

ISO/IEC JTC 1/SC 34

Date: 2005-10-20

ISO/IEC WD N458

ISO/IEC JTC 1/SC 34/WG 3

Secretariat: SCC

Topic Maps Constraint Language

Warning

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

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

Copyright notice

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

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

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

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

Violators may be prosecuted.

Contents

Page

Foreword.....	iv
Introduction.....	v
1 Scope.....	1
2 Normative references.....	1
3 Notation and Conventions.....	1
3.1 Notation and Syntax.....	1
3.2 Informal and Formal Semantics.....	1
4 TMCL.....	1
4.1 Validation Semantics.....	2
4.2 TMCL-Schema.....	2
4.2.1 Topic Map Schema item.....	2
4.2.2 Topic Identification item.....	3
4.2.3 Scope Pattern item.....	4
4.2.4 Subtype Schema Item.....	4
4.2.5 Topic Schema Item.....	5
4.2.6 IRI Schema Item.....	6
4.2.7 Topic Name Schema Item.....	7
4.2.8 Variant Name Schema Item.....	8
4.2.9 Occurrence Schema Item.....	9
4.2.10 Play Role Schema Item.....	9
4.2.11 Role Schema Item.....	10
4.2.12 Association Schema Item.....	10
4.2.13 TMCL-Schema descriptions of warnings.....	10
4.3 Interpretation of TMCL-Schema.....	11
4.3.1 Subject Locator basic constraints.....	11
4.3.1.1 SubjectLocatorCardMinConstraint.....	11
4.3.1.2 SubjectLocatorCardMaxConstraint.....	11
4.3.2 Subject Identifier basic constraints.....	12
4.3.2.1 SubjectIdentifierCardMinConstraint.....	12
4.3.2.2 SubjectIdentifierCardMaxConstraint.....	12
4.3.3 Topic name basic constraints.....	13
4.3.3.1 TopicNameCardMinConstraint.....	13
4.3.3.2 TopicNameCardMaxConstraint.....	13

4.3.3.3	TopicNameMatchConstraint.....	14
4.3.4	Variant name basic constraints.....	14
4.3.4.1	VariantNameCardMinConstraint.....	14
4.3.4.2	VariantNameCardMaxConstraint.....	15
4.3.4.3	VariantNameMatchConstraint.....	16
4.3.5	Occurrence basic constraints.....	17
4.3.5.1	OccurrenceCardMinConstraint.....	17
4.3.5.2	OccurrenceCardMaxConstraint.....	17
4.3.5.3	OccurrenceMatchConstraint.....	18
4.3.5.4	OccurrenceDataTypeConstraint.....	18
4.3.6	PlayRole basic constraints.....	19
4.3.6.1	PlayRoleCardMinConstraint.....	19
4.3.6.2	PlayRoleCardMaxConstraint.....	20
4.3.7	OtherRole basic constraints.....	20
4.3.7.1	20
4.3.7.2	21
4.3.7.3	22
4.3.7.4	23
4.3.7.5	23
4.3.8	Associaton basic constraints.....	24
4.3.8.1	RoleCardMinConstraint.....	24
4.3.8.2	RoleCardMaxConstraint.....	25
4.3.8.3	RoleAllPlayersFromConstraint.....	25
4.3.8.4	RoleOneOfConstraint.....	26
4.4	TMCL-Rule.....	26
4.4.1	RuleItem.....	27
4.4.2	AssertItem.....	27
4.5	Conflict Items.....	27
4.5.1	ConflictItem.....	27
4.6	Syntax for TMCL-Schema.....	28
4.7	Syntax for TMCL-Rule.....	28
4.8	Combining TMCL-Rule and TMCL-Schema.....	28
4.9	Topic Map Representation of Constraints.....	28
4.10	Topic Map Schema References.....	29
4.11	Schema Composition.....	29
	Bibliography.....	30

Foreword

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

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

ISO/IEC N458 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 34, Document Description and Processing Languages.

Introduction

This International Standard defines a means to express constraints on topic maps conforming to the Topic Map Data Model [TMDM].

Topic Maps Constraint Language

1 Scope

This International Standard is designed to allow users to constrain any aspect of topic maps based on topic map data model. This International Standard defines TMCL-Schema and TMCL-Rule. TMCL-Schema provides a type based model of constraints. TMCL-Rule provides a generalized model of constraint based on TMQL [TMQL]. TMQL is used as a means to identify the topic map constructs to be constrained and to define the topic map structures that must exist in order for the constraint to be met.

This International Standard defines a model for representing constraints and formal semantics for the interpretation of the different model structures. It also defines a syntactic form that can be used to represent the model structures.

2 Normative references

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

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

Unicode, *The Unicode Standard, Version 4.0*, The Unicode Consortium, Boston, MA, USA, Addison-Wesley, 2003, ISBN 0-321-18578-1

TMDM, *ISO 13250-2 Topic Maps — Data Model*, ISO, 2005, available at <<http://www.isotopicmaps.org/sam/sam-model/>>

TMQL, *ISO Topic Maps Query Language Working Draft*, ISO, 2005, available at <<http://www.isotopicmaps.org/tmql/>>

XML 1.0, *Extensible Markup Language (XML) 1.0*, W3C, Third Edition, W3C Recommendation, 04 February 2004, available at <<http://www.w3.org/TR/REC-xml/>>

RFC3986, *RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax*, The Internet Society, 2005, available at <<http://www.ietf.org/rfc/rfc3986.txt>>

RFC3987, *RFC 3987 - Internationalized Resource Identifiers (IRIs)*, The Internet Society, 2005, available at <<http://www.ietf.org/rfc/rfc3987.txt>>

3 Notation and Conventions

3.1 Notation and Syntax

TODO: Infoset

3.2 Informal and Formal Semantics

TODO

4 TMCL

TMCL defines TMCL-Schema and TMCL-Rule. TMCL-Schema provides a type-based model of constraints. TMCL-Rule provides a generalized model of constraints based on TMQL.

Both TMCL-Rule and TMCL-Schema define sets of constraints. These constraints consist of terms that identify parts of the Topic Map to be constrained and terms that define the predicates that must hold true for the Topic Map to be considered consistent.

TMCL-Schema and TMCL-Rule use different sub-languages for describing constraints. TMCL-Schema defines grammar-like structures for describing constraints while TMCL-Rule defines constraints in a form of "if ... then ..." rules. Both sub-languages can be combined in schema expressions using specified syntax.

4.1 Validation Semantics

TMCL-Rule and TMCL-Schema are used to constrain instances of the Topic Map Data Model. If the topic map is valid in respect to the constraints being tested then validation is said to have succeeded. More formally it can be said that :

```
Given:
  TopicMap: t
  Schema : s
Then:
  Validate(t, s) => (true) | (false, conflictItem+)
```

Conflict items are information items that reference the constraints which were not valid with respect to the processed topic map. Conflict items can be interpreted by implementations as warnings or errors depending on the localized configuration.

4.2 TMCL-Schema

TMCL-Schema defines a language for constraining types of topics and associations.

TMCL-Schema expressions effectively combine shared context for multiple constraints and allow compact representations of these constraints. However, from logical perspective each TMCL-Schema is equivalent to a set of constraints which can be validated independently

The TMCL schema model is defined using the InfoSet notation. For each constraint an corresponding constraint information item exists. The properties for each constraint are enumerated and described.

4.2.1 Topic Map Schema item

A *topic map schema* is a collection of constraints that are intended to be evaluated together to establish if a given TMDM instance is valid.

Topic schemas consist of an identity, name, a set of TopicSchema items, a set of AssociationSchema items and a set of Rule items.

The schemaID is used to unambiguously differentiate schemas. This International Standard does not specify what format or algorithm is to be used to create such an identity.

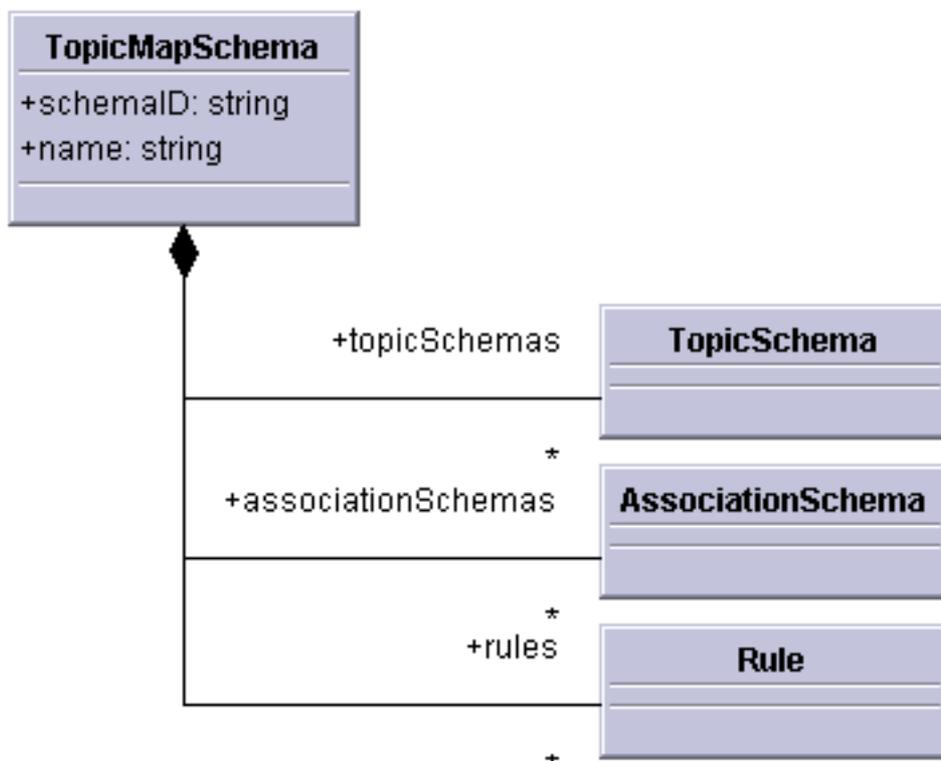


Figure 1 — The TopicMapSchema item

A Collection of constraints. TopicMapSchema items have the following properties:

- 1 **[schemaID]**: A string. The unique identifier for this schema.
- 2 **[name]**: A string. The human readable name for this schema.
- 3 **[topicSchemas]**: A set of TopicSchema items. A set of topic schema constraints.
- 4 **[associationSchemas]**: A set of AssociationSchema items..
- 5 **[rules]**: A set Rule items. A set of rule constraints.

4.2.2 Topic Identification item

A *topic identification item* is a construct used to identify a topic by sourceLocator, subjectIdentifier or subjectLocator.

Constraints need a mechanism by which to identify topics that are part of a constraint. Such as usage would be to specify the type of occurrence that must be present. The Topicidentification item allows for such selection to occur.

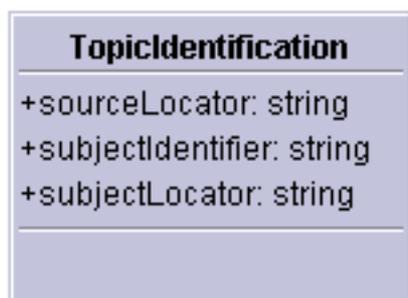


Figure 2 — The TopicIdentification item

A construct to identify topics used in constraints. TopicIdentification items have the following properties:

- 1 **[sourceLocator]**: A string. The sourceLocator value for the topic being identified.
- 2 **[subjectIdentifier]**: A string. The subjectIdentifier value for the topic being identified.
- 3 **[subjectLocator]**: A string. The subjectLocator value for the topic being identified.

4.2.3 Scope Pattern item

A *scope pattern item* is the way in which scope should be interpreted in a given context.

When defining constraints involving scope it is necessary to be precise about the interpretation of scope matching.

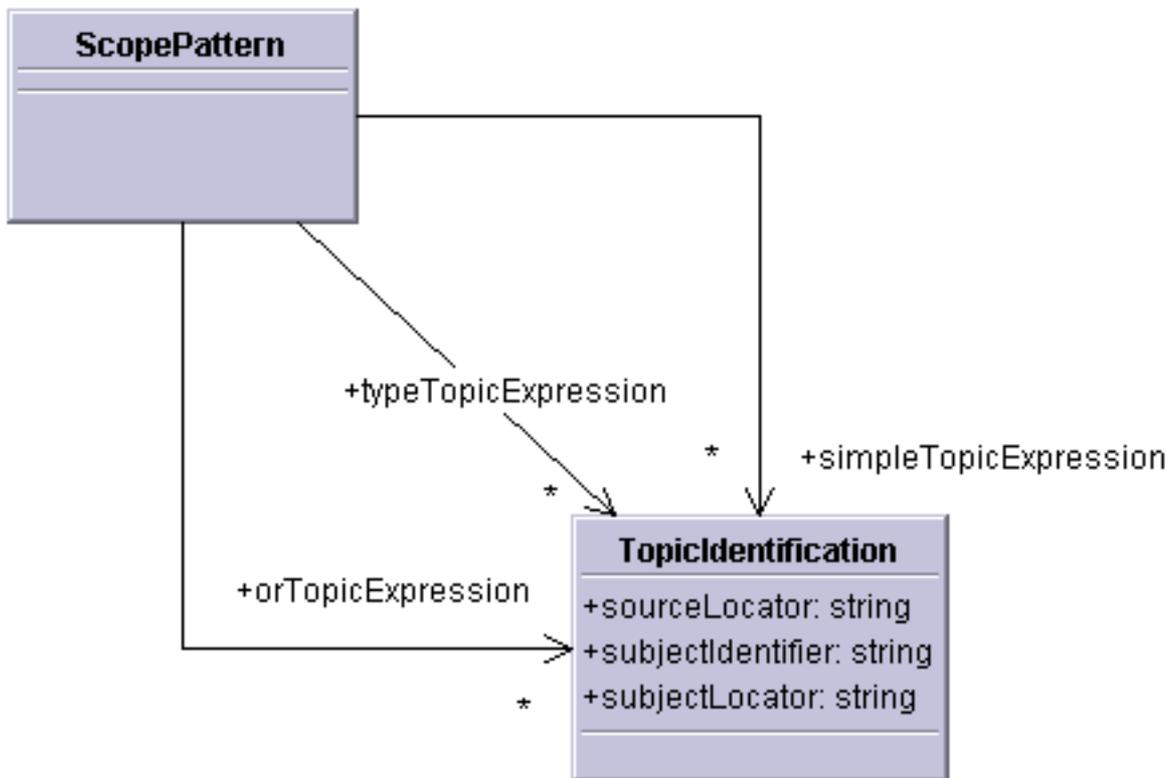


Figure 3 — The ScopePattern item

A construct to define how to match a against a given scope. ScopePattern items have the following properties:

- 1 **[simpleTopicExpression]**: A set of TopicIdentification items. Will match a scope where all of the topics listed are present.
- 2 **[orTopicExpression]**: A set of TopicIdentification items. Will match a scope where any of the topics listed are present.
- 3 **[typeTopicExpression]**: A set of TopicIdentification items. Will match a scope where one of the topics must be an instance of at least one of the identified topics.

4.2.4 Subtype Schema Item

A *subtype schema* provides a way to express the nature of a subtype relationship and instance membership thereof.

This schema constraint allows for the expression of the allowed subtypes of a given topic type. In addition, it conveys how instances of the subtypes relate. The definition of disjoint for example enables an expression that any instance of one subtype cannot be an instance of the other subtype.

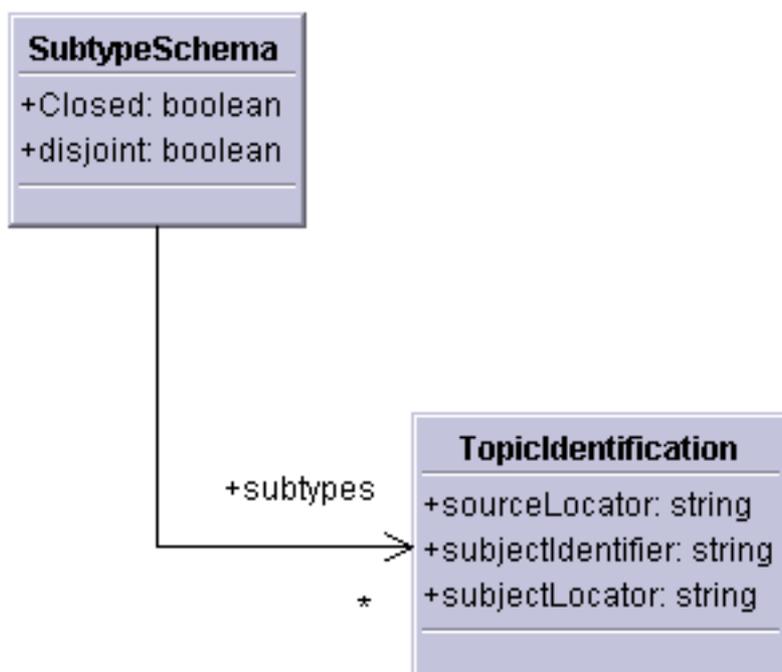


Figure 4 — The Subtype Schema item

A construct to constrain the allowed subtypes of given type. SubtypeSchema items have the following properties:

- 1 **[subtypes]**: A set of Topic Identification items.. The set of topic types that are allowed subtypes.
- 2 **[disjoint]**: A boolean value. Indicates if instances of the subtypes are allowed to be instances of another subtype. The value of 'true' indicates that instances must be of distinct subtypes.
- 3 **[closed]**: A boolean value. Indicates if the list of subtypes represents a closed list. If this value is true then it is a schema conflict if any other subtypes are defined in a topic map.

4.2.5 Topic Schema Item

A *topic schema* defines a collection of constraints that together restrict instances of topics of the specified type.

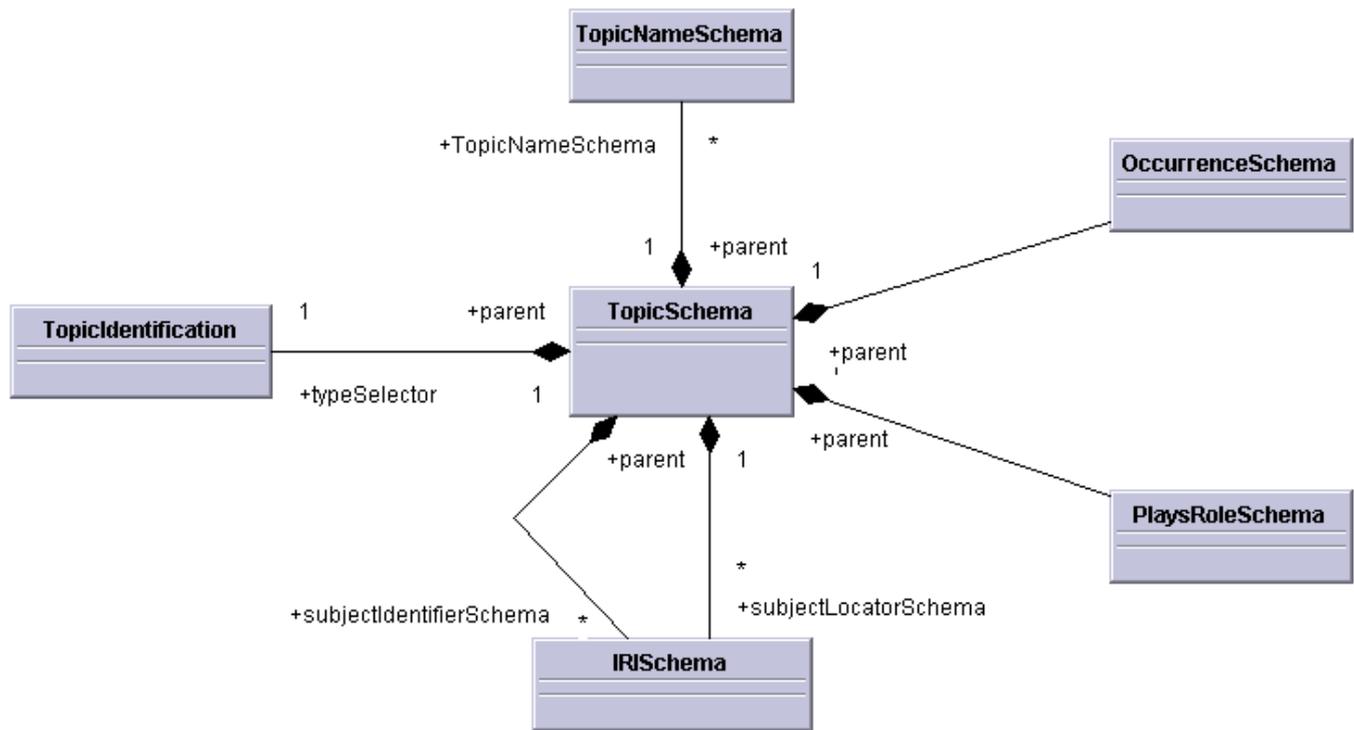


Figure 5 — The Topic Schema item

A construct to group together a collection of constraints that relate to all instances of the identified topic type. TopicSchema items have the following properties:

- 1 **[typeSelector]**: A set of Topic Identification items.. The set of topic types to which the rest of the constraints are relevant[better choice of wording goes here].
- 2 **[subjectAddressSchema]**: A set of IRISchema items. Schema for defining the constraints on subjectLocators of instances of the constrained topic type.
- 3 **[subjectLocatorSchema]**: A set of IRISchema items. Schema for defining the constraints on subjectLocators of instances of the constrained topic type.
- 4 **[subjectIdentifierSchema]**: A set of IRISchema items. A schema for defining the constraints on subjectLocators of instances of the constrained topic type.
- 5 **[topicNameSchema]**: A set of TopicNameSchema items. Schema for defining the constraints on the TopicNames of instances of the constrained topic type.
- 6 **[occurrenceSchema]**: A set of OccurrenceSchema items. Schema for defining the constraints on the Occurrences of instances of the constrained topic type.
- 7 **[playRoleSchema]**: A set of PlayRoleSchema items. Schema for defining the constraints on the roles played by instances of the constrained topic type.
- 8 **[oneOfSchema]**: A set of TopicIdentification items. Schema for defining the set of topics that can be instances of the identified topic type[gdm - er is this really what we mean].
- 9 **[subTypes]**: A set of SubtypeSchema items. Schema for defining the constraints on allowed subtypes of the identified topic type.

4.2.6 IRI Schema Item

A *IRI schema* defines a collection of constrains on the cardinality and string representation of IRIs used as subjectLocators, subjectIdentifiers and subjectAddresses.

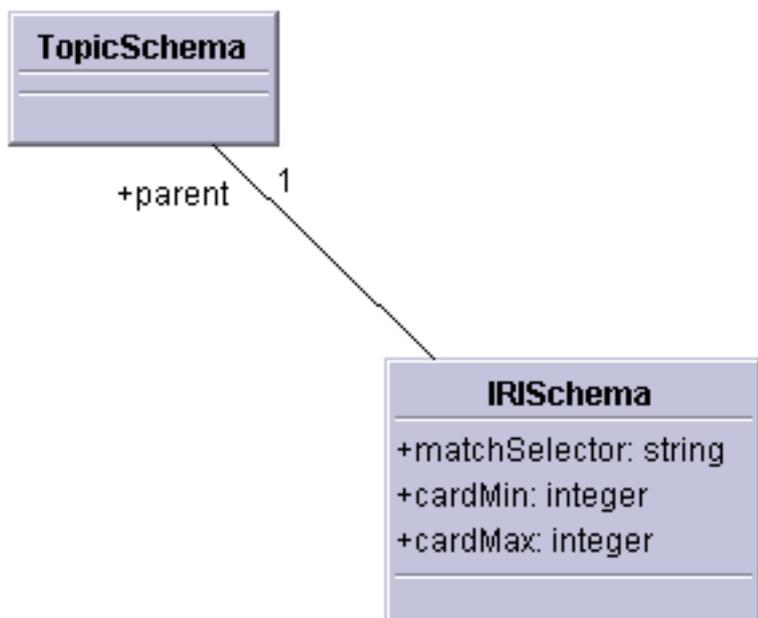


Figure 6 — The IRI Schema item

Construct to constrain the cardinality and string representation of IRIs used as subjectLocators and subjectIdentifiers. IRISchema items have the following properties:

- 1 **[parent]**: A TopicSchema item.. The containing TopicSchema item.
- 2 **[matchSelector]**: A Regular Expression.. The locator value must be conform to the regular expression to be valid.
- 3 **[cardMax]**: An integer value.. The maximum allowed number of identifiers matching this pattern.
- 4 **[cardMin]**: An integer value.. The minimum allowed number of identifiers matching this pattern.

4.2.7 Topic Name Schema Item

A *topic name schema* defines a collection of constrains on names of a given type on topics type of a given type..

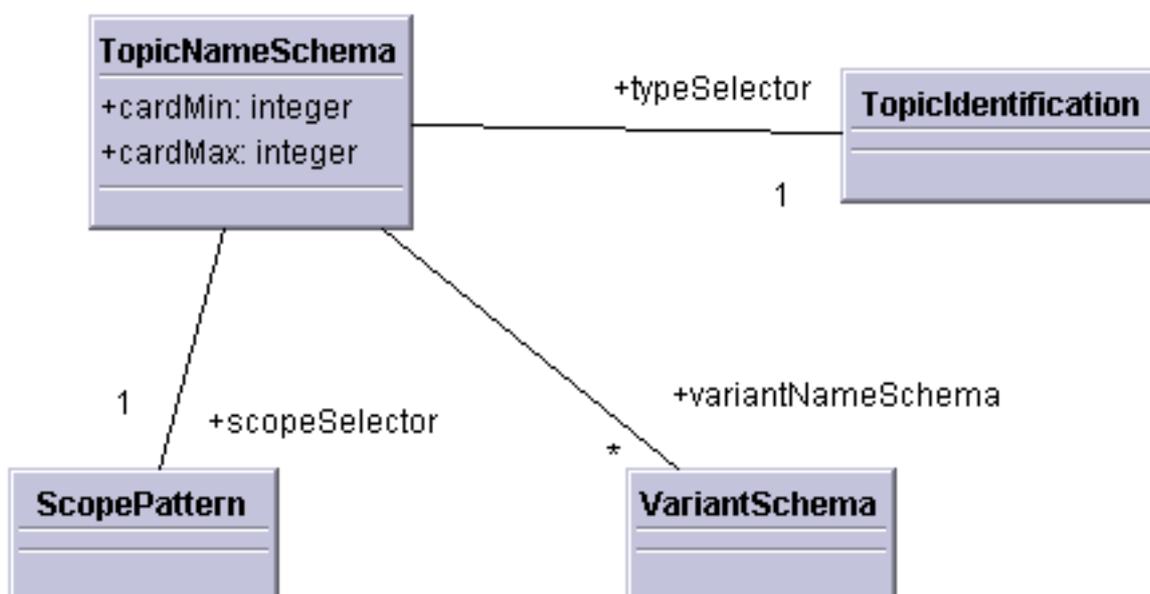


Figure 7 — The Topic Name Schema item

Construct to constrain the names of a topic. TopicNameSchema items have the following properties:

- 1 **[parent]**: A TopicSchema item.. The containing TopicSchema item.
- 2 **[typeSelector]**: A TopicIdentification item.. The type used to identify to which names the constraints should be applied.
- 3 **[scopeSelector]**: An ScopePattern item.. Names whose scope match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. I would prefer scope to be a constraint.
- 4 **[variantSchema]**: A set of Variant Schema items.. Names whose scope match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. I would prefer scope to be a constraint.
- 5 **[cardMin]**: An integer value.. The minimum allowed number of names matching this pattern.
- 6 **[cardMax]**: An integer value.. The maximum allowed number of names matching this pattern.

4.2.8 Variant Name Schema Item

A *variant name schema* defines a collection of constrains on variant names of a given type..

Figure 8 — The Variant Name Schema item

Construct to constrain the variant names of a topic. VariantNameSchema items have the following properties:

- 1 **[parent]**: A VariantNameSchema item.. The containing VariantNameSchema item.
- 2 **[scopeSelector]**: An ScopePattern item.. The Names whose scope must match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. I would prefer scope to be a constraint.
- 3 **[dataTypeSelector]**: A string.. An expression used to constrain the allowed data type. [gdm - maybe we say its an IRI and that can be interpreted locally, perhaps use a regex?]
- 4 **[match]**: A regular expression.. The value of the variant must match this regular expression.
- 5 **[cardMin]**: An integer value.. The minimum allowed number of variant names matching this pattern.
- 6 **[cardMax]**: An integer value.. The maximum allowed number of variant names matching this pattern.

Semantics :

This is the lowest level of model from which we can define the semantics of such a structure.

```
vnc(topictype_s tt,
  nametype_s nt,
  datatype_s dt,
  scopepattern_c sp,
  matchvalue_c mv,
  cardmin_c cmin,
  cardmax_c cmax)
```

=>

Given the above schema model where V is a VariantNameSchema:

The variant name constraint information item is derived as follows:

```
vnc.tt = GetTopic(V.parent.parent.typeSelector)
vnc.nt = GetTopic(V.parent.typeSelector)
```

```

vnc.dt = GetTopic(V.dataTypeSelector)
vnc.sp = V.scopePattern
vnc.mv = V.match
vnc.cmin = V.cardMin
vnc.cmax = V.cardMax

```

GetTopic(TopicIdentification) => Topic Information Item

=> The interpretation is :

For all VariantNames v where
v.TopicName.Topic.Types contains tt &
v.TopicName.NameType = nt &
v.dt = dt

The Following expression must hold:

```

testScopePattern(v.scope, sp) &
count(v) >= cmin &
count(v) <= cmax &
testRegEx(v.value, mv)

```

4.2.9 Occurrence Schema Item

An *occurrence schema* defines a collection of constraints on occurrences of a given type.

Figure 9 — The Occurrence Schema item

Construct to constrain the occurrences of a topic. OccurrenceSchema items have the following properties:

- 1 **[parent]:** A TopicSchema item.. The containing TopicSchema item.
- 2 **[typeSelector]:** A TopicIdentification item.. The selector that defines which types of occurrence are governed by this constraint.
- 3 **[dataType]:** A string.. The data type of the occurrence must adhere to this type definition.
- 4 **[scopeSelector]:** An ScopePattern item.. The Names whose scope must match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. Again, i would prefer scope to be a constraint. here is my rationale, all constrains are type based including assoc. Scope is just poor mans assoc and assoc doesn't use scope as a selector.]
- 5 **[match]:** A regular expression.. The value of the occurrence must match this regular expression.
- 6 **[cardMin]:** An integer value.. The minimum allowed number of occurrence information items matching this pattern.
- 7 **[cardMax]:** An integer value.. The maximum allowed number of occurrence information items matching this pattern.

4.2.10 Play Role Schema Item

A *playrole schema* defines a collection of constraints on the allowed participation in a given role..

Figure 10 — The PlayRole Schema item

Construct to constrain the participation in given roles. PlayRoleSchema items have the following properties:

- 1 **[parent]:** A TopicSchema item.. The containing TopicSchema item.
- 2 **[associationTypeSelector]:** A TopicIdentification item.. The selector that defines which types of associations

are governed by this constraint.

- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[scopeSelector]**: A ScopePattern item.. The occurrence scope must match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. Again, i would prefer scope to be a constraint. here is my rationale, all constrains are type based including assoc. Scope is just poor mans assoc and assoc doesn't use scope as a selector.]
- 5 **[otherRoleSchema]**: A set of RoleSchema items.. A collections of constraints on the other allowed roles in the association.
- 6 **[cardMin]**: An integer value.. The minimum allowed number of roles within a given matching this pattern.
- 7 **[cardMax]**: An integer value.. The maximum allowed number of roles matching this pattern.

4.2.11 Role Schema Item

A *role schema* defines a collection of constraints on roles in a given association.

Figure 11 — The Role Schema item

Construct to constrain the participation in given roles. PlayRoleSchema items have the following properties:

- 1 **[parent]**: A PlayRoleSchema item or AssociationSchema item.. The containing PlayRoleSchema or AssociationSchema item.
- 2 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 3 **[cardMin]**: An integer value.. The minimum allowed number of roles within a given matching this pattern.
- 4 **[cardMax]**: An integer value.. The maximum allowed number of roles matching this pattern.
- 5 **[allPlayersFrom]**: A set of TopicIdentification items.. A list of topic types where the players must be an instance of one of the given types.
- 6 **[oneOf]**: A set of TopicIdentification items.. A list of topics where the players must be one of those contained in the set.
- 7 **[isUnique]**: A boolean value.. Indicates if this role player is unique to main topic [gdm -er?? squeeze me?]

4.2.12 Association Schema Item

An *association schema* defines a collection of constraints on a given association.

Figure 12 — The Association Schema item

Construct to constrain association information items. AssociationRoleSchema items have the following properties:

- 1 **[parent]**: A TopicMapSchema item.. The containing TopicMapSchema item.
- 2 **[typeSelector]**: A TopicIdentification item.. The selector that defines which types of association are governed by this constraint.
- 3 **[roleSchema]**: A set of RoleSchema items.. A collection of constraints governing the AssociationRole information items of the association item.
- 4 **[reifiedBy]**: A TopicIdentification item.. Governs the allowed type of topics that reify this association item.

4.2.13 TMCL-Schema descriptions of warnings

Each constraint has "warning" equivalent. Warnings typically define more detailed constraints.

cardMin	-> cardMinWarning
cardMax	-> cardMaxWarning
dataType	-> dataTypeWarning
oneOf	-> oneOfWarning
match	-> matchWarning
allPlayersFrom	-> allPlayersFromWarning

If 'warning' constraint is violated then ConflictItem with category 'Warning' is generated during the validation process.

4.3 Interpretation of TMCL-Schema

From a logical perspective each TMCL-Schema is equivalent to a set of basic constraints which can be validated independently. A list of predefined basic constraint types and their interpretation is provided below.

4.3.1 Subject Locator basic constraints

4.3.1.1 SubjectLocatorCardMinConstraint

The minimum allowed number of identifiers matching a pattern SubjectLocatorCardMinConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[matchSelector]**: A Regular Expression.. The locator value must be conform to the regular expression to be valid.
- 4 **[cardMin]**: An integer value.. The minimum allowed number of identifiers matching this pattern.

SubjectLocatorCardMinConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $MatchSel, $CardMin suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,SubjectLocatorCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and matchSelector($Constraint,$MatchSel)
  and cardMin($Constraint,$CardMin)
satisfies
  existsAtLeast $CardMin $SubjectLocator suchAs
    subjectLocator($Instance,$SubjectLocator)
    and testRegEx($SubjectIdentifier,$MatchSel)

```

4.3.1.2 SubjectLocatorCardMaxConstraint

The maximum allowed number of identifiers matching a pattern SubjectLocatorCardMaxConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[matchSelector]**: A Regular Expression.. The locator value must be conform to the regular expression to be valid.
- 4 **[cardMax]**: An integer value.. The maximum allowed number of identifiers matching this pattern.

SubjectLocatorCardMaxConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $MatchSel, $CardMax suchAs
  instanceof($Instance,$TopicType)
  and instanceof($Constraint,SubjectLocatorCardMaxConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and matchSelector($Constraint,$MatchSel)
  and cardMax($Constraint,$CardMax)
satisfies
  existsAtMost $CardMax $SubjectLocator suchAs
    subjectLocator($Instance,$SubjectLocator)
    and testRegEx($SubjectIdentifier,$MatchSel)

```

4.3.2 Subject Identifier basic constraints

4.3.2.1 SubjectIdentifierCardMinConstraint

The minimum allowed number of identifiers matching a pattern SubjectIdentifierCardMinConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[matchSelector]**: A Regular Expression.. The locator value must be conform to the regular expression to be valid.
- 4 **[cardMin]**: An integer value.. The minimum allowed number of identifiers matching this pattern.

SubjectIdentifierCardMinConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $MatchSel, $CardMin suchAs
  instanceof($Instance,$TopicType)
  and instanceof($Constraint,SubjectLocatorCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and matchSelector($Constraint,$MatchSel)
  and cardMin($Constraint,$CardMin)
satisfies
  existsAtLeast $CardMin $SubjectIdentifier suchAs
    subjectIdentifier($Instance,$SubjectIdentifier)
    and testRegEx($SubjectIdentifier,$MatchSel)

```

4.3.2.2 SubjectIdentifierCardMaxConstraint

The maximum allowed number of identifiers matching a pattern SubjectIdentifierCardMaxConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[matchSelector]**: A Regular Expression.. The locator value must be conform to the regular expression to be valid.
- 4 **[cardMax]**: An integer value.. The maximum allowed number of identifiers matching this pattern.

SubjectIdentifierCardMaxConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $MatchSel, $CardMax suchAs
  instanceof($Instance,$TopicType)

```

```

and instanceOf($Constraint,SubjectIdentifierCardMaxConstraint)
and topicTypeSelector($Constraint,$TopicType)
and matchSelector($Constraint,$MatchSel)
and cardMax($Constraint,$CardMax)
satisfies
existsAtMost $CardMax $SubjectIdentifier suchAs
  subjectIdentifier($Instance,$SubjectIdentifier)
  and testRegEx($SubjectIdentifier,$MatchSel)

```

4.3.3 Topic name basic constraints

4.3.3.1 TopicNameCardMinConstraint

The minimum allowed number of names matching this pattern. TopicNameCardMinConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[nameTypeSelector]**: A TopicIdentification item.. The type used to identify to which names the constraints should be applied.
- 4 **[scopeSelector]**: An ScopePattern item.. Names whose scope match this pattern must adhere to the topic name constraint
- 5 **[cardMin]**: An integer value.. The minimum allowed number of names matching this pattern.

TopicNameCardMinConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $NameType, $ScopeSel, $CardMin suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,TopicNameCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and nameTypeSelector($Constraint,$NameType)
  and scopeSelector($Constraint,$ScopeSel)
  and cardMin($Constraint,$CardMin)
satisfies
existsAtLeast $CardMin $TopicName suchAs
  topicName($Instance,$TopicName)
  and instanceOf($TopicName,$NameType)
  and scope($TopicName,$Scope)
  and matchScope($Scope,$ScopeSel)

```

4.3.3.2 TopicNameCardMaxConstraint

The maximum allowed number of names matching this pattern. TopicNameCardMaxConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[nameTypeSelector]**: A TopicIdentification item.. The type used to identify to which names the constraints should be applied.
- 4 **[scopeSelector]**: An ScopePattern item.. Names whose scope match this pattern must adhere to the topic name constraint
- 5 **[cardMax]**: An integer value.. The maximum allowed number of names matching this pattern.

TopicNameCardMax Constraint interpretation

```

every $Instance,$Constraint, $TopicType, $NameType, $ScopeSel, $CardMax suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,TopicNameCardMaxConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and nameTypeSelector($Constraint,$NameType)
  and scopeSelector($Constraint,$ScopeSel)
  and cardMax($Constraint,$CardMax)
satisfies
  existsAtMost $CardMax $TopicName suchAs
    topicName($Instance,$TopicName)
    and instanceOf($TopicName,$NameType)
    and scope($TopicName,$Scope)
    and matchScope($Scope,$ScopeSel)

```

4.3.3.3 TopicNameMatchConstraint

The regular expression allowed for topic names. TopicNameMatchConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[nameTypeSelector]**: A TopicIdentification item.. The type used to identify to which names the constraints should be applied.
- 4 **[scopeSelector]**: An ScopePattern item.. Names whose scope match this pattern must adhere to the topic name constraint
- 5 **[match]**: A regular expression.. The value of the name must match this regular expression.

TopicNameMatch Constraint interpretation

```

every $Instance,$Constraint, $TopicType, $NameType, $ScopeSel, $Match suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,TopicNameMatchConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and nameTypeSelector($Constraint,$NameType)
  and scopeSelector($Constraint,$ScopeSel)
  and match($Constraint,$Match)
satisfies
  every $TopicName,$Value suchAs
    topicName($Instance,$TopicName)
    and value($TopicName,$Value)
    and instanceOf($TopicName,$NameType)
    and scope($TopicName,$Scope)
    and matchScope($Scope,$ScopeSel)
  satisfies
    testRegEx($Value,$Match)

```

4.3.4 Variant name basic constraints

4.3.4.1 VariantNameCardMinConstraint

The minimum allowed number of variant names matching this pattern. VariantNameCardMinConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint

- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[nameTypeSelector]**: A TopicIdentification item.. The type used to identify to which names the constraints should be applied.
- 4 **[nameScopeSelector]**: An ScopePattern item.. Names whose scope match this pattern must adhere to the variant name constraint
- 5 **[scopeSelector]**: An ScopePattern item.. Variant Names whose scope match this pattern must adhere to the variant name constraint
- 6 **[dataTypeSelector]**: A string.. An expression used to constrain the allowed data type. [gdm - maybe we say its an IRI and that can be interpreted locally, perhaps use a regex?]
- 7 **[cardMin]**: An integer value.. The minimum allowed number of variant names matching this pattern.

VariantNameCardMinConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $NameType, $NameScopeSel, $ScopeSel, $DataTypeSel, $CardMin suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,TopicNameCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and nameTypeSelector($Constraint,$NameType)
  and nameScopeSelector($Constraint,$NameScopeSel)
  and scopeSelector($Constraint,$ScopeSel)
  and dataTypeSelector($Constraint,$DataTypeSel)
  and cardMin($Constraint,$CardMin)
satisfies
  every $TopicName suchAs
    topicName($Instance,$TopicName)
    and instanceOf($TopicName,$NameType)
    and scope($TopicName,$Scope)
    and matchScope($Scope,$NameScopeSel)
  satisfies
    existsAtLeast $CardMin $VariantName suchAs
      variantName($TopicName,$VariantName)
      and datatype($VariantName,$VariantDataType)
      and $VariantDataType==$DataTypeSel
      and scope($VariantName,$VariantScope)
      and matchScope($VariantScope,$ScopeSel)

```

4.3.4.2 VariantNameCardMaxConstraint

The maximum allowed number of variant names matching this pattern. VariantNameCardMaxConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[nameTypeSelector]**: A TopicIdentification item.. The type used to identify to which names the constraints should be applied.
- 4 **[nameScopeSelector]**: An ScopePattern item.. Names whose scope match this pattern must adhere to the variant name constraint
- 5 **[scopeSelector]**: An ScopePattern item.. Names whose scope match this pattern must adhere to the topic name constraint
- 6 **[dataTypeSelector]**: A string.. An expression used to constrain the allowed datatype. [gdm - maybe we say its an IRI and that can be interpreted locally, perhaps use a regex?]
- 7 **[cardMax]**: An integer value.. The maximum allowed number of variant names matching this pattern.

VariantNameCardMaxConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $NameType, $NameScopeSel, $ScopeSel, $DataTypeSel, $CardMin suchAs
  instanceof($Instance,$TopicType)
  and instanceof($Constraint,TopicNameCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and nameTypeSelector($Constraint,$NameType)
  and nameScopeSelector($Constraint,$NameScopeSel)
  and scopeSelector($Constraint,$ScopeSel)
  and dataTypeSelector($Constraint,$DataTypeSel)
  and cardMin($Constraint,$CardMin)
satisfies
  every $TopicName suchAs
    topicName($Instance,$TopicName)
    and instanceof($TopicName,$NameType)
    and scope($TopicName,$Scope)
    and matchScope($Scope,$NameScopeSel)
  satisfies
    existsAtLeast $CardMax $VariantName suchAs
      variantName($TopicName,$VariantName)
      and datatype($VariantName,$VariantDataType)
      and $VariantDataType==$DataTypeSel
      and scope($VariantName,$VariantScope)
      and matchScope($VariantScope,$ScopeSel)

```

4.3.4.3 VariantNameMatchConstraint

The regular expression allowed for variant names. VariantNameMatchConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[nameTypeSelector]**: A TopicIdentification item.. The type used to identify to which names the constraints should be applied.
- 4 **[nameScopeSelector]**: An ScopePattern item.. Names whose scope match this pattern must adhere to the variant name constraint
- 5 **[scopeSelector]**: An ScopePattern item.. Names whose scope match this pattern must adhere to the topic name constraint
- 6 **[dataTypeSelector]**: A string.. An expression used to constrain the allowed datatype. [gdm - maybe we say its an IRI and that can be interpreted locally, perhaps use a regex?]
- 7 **[match]**: A regular expression.. The value of the variant must match this regular expression.

VariantNameMatchConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $NameType, $NameScopeSel, $ScopeSel, $DataTypeSel, $Match suchAs
  instanceof($Instance,$TopicType)
  and instanceof($Constraint,TopicNameCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and nameTypeSelector($Constraint,$NameType)
  and nameScopeSelector($Constraint,$NameScopeSel)
  and scopeSelector($Constraint,$ScopeSel)
  and dataTypeSelector($Constraint,$DataTypeSel)
  and match($Constraint,$Match)
satisfies
  every $TopicName suchAs
    topicName($Instance,$TopicName)
    and instanceof($TopicName,$NameType)
    and scope($TopicName,$Scope)
    and matchScope($Scope,$NameScopeSel)

```

```

satisfies
  every $VariantName,$Value suchAs
    variantName($TopicName,$VariantName)
    and value($VariantName,$Value)
    and datatype($VariantName,$VariantDataType)
    and $VariantDataType==$DataTypeSel
    and scope($VariantName,$VariantScope)
    and matchScope($VariantScope,$ScopeSel)
  satisfies
    testRegEx($Value,$Match)

```

4.3.5 Occurrence basic constraints

4.3.5.1 OccurrenceCardMinConstraint

The minimum allowed number of occurrences matching this pattern. OccurrenceCardMinConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[occurrenceTypeSelector]**: A TopicIdentification item.. The type used to identify to which occurrences the constraints should be applied.
- 4 **[scopeSelector]**: An ScopePattern item.. Occurrences whose scope match this pattern must adhere to the occurrence constraint
- 5 **[cardMin]**: An integer value.. The minimum allowed number of occurrences matching this pattern.

OccurrenceCardMinConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $OccurrenceType, $ScopeSel, $CardMin suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,OccurrenceCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and occurrenceTypeSelector($Constraint,$OccurrenceType)
  and scopeSelector($Constraint,$ScopeSel)
  and cardMin($Constraint,$CardMin)
satisfies
  existsAtLeast $CardMin $Occurrence suchAs
    occurrence($Instance,$Occurrence)
    and instanceOf($Occurrence,$OccurrenceType)
    and scope($Occurrence,$ScopeSel)
    and matchScope($Scope,$ScopeSel)

```

4.3.5.2 OccurrenceCardMaxConstraint

The maximum allowed number of occurrences matching this pattern. OccurrenceCardMaxConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[occurrenceTypeSelector]**: A TopicIdentification item.. The type used to identify to which occurrences the constraints should be applied.
- 4 **[scopeSelector]**: An ScopePattern item.. Occurrences whose scope match this pattern must adhere to the occurrence constraint
- 5 **[cardMax]**: An integer value.. The maximum allowed number of occurrences matching this pattern.

OccurrenceCardMax Constraint interpretation

```

every $Instance,$Constraint, $TopicType, $OccurrenceType, $ScopeSel, $CardMax suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,OccurrenceCardMaxConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and occurrenceTypeSelector($Constraint,$OccurrenceType)
  and scopeSelector($Constraint,$ScopeSel)
  and cardMax($Constraint,$CardMax)
satisfies
  existsAtMost $CardMax $Occurrence suchAs
    occurrence($Instance,$Occurrence)
    and instanceOf($Occurrence,$OccurrenceType)
    and scope($Occurrence,$Scope)
    and matchScope($Scope,$ScopeSel)

```

4.3.5.3 OccurrenceMatchConstraint

The regular expression allowed for occurrences. OccurrenceMatchConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant
- 3 **[occurrenceTypeSelector]**: A TopicIdentification item.. The type used to identify to which occurrences the constraints should be applied.
- 4 **[scopeSelector]**: An ScopePattern item.. Occurrences whose scope match this pattern must adhere to the occurrence constraint
- 5 **[match]**: A regular expression.. The value of the occurrence must match this regular expression.

OccurrenceMatch Constraint interpretation

```

every $Instance,$Constraint, $TopicType, $OccurrenceType, $ScopeSel, $Match suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,OccurrenceMatchConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and occurrenceTypeSelector($Constraint,$OccurrenceType)
  and scopeSelector($Constraint,$ScopeSel)
  and match($Constraint,$Match)
satisfies
  every $Occurrence,$Value suchAs
    occurrence($Instance,$Occurrence)
    and value($Occurrence,$Value)
    and instanceOf($Occurrence,$OccurrenceType)
    and scope($Occurrence,$Scope)
    and matchScope($Scope,$ScopeSel)
  satisfies
    testRegEx($Value,$Match)

```

4.3.5.4 OccurrenceDataTypeConstraint

The data type allowed for occurrences. OccurrenceDataTypeConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[topicTypeSelector]**: A Topic Identification item.. A topic type to which the this constraint is relevant

- 3 **[occurrenceTypeSelector]**: A TopicIdentification item.. The type used to identify to which occurrences the constraints should be applied.
- 4 **[scopeSelector]**: An ScopePattern item.. Occurrences whose scope match this pattern must adhere to the occurrence constraint
- 5 **[dataType]**: A string.. The data type of the occurrence must adhere to this type definition.

OccurrenceDataType Constraint interpretation

```

every $Instance,$Constraint, $TopicType, $OccurrenceType, $ScopeSel, $DataType suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,OccurrenceMatchConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and occurrenceTypeSelector($Constraint,$OccurrenceType)
  and scopeSelector($Constraint,$ScopeSel)
  and dataType($Constraint,$DataType)
satisfies
  every $Occurrence,$OccurrenceDataType suchAs
    occurrence($Instance,$Occurrence)
    and datatype($Occurrence,$OccurrenceDataType)
    and instanceOf($Occurrence,$OccurrenceType)
    and scope($Occurrence,$Scope)
    and matchScope($Scope,$ScopeSel)
satisfies
  matchDataType($OccurrenceDataType,$DataType)

```

4.3.6 PlayRole basic constraints

4.3.6.1 PlayRoleCardMinConstraint

The minimum allowed number of roles which topic can play PlayRoleCardMinConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[associationTypeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.
- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[scopeSelector]**: A ScopePattern item.. The occurrence scope must match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. Again, i would prefer scope to be a constraint. here is my rationale, all constrains are type based including assoc. Scope is just poor mans assoc and assoc doesn't use scope as a selector.]
- 5 **[cardMin]**: An integer value.. The minimum allowed number of roles within a given matching this pattern.

PlayRoleCardMinConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $AssociationType, $RoleType,$ScopeSel, $CardMin suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,PlayRoleCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and associationTypeSelector($Constraint,$AssociationType)
  and roleTypeSelector($Constraint,$RoleType)
  and scopeSelector($Constraint,$ScopeSel)
  and cardMin($Constraint,$CardMin)
satisfies
  existsAtLeast $CardMin $Role suchAs

```

```

rolesPlayed($Instance,$Role)
and instanceOf($Role,$RoleType)
and role($Role, $Association)
and instanceOf($Association,$AssociationType)
and scope($Association,$Scope)
and matchScope($Scope,$ScopeSel)

```

4.3.6.2 PlayRoleCardMaxConstraint

The maximum allowed number of roles which topic can play PlayRoleCardMaxConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[associationTypeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.
- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[scopeSelector]**: A ScopePattern item.. The occurrence scope must match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. Again, i would prefer scope to be a constraint. here is my rationale, all constrains are type based including assocs. Scope is just poor mans assoc and assoc doesn't use scope as a selector.]
- 5 **[cardMax]**: An integer value.. The maximum allowed number of roles within a given matching this pattern.

PlayRoleCardMaxConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $AssociationType, $RoleType,$ScopeSel, $CardMax suchAs
instanceOf($Instance,$TopicType)
and instanceOf($Constraint,PlayRoleCardMaxConstraint)
and topicTypeSelector($Constraint,$TopicType)
and associationTypeSelector($Constraint,$AssociationType)
and roleTypeSelector($Constraint,$RoleType)
and scopeSelector($Constraint,$ScopeSel)
and cardMax($Constraint,$CardMax)
satisfies
existsAtMost $CardMax $Role suchAs
rolesPlayed($Instance,$Role)
and instanceOf($Role,$RoleType)
and role($Role, $Association)
and instanceOf($Association,$AssociationType)
and scope($Association,$Scope)
and matchScope($Scope,$ScopeSel)

```

4.3.7 OtherRole basic constraints

4.3.7.1

The minimum allowed number of other roles which related topics can play OtherRoleCardMinConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[associationTypeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.
- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[otherRoleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of other association roles are governed by this constraint.

- 5 **[scopeSelector]**: A ScopePattern item.. The occurrence scope must match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. Again, i would prefer scope to be a constraint. here is my rationale, all constrains are type based including assoc. Scope is just poor mans assoc and assoc doesn't use scope as a selector.]
- 6 **[cardMin]**: An integer value.. The minimum allowed number of roles within a given matching this pattern.

OtherRoleCardMinConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $AssociationType, $RoleType, $OtherRoleType,$ScopeSel, $CardMin suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,PlayRoleCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and associationTypeSelector($Constraint,$AssociationType)
  and roleTypeSelector($Constraint,$RoleType)
  and otherRoleTypeSelector($Constraint,$OtherRoleType)
  and scopeSelector($Constraint,$ScopeSel)
  and cardMin($Constraint,$CardMin)
satisfies
  every $Role, $Association suchAs
    rolesPlayed($Instance,$Role)
    and instanceOf($Role,$RoleType)
    and role($Role, $Association)
    and instanceOf($Association,$AssociationType)
    and scope($Association,$Scope)
    and matchScope($Scope,$ScopeSel)
satisfies
  existsAtLeast $CardMin $OtherRole suchAs
    role($OtherRole, $Association)
    and instanceOf($OtherRole,$OtherRoleType)

```

4.3.7.2

The maximum allowed number of other roles which related topics can play OtherRoleCardMaxConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[associationTypeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.
- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[otherRoleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of other association roles are governed by this constraint.
- 5 **[scopeSelector]**: A ScopePattern item.. The occurrence scope must match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. Again, i would prefer scope to be a constraint. here is my rationale, all constrains are type based including assoc. Scope is just poor mans assoc and assoc doesn't use scope as a selector.]
- 6 **[cardMax]**: An integer value.. The maximum allowed number of roles within a given matching this pattern.

OtherRoleCardMaxConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $AssociationType, $RoleType, $OtherRoleType,$ScopeSel, $CardMax suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,PlayRoleCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and associationTypeSelector($Constraint,$AssociationType)
  and roleTypeSelector($Constraint,$RoleType)
  and otherRoleTypeSelector($Constraint,$OtherRoleType)

```

```

and scopeSelector($Constraint,$ScopeSel)
and cardMax($Constraint,$CardMax)
satisfies
  every $Role, $Association suchAs
    rolesPlayed($Instance,$Role)
    and instanceOf($Role,$RoleType)
    and role($Role, $Association)
    and instanceOf($Association,$AssociationType)
    and scope($Association,$Scope)
    and matchScope($Scope,$ScopeSel)
  satisfies
    existsAtMost $CardMax $OtherRole suchAs
      role($OtherRole, $Association)
      and instanceOf($OtherRole,$OtherRoleType)

```

4.3.7.3

A list of topic types where the players must be an instance of one of the given types. OtherRoleAllPlayersFromConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[associationTypeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.
- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[otherRoleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of other association roles are governed by this constraint.
- 5 **[scopeSelector]**: A ScopePattern item.. The occurrence scope must match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. Again, i would prefer scope to be a constraint. here is my rationale, all constrains are type based including assocs. Scope is just poor mans assoc and assoc doesn't use scope as a selector.]
- 6 **[allPlayersFrom]**: A set of TopicIdentification items.. A list of topic types where the players must be an instance of one of the given types.

OtherRoleAllPlayersFromConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $AssociationType, $RoleType, $OtherRoleType,$ScopeSel suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,PlayRoleCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and associationTypeSelector($Constraint,$AssociationType)
  and roleTypeSelector($Constraint,$RoleType)
  and otherRoleTypeSelector($Constraint,$OtherRoleType)
  and scopeSelector($Constraint,$ScopeSel)
satisfies
  every $Role, $Association, $OtherRole, $OtherPlayer suchAs
    rolesPlayed($Instance,$Role)
    and instanceOf($Role,$RoleType)
    and role($Role, $Association)
    and instanceOf($Association,$AssociationType)
    and scope($Association,$Scope)
    and matchScope($Scope,$ScopeSel)
    and role($OtherRole, $Association)
    and instanceOf($OtherRole,$OtherRoleType)
    and rolesPlayed($OtherPlayer,$OtherRole)
  satisfies
    some $PlayerType suchAs
      allPlayersFrom($Constraint,$PlayerType)
    satisfies

```

instanceOf(\$OtherPlayer,\$PlayerType)

4.3.7.4

A list of topics where the players must be one of those contained in the set. OtherRoleOneOfConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[associationTypeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.
- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[otherRoleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of other association roles are governed by this constraint.
- 5 **[scopeSelector]**: A ScopePattern item.. The occurrence scope must match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. Again, i would prefer scope to be a constraint. here is my rationale, all constrains are type based including assoc. Scope is just poor mans assoc and assoc doesn't use scope as a selector.]
- 6 **[oneOf]**: A set of TopicIdentification items.. A list of topics where the players must be one of those contained in the set.

OtherRoleOneOfConstraint interpretation

```

every $Instance,$Constraint, $TopicType, $AssociationType, $RoleType, $OtherRoleType,$ScopeSel suchAs
  instanceOf($Instance,$TopicType)
  and instanceOf($Constraint,PlayRoleCardMinConstraint)
  and topicTypeSelector($Constraint,$TopicType)
  and associationTypeSelector($Constraint,$AssociationType)
  and roleTypeSelector($Constraint,$RoleType)
  and otherRoleTypeSelector($Constraint,$OtherRoleType)
  and scopeSelector($Constraint,$ScopeSel)
satisfies
  every $Role, $Association, $OtherRole, $OtherPlayer suchAs
    rolesPlayed($Instance,$Role)
    and instanceOf($Role,$RoleType)
    and role($Role, $Association)
    and instanceOf($Association,$AssociationType)
    and scope($Association,$Scope)
    and matchScope($Scope,$ScopeSel)
    and role($OtherRole, $Association)
    and instanceOf($OtherRole,$OtherRoleType)
    and rolesPlayed($OtherPlayer,$OtherRole)
satisfies
  some $OneOfPlayer suchAs
    oneOf($Constraint,$OneOfPlayer)
satisfies
  $OtherPlayer===$OneOfPlayer

```

4.3.7.5

Indicates if other role player is unique to a main topic OtherRolesUniqueConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[associationTypeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.

- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[otherRoleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of other association roles are governed by this constraint.
- 5 **[scopeSelector]**: A ScopePattern item.. The occurrence scope must match this pattern must adhere to the topic name constraint. [gdm - not sure this is on the right side of the 80/20 rule. Again, i would prefer scope to be a constraint. here is my rationale, all constrains are type based including assocs. Scope is just poor mans assoc and assoc doesn't use scope as a selector.]
- 6 **[isUnique]**: A boolean value.. Indicates if this role player is unique to main topic [gdm -er?? squeeze me?]

OtherRoleIsUniqueConstraint interpretation

every \$Instance1, \$Instance2, \$Constraint, \$TopicType, \$AssociationType, \$RoleType, \$OtherRoleType,\$ScopeSel suchAs

```
instanceOf($Instance1,$TopicType)
and instanceOf($Instance2,$TopicType)

and instanceOf($Constraint,OtherRoleIsUniqueConstraint)
and topicTypeSelector($Constraint,$TopicType)
and associationTypeSelector($Constraint,$AssociationType)
and roleTypeSelector($Constraint,$RoleType)
and otherRoleTypeSelector($Constraint,$OtherRoleType)
and scopeSelector($Constraint,$ScopeSel)
```

```
and exists $OtherPlayer,
    $Role1, $Association1, $Scope1, $OtherRole1,
    $Role2, $Association2, $Scope2, $OtherRole2
```

```
suchAs
    rolesPlayed($Instance1,$Role1)
    and instanceOf($Role1,$RoleType)
    and role($Role1, $Association1)
    and instanceOf($Association1,$AssociationType)
    and scope($Association1,$Scope1)
    and matchScope($Scope1,$ScopeSel)
    and role($OtherRole1, $Association1)
    and instanceOf($OtherRole1,$OtherRoleType)
    and rolesPlayed($OtherPlayer,$OtherRole1)
```

```
    rolesPlayed($Instance2,$Role2)
    and instanceOf($Role2,$RoleType)
    and role($Role2, $Association2)
    and instanceOf($Association2,$AssociationType)
    and scope($Association2,$Scope2)
    and matchScope($Scope2,$ScopeSel)
    and role($OtherRole2, $Association)
    and instanceOf($OtherRole2,$OtherRoleType)
    and rolesPlayed($OtherPlayer,$OtherRole2)
```

```
satisfies
    $Instance1==$Instance2
```

4.3.8 Associaton basic constraints

4.3.8.1 RoleCardMinConstraint

The minimum allowed number of roles RoleCardMinConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint

- 2 **[typeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.
- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[cardMin]**: An integer value.. The minimum allowed number of roles.

RoleCardMinConstraint interpretation

```

every $Association,$Constraint, $AssociationType, $RoleType, $CardMin suchAs
  instanceOf($Association,$AssociationType)
  and instanceOf($Constraint,RoleCardMinConstraint)
  and typeSelector($Constraint,$AssociationType)
  and roleTypeSelector($Constraint,$RoleType)
  and cardMin($Constraint,$CardMin)
satisfies
  existsAtLeast $CardMin $Role suchAs
    role($Role, $Association)
    and instanceOf($Role,$RoleType)

```

4.3.8.2 RoleCardMaxConstraint

The maximum allowed number of roles RoleCardMaxConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[typeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.
- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[cardMax]**: An integer value.. The maximum allowed number of roles.

RoleCardMaxConstraint interpretation

```

every $Association,$Constraint, $AssociationType, $RoleType, $CardMax suchAs
  instanceOf($Association,$AssociationType)
  and instanceOf($Constraint,RoleCardMaxConstraint)
  and typeSelector($Constraint,$AssociationType)
  and roleTypeSelector($Constraint,$RoleType)
  and cardMax($Constraint,$CardMax)
satisfies
  existsAtMost $CardMax $Role suchAs
    role($Role, $Association)
    and instanceOf($Role,$RoleType)

```

4.3.8.3 RoleAllPlayersFromConstraint

A list of topic types where the players must be an instance of one of the given types. RoleAllPlayersFromConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[typeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.
- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are

governed by this constraint.

- 4 **[allPlayersFrom]**: A set of TopicIdentification items.. A list of topic types where the players must be an instance of one of the given types.

RoleAllPlayersFromConstraint interpretation

every \$Association, \$Role, \$Player, \$Constraint, \$AssociationType, \$RoleType suchAs

and instanceOf(\$Constraint,RoleAllPlayersFromConstraint)
 and typeSelector(\$Constraint,\$AssociationType)
 and roleTypeSelector(\$Constraint,\$RoleType)

instanceOf(\$Association,\$AssociationType)
 and role(\$Role, \$Association)
 and instanceOf(\$Role,\$RoleType)
 and rolesPlayed(\$Player,\$Role)

satisfies

some \$PlayerType suchAs
 allPlayersFrom(\$Constraint,\$PlayerType)
 satisfies
 instanceOf(\$Player,\$PlayerType)

4.3.8.4 RoleOneOfConstraint

A list of topics where the players must be one of those contained in the set. RoleOneOfConstraint items have the following properties:

- 1 **[schemaID]**: A string.. Reference to a Topic Map Schema which contains this constraint
- 2 **[typeSelector]**: A TopicIdentification item.. The selector that defines which types of associations are governed by this constraint.
- 3 **[roleTypeSelector]**: A TopicIdentification item.. The selector that defines which types of association roles are governed by this constraint.
- 4 **[oneOf]**: A set of TopicIdentification items.. A list of topics where the players must be one of those contained in the set.

RoleOneOfConstraint interpretation

every \$Association, \$Role, \$Player, \$Constraint, \$AssociationType, \$RoleType suchAs

and instanceOf(\$Constraint,RoleOneOfConstraint)
 and typeSelector(\$Constraint,\$AssociationType)
 and roleTypeSelector(\$Constraint,\$RoleType)

instanceOf(\$Association,\$AssociationType)
 and role(\$Role, \$Association)
 and instanceOf(\$Role,\$RoleType)
 and rolesPlayed(\$Player,\$Role)

satisfies

some \$OneOfPlayer suchAs
 oneOf(\$Constraint,\$OneOfPlayer)
 satisfies
 \$Player==\$OneOfPlayer

4.4 TMCL-Rule

TMCL-Rule allows to declare set of assertions about topic maps. It is a rule-based language which leverages TMQL constructs for specifying conditions and assertions.

TMCL-Rule is close to ISO/IEC 19757-3 (Document Schema Definition Languages (DSDL)- Part 3: Rule-based validation Schematron). Schematron allows to define validation rules for XML documents. TMCL-Rule leverages experience from other rule-based languages and allows specifying constraints based on TMDM.

4.4.1 RuleItem

A *RuleItem* defines a set of assertions about a topic map. The *RuleItem* consists of an optional selector, and one or more assertions.

A constraint in a form of a if-then rule *RuleItem* items have the following properties:

- 1 **[parent]**: TopicMapSchema. Reference to a parent topic map schema
- 2 **[ruleID]**: A string.. Is used to identify a rule in a schema
- 3 **[ruleName]**: A string.. Is used to provide a name for a rule
- 4 **[selector]**: A string, optional.. A TMQL expression which generates a list of tuples
- 5 **[assertItem]**: A set of AssertItems. AssertItems specify constraints in a context of a selector and specify error messages

4.4.2 AssertItem

A constraint in a form of a if-then rule *AssertItem* items have the following properties:

- 1 **[parent]**: RuleItem. Reference to a parent RuleItem
- 2 **[constraintID]**: A string.. Is used to identify a constraint in a schema
- 3 **[type]**: TopicIdentification, optional.. reference to a category of an assertion, default category is "Error"
- 4 **[test]**: A string, optional.. TMQL expression, can include variable \$self for addressing list of tuples produced by selector
- 5 **[every]**: A string, optional. TMQL expression, requires a selector. It is a shortcut for an expression "every \$1, \$2... in 'tuple list' satisfies P(\$1,\$2,...)"
- 6 **[some]**: A string, optional.. TMQL expression, requires a selector. It is a shortcut for an expression "some \$1, \$2... in 'tuple list' satisfies P(\$1,\$2,...)"
- 7 **[message]**: A string. Contains a human-oriented explanation of a conflict . It can include TMQL sub-expressions with \$self, \$1,\$2... variables

Example 1: Topic map must have more than 20 topics of a "musician" type.

Example 2: Topic map must have a topic for a composer who was born in Milan.

4.5 Conflict Items

Conflict Items allow to represent results of a validation process in a standard way. InfoSet notation is used to describe Conflict Items

Basic Conflict Item has a 'message' property which describes a conflict in a human-readable format. Specific conflicts can record additional information in Conflict Items

4.5.1 ConflictItem

A *Conflict item* represents a conflict generated by a constraint.

A construct to represent a result of a constraint validation. *ConflictItem* items have the following properties:

- 1 **[category]**: TopicIdentification. Reference to a category of a conflict such as 'Error', 'Warning'
- 2 **[schemaID]**: A string.. Reference to Topic Map Schema which contains a constraint which generated a conflict.
- 3 **[constraintID]**: A string.. Reference to a constraint which generates a conflict.
- 4 **[message]**: A string.. Contains a human-oriented explanation of a conflict .

TMCL-Schema constraints generate subtypes of a ConflictItem which correspond to specific constraints:

- SubjectLocatorCardMinConflictItem
- SubjectLocatorCardMaxConflictItem
- SubjectIdentifierCardMinConflictItem
- SubjectIdentifierCardMaxConflictItem
- TopicNameCardMinConflictItem
- TopicNameCardMaxConflictItem
- TopicNameMatchConflictItem
- VariantNameCardMinConflictItem
- VariantNameCardMaxConflictItem
- VariantNameMatchConflictItem
- OccurrenceCardMinConflictItem
- OccurrenceCardMaxConflictItem
- OccurrenceCardMatchConflictItem
- OccurrenceDataTypeConflictItem
- PlayRoleCardMinConflictItem
- PlayRoleCardMaxConflictItem
- OtherRoleCardMinConflictItem
- OtherRoleCardMaxConflictItem
- OtherRoleAllPlayersFromConflictItem
- OtherRoleOneOfConflictItem
- OtherRoleIsUniqueConflictItem
- RoleCardMinConflictItem
- RoleCardMaxConflictItem
- RoleAllPlayersFromConflictItem
- RoleOneOfConflictItem
- RoleIsUniqueConflictItem

4.6 Syntax for TMCL-Schema

Relax NG Schema

TODO: update based on current schema

TODO: allow mixing TMCL-rules

4.7 Syntax for TMCL-Rule

To be done....

4.8 Combining TMCL-Rule and TMCL-Schema

TMCL-Rule and TMCL-Schema expressions can be combined in the same TMCL schema. It is also possible to insert rules inside of type descriptions. In this case rules have simplified syntax.

TODO: describe simplified syntax for embedding rules into type description In this case there is an implicit context - type

4.9 Topic Map Representation of Constraints

To be done...

4.10 Topic Map Schema References

TMCL enables topic map authors to specify a schema to which the topic map is conformant. This is achieved by reifying the topic map with a topic and assigning an occurrence to that topic of type `TMCLSchemaReference`.

The following PSI is used to denote the occurrence type for schema references.

<http://www.isotopicmaps.org/tmcl/#TMCLSchemaReference>

This is used to type an occurrence on a topic that reifies the topic map in order to reference the schema for this topic map instance. The value of the occurrence must reference a valid TMCL XML representation or a topic of type `Schema`.

4.11 Schema Composition

Schema composition is the ability to take two or more schemas and compose them into a single schema. Given that a schema consists of a set of constraints, schema composition merely takes all the constraints from all schemas being composed and returns a single schema that consists of all constraints. Applications are free to identify and remove redundant constraints and generate conflicts should any constraints be contradictory.

More formally:

```

Given
  Schema : s1, s2
  Constraint : c1, c2, c3, c4, c5
  s1 := {c1, c2, c3}
  s2 := {c4, c5}
That
  Compose(s1, s2) => s3
  s3 := {c1, c2, c3, c4, c5}

```

Bibliography

- [1] *TMCL Requirements*, ISO, 2004
- [2] *TMCL Use Cases*, ISO, 2004
- [3] *TMQL Requirements*, ISO, 2003