Constraining Topic Maps

A TMCL declarative implementation

José Carlos Ramalho
Giovani Librelotto
Pedro Rangel Henriques

Computer Science and Technology Research Center
CCTC - Portugal

Agenda

- Questions, lots of...
- Metamorphosis overview (the umbrella)
- Talk about TM model
- XTche design implementation
- Examples and demos
- Conclusion

Motivation or ...!?

- Constrains... Why?
- Is a Schema a Constraint Language?
- Creating a new language ... Why?
  - We are always creating new DSL arguing that we will hide complexity
  - Aren´t we introducing complexity?
- If you found interesting answers to the questions above you can try answering the next one:
  - How should we implement this new language?

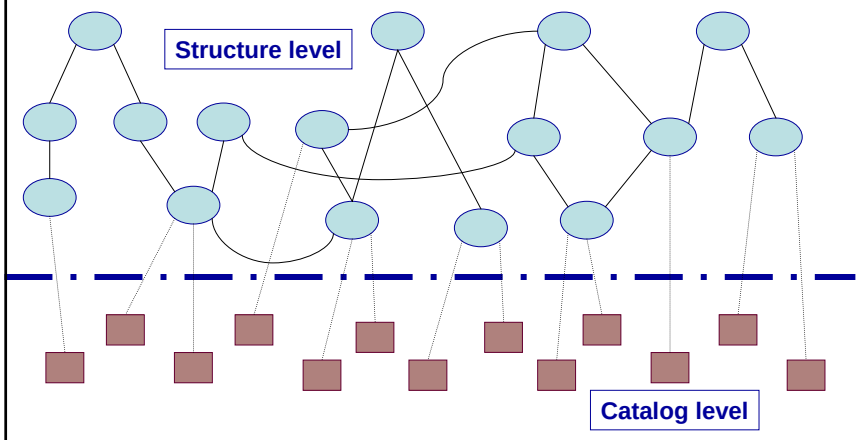
Questions for the moment...

- Why are Topic Maps so unsuccessful?
- Why an interesting model is not used more often?
- What are they used for?
- Is it the model? Is it hard to understand? Is it hard to process?
- Are Topic Maps still far from the user comunity? Is there a gap to be filled?

What are TM used for?

- Knowledge Management
- Semantic Web (foggy buzzword)
  - Semantic Network of Resources
  - Ontologies

What is an ontology? (IMHO)

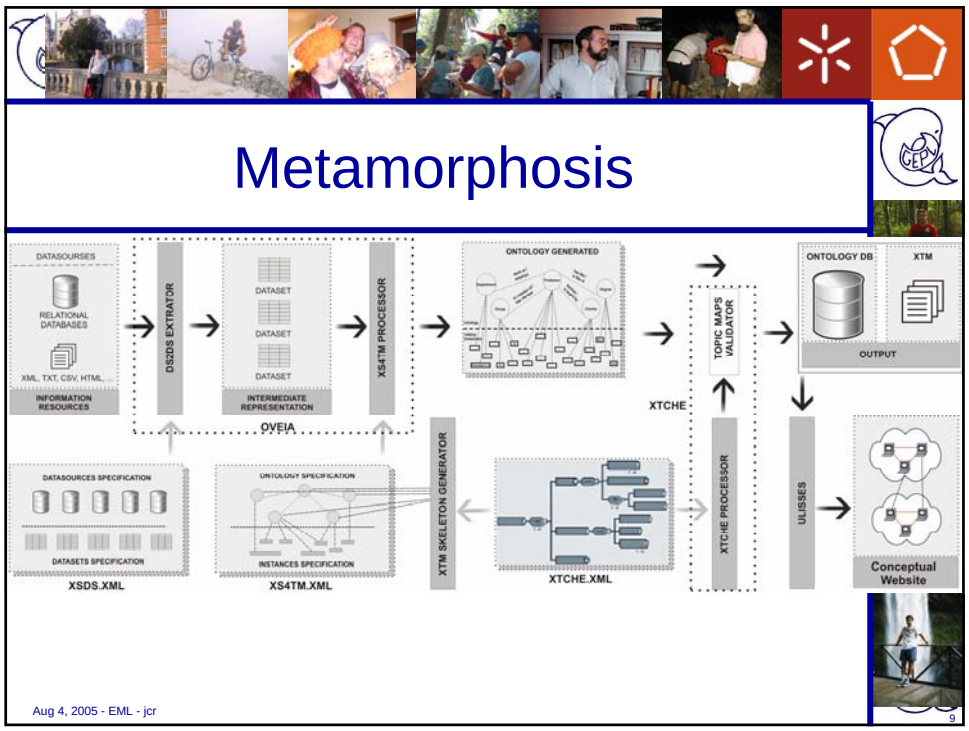


Getting started

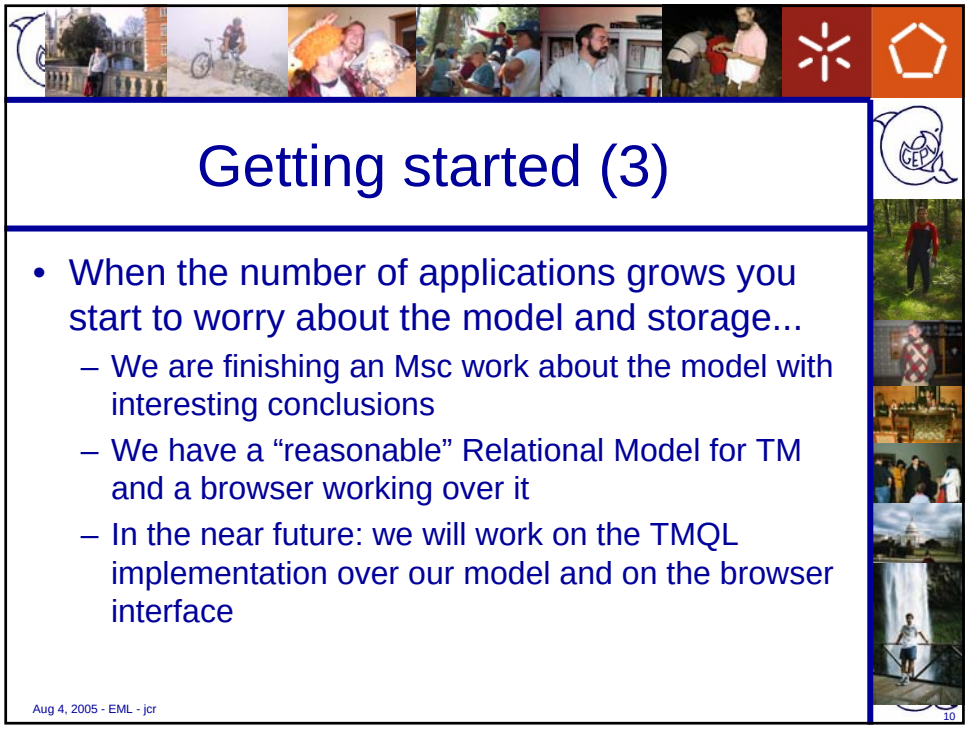
- Why Topic Maps?
  - These conferences ...
- Creating a TM is not so easy!
  - Development of some editing tools and methodologies
  - Development of some browsers

Getting started (2)

- When trying to apply the concepts to some real cases...
  - Creating a TM can become an hard task
  - Development of extraction and conversion tools that create TM automatically
  - An empty model is not useful -> harvesting



Metamorphosis



Getting started (3)

- When the number of applications grows you start to worry about the model and storage...
  - We are finishing an Msc work about the model with interesting conclusions
  - We have a “reasonable” Relational Model for TM and a browser working over it
  - In the near future: we will work on the TMQL implementation over our model and on the browser interface

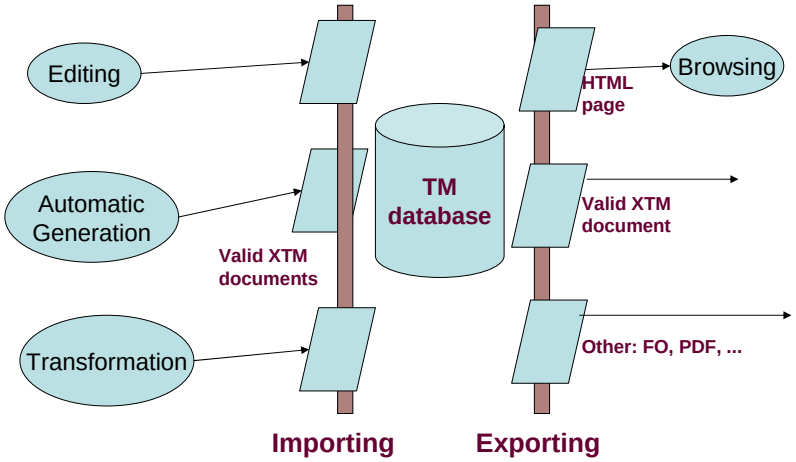
Returning to the beginning

- What is really a TM?
  - According to the XTM DTD
    - A list of ... Topics, associations, ocurrences
    - Everything else is constructed by reference
    - **It is easy to get lost in a TM!**
  - That DTD is completely abstract
    - There is room for “Specific Semantic Networks Schemas”

Can we throw away the TM?

- Once instanciated what is the use of it?
  - Add
  - Edit
  - Merge
- If yes, we could derive the specific models, we could implement a tool to do that automatically

TM lifecycle



Why TMCL?

- In some cases we felt the need to restrict the model
- It looked promising and challenging (we had an undergoing phd)...

Constraints?

- Enforcing a structure over a set of objects referencing eachother
- Generating reports about the concrete TM: creating a reporting service

XTM or ...?

- What should be the target of our work?
  - Some model implemented in some Database (TMDM based)?
    - Complex to manipulate: do any of these scale?
  - XTM files?
    - Do not scale
- For the moment: XTM files

About the model...

- Msc thesis: TM Repository Specification and Implementation
  - Starting goal: take XTM and try to simplify it; compare the result with TMDM; create a relational model; test it and reason about it.
- Eureka project: Information Knowledge Fusion
  - XTM, OWL, RDF, IKF-Imodel

The model: Starting point

topicMap	topic *	instanceOf *	(topicRef subjectIndicatorRef)		
		subjectIdentity ?	resourceRef ?		
			(topicRef subjectIndicatorRef) *		
		baseName *	scope ?	(topicRef subjectIndicatorRef resourceRef) +	
			baseNameString		
			variant *	parameters	(topicRef subjectIndicatorRef) +
				variantName ?	(resourceRef resourceData)
				variant * (recursive)	
		ocurrence *	instanceOf ?	(topicRef subjectIndicatorRef)	
			scope ?	(topicRef subjectIndicatorRef resourceRef) +	
			(resourceRef resourceData)		
	association *	instanceOf ?	(topicRef subjectIndicatorRef)		
		scope ?	(topicRef subjectIndicatorRef resourceRef) +		
		member +	roleSpec ?	(topicRef subjectIndicatorRef)	
			(topicRef subjectIndicatorRef resourceRef) *		
	mergeMap *	(topicRef subjectIndicatorRef resourceRef) *			

R - Relationship: R - Mandatory Single, R+ - Mandatory Repeatable, R? - Optional Single, R* - Optional Repeatable

The model: looking at the leaves

(topicRef | subjectIndicatorRef | resourceRef)
(topicRef | subjectIndicatorRef)
(resourceRef | resourceData)
resourceRef
baseNameString

subjectIdentity = (resourceRef ?, (topicRef subjectIndicatorRef) *)

subjectIdentity ?	resourceRef ?
	(topicRef subjectIndicatorRef) *

step 1

subjectIdentity = ((ic type value) ? | (ic type value) *)

subjectIdentity ?	(ic type value) ?
	(ic type value) *

step 2

subjectIdentity = (ic type value) *

subjectIdentity ?	(ic type value) *

step 3

subjectIdentity = subjectIdentityTV *
subjectIdentityTV = (ic type value)

subjectIdentity ?	subjectIdentityTV *	(ic type value)

The model: normalized

E: topicMap	E: topic *	E: instanceOf *	A: (id type value)			
		E: subjectIdentity ?	A: (id type value) ?			
			E: subjectIdentityTV *			A: (id type value)
		E: baseName *	E: scope ?	E: scopeTV +		A: (id type value)
			A: (id type value)			
			E: variant *	E: parameters	E: parametersTV +	A: (id type value)
				E: variantName ?		A: (id type value)
				E: variant * (recursive)		
		E: ocurrence *	E: instanceOf ?			A: (id type value)
			E: scope ?	E: scopeTV +		A: (id type value)
			A: (id type value)			
	E: association *	E: instanceOf ?	A: (id type value)			
		E: scope ?	E: scopeTV +			A: (id type value)
		E: member +	E: roleSpec ?			A: (id type value)
			E: memberTV *			A: (id type value)
	E: mergeMap *	E: mergeMapTV *	A: (id type value)			

E - Entity, A - Attribute, R - Relationship
R - Mandatory Single, R+ - Mandatory Repeatable, R? - Optional Single, R* - Optional Repeatable

TMDiscovery

- Demo: TMDisovery

Aug 4, 2005 - EML - jcr

21

TMCL: Implementation wishes

- Do it simple
- Do it with existing technology
- Do it user friendly
  - If possible do not force the user to learn a new language

Schema constraints:

Topic of type T must have a specified number of explicit names/occurrences/subject-indicators (cardinality);
Topic of type T must have as name/occurrence/subject-indicators a value matching a particular pattern;
Topic of type T must (not) have a name/occurrence with scope S;
Topic of type T must have a name/occurrence, that is instance of topic type T, in scope S;
Topic of type T must (not) have an occurrence that is of type O;
Topic T can (only/not) be used as an association role topic in association with association type A;
Topic of type T can (only/not) be used as an association player topic in association with association type A;
A list of topics are instances of topic type T;
Association with association type A must be in scope S;
Association with association type A has (only/at least/not) roles R1 and R2;
Association of type A must have (only/at least) two participating topics where one is of type T1 and the other is of type T2;
Association of type A must (not) have the role R being played by a topic of type T;
Association of type A has role R played by exactly two topics of type T (cardinality);
Association of type A has role R1 played by topic of type T1 and role R2 played by topic of type T1 or T2;
Association of type A must have dependencies between player types;
Occurrence of type O can (only/not) be a characteristic of topics of type T;
Occurrence of type O can (only/not) be used within scope S;
Occurrence of type O must have locators that match a URI pattern P;

Contextual constraints:

Topic T can (only/not) be used for typing other topics;
Topic T can (only/not) be used for typing subject indicator;
Topic T can (only/not) be used for typing basenames;
Topic T can (only/not) be used for typing occurrences;
Topic T can (only/not) be used for typing associations;
Topic T can (only/not) be used as an association scope;
Topic T can (only/not) be used as an association role topic;
Topic of type T can (only/not) be used for scoping occurrences;
Topic of type T can (only/not) be used for scoping base names;
Topic of type T can (only/not) be used for scoping associations;
Topic of type T can (only/not) be used as an association player topic;

Aug 4, 2005 - EML - jcr

22

TMCL: Requirements from ISO JTC1 SC34

Schema constraints:

- •Topic of type T must have a specified number of explicit names/occurrences/subject-indi
- •Topic of type T must have as name/occurrence/subject-indicators a value matching a pat
- •Topic of type T must (not) have a name/occurrence with scope S;
- •Topic of type T must have a name/occurrence, that is instance of topic type T in scope S

...

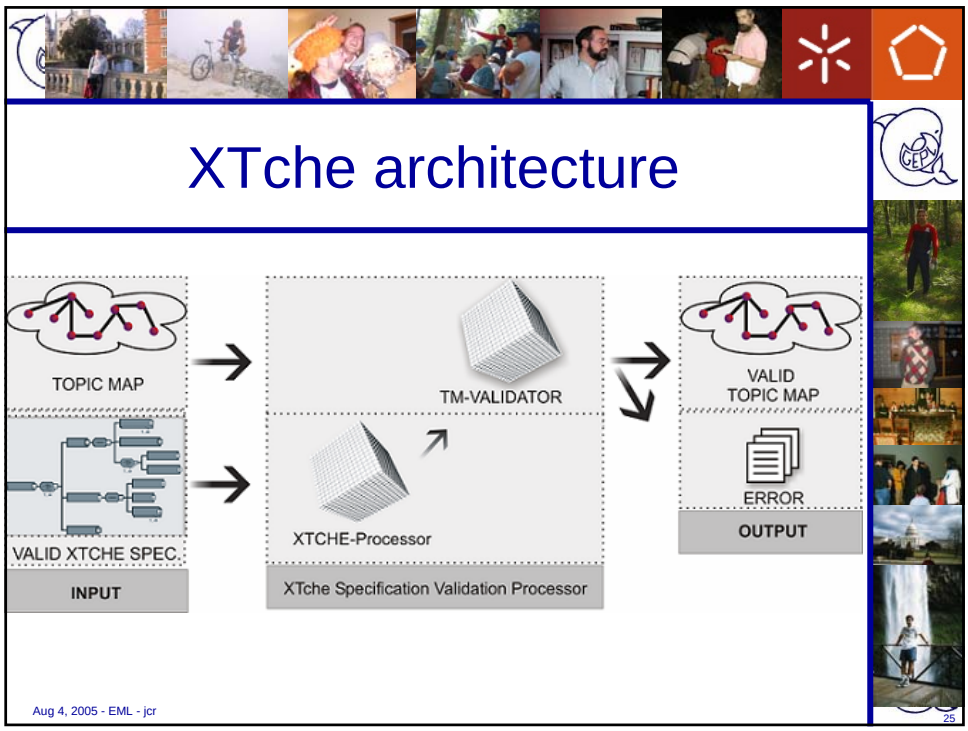
Contextual constraints:

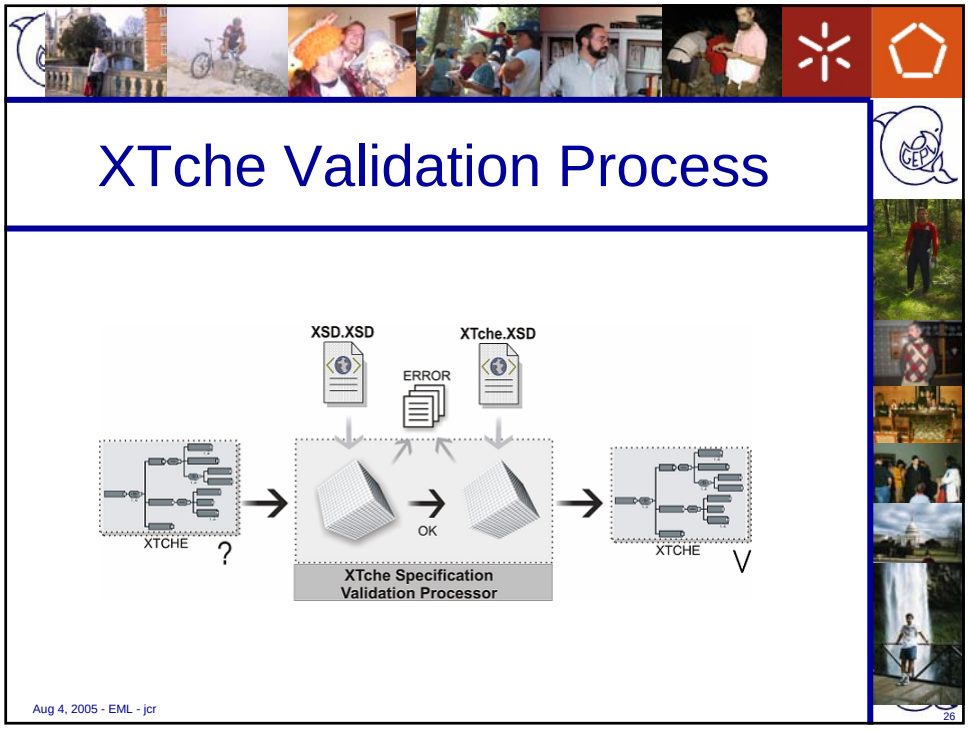
- •Topic T can (only/not) be used for typing other topics;
- •Topic T can (only/not) be used for typing subject indicator;
- •Topic T can (only/not) be used for typing basenames;
- •Topic T can (only/not) be used for typing occurrences;

...

How to ...?

- Schematron/XCSL processing model has been used in similar contexts
- For prototyping this processing model we would only need open “standards”:
  - XML
  - XSLT
  - XML Schemas

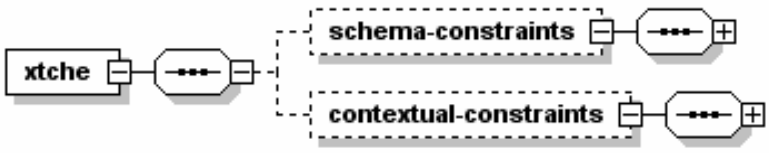

XTche architecture
TOPIC MAP
VALID XTCHE SPEC.
INPUT
TM-VALIDATOR
XTCHE-Processor
XTche Specification Validation Processor
VALID
TOPIC MAP
ERROR
OUTPUT
Aug 4, 2005 - EML - jcr
25


XTche Validation Process
XSD.XSD
XTche.XSD
ERROR
XTCHE
?
OK
XTche Specification
Validation Processor
XTCHE
V
Aug 4, 2005 - EML - jcr
26

XTche Specification Language

- XML Schema based
  - Constraint lang. = Schema lang.
  - We have some experience with XML Schemas
  - The tools we use have an interesting graphical support that we wanted to use
  - If we choose an existing language to host ours part of the work is already done
- **XTche Spec. = XML Schema + semantic stamps**

XTche skeleton



- Namespace: **xmlns:xtche=http://www.di.uminho.pt/~gepl/xtche**
- Imports: http://www.di.uminho.pt/~gepl/xtche/xtche-schema.xd

xtche-schema.xsd

- A set of flags / semantic stamps
- Each flag represents a semantic validation
- The user will associate flags to concrete topics and these will map to semantic constraints that will be verified
- Remaining: **Did we cover the wole TMCL?**

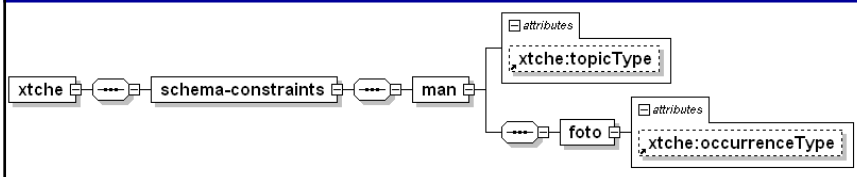
Flags / Semantic Stamps

- Schema constraints
  - <xs:attribute name="topicType"/>
  - "this element represents a Topic type"
- Contextual constraints
  - <xs:attribute name="associationTypeExclusive"/>
  - "this topic can only be used to qualify associations"

A small case study

- JCR's genealogical tree
- Using TMDiscovery (a TM browser soon freely available)

Example: schema constraint



- **Each topic of type "man" must have an occurrence of type "foto"**

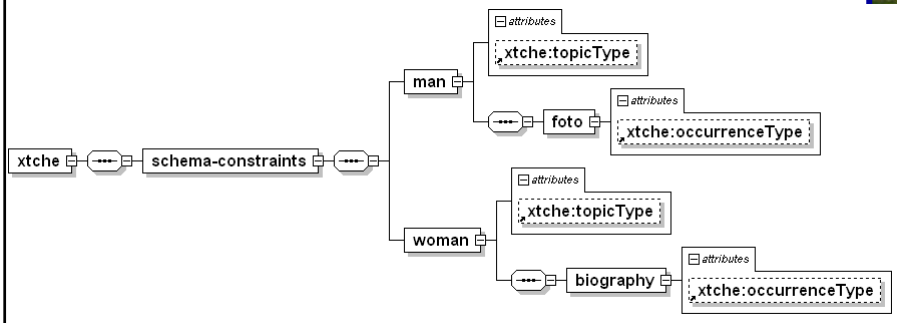
Generated processor

```
<xtche:template match="*" mode="T-occurrences-cardinal">
    <xtche:for-each
select="xtm:topic[xtm:instanceOf/xtm:topicRef/@xlink:href='#man']">
       <xtche:if
test="count(xtm:occurrence[xtm:instanceOf/xtm:topicRef/@xlink:href =
'#foto']) = 0">
         <err-message>
           : 4.1.2 - U2 - (c1) : Topic #<xtche:value-of select="@id"/> must
have an occurrence that needs to be an instance of #foto.
         </err-message>
       </xtche:if>
    </xtche:for-each>
  </xtche:template>
```


Results: foto

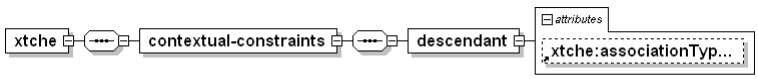
```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<doc-status
xmlns="http://www.di.uminho.pt/~gepl/xtche/namespace">
  <err-message number="1">Constraint: 4.1.2 - U2 - (c1)
: Topic #leo must have an occurrence that needs to be an
instance of #foto.</err-message>
  <err-message number="1">Constraint: 4.1.2 - U2 - (c1)
: Topic #zeca must have an occurrence that needs to be
an instance of #foto.</err-message>
</doc-status>
```

Example: schema constraint



- Each topic of type “man” must have an occurrence of type “foto”
- Each topic of type “woman” must have an occurrence of type “biography”

Example: contextual constraint



- Semantic Stamp = xtche:associationType-Exclusive
- topic “descendant” can only be used to type associations
- demo: TMDiscovery

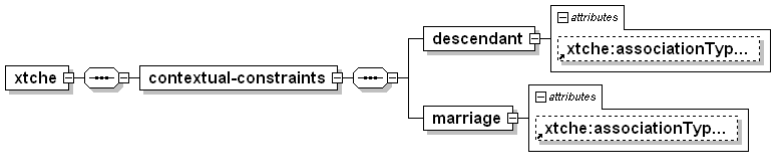
Results: descendant

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<doc-status xmlns="http://www.di.uminho.pt/~gepl/xtche/namespace">
  <err-message number="1">Constraint: 4.1.1 - U1 - (a2) : Topic
#descendant can be used for typing associations and nothing else.
Error in Topic #zp.</err-message>

  <err-message number="3">Constraint: 4.1.1 - U1 - (a1) : Topic
#marriage must be declared for typing associations.</err-message>
</doc-status>
```

There are bugs somewhere in the middle of 40 XSLT stylesheets!

Example: reporting services



- **Semantic Stamp = xtche:associationType-Forbidden**
- **topic "descendant" cannot be used to type associations**
- **topic "marriage" cannot be used to type associations**

- **Expected result: number of descendant and marriage associations**

Results from reporting

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<doc-status xmlns="http://www.di.uminho.pt/~gepl/xtche/namespace">

  <err-message number="6">Constraint: 4.1.1 - U1 - (a3) : Topic
#descendant can not be used for typing associations.</err-message>

  <err-message number="3">Constraint: 4.1.1 - U1 - (a3) : Topic
#marriage can not be used for typing associations.</err-message>

</doc-status>
```

Related Work

- AsTMa! – Robert Barta
- Eric Freese's proposal: DAML+OIL
- ...

Future Work

- Test, test, test ...
- Correct, correct, correct ...
- Optimize
- To implement this in the model behind TMDiscovery