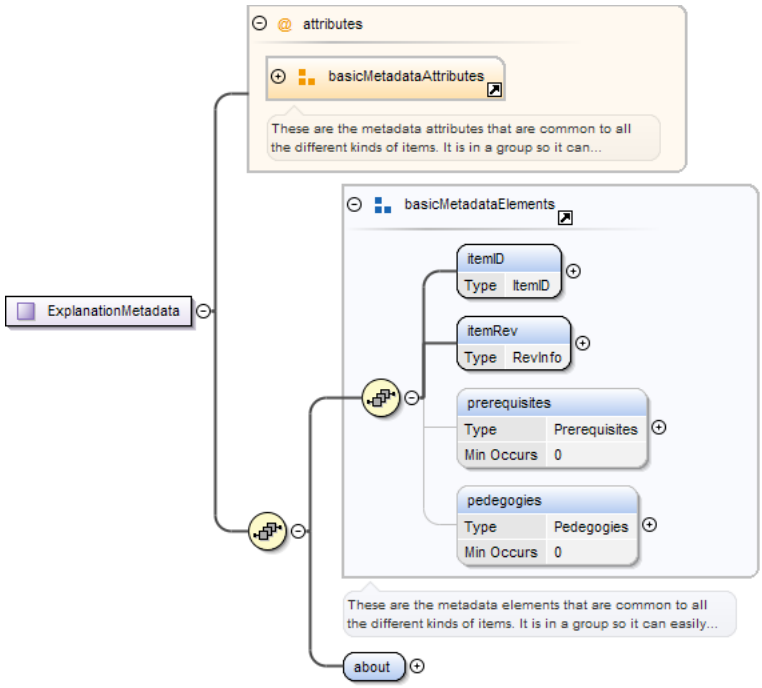
Complex Type `ExplanationItem`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	extension of Item
Type hierarchy	• Item • ExplanationItem
Model	explanationMetadata , content
Children	content, explanationMetadata

`Source`	`<xs:complexType name="ExplanationItem">`
	`  <xs:complexContent>`
	`  <xs:extension base="Item">`
	`  <xs:sequence>`
	`  <xs:element name="explanationMetadata" type="ExplanationMetadata"/>`
	`  <xs:element name="content" type="BasicContent"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Complex Type `ExplanationMetadata`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Used by	Element ExplanationItem/explanationMetadata
Model	itemID , itemRev , prerequisites{0,1} , pedegogies{0,1} , about
Children	about, itemID, itemRev, pedegogies, prerequisites

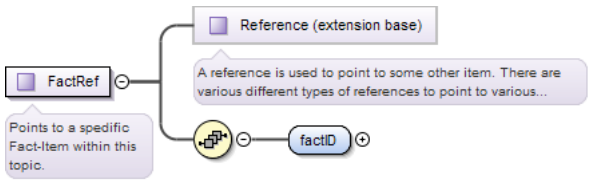
Attributes	**QName**	**Type**	**Fixed**	**Default**	**Use**
	aETM	AETM			optional
	difficulty	Percent			optional

Source

```
<xs:complexType name="ExplanationMetadata">
  <xs:sequence>
    <xs:group ref="basicMetadataElements"/>
    <xs:element name="about">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:group ref="standardRefs"/>
          <xs:element name="qaPairRef" type="QApairRef"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="basicMetadataAttributes"/>
</xs:complexType>
```

Schema location: file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

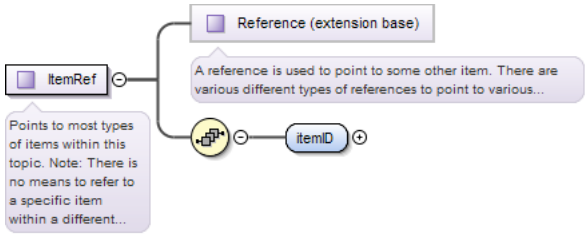
Complex Type `FactRef`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Points to a spedific Fact-Item within this topic.
Diagram	
Type	extension of Reference
Type hierarchy	• Reference • FactRef
Used by	Element standardRefs/factRef
Model	factID
Children	factID

Source	<pre><xs:complexType name="FactRef">
	  <xs:annotation>
	  <xs:documentation>Points to a spedific Fact-Item within this topic.</xs:documentation>
	  </xs:annotation>
	  <xs:complexContent>
	  <xs:extension base="Reference">
	  <xs:sequence>
	  <xs:element name="factID" type="FactID"/>
	  </xs:sequence>
	  </xs:extension>
	  </xs:complexContent>
	</xs:complexType></pre>

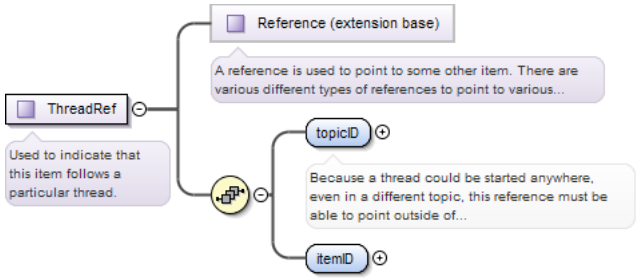
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `ItemRef`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Points to most types of items within this topic. Note: There is no means to refer to a specific item within a different topic.
Diagram	
Type	extension of Reference

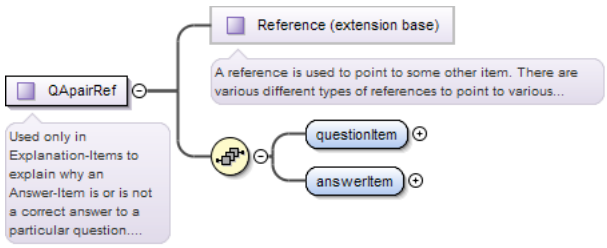
Type hierarchy	• Reference • ItemRef
Used by	Element standardRefs/itemRef
Model	itemID
Children	itemID
Source	<xs:complexType name="ItemRef"> <xs:annotation> <xs:documentation>Points to most types of items within this topic. Note: There is no means to refer to a specific item within a different topic.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="Reference"> <xs:sequence> <xs:element name="itemID" type="ItemID"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type ThreadRef

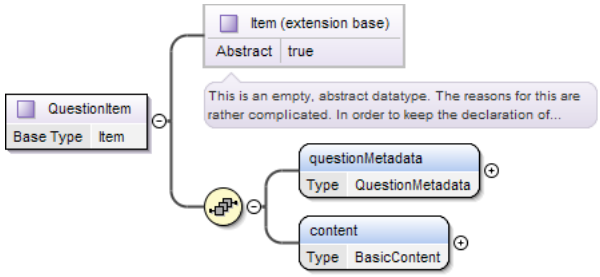
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Used to indicate that this item follows a particular thread.
Diagram	
Type	extension of Reference
Type hierarchy	• Reference • ThreadRef
Used by	Element standardRefs/threadRef
Model	topicID , itemID
Children	itemID, topicID
Source	<xs:complexType name="ThreadRef"> <xs:annotation> <xs:documentation>Used to indicate that this item follows a particular thread.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="Reference"> <xs:sequence> <xs:element name="topicID" type="TopicID"> <xs:annotation> <xs:documentation>Because a thread could be started anywhere, even in a different topic, this reference must be able to point outside of the topic where it is used.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="itemID" type="ItemID"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type QApairRef

Namespace	http://www.demml.org/schemas/alpha/0.1/demml

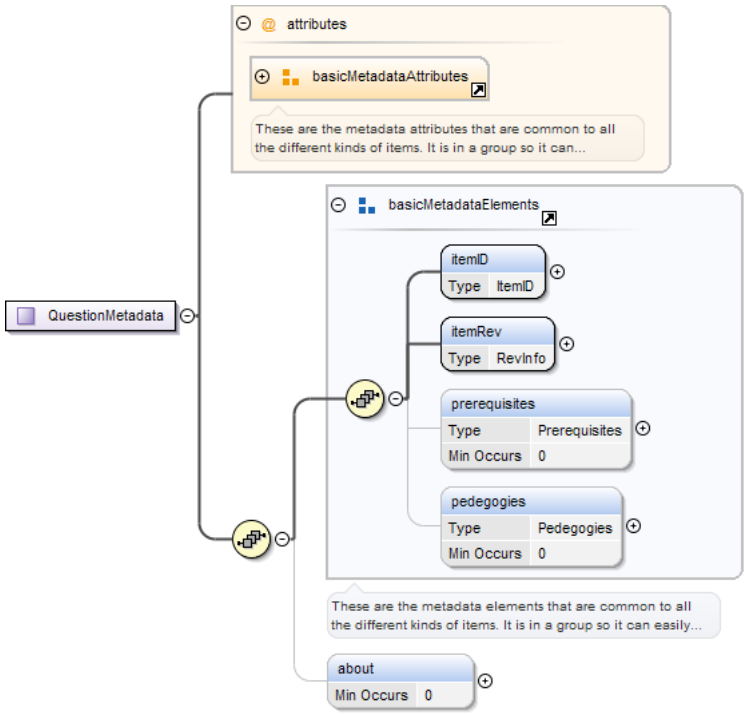
Annotations	Used only in Explanation-Items to explain why an Answer-Item is or is not a correct answer to a particular question. Because questions and answers can be worded many different ways, while still meaning the same thing, some Explanation-Items may point to more than one question-answer pairs.
Diagram	
Type	extension of Reference
Type hierarchy	• Reference • QApairRef
Used by	Element ExplanationMetadata/about/qaPairRef
Model	questionItem , answerItem
Children	answerItem, questionItem
Source	<xs:complexType name="QApairRef"> <xs:annotation> <xs:documentation>Used only in Explanation-Items to explain why an Answer-Item is or is not a correct answer to a particular question. Because questions and answers can be worded many different ways, while still meaning the same thing, some Explanation-Items may point to more than one question-answer pairs.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="Reference"> <xs:sequence> <xs:element name="questionItem" type="ItemID"/> <xs:element name="answerItem" type="ItemID"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `QuestionItem`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	extension of Item
Type hierarchy	• Item • QuestionItem
Model	questionMetadata , content
Children	content, questionMetadata
Source	<xs:complexType name="QuestionItem"> <xs:complexContent> <xs:extension base="Item"> <xs:sequence> <xs:element name="questionMetadata" type="QuestionMetadata"/> <xs:element name="content" type="BasicContent"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

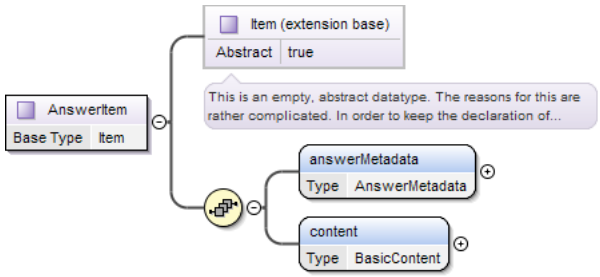
Complex Type `QuestionMetadata`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Used by	Element QuestionItem/questionMetadata
Model	itemID , itemRev , prerequisites{0,1} , pedegogies{0,1} , about{0,1}
Children	about, itemID, itemRev, pedegogies, prerequisites
Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **aETM** / AETM / / / optional **difficulty** / Percent / / / optional

Source	`<xs:complexType name="QuestionMetadata">`
	`  <xs:sequence>`
	`  <xs:group ref="basicMetadataElements"/>`
	`  <xs:element name="about" minOccurs="0">`
	`  <xs:complexType>`
	`  <xs:group ref="standardRefs" maxOccurs="unbounded"/>`
	`  </xs:complexType>`
	`  </xs:element>`
	`  </xs:sequence>`
	`  <xs:attributeGroup ref="basicMetadataAttributes"/>`
	`</xs:complexType>`

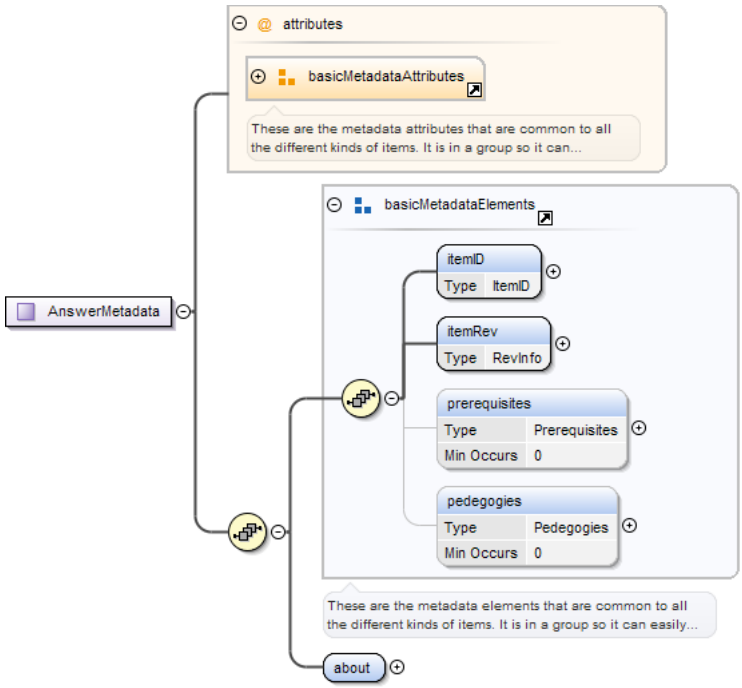
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Complex Type `AnswerItem`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	extension of Item

Type hierarchy	• Item • AnswerItem
Model	answerMetadata , content
Children	answerMetadata, content
Source	<xs:complexType name="AnswerItem"> <xs:complexContent> <xs:extension base="Item"> <xs:sequence> <xs:element name="answerMetadata" type="AnswerMetadata"/> <xs:element name="content" type="BasicContent"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Complex Type AnswerMetadata

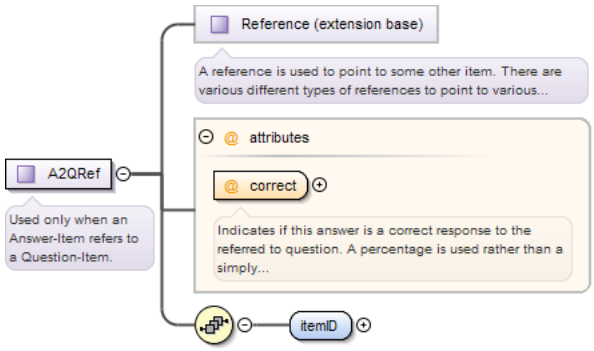
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Used by	Element AnswerItem/answerMetadata
Model	itemID , itemRev , prerequisites{0,1} , pedegogies{0,1} , about
Children	about, itemID, itemRev, pedegogies, prerequisites

Attributes

QName	Type	Fixed	Default	Use
aETM	AETM			optional
difficulty	Percent			optional

Source	<xs:complexType name="AnswerMetadata"> <xs:sequence> <xs:group ref="basicMetadataElements"/> <xs:element name="about"> <xs:complexType> <xs:choice maxOccurs="unbounded"> <xs:element name="a2QRef" type="A2QRef"/> </xs:choice> </xs:complexType> </xs:element> </xs:sequence> <xs:attributeGroup ref="basicMetadataAttributes"/> </xs:complexType>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Complex Type A2QRef

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Used only when an Answer-Item refers to a Question-Item.
Diagram	
Type	extension of Reference
Type hierarchy	• Reference • A2QRef
Used by	Element AnswerMetadata/about/a2QRef
Model	itemID
Children	itemID

Attributes

QName	Type	Fixed	Default	Use
correct	Percent			required
	Indicates if this answer is a correct response to the referred to question. A percentage is used rather than a simply boolean because some answers are worth partial credit. Note: it is not necessary to create a reference from every Answer-Item to every Question-Item indicating the degree of correctness. Instead, only create these references when the answer is a plausible answer to the referred to question or is a good distractor for that question. This way software can automatically create quizes that are different each time.			

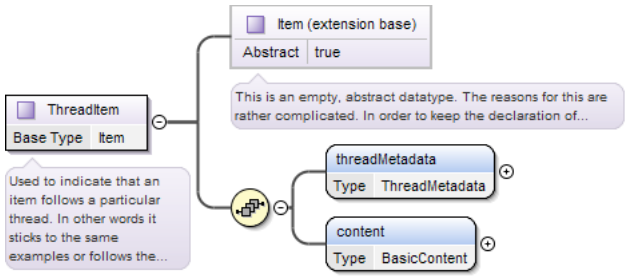
Source

```
<xs:complexType name="A2QRef">
  <xs:annotation>
    <xs:documentation>Used only when an Answer-Item refers to a Question-Item.</
xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Reference">
      <xs:sequence>
        <xs:element name="itemID" type="ItemID"/>
      </xs:sequence>
      <xs:attribute name="correct" type="Percent" use="required">
        <xs:annotation>
          <xs:documentation>Indicates if this answer is a correct response to the referred
 to question. A percentage is used rather than a simply boolean because some answers are
 worth partial credit. Note: it is not necessary to create a reference from every Answer-
Item to every Question-Item indicating the degree of correctness. Instead, only create
 these references when the answer is a plausible answer to the referred to question or is
 a good distractor for that question. This way software can automatically create quizes
 that are different each time.</xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type ThreadItem

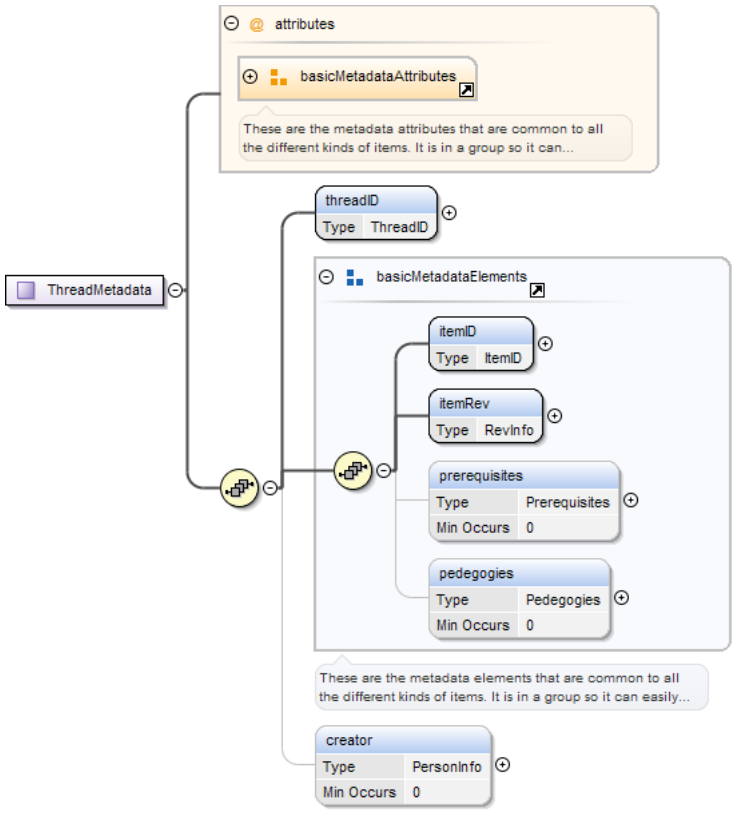
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Used to indicate that an item follows a particular thread. In other words it sticks to the same examples or follows the same storyline of other items. Threads are used to provide continuity to the learning experience. Threads should always be placed in the farthest out branch of the tree which still includes all or most of the items that refer to it.

Diagram	
Type	extension of Item
Type hierarchy	• Item • ThreadItem
Model	threadMetadata , content
Children	content, threadMetadata

Source	`<xs:complexType name="ThreadItem">`
	`  <xs:annotation>`
	`  <xs:documentation>Used to indicate that an item follows a particular thread. In other words it sticks to the same examples or follows the same storyline of other items. Threads are used to provide continuity to the learning experience.</xs:documentation>`
	`  <xs:documentation>Threads should always be placed in the farthest out branch of the tree which still includes all or most of the items that refer to it.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="Item">`
	`  <xs:sequence>`
	`  <xs:element name="threadMetadata" type="ThreadMetadata"/>`
	`  <xs:element name="content" type="BasicContent"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Complex Type `ThreadMetadata`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	

Used by	Element ThreadItem/threadMetadata
Model	threadID , itemID , itemRev , prerequisites{0,1} , pedegogies{0,1} , creator{0,1}
Children	creator, itemID, itemRev, pedegogies, prerequisites, threadID

Attributes

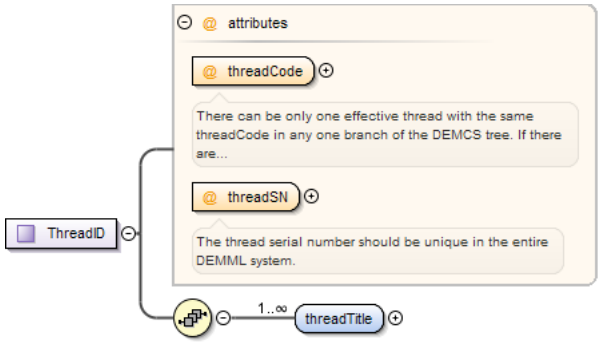
QName	Type	Fixed	Default	Use
aETM	AETM			optional
difficulty	Percent			optional

Source

```
<xs:complexType name="ThreadMetadata">
  <xs:sequence>
    <xs:element name="threadID" type="ThreadID"/>
    <xs:group ref="basicMetadataElements"/>
    <xs:element name="creator" minOccurs="0" type="PersonInfo"/>
  </xs:sequence>
  <xs:attributeGroup ref="basicMetadataAttributes"/>
</xs:complexType>
```

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Complex Type `ThreadID`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	

Used by	Element ThreadMetadata/threadID
Model	threadTitle+
Children	threadTitle

Attributes

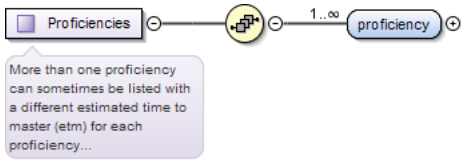
QName	Type	Fixed	Default	Use
threadCode	xs:token			required
	There can be only one effective thread with the same threadCode in any one branch of the DEMCS tree. If there are threads with the same threadCode in more than one level of the same branch of the tree, the thread in the branch farthest from the root of the tree takes precedence. This allows a type of inheritance in that a thread can be more general closer to the root of the tree but then be modified to be more specific to the topics in the branches farther out from the root. There is no formal mechanism for controling inheritance. Whichever thread is referenced is the thread that will be applied to the referencing item.			
threadSN	SerialNumber			required
	The thread serial number should be unique in the entire DEMML system.			

Source

```
<xs:complexType name="ThreadID">
  <xs:sequence>
    <xs:element name="threadTitle" type="Title" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="threadCode" type="xs:token" use="required">
    <xs:annotation>
      <xs:documentation>There can be only one effective thread with the same threadCode in
 any one branch of the DEMCS tree. If there are threads with the same threadCode in more
 than one level of the same branch of the tree, the thread in the branch farthest from the
 root of the tree takes precedence. This allows a type of inheritance in that a thread can
 be more general closer to the root of the tree but then be modified to be more specific
 to the topics in the branches farther out from the root. There is no formal mechanism for
 controling inheritance. Whichever thread is referenced is the thread that will be applied
 to the referencing item.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="threadSN" type="SerialNumber" use="required">
```

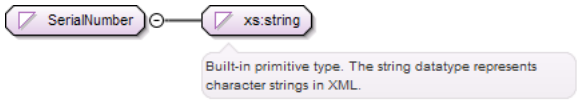
	<xs:annotation> <xs:documentation>The thread serial number should be unique in the entire DEMML system.</xs:documentation> </xs:annotation> </xs:attribute> </xs:complexType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Complex Type `Proficiencies`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	More than one proficiency can sometimes be listed with a different estimated time to master (etm) for each proficiency level. There is no need to list more than a few different proviciencies as software will normally be able to extrapolate the etm values for various proficiency levels. Multiple proficiencies should be used when a subject, topic, or item has a much steeper "learning curve" for the more advanced material than one would normally expect.
Diagram	
Model	proficiency+
Children	proficiency
Source	<xs:complexType name="Proficiencies"> <xs:annotation> <xs:documentation>More than one proficiency can sometimes be listed with a different estimated time to master (etm) for each proficiency level. There is no need to list more than a few different proviciencies as software will normally be able to extrapolate the etm values for various proficiency levels. Multiple proficiencies should be used when a subject, topic, or item has a much steeper "learning curve" for the more advanced material than one would normally expect.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="proficiency" maxOccurs="unbounded" type="Proficiency"/> </xs:sequence> </xs:complexType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

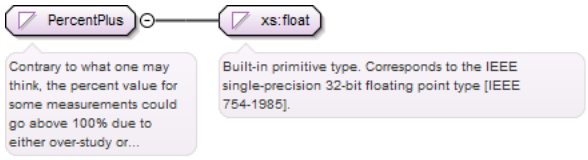
Simple Types

Simple Type `SerialNumber`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Diagram	
Type	restriction of xs:string
Facets	pattern [0-9A-Z]+
Used by	Attributes ItemID/@itemNumber, ThreadID/@threadSN, TopicID/@topicSN
Source	<xs:simpleType name="SerialNumber"> <xs:restriction base="xs:string"> <xs:pattern value="[0-9A-Z]+"/> </xs:restriction> </xs:simpleType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Simple Type `PercentPlus`

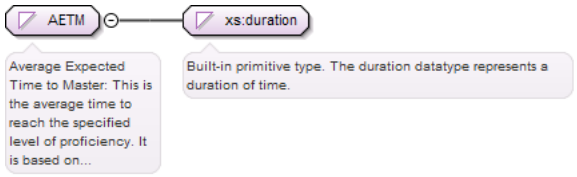
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Contrary to what one may think, the percent value for some measurements could go above 100% due to either over-study or learning more sub-topics with more proficiency than required by a DEMML syllabus.

Diagram	
Type	restriction of xs:float
Facets	minInclusive `0.00`
Used by	Attribute: Proficiency/@percent Simple Type: Percent

Source	`<xs:simpleType name="PercentPlus">`
	`  <xs:annotation>`
	`  <xs:documentation>Contrary to what one may think, the percent value for some measurements could go above 100% due to either over-study or learning more sub-topics with more proficiency than required by a DEMML syllabus.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:float">`
	`  <xs:minInclusive value="0.00"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

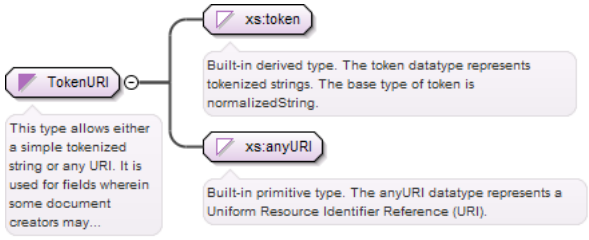
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Simple Type `AETM`

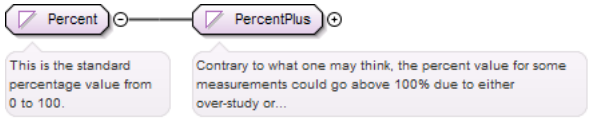
Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	Average Expected Time to Master: This is the average time to reach the specified level of proficiency. It is based on the assumption that the student exactly meets all of the prerequisites for this item or topic but has not studied any of the item, topic, or its sub-topics yet. The duration should be set to a reasonable value. If aETM is not given then software is expected to ignore the etm for this item or rather than attempt to force the student to learn the item in zero seconds.
Diagram	
Type	restriction of xs:duration
Facets	minInclusive `PT0M` — Naturally, the aETM shouldn't be less than zero minutes. pattern `PT\d+M`
Used by	Attributes: Proficiency/@aETM, basicMetadataAttributes/@aETM
Source	`<xs:simpleType name="AETM">` `  <xs:annotation>` ` <xs:documentation>Average Expected Time to Master: This is the average time to reach the specified level of proficiency. It is based on the assumption that the student exactly meets all of the prerequisites for this item or topic but has not studied any of the item, topic, or its sub-topics yet.</xs:documentation>` ` <xs:documentation>The duration should be set to a reasonable value. If aETM is not given then software is expected to ignore the etm for this item or rather than attempt to force the student to learn the item in zero seconds.</xs:documentation>` ` </xs:annotation>` ` <xs:restriction base="xs:duration">` ` <!--<xs:pattern value="P\d+Y\d+M\d+DT\d+H\d+M0S"/>-->` ` <!--<xs:pattern value="PT\p{Nd}+M"/>-->` ` <xs:pattern value="PT\d+M">` ` <xs:annotation>` ` <xs:documentation>This pattern should limit the aETM to being expressed in minutes only.</xs:documentation>` ` </xs:annotation>` ` </xs:pattern>` ` <xs:minInclusive value="PT0M">` ` <xs:annotation>` ` <xs:documentation>Naturally, the aETM shouldn't be less than zero minutes.</xs:documentation>` ` </xs:annotation>` ` </xs:minInclusive>`

	</xs:restriction> </xs:simpleType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Simple Type `TokenURI`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	This type allows either a simple tokenized string or any URI. It is used for fields wherein some document creators may want to just just a standardized value and others may want to use a URI to provide more specific information.
Diagram	
Type	union of(xs:token, xs:anyURI)
Used by	Attributes: Pedegogies/pedegogy/@Classification, Pedegogies/pedegogy/@domain, Pedegogies/pedegogy/@model
Source	<xs:simpleType name="TokenURI"> <xs:annotation> <xs:documentation>This type allows either a simple tokenized string or any URI. It is used for fields wherein some document creators may want to just just a standardized value and others may want to use a URI to provide more specific information.</xs:documentation> </xs:annotation> <xs:union memberTypes="xs:token xs:anyURI"/> </xs:simpleType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

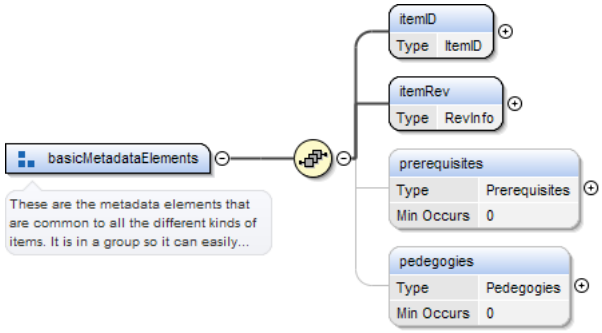
Simple Type `Percent`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	This is the standard percentage value from 0 to 100.
Diagram	
Type	restriction of PercentPlus
Type hierarchy	• xs:float • PercentPlus • Percent
Facets	maxInclusive: 100 minInclusive: 0.00
Used by	Attributes: A2QRef/@correct, Pedegogies/pedegogy/@percent, basicMetadataAttributes/@difficulty
Source	<xs:simpleType name="Percent"> <xs:annotation> <xs:documentation>This is the standard percentage value from 0 to 100.</ xs:documentation> </xs:annotation> <xs:restriction base="PercentPlus"> <xs:maxInclusive value="100"/> </xs:restriction> </xs:simpleType>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

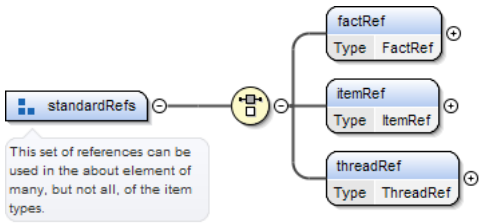
Element Groups

Element Group `basicMetadataElements`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml

Annotations	These are the metadata elements that are common to all the different kinds of items. It is in a group so it can easily be inserted into any of the metadata types.
Diagram	
Used by	Complex Types AnswerMetadata, ExplanationMetadata, FactMetadata, QuestionMetadata, ThreadMetadata
Model	itemID , itemRev , prerequisites{0,1} , pedegogies{0,1}
Children	itemID, itemRev, pedegogies, prerequisites
Source	<xs:group name="basicMetadataElements"> <xs:annotation> <xs:documentation>These are the metadata elements that are common to all the different kinds of items. It is in a group so it can easily be inserted into any of the metadata types.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="itemID" type="ItemID"/> <xs:element name="itemRev" type="RevInfo"/> <xs:element name="prerequisites" type="Prerequisites" minOccurs="0"/> <xs:element name="pedegogies" type="Pedegogies" minOccurs="0"/> </xs:sequence> </xs:group>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

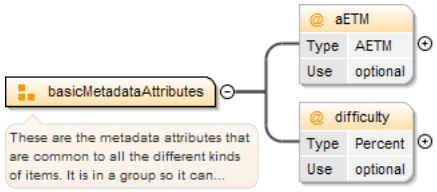
Element Group `standardRefs`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Annotations	This set of references can be used in the about element of many, but not all, of the item types.
Diagram	
Used by	Elements ExplanationMetadata/about, QuestionMetadata/about
Model	factRef | itemRef | threadRef
Children	factRef, itemRef, threadRef
Source	<xs:group name="standardRefs"> <xs:annotation> <xs:documentation>This set of references can be used in the about element of many, but not all, of the item types.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="factRef" type="FactRef"/> <xs:element name="itemRef" type="ItemRef"/> <xs:element name="threadRef" type="ThreadRef"/> </xs:choice> </xs:group>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Attribute Groups

Attribute Group `basicMetadataAttributes`

Namespace	http://www.demml.org/schemas/alpha/0.1/demml

Annotations	These are the metadata attributes that are common to all the different kinds of items. It is in a group so it can easily be inserted into any of the metadata types.
Diagram	
Used by	Complex Types AnswerMetadata, ExplanationMetadata, FactMetadata, QuestionMetadata, ThreadMetadata

Attributes

QName	Type	Fixed	Default	Use
aETM	AETM			optional
difficulty	Percent			optional

Source

```
<xs:attributeGroup name="basicMetadataAttributes">
  <xs:annotation>
    <xs:documentation>These are the metadata attributes that are common to all the
 different kinds of items. It is in a group so it can easily be inserted into any of the
 metadata types.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="aETM" type="AETM"/>
  <xs:attribute name="difficulty" type="Percent"/>
</xs:attributeGroup>
```

Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Namespace: "http://www.w3.org/XML/1998/namespace"

Schemas

Imported schema `xml.xsd`

Namespace	http://www.w3.org/XML/1998/namespace
Annotations	See http://www.w3.org/XML/1998/namespace.html and http://www.w3.org/TR/REC-xml for information about this namespace. This schema document describes the XML namespace, in a form suitable for import by other schema documents. Note that local names in this namespace are intended to be defined only by the World Wide Web Consortium or its subgroups. The following names are currently defined in this namespace and should not be used with conflicting semantics by any Working Group, specification, or document instance: base (as an attribute name): denotes an attribute whose value provides a URI to be used as the base for interpreting any relative URIs in the scope of the element on which it appears; its value is inherited. This name is reserved by virtue of its definition in the XML Base specification. id (as an attribute name): denotes an attribute whose value should be interpreted as if declared to be of type ID. This name is reserved by virtue of its definition in the xml:id specification. lang (as an attribute name): denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification. space (as an attribute name): denotes an attribute whose value is a keyword indicating what whitespace processing discipline is intended for the content of the element; its value is inherited. This name is reserved by virtue of its definition in the XML specification. Father (in any context at all): denotes Jon Bosak, the chair of the original XML Working Group. This name is reserved by the following decision of the W3C XML Plenary and XML Coordination groups: In appreciation for his vision, leadership and dedication the W3C XML Plenary on this 10th day of February, 2000 reserves for Jon Bosak in perpetuity the XML name

	xml:Father This schema defines attributes and an attribute group suitable for use by schemas wishing to allow xml:base, xml:lang, xml:space or xml:id attributes on elements they define. To enable this, such a schema must import this schema for the XML namespace, e.g. as follows: <schema . . .> . . . <import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="http://www.w3.org/2001/xml.xsd"/> Subsequently, qualified reference to any of the attributes or the group defined below will have the desired effect, e.g. <type . . .> . . . <attributeGroup ref="xml:specialAttrs"/> will define a type which will schema-validate an instance element with any of those attributes In keeping with the XML Schema WG's standard versioning policy, this schema document will persist at http://www.w3.org/2007/08/xml.xsd. At the date of issue it can also be found at http://www.w3.org/2001/xml.xsd. The schema document at that URI may however change in the future, in order to remain compatible with the latest version of XML Schema itself, or with the XML namespace itself. In other words, if the XML Schema or XML namespaces change, the version of this document at http://www.w3.org/2001/xml.xsd will change accordingly; the version at http://www.w3.org/2007/08/xml.xsd will not change.
Properties	attribute form default: unqualified element form default: unqualified
Schema location	file:/C:/Program Files/Oxygen XML Editor 11/frameworks/xml/xml.xsd

Attributes

Attribute @xml:lang

Namespace	http://www.w3.org/XML/1998/namespace
Annotations	Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.
Type	union of(xs:language, restriction of xs:string)
Properties	content: simple
Used by	Complex Type: Title Attribute Group: xml:specialAttrs
Source	<xs:attribute name="lang"> <xs:annotation> <xs:documentation>Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.</xs:documentation> </xs:annotation> <xs:simpleType> <xs:union memberTypes="xs:language"> <xs:simpleType> <xs:restriction base="xs:string">

```
          <xs:enumeration value=""/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>
</xs:attribute>
```

Schema location	file:/C:/Program Files/Oxygen XML Editor 11/frameworks/xml/xml.xsd

Attribute `@xml:space`

Namespace	http://www.w3.org/XML/1998/namespace	
Type	restriction of xs:NCName	
Properties	content:	simple
Facets	enumeration	default
	enumeration	preserve
Used by	Attribute Group	xml:specialAttrs

Source

```
<xs:attribute name="space">
  <xs:simpleType>
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="default"/>
      <xs:enumeration value="preserve"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

Schema location	file:/C:/Program Files/Oxygen XML Editor 11/frameworks/xml/xml.xsd

Attribute `@xml:base`

Namespace	http://www.w3.org/XML/1998/namespace	
Annotations	See http://www.w3.org/TR/xmlbase/ for information about this attribute.	
Type	xs:anyURI	
Properties	content:	simple
Used by	Attribute Group	xml:specialAttrs

Source

```
<xs:attribute name="base" type="xs:anyURI">
  <xs:annotation>
    <xs:documentation>See http://www.w3.org/TR/xmlbase/ for
                      information about this attribute.</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Schema location	file:/C:/Program Files/Oxygen XML Editor 11/frameworks/xml/xml.xsd

Attribute `@xml:id`

Namespace	http://www.w3.org/XML/1998/namespace	
Annotations	See http://www.w3.org/TR/xml-id/ for information about this attribute.	
Type	xs:ID	
Properties	content:	simple
Used by	Attribute Group	xml:specialAttrs

Source

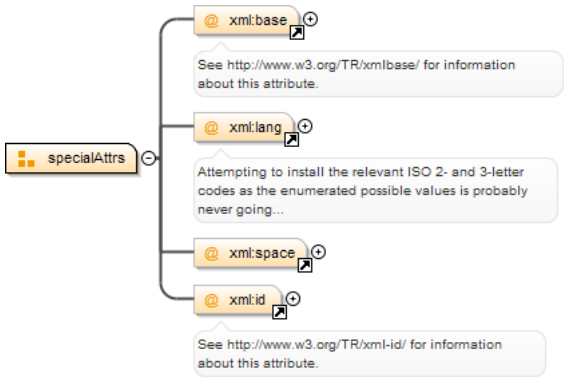
```
<xs:attribute name="id" type="xs:ID">
  <xs:annotation>
    <xs:documentation>See http://www.w3.org/TR/xml-id/ for
                      information about this attribute.</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Schema location	file:/C:/Program Files/Oxygen XML Editor 11/frameworks/xml/xml.xsd

Attribute Groups

Attribute Group `xml:specialAttrs`

Namespace	http://www.w3.org/XML/1998/namespace

Diagram	

Attributes

QName	Type	Fixed	Default	Use
xml:base	xs:anyURI			optional
	See http://www.w3.org/TR/xmlbase/ for information about this attribute.			
xml:id	xs:ID			optional
	See http://www.w3.org/TR/xml-id/ for information about this attribute.			
xml:lang	union of(xs:language, restriction of xs:string)			optional
	Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.			
xml:space	restriction of xs:NCName			optional

Source

```
<xs:attributeGroup name="specialAttrs">
  <xs:attribute ref="xml:base"/>
  <xs:attribute ref="xml:lang"/>
  <xs:attribute ref="xml:space"/>
  <xs:attribute ref="xml:id"/>
</xs:attributeGroup>
```

Schema location: file:/C:/Program Files/Oxygen XML Editor 11/frameworks/xml/xml.xsd

Namespace: ""

Attributes

Attribute `TopicID / @topicCode`

Namespace	No namespace
Annotations	This is the Distributable Educational Material Classification System (DEMCS) classification code. This code includes the full path to the topic within the DEMCS. See http://demml.org/standard/classification/for more information. Do not include the demml:// protocol prefix in the value for this attribute.
Type	xs:token
Properties	use: required
Used by	Complex Type TopicID

Source

```
<xs:attribute name="topicCode" type="xs:token" use="required">
  <xs:annotation>
    <xs:documentation>
      <p>This is the Distributable Educational Material Classification System (DEMCS) classification code. This code includes the full path to the topic within the DEMCS. See <a href="http://demml.org/standard/classification/">http://demml.org/standard/classification/</a>for more information.</p>
```

	Do not include the demml:// protocol prefix in the value for this attribute. </xs:documentation> </xs:annotation> </xs:attribute>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Attribute TopicID / @topicSN

Namespace	No namespace
Annotations	The Topic Serial Number must be unique among all the topics within the entire DEMML system. Serial Numbers are assigned sequentially as each topic is added the DEMCS. In case some topics need to be rearranged at some time in the future (hopefully very rarely) the serial numbers can be used by software to locate the correct topics and update their locations within individual users' databases of DEMML content.
Type	SerialNumber
Properties	use: required
Facets	pattern [0-9A-Z]+
Used by	Complex Type TopicID
Source	<xs:attribute name="topicSN" type="SerialNumber" use="required"> <xs:annotation> <xs:documentation>The Topic Serial Number must be unique among all the topics within the entire DEMML system. Serial Numbers are assigned sequentially as each topic is added the DEMCS. In case some topics need to be rearranged at some time in the future (hopefully very rarely) the serial numbers can be used by software to locate the correct topics and update their locations within individual users' databases of DEMML content.</xs:documentation> </xs:annotation> </xs:attribute>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Attribute contentRevInfo / RevInfo / @contentVersion

Namespace	No namespace
Type	xs:token
Properties	use: required
Used by	Element RevInfo/contentRevInfo
Source	<xs:attribute name="contentVersion" type="xs:token" use="required"/>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Attribute contentRevInfo / RevInfo / @contentModDate

Namespace	No namespace
Type	xs:date
Properties	use: required
Used by	Element RevInfo/contentRevInfo
Source	<xs:attribute name="contentModDate" type="xs:date" use="required"/>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Attribute dataRevInfo / RevInfo / @dataVersion

Namespace	No namespace
Type	xs:token
Properties	use: required
Used by	Element RevInfo/dataRevInfo
Source	<xs:attribute name="dataVersion" type="xs:token" use="required"/>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Attribute `dataRevInfo / RevInfo / @dataModDate`

Namespace	No namespace
Type	xs:date
Properties	use: `required`
Used by	Element RevInfo/dataRevInfo
Source	`<xs:attribute name="dataModDate" type="xs:date" use="required"/>`
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Attribute `RevInfo / @createDate`

Namespace	No namespace
Annotations	`Date the file was first created. Once the file is created, this date should not be changed.`
Type	xs:date
Properties	use: `required`
Used by	Complex Type RevInfo

Source	`<xs:attribute name="createDate" type="xs:date" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation>Date the file was first created. Once the file is created, this date should not be changed.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Attribute `FactID / @factCode`

Namespace	No namespace
Annotations	`Fact codes consist of one or more digits or capital letters suffixed by zero or one lower-case letter. The first letter is always a capital 'F' to make them easier to recognize when written down. In order to allow additional facts to be inserted into an ordered list of preexisting facts, fact codes follow the same rules and pattern as the DEMCS coding system (see http://demml.org/standard/classification/4-assign_codes.htm). In some cases one level of sub-facts will be appropriate. In which case the sub-facts will have the code of the parent fact followed directly by one lower-case letter. For instance, N7b or 3La. If at any time one feels it necessary to create a third level of facts then serious consideration should be given to creating sub-topics instead, putting each of the primary parent facts in its own topic. Sometimes the order of facts will not matter. But for some topics where the order the facts are listed helps provide meaning to these facts then make sure the facts are numbered in an appropriate order.`  `Fact F0 is the statement of the main point of the topic.`
Type	restriction of xs:string
Properties	use: `required`
Facets	pattern `F[0-9A-Z]+[a-z]*`
Used by	Complex Type FactID
Source	`<xs:attribute name="factCode" use="required">` `  <xs:annotation>` ` <xs:documentation>Fact codes consist of one or more digits or capital letters suffixed by zero or one lower-case letter. The first letter is always a capital 'F' to make them easier to recognize when written down. In order to allow additional facts to be inserted into an ordered list of preexisting facts, fact codes follow the same rules and pattern as the DEMCS coding system (see http://demml.org/standard/classification/4-assign_codes.htm). In some cases one level of sub-facts will be appropriate. In which case the sub-facts will have the code of the parent fact followed directly by one lower-case letter. For instance, N7b or 3La. If at any time one feels it necessary to create a third level of facts then serious consideration should be given to creating sub-topics instead, putting each of the primary parent facts in its own topic. Sometimes the order of facts will not matter. But for some topics where the order the facts are listed helps provide meaning to these facts then make sure the facts are numbered in an appropriate order.</xs:documentation>` ` <xs:documentation>Fact F0 is the statement of the main point of the topic.</xs:documentation>` ` </xs:annotation>` ` <xs:simpleType>` ` <xs:restriction base="xs:string">`

	<xs:pattern value="F[0-9A-Z]+[a-z]*"/> </xs:restriction> </xs:simpleType> </xs:attribute>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Attribute `ItemID / @itemNumber`

Namespace	No namespace	
Annotations	This is simply a sequentialy assigned code number comprised of digits and capital letters. It must be unique among all the other items in the same topic folder.	
Type	SerialNumber	
Properties	use:	required
Facets	pattern	[0-9A-Z]+
Used by	Complex Type	ItemID
Source	<xs:attribute name="itemNumber" type="SerialNumber" use="required"> <xs:annotation> <xs:documentation>This is simply a sequentialy assigned code number comprised of digits and capital letters. It must be unique among all the other items in the same topic folder.</xs:documentation> </xs:annotation> </xs:attribute>	
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd	

Attribute `ItemID / @itemType`

Namespace	No namespace	
Annotations	This attribute is for informational purposes only, rather than to define a datatype. It is included so when an itemID is used in a reference, software or human readers can tell what type of item is being referenced without retrieving that actual item.	
Type	restriction of xs:string	
Properties	content:	simple
Facets	enumeration	fact
	enumeration	explanation
	enumeration	question
	enumeration	answer
	enumeration	thread
Used by	Complex Type	ItemID
Source	<xs:attribute name="itemType"> <xs:annotation> <xs:documentation>This attribute is for informational purposes only, rather than to define a datatype. It is included so when an itemID is used in a reference, software or human readers can tell what type of item is being referenced without retrieving that actual item.</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="fact"/> <xs:enumeration value="explanation"/> <xs:enumeration value="question"/> <xs:enumeration value="answer"/> <xs:enumeration value="thread"/> <!--<xs:enumeration value="problem"/>--> <!--<xs:enumeration value="solution"/>--> <!--<xs:enumeration value="definition"/>--> <!--<xs:enumeration value="exercise"/>--> <!--<xs:enumeration value="exerciseExternal"/>--> <!--<xs:enumeration value="lab"/>--> <!--<xs:enumeration value="project"/>--> <!--<xs:enumeration value="identification"/>--> <!--<xs:enumeration value="demmlLink"/>--> <!--<xs:enumeration value="externalLink"/>--> </xs:restriction> </xs:simpleType> </xs:attribute>	
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd	

Attribute `Proficiency / @percent`

Namespace	No namespace	
Type	PercentPlus	
Properties	use:	`required`
Facets	minInclusive	`0.00`
Used by	Complex Type	Proficiency
`Source`	`<xs:attribute name="percent" type="PercentPlus" use="required"/>`	
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd	

Attribute `Proficiency / @aETM`

Namespace	No namespace	
Type	AETM	
Properties	content:	`simple`
Facets	minInclusive	`PT0M` `Naturally, the aETM shouldn't be less than zero minutes.`
	pattern	`PT\d+M`
Used by	Complex Type	Proficiency
`Source`	`<xs:attribute name="aETM" type="AETM"/>`	
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd	

Attribute `pedegogy / Pedegogies / @model`

Namespace	No namespace	
Type	TokenURI	
Properties	use:	`required`
Used by	Element	Pedegogies/pedegogy
`Source`	`<xs:attribute name="model" type="TokenURI" use="required"/>`	
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd	

Attribute `pedegogy / Pedegogies / @domain`

Namespace	No namespace	
Type	TokenURI	
Properties	content:	`simple`
Used by	Element	Pedegogies/pedegogy
`Source`	`<xs:attribute name="domain" type="TokenURI"/>`	
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd	

Attribute `pedegogy / Pedegogies / @Classification`

Namespace	No namespace	
Type	TokenURI	
Properties	use:	`required`
Used by	Element	Pedegogies/pedegogy
`Source`	`<xs:attribute name="Classification" type="TokenURI" use="required"/>`	
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd	

Attribute `pedegogy / Pedegogies / @percent`

Namespace	No namespace

Type	Percent

Type hierarchy	• xs:float
	  • PercentPlus
	  • Percent

Properties	use: required
Facets	maxInclusive 100 minInclusive 0.00
Used by	Element Pedegogies/pedegogy
Source	<xs:attribute name="percent" type="Percent" use="required"/>
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

Attribute basicMetadataAttributes / @aETM

Namespace	No namespace
Type	AETM
Properties	content: simple
Facets	minInclusive PT0M Naturally, the aETM shouldn't be less than zero minutes. pattern PT\d+M
Used by	Attribute Group basicMetadataAttributes
Source	<xs:attribute name="aETM" type="AETM"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Attribute basicMetadataAttributes / @difficulty

Namespace	No namespace
Type	Percent

Type hierarchy	• xs:float
	  • PercentPlus
	  • Percent

Properties	content: simple
Facets	maxInclusive 100 minInclusive 0.00
Used by	Attribute Group basicMetadataAttributes
Source	<xs:attribute name="difficulty" type="Percent"/>
Schema location	file:/D:/My Notebooks/Projects/DEMML/XML Projects/DEMML_0.1/DEMML_0.1.xsd

Attribute A2QRef / @correct

Namespace	No namespace
Annotations	Indicates if this answer is a correct response to the referred to question. A percentage is used rather than a simply boolean because some answers are worth partial credit. Note: it is not necessary to create a reference from every Answer-Item to every Question-Item indicating the degree of correctness. Instead, only create these references when the answer is a plausible answer to the referred to question or is a good distractor for that question. This way software can automatically create quizes that are different each time.
Type	Percent

Type hierarchy	• xs:float
	  • PercentPlus
	  • Percent

Properties	use: required

Facets	maxInclusive	100
	minInclusive	0.00
Used by	Complex Type	A2QRef
Source	<xs:attribute name="correct" type="Percent" use="required"> <xs:annotation> <xs:documentation>Indicates if this answer is a correct response to the referred to question. A percentage is used rather than a simply boolean because some answers are worth partial credit. Note: it is not necessary to create a reference from every Answer-Item to every Question-Item indicating the degree of correctness. Instead, only create these references when the answer is a plausible answer to the referred to question or is a good distractor for that question. This way software can automatically create quizes that are different each time.</xs:documentation> </xs:annotation> </xs:attribute>	
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd	

Attribute ThreadID / @threadCode

Namespace	No namespace	
Annotations	There can be only one effective thread with the same threadCode in any one branch of the DEMCS tree. If there are threads with the same threadCode in more than one level of the same branch of the tree, the thread in the branch farthest from the root of the tree takes precedence. This allows a type of inheritance in that a thread can be more general closer to the root of the tree but then be modified to be more specific to the topics in the branches farther out from the root. There is no formal mechanism for controling inheritance. Whichever thread is referenced is the thread that will be applied to the referencing item.	
Type	xs:token	
Properties	use:	required
Used by	Complex Type	ThreadID
Source	<xs:attribute name="threadCode" type="xs:token" use="required"> <xs:annotation> <xs:documentation>There can be only one effective thread with the same threadCode in any one branch of the DEMCS tree. If there are threads with the same threadCode in more than one level of the same branch of the tree, the thread in the branch farthest from the root of the tree takes precedence. This allows a type of inheritance in that a thread can be more general closer to the root of the tree but then be modified to be more specific to the topics in the branches farther out from the root. There is no formal mechanism for controling inheritance. Whichever thread is referenced is the thread that will be applied to the referencing item.</xs:documentation> </xs:annotation> </xs:attribute>	
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd	

Attribute ThreadID / @threadSN

Namespace	No namespace	
Annotations	The thread serial number should be unique in the entire DEMML system.	
Type	SerialNumber	
Properties	use:	required
Facets	pattern	[0-9A-Z]+
Used by	Complex Type	ThreadID
Source	<xs:attribute name="threadSN" type="SerialNumber" use="required"> <xs:annotation> <xs:documentation>The thread serial number should be unique in the entire DEMML system.</xs:documentation> </xs:annotation> </xs:attribute>	
Schema location	http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd	

schema

Target Namespace	http://www.demml.org/schemas/alpha/0.1/demml
Element Form Default	qualified

include: http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd

import: http://www.w3.org/XML/1998/namespace (http://www.w3.org/2001/xml.xsd)

Necessary to enable use of the xml:lang attribute.

FileInfo

Every DEMML content file will contain a fileInfo element at the top to indicate information about the file as a...

fileRevInfo

When the RevInfo type is used for the fileRevInfo element then that info pertains to the file as a whole. If any...

RevInfo

Reflects the date and version for this file or item. If this is the first version then the version should be 1.0 and...

1..∞ creator

This is the person who created or compiled the file. Not necessarily who created every item within the file.

PersonInfo

personName

xs:token

Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.

ToDo: Look up or create a more detailed PersonInfo type to use here.

TopicID

The topic ID is used to uniquely identify a DEMML topic. It is used both to lable specific topics as well as to refer...

attributes

@ topicCode — xs:token

This is the Distributable Educational Material Classification System (DEMCS) classification code. This code includes...

Built-in derived type. The token datatype represents tokenized strings. The base type of token is normalizedString.

@ topicSN — SerialNumber

The Topic Serial Number must be unique among all the topics within the entire DEMML system. Serial Numbers are assigned...

1..∞ topicTitle

When used to label a topic within the Topic Info file there must be one topic title for each of the language-specifi c...

Title

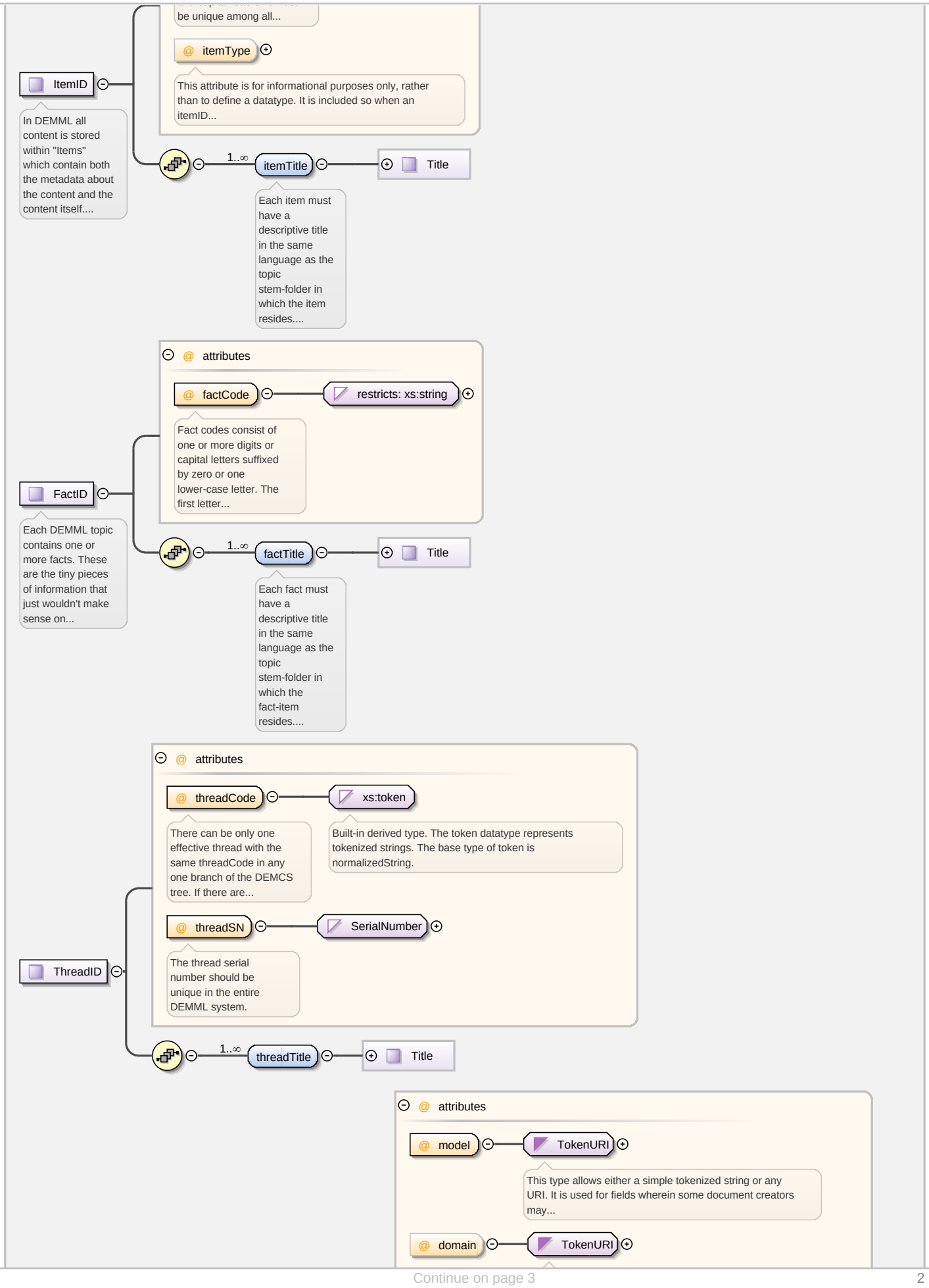
attributes

@ itemNumber — SerialNumber

This is simply a sequentialy assigned code number comprised of digits and capital letters. It must

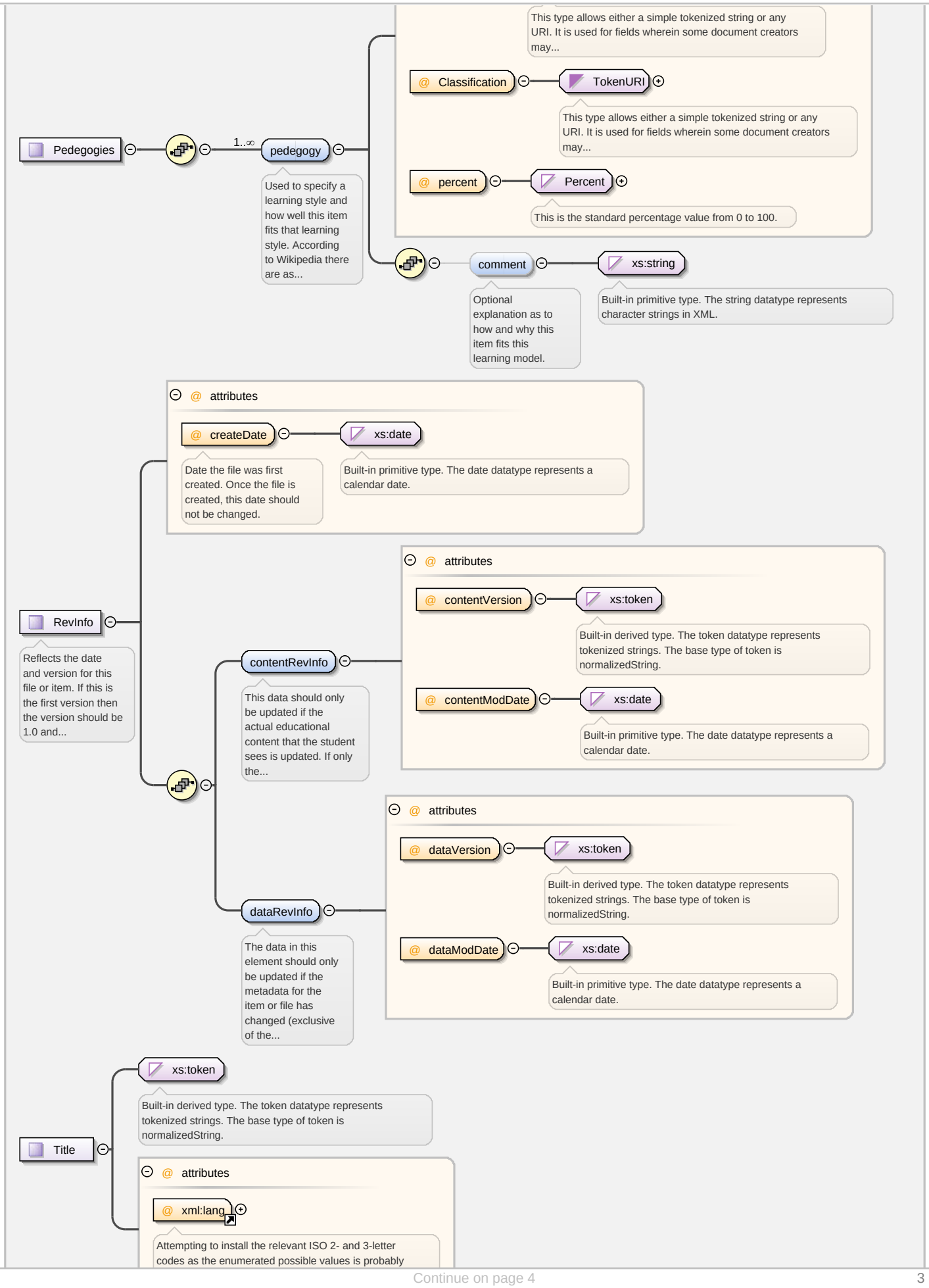
Continue on page 2

Continue on page 1



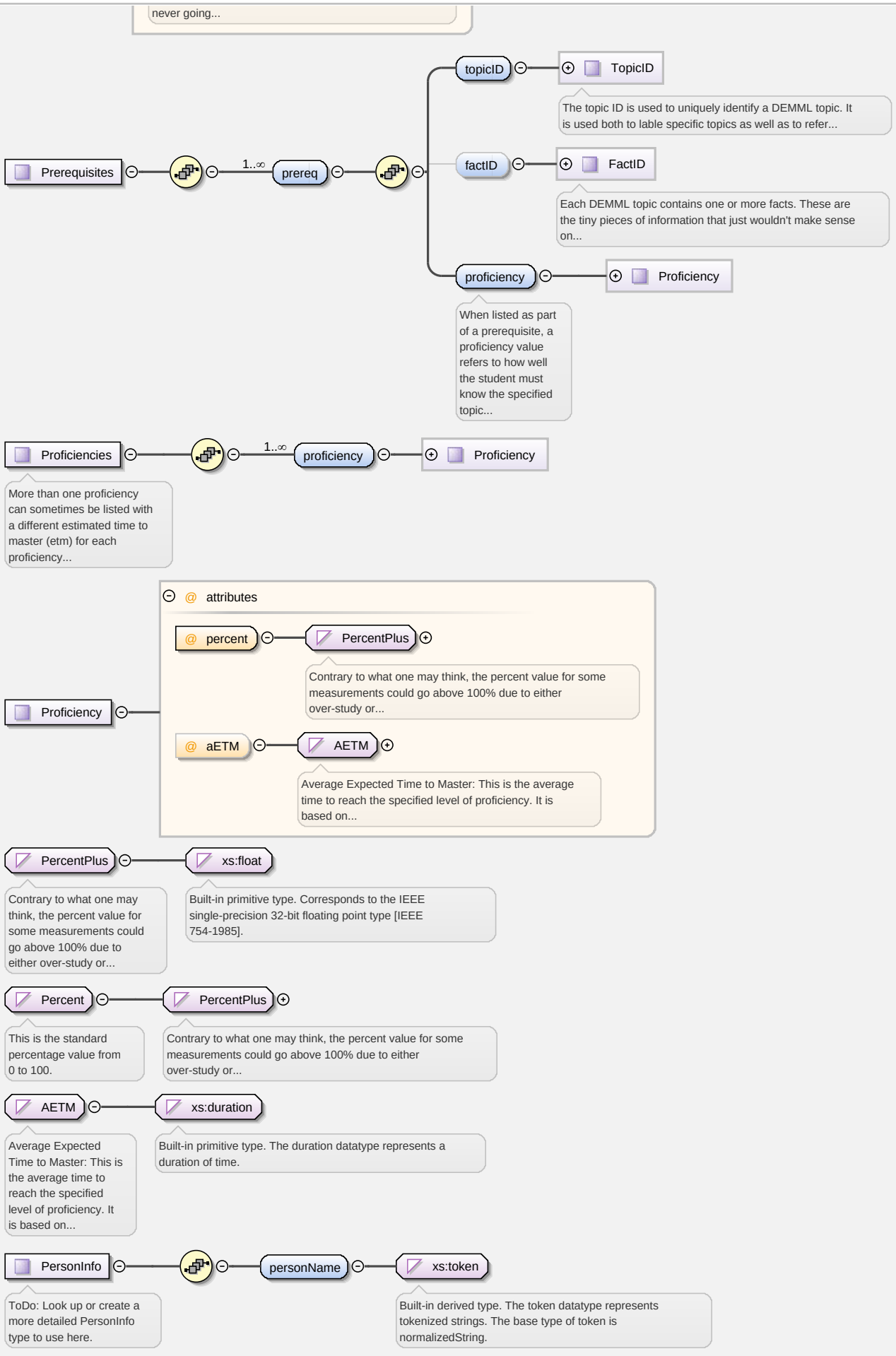
Continue on page 3

Continue on page 2



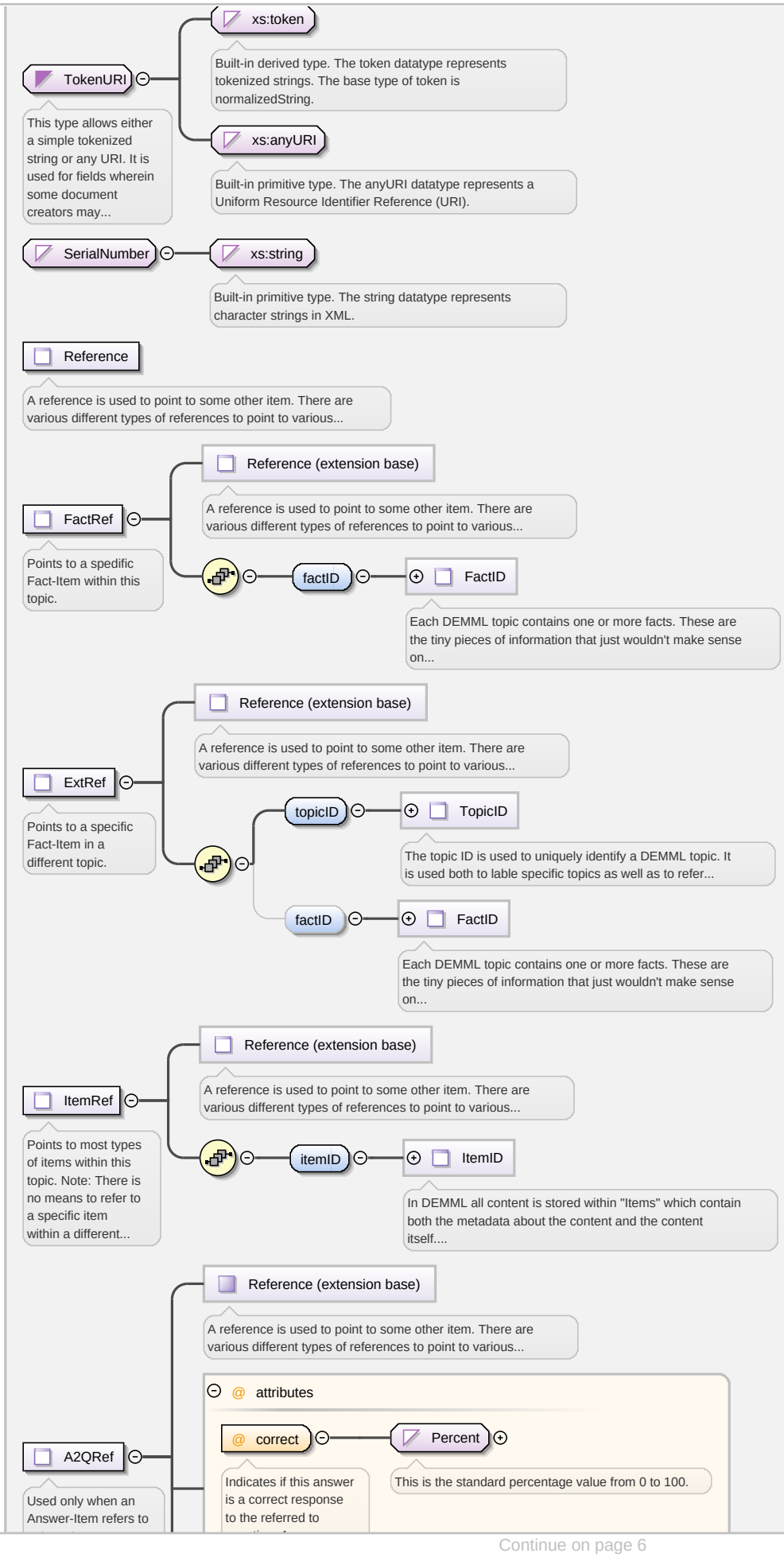
Continue on page 4

Continue on page 3



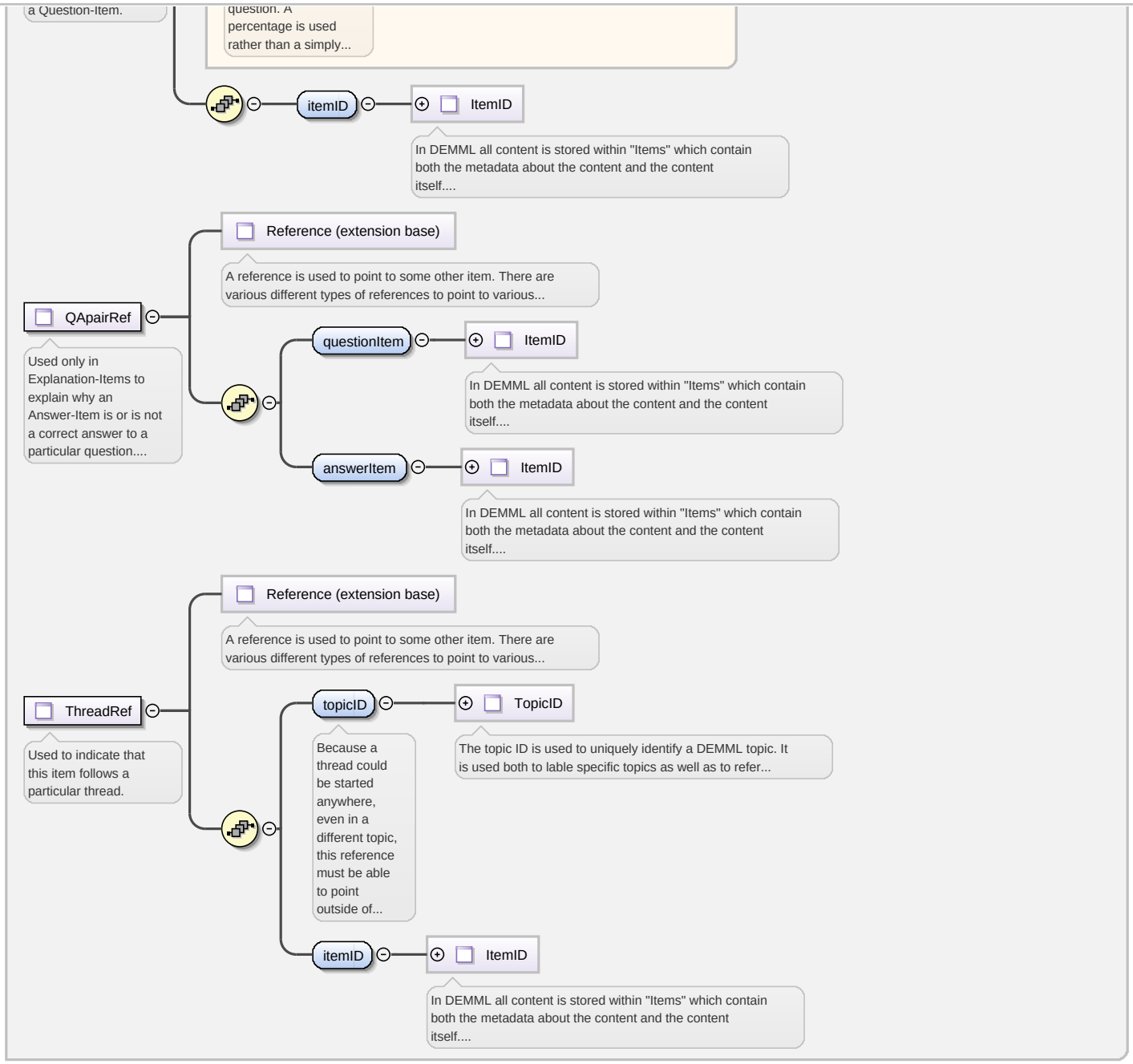
Continue on page 5

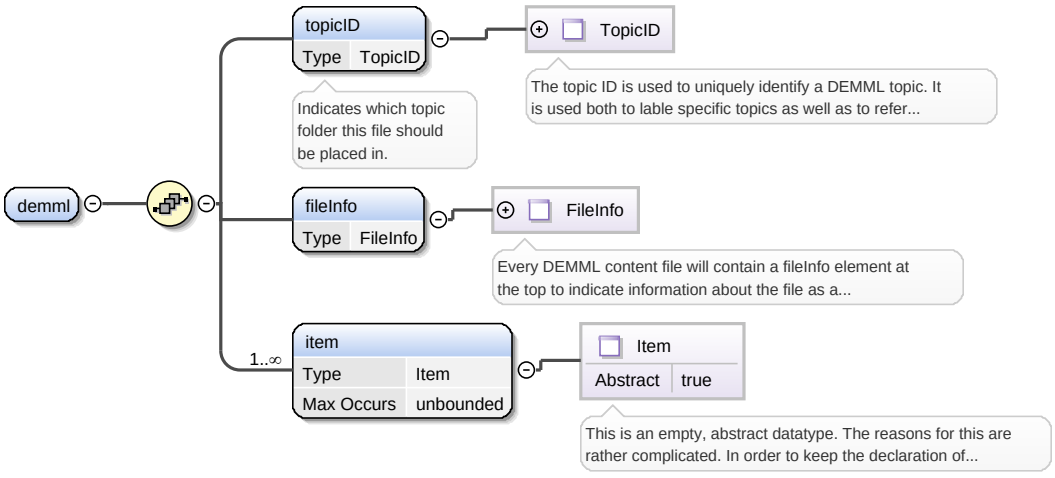
Continue on page 4

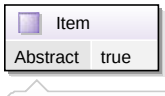


Continue on page 6

Continue on page 5

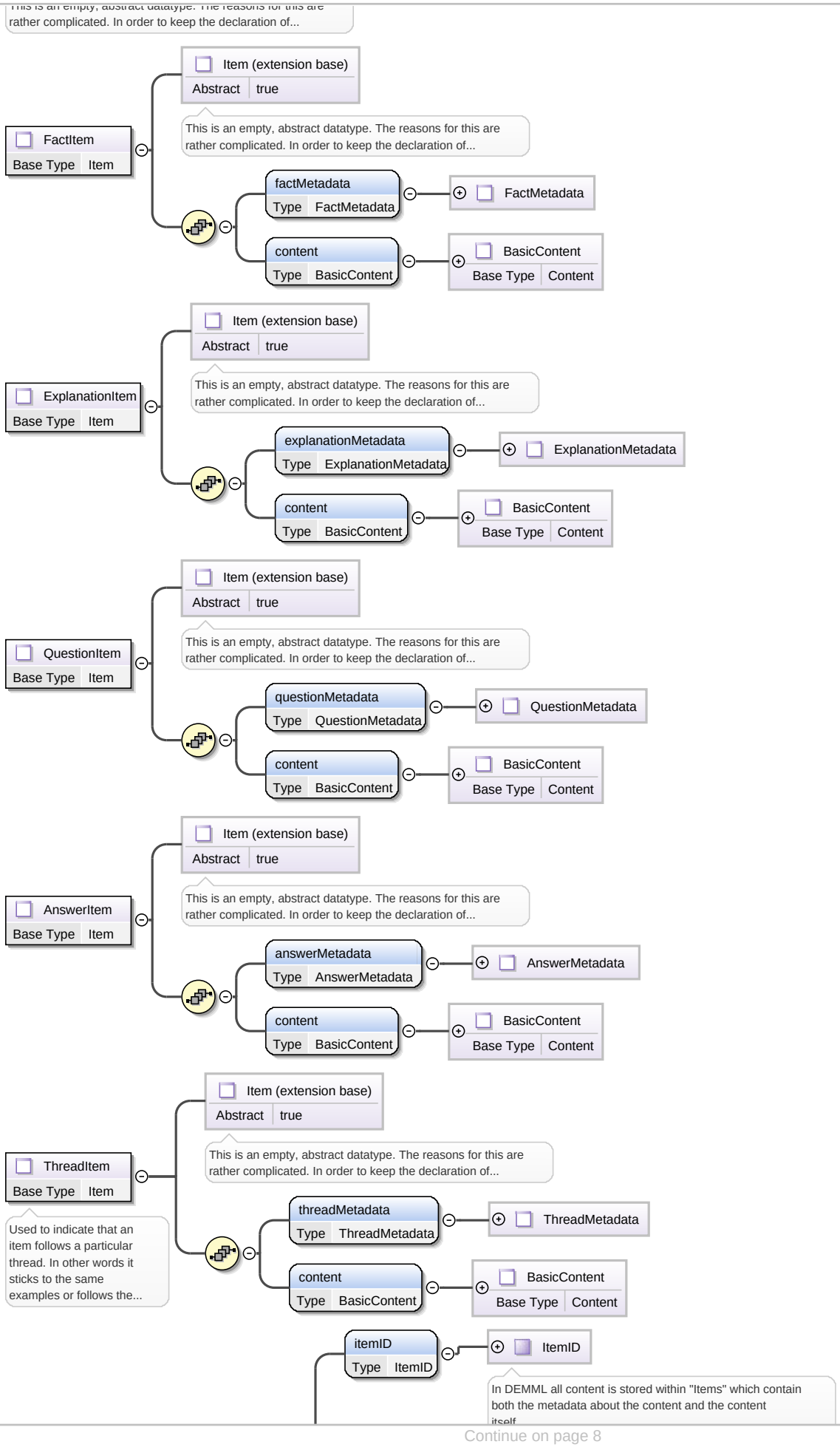






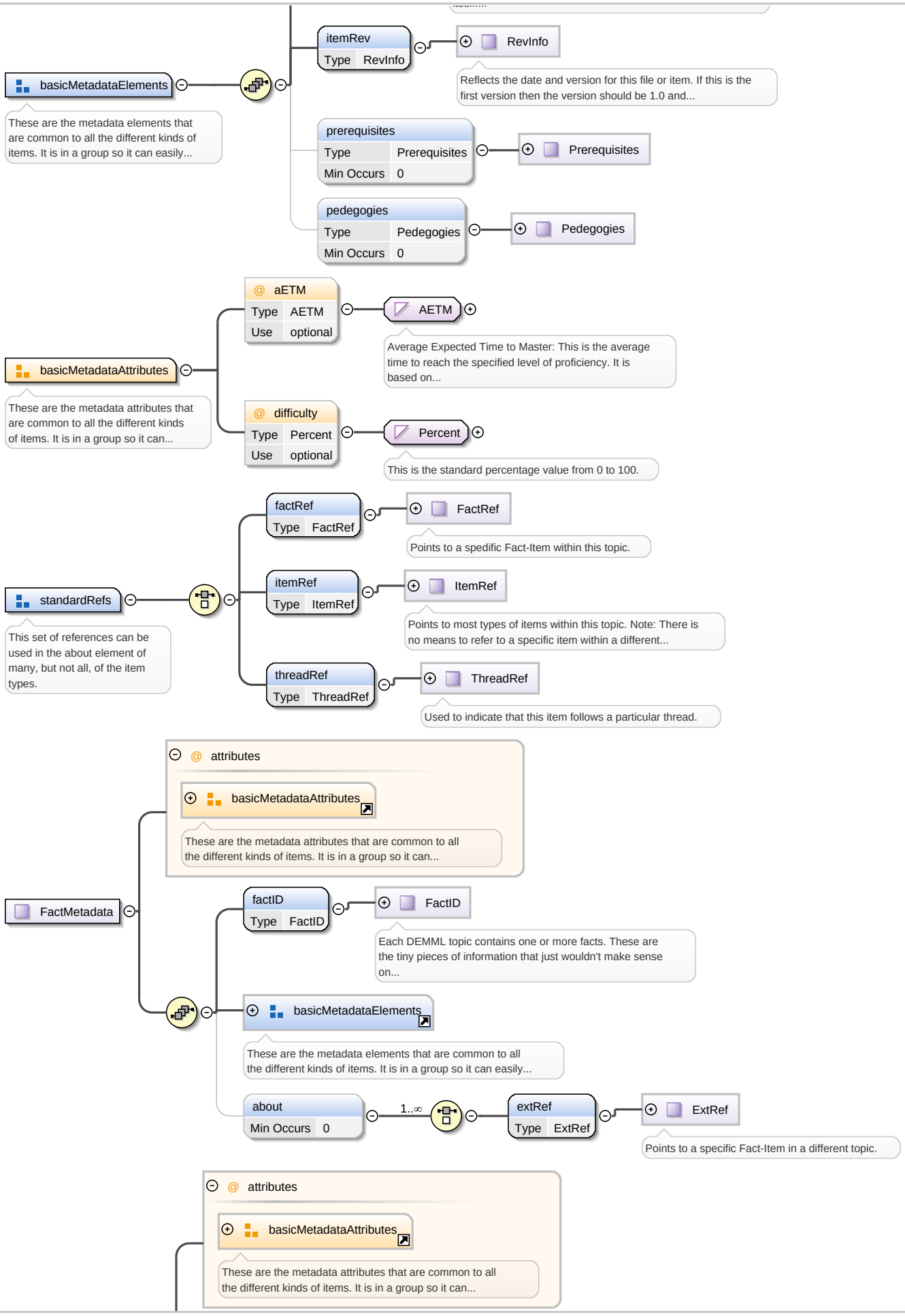
Continue on page 7

Continue on page 6



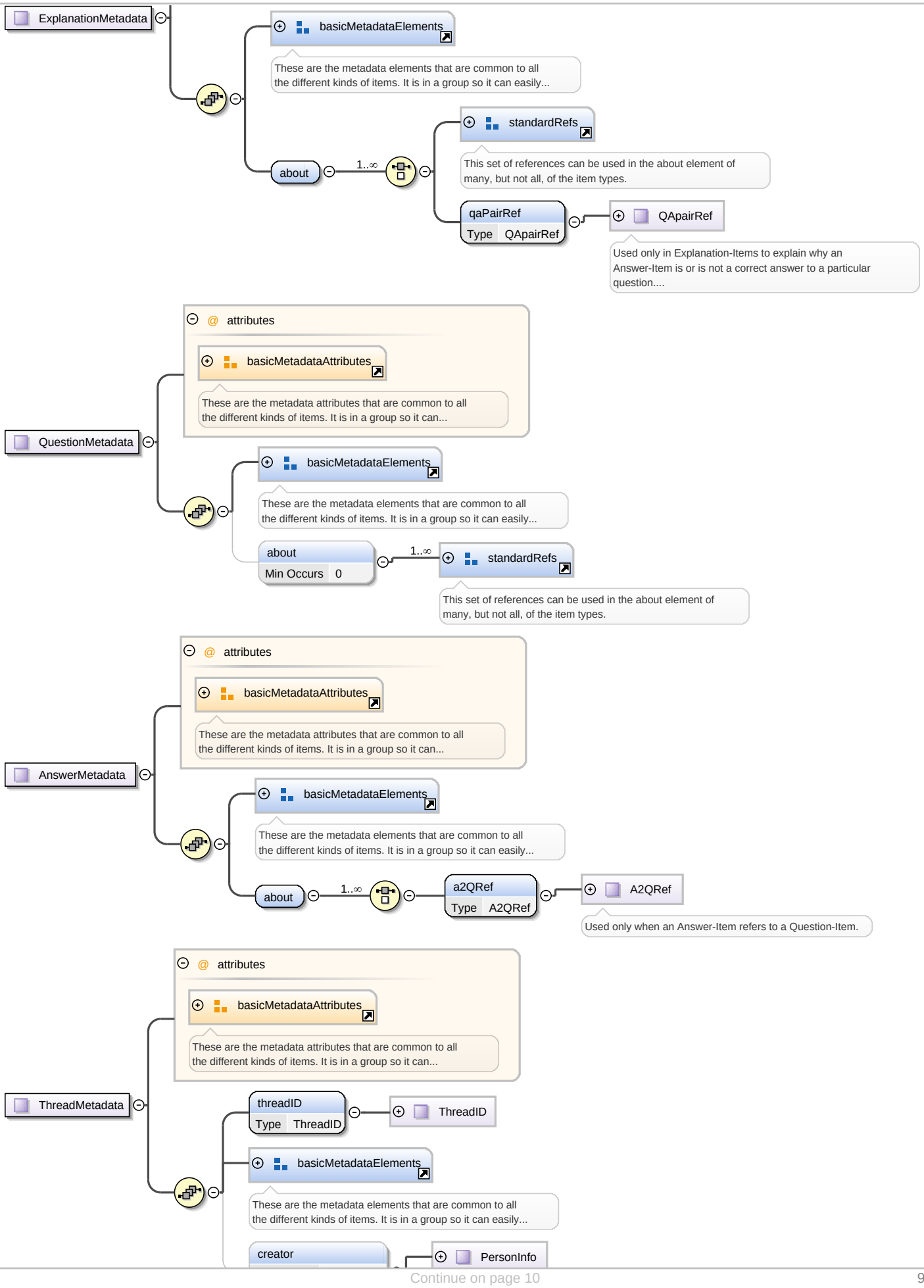
Continue on page 8

Continue on page 7



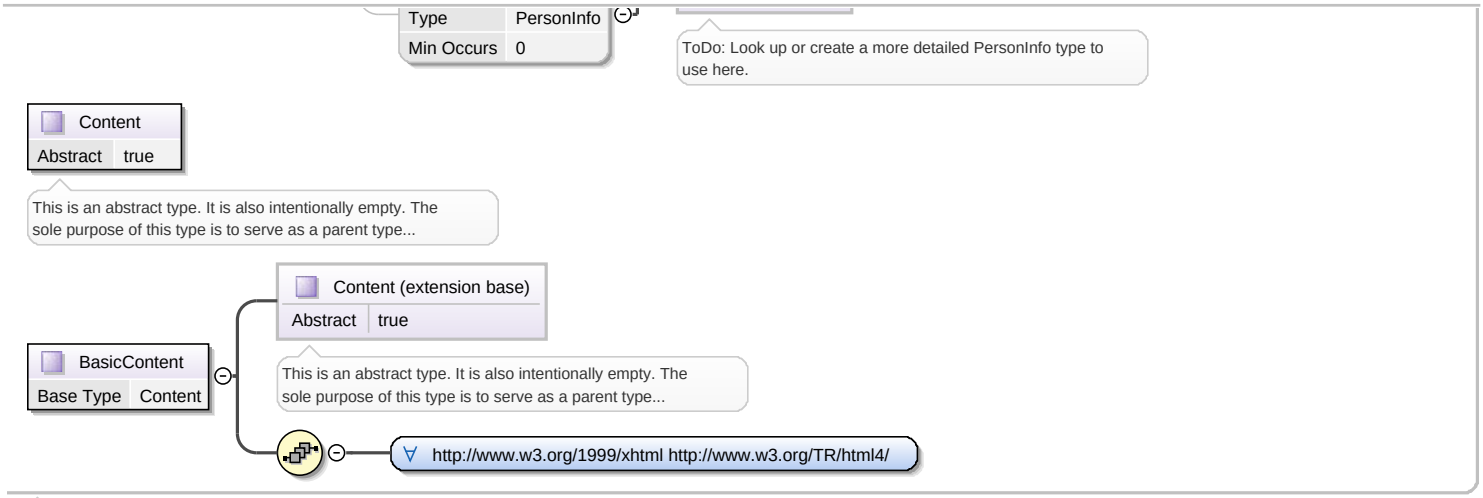
Continue on page 9

Continue on page 8



Continue on page 10

Continue on page 9



This is the first alpha version of the DEMML(tm) schema for content (version 0.1). This schema is far from complete and...

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema  version="0.1" xml:lang="en"
    xmlns="http://www.demml.org/schemas/alpha/0.1/demml"
    targetNamespace="http://www.demml.org/schemas/alpha/0.1/demml"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">

    <xs:include schemaLocation="http://www.demml.org/schemas/alpha/0.1/common/DEMMLcommon.xsd"/>

    <xs:annotation>
        <xs:documentation>
            <p>This is the first alpha version of the DEMML(tm) schema for content (version 0.1). This schema
is far from complete and the structure may need to be modified considerably in order to incorporate all the
features of the DEMML(tm). While I plan to make future versions of this schema a semi-open standard in the
future, the entire schema is currently copyright (c) 2010 by Grant Sheridan Robertson. It is currently not
permitted to use this schema - or any  portions thereof - in any manner without written permission. However,
if you write, I will probably give permision. I simply want to keep track of what is being done with
DEMML(tm) so I can keep a handle on it till it is ready for release. Thank you for your cooperation.
            </p>
            <p>To learn more about this DEMML(tm) go to www.demml.org. To learn more about the schema itself
(past,  present,&amp;amp; future)  as well as a general description of how the DEMML system is organized see
www.demml.org/standard/.</p>
        </xs:documentation>
    </xs:annotation>

    <!--   Root  Element-->
    <xs:element name="demml">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="topicID" type="TopicID">
                    <xs:annotation>
                        <xs:documentation>
                           Indicates which topic folder this file should be placed in.
                        </xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="fileInfo" type="FileInfo"/>
                <xs:element name="item" type="Item" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <!--   Item  Types-->
    <xs:complexType name="Item" abstract="true">
        <xs:annotation>
            <xs:documentation>
               This is an empty, abstract datatype. The reasons for this are rather complicated. In order to
keep the declaration of the demml element simple, even though there will be many different types of items, it
was necessary for demml/item element to use an abstract datatype allowing type substitution of the child
datatypes in the place of this element. In order to keep the structure of the item elements consistent it is
best to keep the metadata on top and the content last. Therefore, it is also best to keep the metadata and
content in their own separate elements, allowing each of their datatypes to also be abstracted and extended.
While itemMetadata and content elements could have been placed here to then reference those abstract
datatypes, allowing their replacement with concrete datatypes, this would not have ensured that the correct
sets of metadata were always paired with the correct type of content. It is also the case that not all items
will actually contain content. Others may have more complicated content than can be expressed by a simple
```

element with some HTML in it. Therefore, it is necessary to declare each different item type as an extension of this type with a metadata element and content element designed just for that item type.

```
            </xs:documentation>
        </xs:annotation>
    </xs:complexType>

    <xs:complexType name="FactItem">
        <xs:complexContent>
            <xs:extension base="Item">
                <xs:sequence>
                    <xs:element name="factMetadata" type="FactMetadata"/>
                    <xs:element name="content" type="BasicContent"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ExplanationItem">
        <xs:complexContent>
            <xs:extension base="Item">
                <xs:sequence>
                    <xs:element name="explanationMetadata" type="ExplanationMetadata"/>
                    <xs:element name="content" type="BasicContent"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="QuestionItem">
        <xs:complexContent>
            <xs:extension base="Item">
                <xs:sequence>
                    <xs:element name="questionMetadata" type="QuestionMetadata"/>
                    <xs:element name="content" type="BasicContent"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AnswerItem">
        <xs:complexContent>
            <xs:extension base="Item">
                <xs:sequence>
                    <xs:element name="answerMetadata" type="AnswerMetadata"/>
                    <xs:element name="content" type="BasicContent"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ThreadItem">
        <xs:annotation>
            <xs:documentation>
               Used to indicate that an item follows a particular thread. In other words it sticks to the same examples or follows the same storyline of other items. Threads are used to provide continuity to the learning experience.
```

```
            </xs:documentation>
            <xs:documentation>
                Threads should always be placed in the farthest out branch of the tree which still includes
all or most of the items that refer to it.
            </xs:documentation>
        </xs:annotation>
        <xs:complexContent>
            <xs:extension base="Item">
                <xs:sequence>
                    <xs:element name="threadMetadata" type="ThreadMetadata"/>
                    <xs:element name="content" type="BasicContent"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>


    <!--   Metadata  Types-->
    <xs:group name="basicMetadataElements">
        <xs:annotation>
            <xs:documentation>
                These are the metadata elements that are common to all the different kinds of items. It is in
a group so it can easily be inserted into any of the metadata types.
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="itemID" type="ItemID"/>
            <xs:element name="itemRev" type="RevInfo"/>
            <xs:element name="prerequisites" type="Prerequisites" minOccurs="0"/>
            <xs:element name="pedegogies" type="Pedegogies" minOccurs="0"/>
        </xs:sequence>
    </xs:group>

    <xs:attributeGroup name="basicMetadataAttributes">
        <xs:annotation>
            <xs:documentation>
                These are the metadata attributes that are common to all the different kinds of items. It is
in a group so it can easily be inserted into any of the metadata types.
            </xs:documentation>
        </xs:annotation>
        <xs:attribute name="aETM" type="AETM"/>
        <xs:attribute name="difficulty" type="Percent"/>
    </xs:attributeGroup>

    <xs:group name="standardRefs">
        <xs:annotation>
            <xs:documentation>
                This set of references can be used in the about element of many, but not all, of the item
types.
            </xs:documentation>
        </xs:annotation>
        <xs:choice>
            <xs:element name="factRef" type="FactRef"></xs:element>
            <xs:element name="itemRef" type="ItemRef"></xs:element>
            <xs:element name="threadRef" type="ThreadRef"></xs:element>
        </xs:choice>
```

```
    </xs:group>


    <xs:complexType name="FactMetadata">
        <xs:sequence>
            <xs:element name="factID" type="FactID"/>
            <xs:group ref="basicMetadataElements"/>
            <xs:element name="about" minOccurs="0">
                <xs:complexType>
                    <xs:choice maxOccurs="unbounded">
                        <xs:element name="extRef" type="ExtRef"/>
                    </xs:choice>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="basicMetadataAttributes"/>
    </xs:complexType>

    <xs:complexType name="ExplanationMetadata">
        <xs:sequence>
            <xs:group ref="basicMetadataElements"/>
            <xs:element name="about">
                <xs:complexType>
                    <xs:choice maxOccurs="unbounded">
                        <xs:group ref="standardRefs"/>
                        <xs:element name="qaPairRef" type="QApairRef"/>
                    </xs:choice>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="basicMetadataAttributes"/>
    </xs:complexType>

    <xs:complexType name="QuestionMetadata">
        <xs:sequence>
            <xs:group ref="basicMetadataElements"/>
            <xs:element name="about" minOccurs="0">
                <xs:complexType>
                    <xs:group ref="standardRefs" maxOccurs="unbounded"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="basicMetadataAttributes"/>
    </xs:complexType>

    <xs:complexType name="AnswerMetadata">
        <xs:sequence>
            <xs:group ref="basicMetadataElements"/>
            <xs:element name="about">
                <xs:complexType>
                    <xs:choice maxOccurs="unbounded">
                        <xs:element name="a2QRef" type="A2QRef"/>
                    </xs:choice>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
```

```
        <xs:attributeGroup ref="basicMetadataAttributes"/>
    </xs:complexType>

    <xs:complexType name="ThreadMetadata">
        <xs:sequence>
            <xs:element name="threadID" type="ThreadID"/>
            <xs:group ref="basicMetadataElements"/>
            <xs:element name="creator" minOccurs="0" type="PersonInfo"/>
        </xs:sequence>
        <xs:attributeGroup ref="basicMetadataAttributes"/>
    </xs:complexType>


    <!--   Content  Types-->
    <xs:complexType name="Content" abstract="true">
        <xs:annotation>
            <xs:documentation>
                This is an abstract type. It is also intentionally empty. The sole purpose of this type is t
serve as a parent type for other types of content elements. This allows the demml/item element to remain
relatively simple. It also allows software developers to create a parent or interface class from which child
classes can be extended or implemented. This way any of these content types can be stored in the same
container if necessary.
            </xs:documentation>
        </xs:annotation>
    </xs:complexType>

    <xs:complexType name="BasicContent">
        <xs:complexContent>
            <xs:extension base="Content">
                <xs:sequence>
                    <xs:any namespace="http://www.w3.org/1999/xhtml http://www.w3.org/TR/html4/"
                        processContents="skip"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

</xs:schema>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"xml:lang="en" elementFormDefault="qualified">
    <xs:annotation>
        <xs:documentation>
           This schema file contains types which are common to many of the different schemas used by the
DEMML system. It has no target namespace so, when it is included in another schema document, it will take on
the namespace of the including document.
        </xs:documentation>
        <xs:documentation>
            <p>This is the first alpha version of the DEMML(tm) schema for content (version 0.1). This schema
is far from complete and the structure may need to be modified considerably in order to incorporate all the
features of the DEMML(tm). While I plan to make future versions of this schema a semi-open standard in the
future, the entire schema is currently copyright (c) 2010 by Grant Sheridan Robertson. It is currently not
permitted to use this schema - or any  portions thereof - in any manner without written permission. However,
if you write, I will probably give permision. I simply want to keep track of what is being done with
DEMML(tm) so I can keep a handle on it till it is ready for release.
            </p>
            <p>To learn more about this DEMML(tm) go to www.demml.org. To learn more about the schema itself
(past,  present,&amp;amp; future)  as well as a general description of how the DEMML system is organized see
www.demml.org/standard/.</p>
        </xs:documentation>

    </xs:annotation>
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
       schemaLocation="http://www.w3.org/2001/xml.xsd">
        <xs:annotation>
            <xs:documentation>
               Necessary to enable use of the xml:lang attribute.
            </xs:documentation>
        </xs:annotation>
    </xs:import>

    <!--   File  Metadata-->
    <xs:complexType name="FileInfo">
        <xs:annotation>
            <xs:documentation>
               Every DEMML content file will contain a fileInfo element at the top to indicate information
about the file as a whole.
            </xs:documentation>
            <xs:documentation>
               Any elements declared using this type should be called 'fileInfo'
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="fileRevInfo" type="RevInfo">
                <xs:annotation>
                    <xs:documentation>
                       When the RevInfo type is used for the fileRevInfo element then that info pertains to
the file as a whole. If any content for any of the items within the file is updated then the
fileRefInfo/contentRevInfo element should be updated. Similarly if any of the metadata for any of the items
within the file or if the file's metadata (exclusive of the fileRefInfo/contentRevInfo element) is modified,
then the fileRefInfo/dataRevInfo element should be updated.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="creator" maxOccurs="unbounded" type="PersonInfo">
```

```
                <xs:annotation>
                    <xs:documentation>
                        This is the person who created or compiled the file. Not necessarilly who created
every item within the file.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

    <!--   Topic  Identifier-->
    <xs:complexType name="TopicID">
        <xs:annotation>
            <xs:documentation>
                The topic ID is used to uniquely identify a DEMML topic. It is used both to lable specific
topics as well as to refer to them.
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="topicTitle" type="Title" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>
                        When used to label a topic within the Topic Info file there must be one topic title
for each of the language-specific stem folders in the topic folder. When used to indicate the topic in which
a DEMML content file belongs, the topicID must include a descriptive title in the same language as the topic
stem-folder in which the file belongs or resides. However, one may provide optional additional titles in
other languages if desired.
                    </xs:documentation>
                    <xs:documentation>
                        Remember, the title is purely for human consumption. All software must use only the
topicCode and topicSN to refer to and index the topics.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="topicCode" type="xs:token" use="required">
            <xs:annotation>
                <xs:documentation>
                    <p>This is the Distributable Educational Material Classification System (DEMCS)
classification code. This code includes the full path to the topic within the DEMCS. See<a
href="http://demml.org/standard/classification/">http://demml.org/standard/classification/</a> for  more
information.</p>
                    <p>Do not include the demml:// protocol prefix in the value for this attribute.</p>
                </xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attribute name="topicSN" type="SerialNumber" use="required">
            <xs:annotation>
                <xs:documentation>
                    The Topic Serial Number must be unique among all the topics within the entire DEMML
system. Serial Numbers are assigned sequentially as each topic is added the DEMCS. In case some topics need
to be rearranged at some time in the future (hopefully very rarely) the serial numbers can be used by
software to locate the correct topics and update their locations within individual users' databases of DEMML
content.
                </xs:documentation>
            </xs:annotation>
```

```
        </xs:attribute>
    </xs:complexType>

    <!--   Item  Identifier-->
    <xs:complexType name="ItemID">
        <xs:annotation>
            <xs:documentation>
                <p>In DEMML all content is stored within "Items" which contain both the metadata about the
content and the content itself. All items about a particular topic are stored in the appropriate language
stem folder under the topic folder. However, it is possible for an item to be associated with more than one
topic. In that case the item will be stored under the most closely associated topic (or the topic closest to
the root of the tree, or just choose one at random). More than one item can be stored in a particular DEMML
content file but all items in a file obviously must be associated with the same topic in order to meet the
above requirement.</p>
                <p>DEMML items can be of various types. Most important are the Fact-Items which state one of
the facts that must be learned about the topic. Fact-Items should be concise and to the point.
Explanation-Items are items which explain something within the topic. This is where most of the learning
takes place. An Explanation-Item can explain a fact or even why a particular answer is right or wrong. There
are Question-Items and Answer-Items as well as Problem-Items and Solution-Items. There can be questions about
one or more facts or even about one particular explanation. Explanations can be about almost any other kind
of item. However, there should never be questions about explanations which are already about some other
question or answer. One has to draw the line somewhere.</p>
                <p>While most XML developers have become accustomed to mirroring the hierarchical structure
of their data within the XML document itself, DEMML takes a different tack for the structure of items within
a topic. One of the primary goals of DEMML is to allow easy distribution of the content in the form of
relatively small files. This way a student does not need to download (or copy) and store much more than they
really need and the transfer will be as efficient as possible. In addition, there is the possibility of there
being hundreds - or even thousands - of items within a topic, especially when you consider all the different
problems, solutions, and explanations for those solutions for a single mathematical concept. Therefore, it is
not reasonable to attempt to store a topic's entire structure of items within one XML document.</p>
                <p>All this means the structure of the relationships between DEMML items is a limited sort of
directed graph with the references pointing backwards. Rather than have each fact contain a reference to
every question about that fact, as many software developers are accustomed to seeing, it is far more
efficient to have each question contain references to the few items that question is about. This way the
Fact-Items do not need to be updated every time a new Question-Item is published. All a user has to do is
copy the file with the Question-Item into the appropriate folder on their computer (which can be done
automatically by some software) and the software can read the file and update its internal database.</p>
            </xs:documentation>
            <xs:documentation>Any time an element is declared using this type it should be called
                'itemID'</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="itemTitle" type="Title" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>
                        Each item must have a descriptive title in the same language as the topic stem-folder
in which the item resides. However, one may provide optional additional titles in other languages if desired.
                    </xs:documentation>
                    <xs:documentation>
                        If the item is a question or problem then the title should not give away the answer,
just in case some software chooses to display the titles of each item.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="itemNumber" type="SerialNumber" use="required">
```

```
            <xs:annotation>
                <xs:documentation>
                    This is simply a sequentialy assigned code number comprised of digits and capital
letters. It must be unique among all the other items in the same topic folder.
                </xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attribute name="itemType">
            <xs:annotation>
                <xs:documentation>
                    This attribute is for informational purposes only, rather than to define a datatype. It
is included so when an itemID is used in a reference, software or human readers can tell what type of item is
being referenced without retrieving that actual item.
                </xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="fact"/>
                    <xs:enumeration value="explanation"/>
                    <xs:enumeration value="question"/>
                    <xs:enumeration value="answer"/>
                    <xs:enumeration value="thread"/>
                    <!--<xs:enumerationvalue="problem"/>-->
                    <!--<xs:enumerationvalue="solution"/>-->
                    <!--<xs:enumerationvalue="definition"/>-->
                    <!--<xs:enumerationvalue="exercise"/>-->
                    <!--<xs:enumerationvalue="exerciseExternal"/>-->
                    <!--<xs:enumeration value="lab"/>->
                    <!--<xs:enumerationvalue="project"/>-->
                    <!--<xs:enumerationvalue="identification"/>-->
                    <!--<xs:enumerationvalue="demmlLink"/>-->
                    <!--<xs:enumerationvalue="externalLink"/>-->
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

    <!--   Fact  Identifier-->
    <xs:complexType name="FactID">
        <xs:annotation>
            <xs:documentation>
                Each DEMML topic contains one or more facts. These are the tiny pieces of information that
just wouldn't make sense on their own. A fact is one of the many special types of items. For each fact there
could be more than one way to state that fact. All the different ways to state the same fact will each be in
their own individual items but will all have the same fact code. This way software can allow students or
teachers to choose the way of stating that fact that makes the most sense to them.
            </xs:documentation>
            <xs:documentation>
                Any elements created from this type should have the name 'factID'
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="factTitle" type="Title" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>
                        Each fact must have a descriptive title in the same language as the topic stem-folde
```

```
in which the fact-item resides. However, one may provide optional additional titles in other languages if
desired.
                    </xs:documentation>
                    <xs:documentation>
                        Because each fact is contained in a separate item, the Item Title for the item
containing the fact shuld be the same as the Fact Title.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="factCode" use="required">
            <xs:annotation>
                <xs:documentation>
                    Fact codes consist of one or more digits or capital letters suffixed by zero or one
lower-case letter. The first letter is always a capital 'F' to make them easier to recognize when written
down. In order to allow additional facts to be inserted into an ordered list of preexisting facts, fact codes
follow the same rules and pattern as the DEMCS coding system (see
http://demml.org/standard/classification/4-assign_codes.htm). In some cases one level of sub-facts will be
appropriate. In which case the sub-facts will have the code of the parent fact followed directly by one
lower-case letter. For instance, N7b or 3La. If at any time one feels it necessary to create a third level of
facts then serious consideration should be given to creating sub-topics instead, putting each of the primary
parent facts in its own topic. Sometimes the order of facts will not matter. But for some topics where the
order the facts are listed helps provide meaning to these facts then make sure the facts are numbered in an
appropriate order.
                </xs:documentation>
                <xs:documentation>
                    Fact F0 is the statement of the main point of the topic.
                </xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern value="F[0-9A-Z]+[a-z]*"></xs:pattern>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

    <xs:complexType name="ThreadID">
        <xs:sequence>
            <xs:element name="threadTitle" type="Title" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="threadCode" type="xs:token" use="required">
            <xs:annotation>
                <xs:documentation>
                    There can be only one effective thread with the same threadCode in any one branch of the
DEMCS tree. If there are threads with the same threadCode in more than one level of the same branch of the
tree, the thread in the branch farthest from the root of the tree takes precedence. This allows a type of
inheritance in that a thread can be more general closer to the root of the tree but then be modified to be
more specific to the topics in the branches farther out from the root. There is no formal mechanism for
controling inheritance. Whichever thread is referenced is the thread that will be applied to the referencing
item.
                </xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attribute name="threadSN" type="SerialNumber" use="required">
            <xs:annotation>
```

```
                <xs:documentation>
                   The thread serial number should be unique in the entire DEMML system.
                </xs:documentation>
            </xs:annotation>
        </xs:attribute>
    </xs:complexType>

    <!--    Pedagogy-->
    <xs:complexType name="Pedegogies">
        <xs:sequence>
            <xs:element name="pedegogy" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>
                       Used to specify a learning style and how well this item fits that learning style.
According to Wikipedia there are as many as 71 different "learning style" models, none of which has been
proven valid in any way (Pashler, H.; McDaniel, M.; Rohrer, D.; Bjork, R. (2009). "Learning styles: Concepts
and evidence". Psychological Science in the Public Interest 9: 105–119. cited in:
http://en.wikipedia.org/wiki/Learning_styles#cite_ref-pashler_1-2). However, this element is made available
for those who either subscribe to the Learning Style theories or simply want to provide more information
about their content. At the very least, software can use these tags to ensure that students get a good mix of
content based on different learning styles for any particular topic or fact.
                    </xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="comment" type="xs:string" minOccurs="0">
                            <xs:annotation>
                                <xs:documentation>
                                    Optional explanation as to how and why this item fits this learning
model.
                                </xs:documentation>
                            </xs:annotation>
                        </xs:element>
                    </xs:sequence>
                    <xs:attribute name="model" type="TokenURI" use="required"/>
                    <xs:attribute name="domain" type="TokenURI"/>
                    <xs:attribute name="Classification" type="TokenURI" use="required"/>
                    <xs:attribute name="percent" type="Percent" use="required"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

    <!--   General  Metadata-->
    <xs:complexType name="RevInfo">
        <xs:annotation>
            <xs:documentation>
               Reflects the date and version for this file or item. If this is the first version then the
version should be 1.0 and the dates should match the createDate attribute of the fileRevInfo element.
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="contentRevInfo">
                <xs:annotation>
                    <xs:documentation>
                       This data should only be updated if the actual educational content that the student
```

```
sees is updated. If only the metadata changes then do not update this element.
                    </xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:attribute name="contentVersion" type="xs:token" use="required"/>
                    <xs:attribute name="contentModDate" type="xs:date" use="required"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="dataRevInfo">
                <xs:annotation>
                    <xs:documentation>
                        The data in this element should only be updated if the metadata for the item or file
has changed (exclusive of the contentRevInfo element). If only the content has been changed then do not
update this element.
                    </xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:attribute name="dataVersion" type="xs:token" use="required"/>
                    <xs:attribute name="dataModDate" type="xs:date" use="required"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="createDate" type="xs:date" use="required">
            <xs:annotation>
                <xs:documentation>
                    Date the file was first created. Once the file is created, this date should not be
changed.
                </xs:documentation>
            </xs:annotation>
        </xs:attribute>
    </xs:complexType>

    <xs:complexType name="Title">
        <xs:simpleContent>
            <xs:extension base="xs:token">
                <xs:attribute use="required" ref="xml:lang"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>

    <xs:complexType name="Prerequisites">
        <xs:sequence>
            <xs:element name="prereq" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="topicID" type="TopicID"/>
                        <xs:element name="factID" type="FactID" minOccurs="0"/>
                        <xs:element name="proficiency" type="Proficiency">
                            <xs:annotation>
                                <xs:documentation>
                                    When listed as part of a prerequisite, a proficiency value refers to how
well the student must know the specified topic or fact within a topic. In addition, the estimated time to
master (etm) attribute should be eiter set to "PT0S" for zero seconds or left out.
                                </xs:documentation>
                            </xs:annotation>
                        </xs:element>
```

```
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="Proficiencies">
        <xs:annotation>
            <xs:documentation>
                More than one proficiency can sometimes be listed with a different estimated time to master
(etm) for each proficiency level. There is no need to list more than a few different proviciencies as
software will normally be able to extrapolate the etm values for various proficiency levels. Multiple
proficiencies should be used when a subject, topic, or item has a much steeper "learning curve" for the more
advanced material than one would normally expect.
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="proficiency" maxOccurs="unbounded" type="Proficiency"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="Proficiency">
        <xs:attribute name="percent" type="PercentPlus" use="required"/>
        <xs:attribute name="aETM" type="AETM"/>
    </xs:complexType>

    <xs:simpleType name="PercentPlus">
        <xs:annotation>
            <xs:documentation>
                Contrary to what one may think, the percent value for some measurements could go above 100%
due to either over-study or learning more sub-topics with more proficiency than required by a DEMML syllabus.
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:float">
            <xs:minInclusive value="0.00"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="Percent">
        <xs:annotation>
            <xs:documentation>
                This is the standard percentage value from 0 to 100.
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="PercentPlus">
            <xs:maxInclusive value="100"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="AETM">
        <xs:annotation>
            <xs:documentation>
                Average Expected Time to Master: This is the average time to reach the specified level of
proficiency. It is based on the assumption that the student exactly meets all of the prerequisites for this
item or topic but has not studied any of the item, topic, or its sub-topics yet.
            </xs:documentation>
```

```
            <xs:documentation>
               The duration should be set to a reasonable value. If aETM is not given then software is
expected to ignore the etm for this item or rather than attempt to force the student to learn the item in
zero seconds.
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:duration">
            <!--<xs:patternvalue="P\d+Y\d+M\d+DT\d+H\d+M0S"/>-->
            <!--<xs:patternvalue="PT\p{Nd}+M"/>-->
            <xs:pattern value="PT\d+M">
                <xs:annotation>
                    <xs:documentation>
                       This pattern should limit the aETM to being expressed in minutes only.
                    </xs:documentation>
                </xs:annotation>
            </xs:pattern>
            <xs:minInclusive value="PT0M">
                <xs:annotation>
                    <xs:documentation>
                       Naturally, the aETM shouldn't be less than zero minutes.
                    </xs:documentation>
                </xs:annotation>
            </xs:minInclusive>
        </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="PersonInfo">
        <xs:annotation>
            <xs:documentation>
               ToDo: Look up or create a more detailed PersonInfo type to use here.
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="personName" type="xs:token"/>
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="TokenURI">
        <xs:annotation>
            <xs:documentation>
               This type allows either a simple tokenized string or any URI. It is used for fields wherein
some document creators may want to just just a standardized value and others may want to use a URI to provide
more specific information.
            </xs:documentation>
        </xs:annotation>
        <xs:union memberTypes="xs:token xs:anyURI"/>
    </xs:simpleType>

    <xs:simpleType name="SerialNumber">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9A-Z]+"/>
        </xs:restriction>
    </xs:simpleType>

    <!--   Reference  Types-->
    <xs:complexType name="Reference" abstract="true">
```

```
        <xs:annotation>
            <xs:documentation>
                A reference is used to point to some other item. There are various different types of
references to point to various different categories of items.
            </xs:documentation>
            <xs:documentation>
                This datatype is an empty, abstract datatype. It exists solely to indicate that all the
different types of references are of one family. Though it would have been possible to create one reference
datatype which optionally allowed all the different types of identifiers, that method would have resulted in
DEMML content documents that were not as easy to read. It would also have allowed invalid combinations of the
various identifiers. This method enforces a specific set of combinations of identifiers. It also forces
content creators to use the xsi:type= attribute in their reference elements to indicate the actual datatype
being used, which allows human readers to easily see what that reference is pointing to. Finally, it allows
developers to make use of inheritance and polymorphism, specifying this datatype in an object which then
allows all the child datatypes.
            </xs:documentation>
        </xs:annotation>
    </xs:complexType>

    <xs:complexType name="FactRef">
        <xs:annotation>
            <xs:documentation>
                Points to a spedific Fact-Item within this topic.
            </xs:documentation>
        </xs:annotation>
        <xs:complexContent>
            <xs:extension base="Reference">
                <xs:sequence>
                    <xs:element name="factID" type="FactID"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ExtRef">
        <xs:annotation>
            <xs:documentation>
                Points to a specific Fact-Item in a different topic.
            </xs:documentation>
        </xs:annotation>
        <xs:complexContent>
            <xs:extension base="Reference">
                <xs:sequence>
                    <xs:element name="topicID" type="TopicID"/>
                    <xs:element name="factID" type="FactID" minOccurs="0"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ItemRef">
        <xs:annotation>
            <xs:documentation>
                Points to most types of items within this topic. Note: There is no means to refer to a
specific item within a different topic.
            </xs:documentation>
```

```
        </xs:annotation>
        <xs:complexContent>
            <xs:extension base="Reference">
                <xs:sequence>
                    <xs:element name="itemID" type="ItemID"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="A2QRef">
        <xs:annotation>
            <xs:documentation>
                Used only when an Answer-Item refers to a Question-Item.
            </xs:documentation>
        </xs:annotation>
        <xs:complexContent>
            <xs:extension base="Reference">
                <xs:sequence>
                    <xs:element name="itemID" type="ItemID"/>
                </xs:sequence>
                <xs:attribute name="correct" type="Percent" use="required">
                    <xs:annotation>
                        <xs:documentation>
                            Indicates if this answer is a correct response to the referred to question. A
percentage is used rather than a simply boolean because some answers are worth partial credit. Note: it is
not necessary to create a reference from every Answer-Item to every Question-Item indicating the degree of
correctness. Instead, only create these references when the answer is a plausible answer to the referred to
question or is a good distractor for that question. This way software can automatically create quizes that
are different each time.
                        </xs:documentation>
                    </xs:annotation>
                </xs:attribute>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="QApairRef">
        <xs:annotation>
            <xs:documentation>
                Used only in Explanation-Items to explain why an Answer-Item is or is not a correct answer to
a particular question. Because questions and answers can be worded many different ways, while still meaning
the same thing, some Explanation-Items may point to more than one question-answer pairs.
            </xs:documentation>
        </xs:annotation>
        <xs:complexContent>
            <xs:extension base="Reference">
                <xs:sequence>
                    <xs:element name="questionItem" type="ItemID"/>
                    <xs:element name="answerItem" type="ItemID"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ThreadRef">
```

```
        <xs:annotation>
            <xs:documentation>
               Used to indicate that this item follows a particular thread.
            </xs:documentation>
        </xs:annotation>
        <xs:complexContent>
            <xs:extension base="Reference">
                <xs:sequence>
                    <xs:element name="topicID" type="TopicID">
                        <xs:annotation>
                            <xs:documentation>
                               Because a thread could be started anywhere, even in a different topic, this
reference must be able to point outside of the topic where it is used.
                            </xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="itemID" type="ItemID"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

</xs:schema>
```