

# NetworkedPlanet

**Networked Information – Networked Knowledge**

Tuesday 3<sup>rd</sup> May 2005

TMCL – Topic Maps Constraint Language

Graham Moore

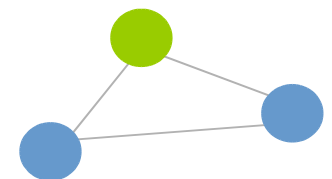
[graham.moore@networkedplanet.com](mailto:graham.moore@networkedplanet.com)

# NetworkedPlanet

## Networked Information – Networked Knowledge

### TMCL Topic Map Constraint Language

- Requirements revisited
- Key concepts and paradigms
- TMCL Schema
- TMCL Rule

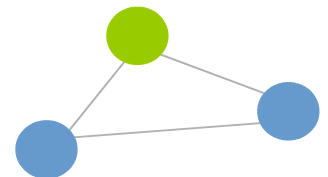


# NetworkedPlanet

## Networked Information – Networked Knowledge

### TMCL Topic Map Constraint Language - Requirements

- Constraint of topic maps
- All aspects of the TMDM should be constrainable
- Introspectable
- In harmony with TMQL
- Well defined model with semantics
- High level syntax, yet
  - Representable as a Topic Map

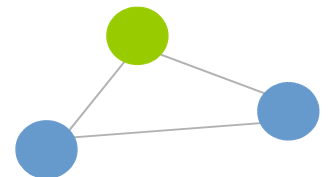


# NetworkedPlanet

## Networked Information – Networked Knowledge

### TMCL Topic Map Constraint Language – Key Concepts

- Class level constraints
- Map level constraints
- Map Validation
- A Schema

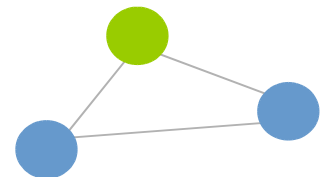


# NetworkedPlanet

## Networked Information – Networked Knowledge

### TMCL Topic Map Constraint Language – Class Level Constraints

- Topics of type ‘person’ must have:
  - at least one name
  - 1 occurrence of type ‘age’
  - be allowed to play the role of ‘employee’ in association ‘employment’
- Associations of type ‘employment’ must have:
  - exactly 1 member with role type ‘employee’, where the player must be of type ‘person’
  - exactly 1 member with role type ‘employer’ where the player must be of type ‘company’



# NetworkedPlanet

## Networked Information – Networked Knowledge

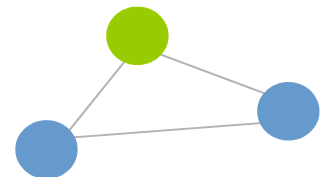
### TMCL Topic Map Constraint Language – Map Level Constraints

Wide ranging and not confined to any one class.

- If a 'person' is related to a 'company' via 'employment' and related to a 'standard' as an 'editor' then they must also be connected to a 'presentation' about that standard.

Note:

1. the class schema can be flexible not to mandate this
2. the map level schema is a kind of business rule, with conditionals.



# NetworkedPlanet

## Networked Information – Networked Knowledge

### TMCL Topic Map Constraint Language – Map Validation

TMCL aims provides a language to detect if something is not valid in respect to a given constraint.

TMCL does not say what it means for a constraint to be broken. In different contexts the meaning of not being valid could be different for the same set of constraints. E.g different map, user, application.

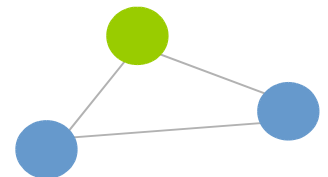
Given:

TopicMap: tm1

Schema : sc1

Then:

Validate(tm1, sc1) => (true, ReportItem\*)



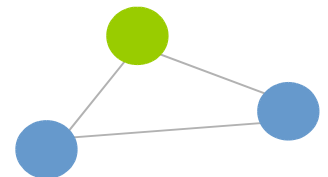
# NetworkedPlanet

## Networked Information – Networked Knowledge

### TMCL Topic Map Constraint Language – A Schema

A Schema is a collection of constraints. Each constraint is a standalone atomic unit that can be Validated against a topic map.

When validating a Map against a schema all the individual constraints must hold true for the map to validate.





# NetworkedPlanet

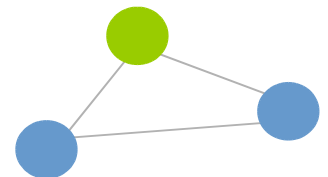
## Networked Information – Networked Knowledge

### TMCL Schema

TMCL Schema is designed to allow the definition of class level constraints.

The standard defines an abstract model and syntax for these constraint definitions.

The following slides show the key constructs.



# NetworkedPlanet

## Networked Information – Networked Knowledge

### Topic Map Schema

Used to group together a collection of constraints.

TopicMapSchema:

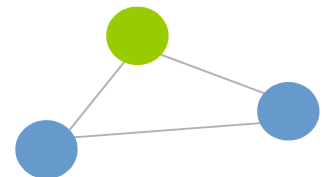
SchemaID # ID which identifies this schema

Name? # defines schema Name

Include \* # URI - list of schemas to include

TopicSchema \* # TopicSchema

AssociationSchema \* # AssociationSchema



# NetworkedPlanet

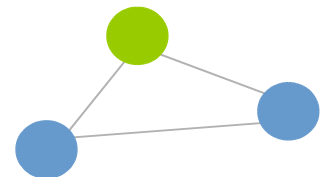
## Networked Information – Networked Knowledge

### Topic Identification

Topic Identification is used to identify exactly 1 topic.

TopicIdentification:

srcLocators	* # URI
subjectIdentifier	* # URI
subjectLocator	* # URI



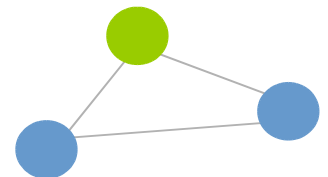
# NetworkedPlanet

**Networked Information – Networked Knowledge**

**Set of topic identification constructs**

TopicSet:

topicIdentification \* # TopicIdentification



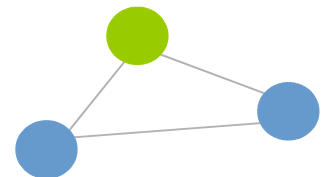
# NetworkedPlanet

Networked Information – Networked Knowledge

## Scope pattern

ScopePattern:

simpleTopicExpression \* # TopicIdentificaiton  
orTopicExpression \* # OrTopicExpression  
typeTopicExpression \* # TopicIdentificaiton



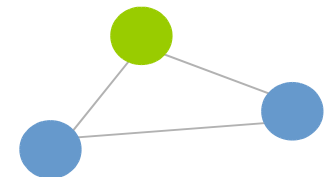
# NetworkedPlanet

**Networked Information – Networked Knowledge**

**Or Topic Expression**

OrTopicExpression:

topicIdentification \* # TopicIdentificaiton



# NetworkedPlanet

## Networked Information – Networked Knowledge

### Topic Schema

TopicSchema:

schemaID # ID which identifies this partial type description

type # TopicIdentification

subjectAddressSchema \* # SubjectAddressSchema

subjectIndicatorSchema \* # SubjectIndicatorSchema

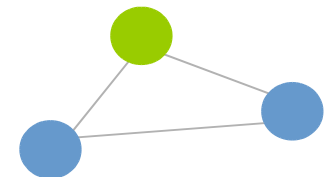
baseNameSchema \* # BaseNameSchema

internalOccurrenceSchema \* # InternalOccurrenceSchem

externalOccurrenceSchema \* # ExternalOccurrenceSchema

playRoleSchema \* # PlayRoleSchema

oneOfSchema? # OneOfSchema



# NetworkedPlanet

**Networked Information – Networked Knowledge**

## **Subject Indicator Schema**

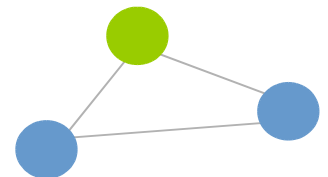
Constrains the cardinality and shape of subject indicator locator.

SubjectIndicatorSchema:

cardMin # Integer?

cardMax # Integer?

match # Regular Expression\*





# NetworkedPlanet

**Networked Information – Networked Knowledge**

## **Subject Address Schema**

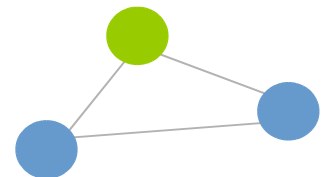
Constrains the cardinality and shape of subject address locator.

SubjectAddressSchema:

cardMin # Integer?

cardMax # Integer?

match # Regular Expression?



# NetworkedPlanet

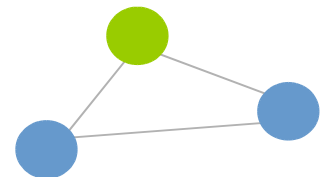
## Networked Information – Networked Knowledge

### Base Name Schema

Constrains topic names.

BaseNameSchema:

- type # TopicIdentification
- scope # ScopePattern?
- cardMin # Integer?
- cardMax # Integer?
- dataType # xsd and custom xml schemas?
- oneOf # String\*
- match # Regular Expression\*
- variantSchema # VariantSchema\*



# NetworkedPlanet

## Networked Information – Networked Knowledge

### Variant Schema

Constrains on variants.

VariantSchema:

scope # ScopePattern?

dataType # xsd and custom xml schemas?

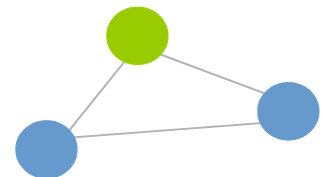
cardMin # Integer?

cardMax # Integer?

oneOf # String\*

match # Regular Expression\*

Note: dataType for variants is part of a pattern, not a constraint



# NetworkedPlanet

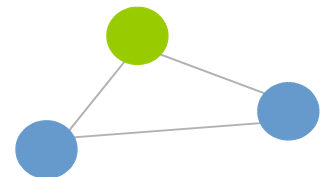
## Networked Information – Networked Knowledge

### Internal Occurrence Schema

Constrains internal occurrences.

InternalOccurrenceSchema:

- type # TopicIdentification
- scope # ScopePattern?
- cardMin # Integer?
- cardMax # Integer?
- dataType # xsd and custom schemas?
- oneOf # String\*
- match # Regular Expression \*
- minExclusive # String?
- minInclusive # String?
- maxExclusive # String?
- maxInclusive # String?



# NetworkedPlanet

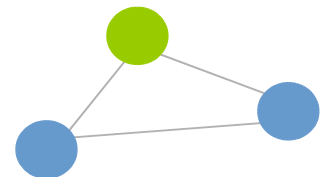
## Networked Information – Networked Knowledge

### External Occurrence Schema

Constrains external occurrences.

ExternalOccurrenceSchema:

- type # TopicIdentification
- scope # ScopePattern?
- cardMin # Integer?
- cardMax # Integer?
- oneOf # URI\*
- match # Regular Expression \*



# NetworkedPlanet

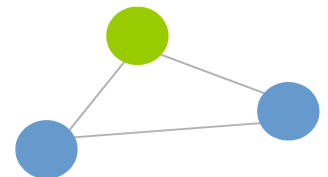
**Networked Information – Networked Knowledge**

## **One Of Schema**

One of is used to defined a controlled vocabulary.

OneOfSchema:

oneOf # TopicSet



# NetworkedPlanet

## Networked Information – Networked Knowledge

### Plays Role Schema

Constraints the nature of participation in associations.

PlayRoleSchema:

associationType # TopicIdentification

roleType # TopicIdentification

otherRoleType # TopicSet //association type, roleType and

otherRoleTypes define signature

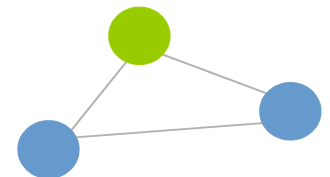
scope # ScopePattern

cardMin # Integer

cardMax # Integer

otherRoles # RoleSchema\*

otherPlayers # OtherPlayerSchema\*



# NetworkedPlanet

Networked Information – Networked Knowledge

## Other Player Schema

Constraints the nature of other players of the association

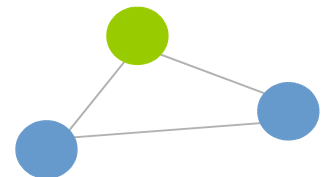
OtherPlayerSchema:

cardMin # Integer

cardMax # Integer

allPlayersFrom # TopicSet // list of Types

oneOf # TopicSet // list of topics





# NetworkedPlanet

Networked Information – Networked Knowledge

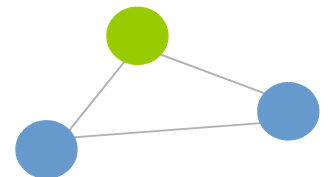
## Association Schema

Constrains classes of association.

AssociationSchema:

    schemaID # ID which identifies this partial association type  
description  
    type # TopicIdentification  
    associationSignature # AssociationSignatureSchema+

Note: It is possible to have different signatures with associations of the same type. It allows to use 'generalized' associations



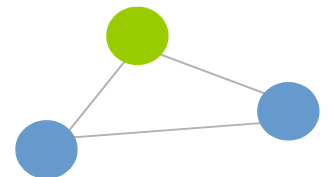
# NetworkedPlanet

Networked Information – Networked Knowledge

## Association Signature Schema

Constrains specific association signature

AssociationSignatureSchema:  
signature # TopicSet  
roleSchema # RoleSchema+



# NetworkedPlanet

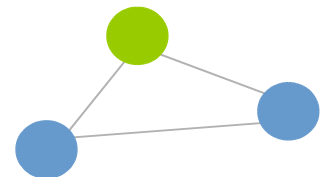
Networked Information – Networked Knowledge

## Role Schema

Constraints the nature of roles on associations of specific types.

RoleSchema:

```
roleType # TopicIdentification  
cardMin # Integer  
cardMax # Integer  
allPlayersFrom # TopicSet // list of Types  
oneOf # TopicSet // list of topics
```



# NetworkedPlanet

Networked Information – Networked Knowledge

## Formal Explanation of the Constraint Types

InternalOccurrenceMinExclusive

TopicSchema:

Type = \$Type

internalOccurrenceSchema:

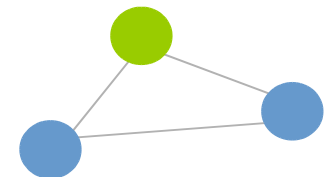
type = \$OcType

scope = \$ScopePattern

match = \$PatternSet

minExclusive = \$MinVal

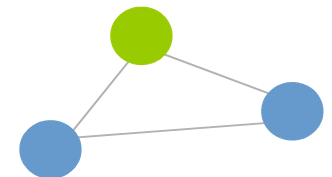
Every \$Inst SuchAs instanceOf(\$Inst,\$Type) Satisfies  
Every \$OcVal SuchAs internalOccurrence  
(\$Inst,\$OcType,\$OcVal)@\$Scope and  
matchScope(\$Scope,\$ScopePattern) Satisfies  
\$OcVal > \$MinVal



# NetworkedPlanet

## Networked Information – Networked Knowledge

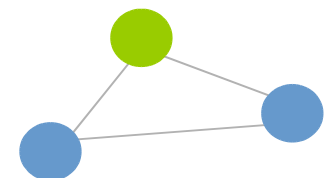
tmcl:TopicMapSchema(\$X)  
tmcl:SchemaID(\$Schema, \$ID)  
tmcl:SchemaName(\$Schema,\$Name)  
tmcl:Include(\$Schema,\$IncludedSchema)  
tmcl:TopicSchema(\$TMSchema,\$TopicSchema)  
tmcl:AssociationSchema(\$TMSchema,\$AssociationSchema)  
tmcl:SubjectAddressSchema(\$TopicSchema,\$SubjectAddressSchema)  
tmcl:SubjectIndicatorSchemas(\$TopicSchema,\$SubjectIndicatorSchema)  
tmcl:BaseNameSchema(\$TopicSchema,\$BaseNameSchema)  
tmcl:InternalOccurrenceSchema(\$TopicSchema,\$InternalOccurrenceSchema)  
tmcl:ExternalOccurrenceSchema(\$TopicSchema,\$ExternalOccurrenceSchema)  
tmcl:PlayRoleSchema(\$TopicSchema,\$PlayRoleSchema)  
tmcl:OneOfSchema(\$TopicSchema,\$OneOfSchema)  
tmcl:Type(\$Schema,\$Topic)  
tmcl:CardMin(\$Schema,\$CardMin)  
tmcl:CardMax(\$Schema,\$CardMax)  
tmcl:Match(\$Schema,\$Match)



# NetworkedPlanet

## Networked Information – Networked Knowledge

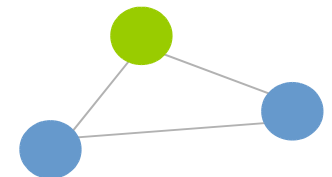
tmcl:CardMinNotify(\$Schema,\$CardMinNotify)  
tmcl:CardMaxNotify(\$Schema,\$CardMaxNotify)  
tmcl:MatchNotify(\$Schema,\$MatchNotify)  
tmcl:MinExclusive(\$Schema,\$MinExclusive)  
tmcl:MinInclusive(\$Schema,\$MinInclusive)  
tmcl:MinInclusive(\$Schema,\$MinInclusive)  
tmcl:MaxInclusive(\$Schema,\$MaxInclusive)  
tmcl:MinExclusiveNotify(\$Schema,\$MinExclusiveNotify)  
tmcl:MinInclusiveNotify(\$Schema,\$MinInclusiveNotify)  
tmcl:MinInclusiveNotify(\$Schema,\$MinInclusiveNotify)  
tmcl:MaxInclusiveNotify(\$Schema,\$MaxInclusiveNotify)  
tmcl:DataType(\$Schema,\$DataType)  
tmcl:OneOfValue(\$Schema,\$OneOfvalue),  
tmcl:DataTypeNotify(\$Schema,\$DataTypeNotify),



# NetworkedPlanet

## Networked Information – Networked Knowledge

tmcl:OneOfvalueNotify(\$Schema,\$OneOfvalueNotify),  
tmcl:VariantSchemas(\$BaseNameSchema,\$VariantSchema)  
tmcl:ScopePattern(\$Schema,\$ScopePatternt)  
tmcl:SimpleTopicExpression(\$Schema,\$Topic)  
tmcl:OrTopicExpression(\$Schema,\$Topic)  
tmcl:TypeTopicExpression(\$Schema,\$Topic)  
tmcl:AllPlayersFrom(\$Schema,\$Topic)  
tmcl:oneOfTopic(\$Schema,\$Topic)  
tmcl:OtherRoleType(\$Schema,\$Topic)  
tmcl:OtherRole(\$Schema,\$RoleSchema)  
tmcl:OtherPlayer(\$Schema,\$PlayerSchema)  
tmcl:AssociationSignature(\$Schema,\$AssociationSignature)  
tmcl:Signature(\$Schema,\$Topic)  
tmcl:RoleSchema(\$Schema,\$RoleSchema)



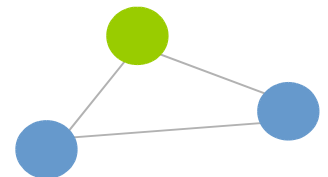
# NetworkedPlanet

## Networked Information – Networked Knowledge

### 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).





# NetworkedPlanet

Networked Information – Networked Knowledge

## TopicMapSchema

The TopicMapSchema collects together a set of rules that can be used to validate a topic map.

TopicMapSchema:

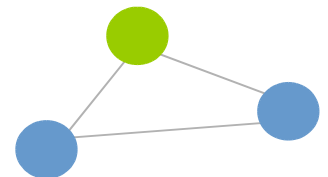
SchemaID # defines schema ID

Name? # defines schema Name

Includes \* # URI - list of schemas to include

RuleItem\* # set of rules

DiagnosticItem\* # provides more specific details for  
assertions and reports



# NetworkedPlanet

## Networked Information – Networked Knowledge

### RuleItem

The RuleItem defines set of assertions about topic map.

RuleItem:

ID #defines rule ID

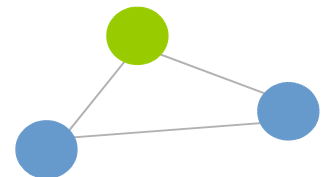
Name? #defines rule Name

ContextItem? #locates topic map data model information  
items to be constrained.

LetItem\* #introduces local variables which  
can be used in assertions and report items

AssertItem\* #if test is negative AssertItem generates ConflictItem

ReportItem\* #if test is positive ReportItem generates NotifyItem



# NetworkedPlanet

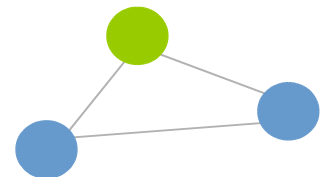
Networked Information – Networked Knowledge

## ContextItem

ContextItem:

ForEvery+ #list of variables

Where #TMQL predicate expression with free  
variables from ForEvery list



# NetworkedPlanet

## Networked Information – Networked Knowledge

AssertItem:

Test #TMQL expression which can include variables from ContextItem and LetItems and returns true or false

Message #string which can include variables(and simple path expressions) from ContextItem and LetItems

Diagnostics #list of DiagnosticItem IDs, is used for detailed notification

Note 1: Rules without ContextItem allow to express constraints defined on full topic map

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

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

Note 2: If constraint can be formulated in a form of "forevery X,Y... where P(X,Y...) satisfies Q(X,Y,...)" preferable form of a rule includes explicit ContextItem.



# NetworkedPlanet

Networked Information – Networked Knowledge

## ConflictItem

ConflictItem :- RuleConflictItem | SchemaConflictItem

## RuleConflictItem

RuleConflictItem:

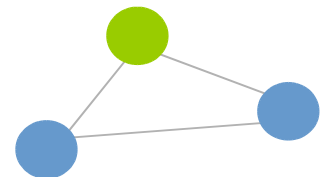
RuleID # reference to rule which generates conflict

TestMessage # string representing Test item from assertion

ContextBinding\* # defines binding for variables from ContextItem

Message # string

DiagosticMessage\* # string



# NetworkedPlanet

## Networked Information – Networked Knowledge

### SchemaConflictItem

SchemaConflictItem:

ConflictTypeID # reference to constraint type which generates conflict

TopicSchemaID\* # reference to topic schemas which contain this constraint

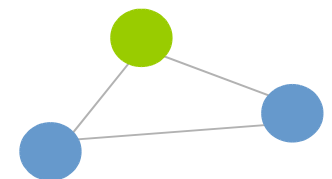
AssociationSchemaID\* # reference to association schemas which  
contain this constraint

TestMessage # string representing logical interpretation of constraint

ContextBinding\* # defines binding for variables  
from constraint type definition

Message # string

DiagnosticMessage\* # string



# NetworkedPlanet

**Networked Information – Networked Knowledge**

## **Summary:**

**TMCL is on its way. The key concepts are in place and we are into a refinement and resolution stage of the process.**

