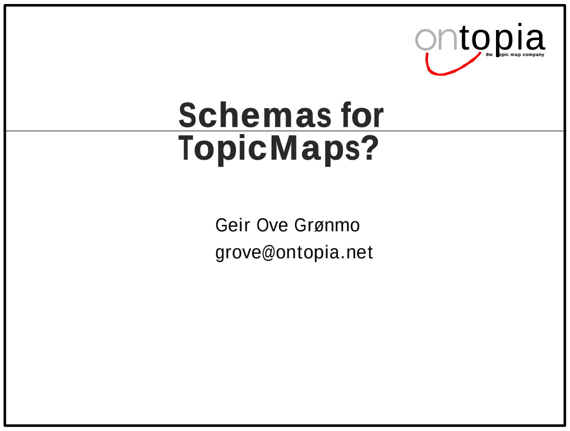
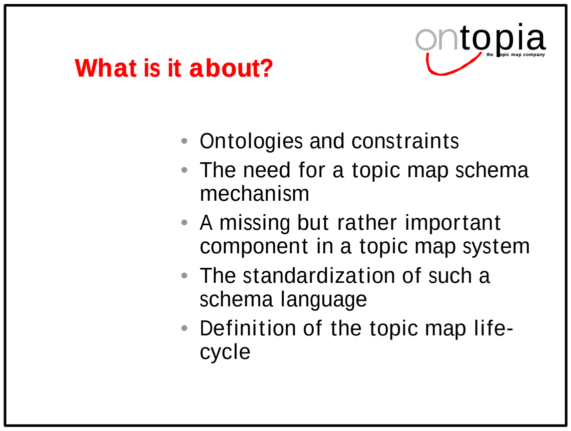


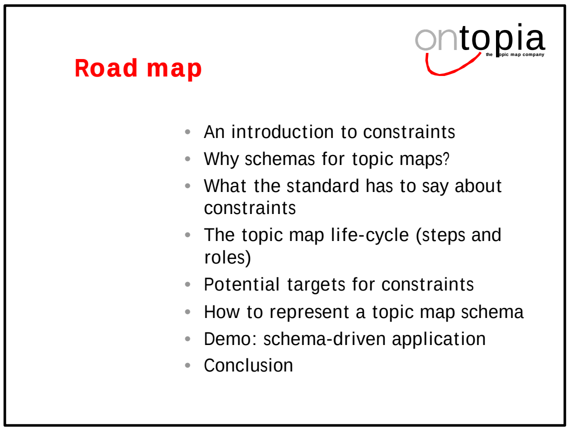
Schemas for TopicMaps?

Geir Ove Grønmo
grove@ontopia.net



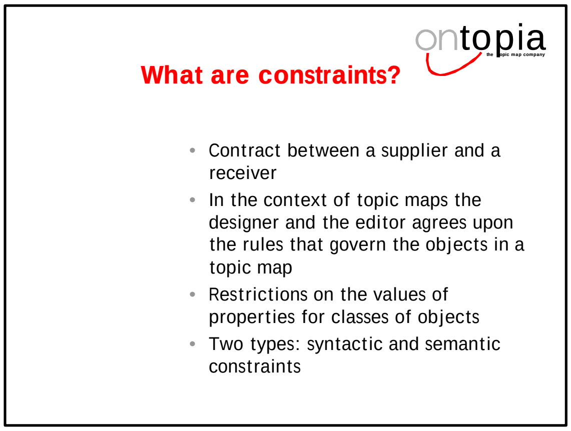
What is it about?

- Ontologies and constraints
- The need for a topic map schema mechanism
- A missing but rather important component in a topic map system
- The standardization of such a schema language
- Definition of the topic map life-cycle



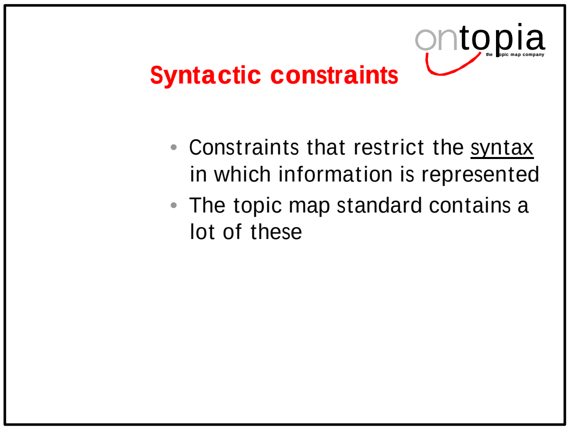
Road map

- An introduction to constraints
- Why schemas for topic maps?
- What the standard has to say about constraints
- The topic map life-cycle (steps and roles)
- Potential targets for constraints
- How to represent a topic map schema
- Demo: schema-driven application
- Conclusion



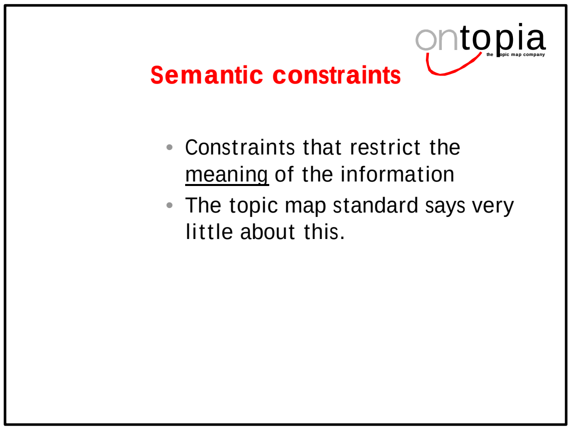
What are constraints?

- Contract between a supplier and a receiver
- In the context of topic maps the designer and the editor agrees upon the rules that govern the objects in a topic map
- Restrictions on the values of properties for classes of objects
- Two types: syntactic and semantic constraints



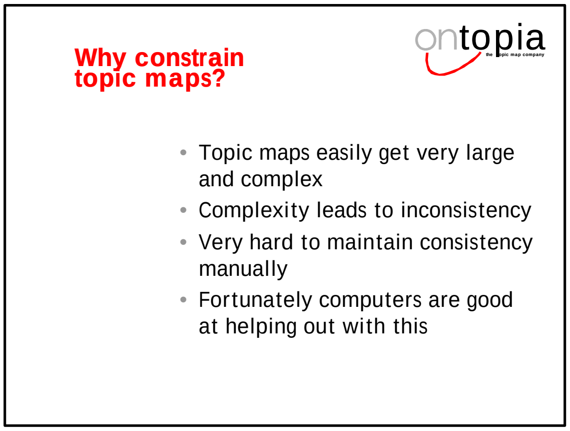
Syntactic constraints

- Constraints that restrict the syntax in which information is represented
- The topic map standard contains a lot of these



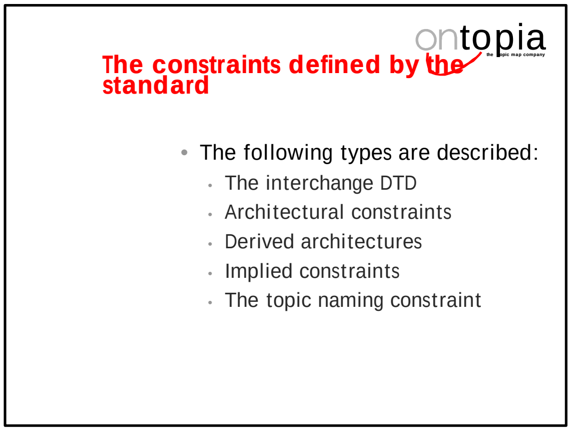
Semantic constraints

- Constraints that restrict the meaning of the information
- The topic map standard says very little about this.



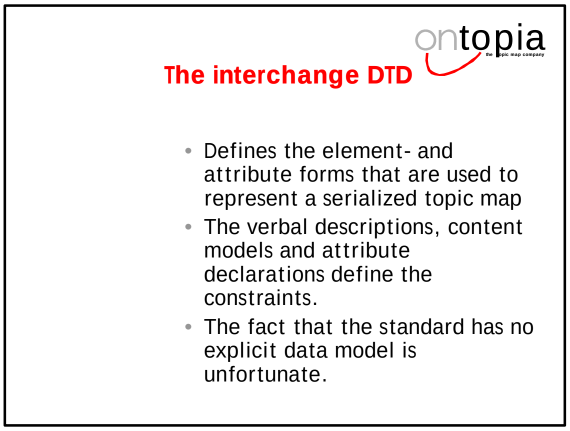
Why constrain topic maps?

- Topic maps easily get very large and complex
- Complexity leads to inconsistency
- Very hard to maintain consistency manually
- Fortunately computers are good at helping out with this



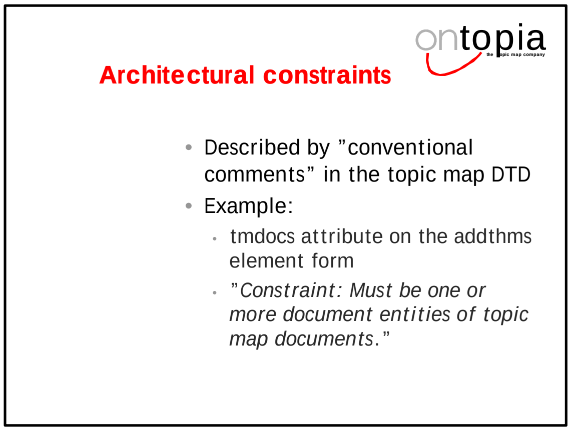
The constraints defined by the standard

- The following types are described:
  - The interchange DTD
  - Architectural constraints
  - Derived architectures
  - Implied constraints
  - The topic naming constraint



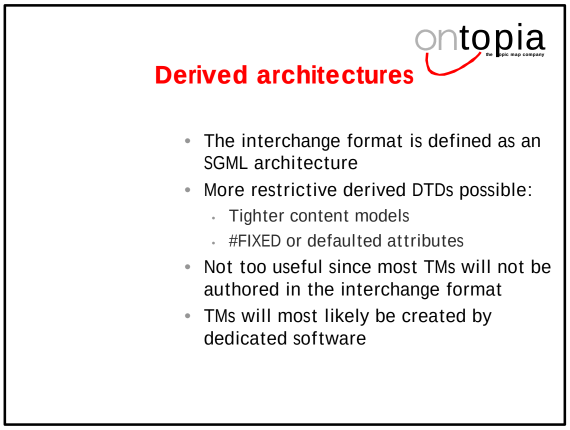
The interchange DTD

- Defines the element- and attribute forms that are used to represent a serialized topic map
- The verbal descriptions, content models and attribute declarations define the constraints.
- The fact that the standard has no explicit data model is unfortunate.



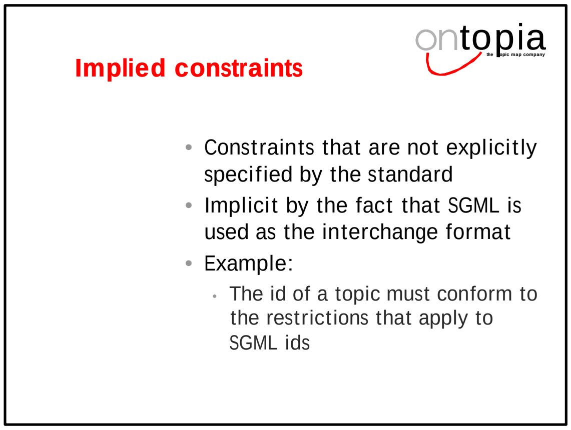
Architectural constraints

- Described by ”conventional comments” in the topic map DTD
- Example:
  - tmdocs attribute on the addthms element form
  - ”*Constraint: Must be one or more document entities of topic map documents.*”



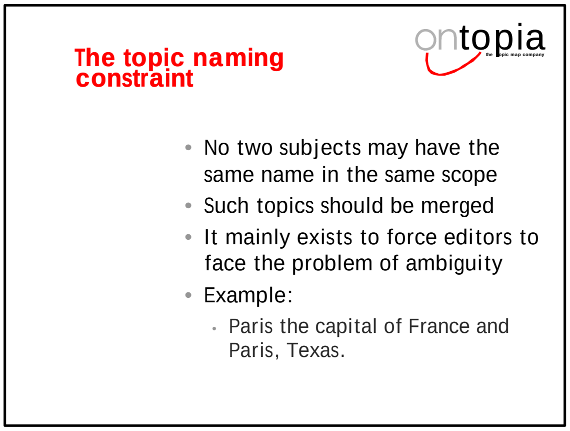
Derived architectures

- The interchange format is defined as an SGML architecture
- More restrictive derived DTDs possible:
  - Tighter content models
  - #FIXED or defaulted attributes
- Not too useful since most TMs will not be authored in the interchange format
- TMs will most likely be created by dedicated software



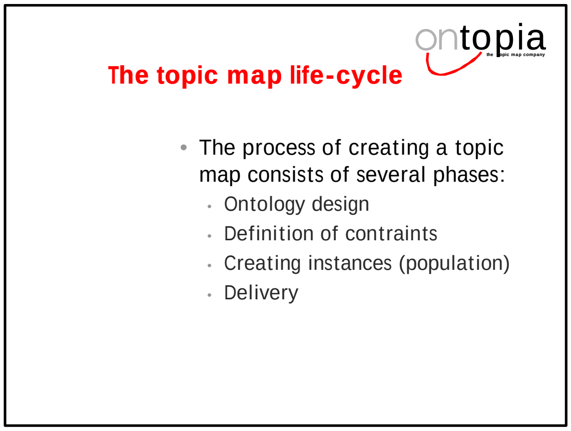
Implied constraints

- Constraints that are not explicitly specified by the standard
- Implicit by the fact that SGML is used as the interchange format
- Example:
  - The id of a topic must conform to the restrictions that apply to SGML ids



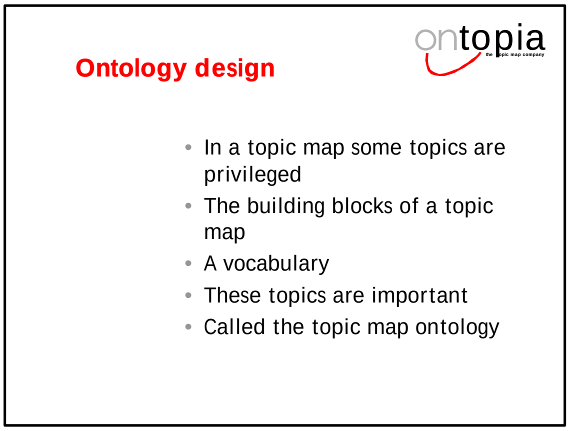
The topic naming constraint

- No two subjects may have the same name in the same scope
- Such topics should be merged
- It mainly exists to force editors to face the problem of ambiguity
- Example:
  - Paris the capital of France and Paris, Texas.



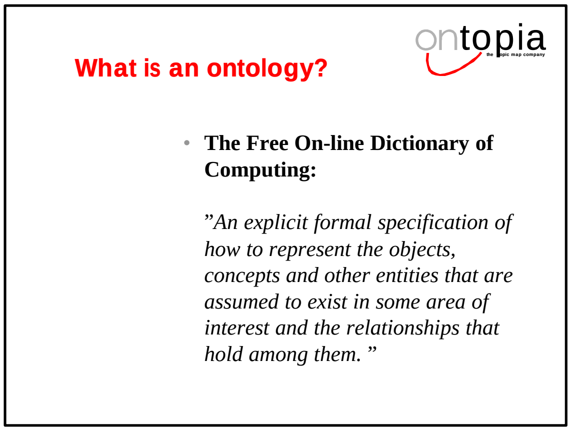
The topic map life-cycle

- The process of creating a topic map consists of several phases:
  - Ontology design
  - Definition of contraints
  - Creating instances (population)
  - Delivery



Ontology design

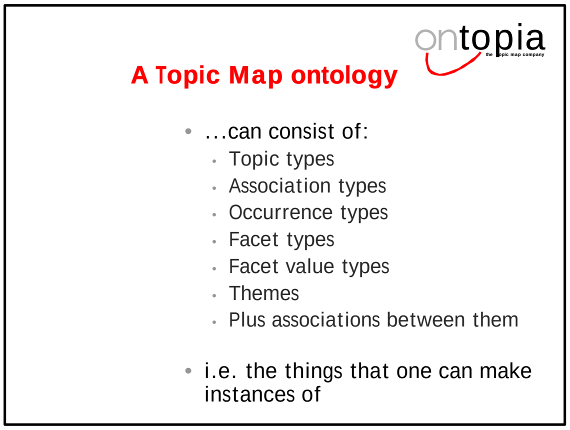
- In a topic map some topics are privileged
- The building blocks of a topic map
- A vocabulary
- These topics are important
- Called the topic map ontology



What is an ontology?

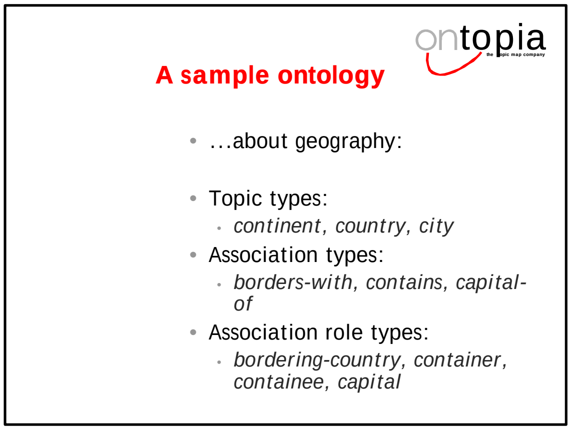
- **The Free On-line Dictionary of Computing:**

  ”An explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them. ”



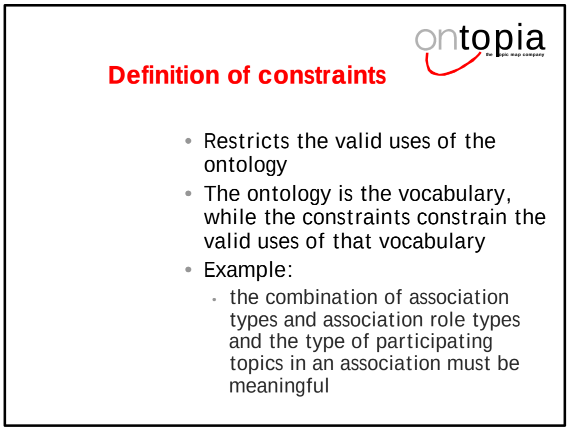
A Topic Map ontology

- ...can consist of:
  - Topic types
  - Association types
  - Occurrence types
  - Facet types
  - Facet value types
  - Themes
  - Plus associations between them
- i.e. the things that one can make instances of



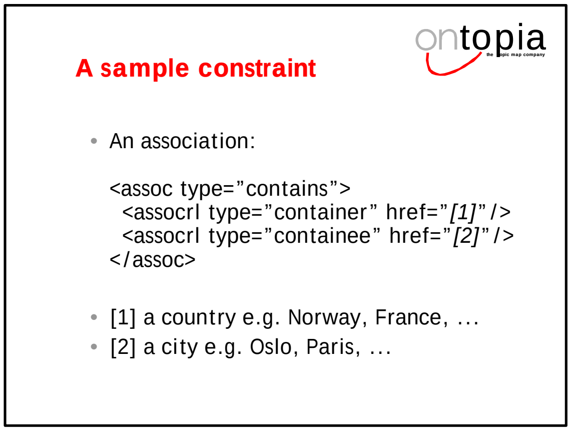
A sample ontology

- ...about geography:
- Topic types:
  - *continent, country, city*
- Association types:
  - *borders-with, contains, capital-of*
- Association role types:
  - *bordering-country, container, containee, capital*



Definition of constraints

- Restricts the valid uses of the ontology
- The ontology is the vocabulary, while the constraints constrain the valid uses of that vocabulary
- Example:
  - the combination of association types and association role types and the type of participating topics in an association must be meaningful

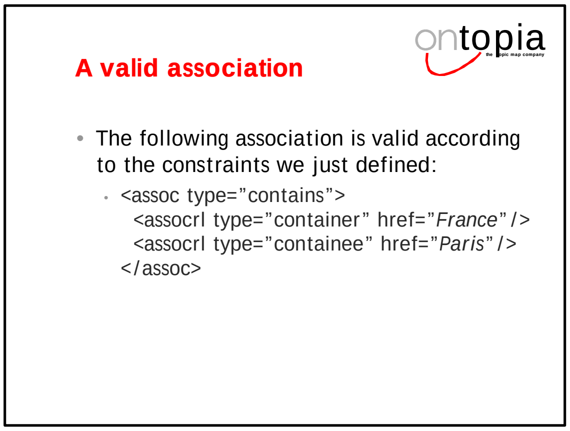


A sample constraint

- An association:

```
<assoc type="contains">
  <assocrl type="container" href="[1]"/>
  <assocrl type="containee" href="[2]"/>
</assoc>
```

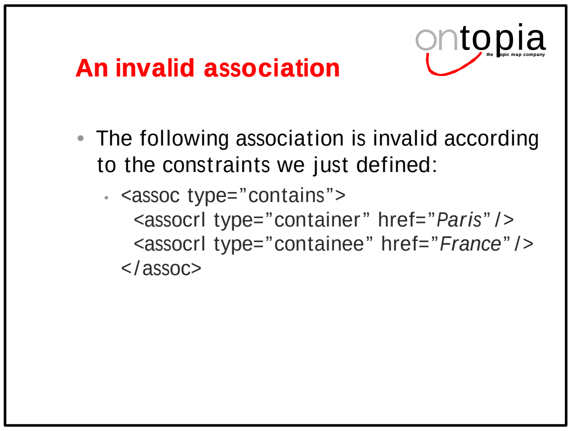
- [1] a country e.g. Norway, France, ...
- [2] a city e.g. Oslo, Paris, ...



A valid association

- The following association is valid according to the constraints we just defined:

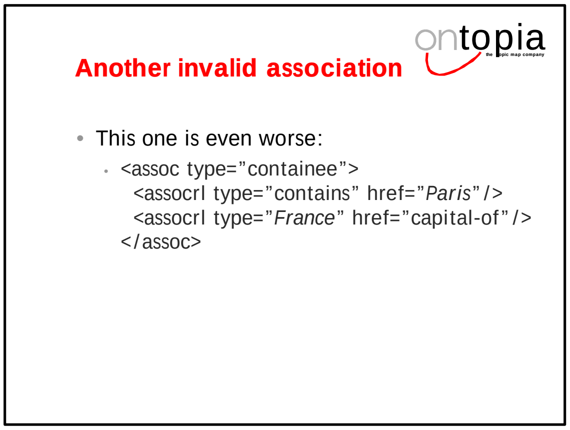
```
<assoc type="contains">
  <assocrl type="container" href="France"/>
  <assocrl type="containee" href="Paris"/>
</assoc>
```



An invalid association

- The following association is invalid according to the constraints we just defined:

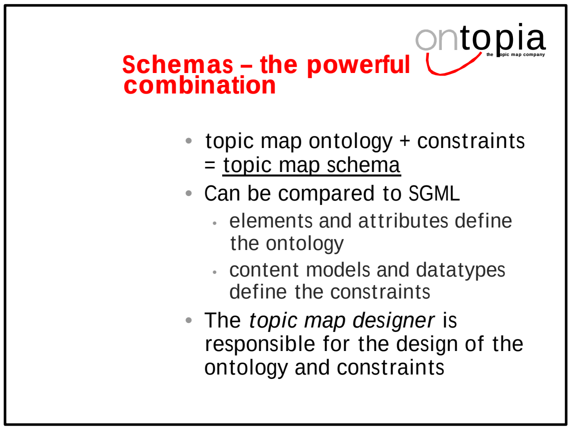
```
<assoc type="contains">
  <assocrl type="container" href="Paris"/>
  <assocrl type="containee" href="France"/>
</assoc>
```



Another invalid association

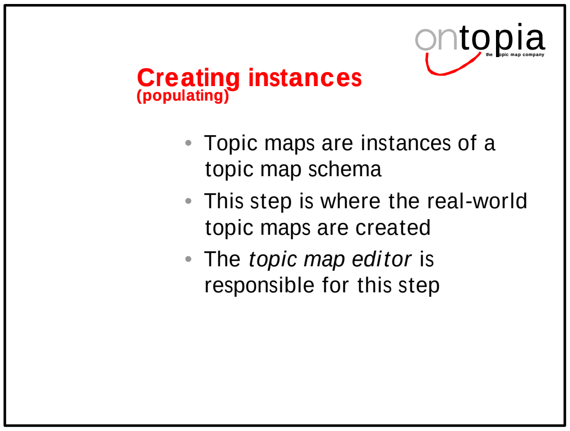
- This one is even worse:

```
<assoc type="containee">
  <assocrl type="contains" href="Paris"/>
  <assocrl type="France" href="capital-of"/>
</assoc>
```



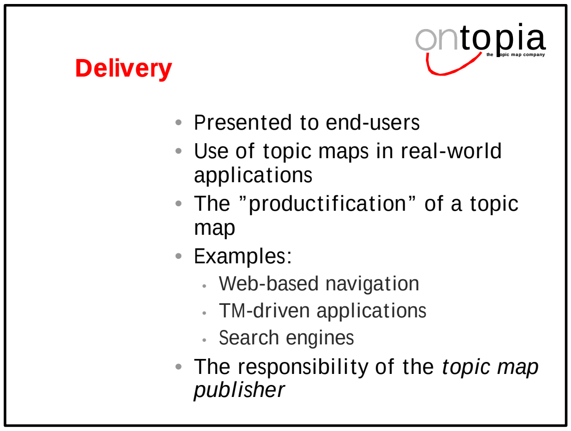
Schemas – the powerful combination

- topic map ontology + constraints = topic map schema
- Can be compared to SGML
  - elements and attributes define the ontology
  - content models and datatypes define the constraints
- The *topic map designer* is responsible for the design of the ontology and constraints



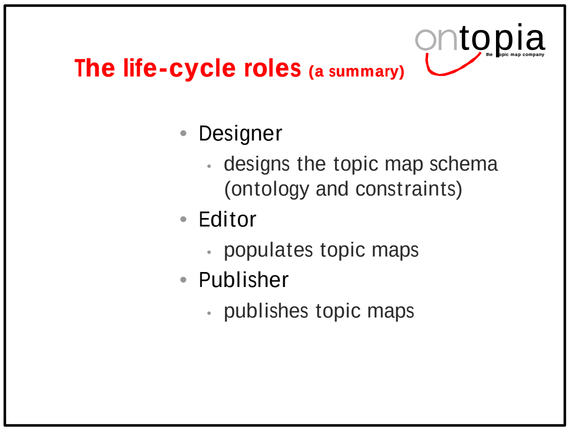
Creating instances
(populating)

- Topic maps are instances of a topic map schema
- This step is where the real-world topic maps are created
- The *topic map editor* is responsible for this step



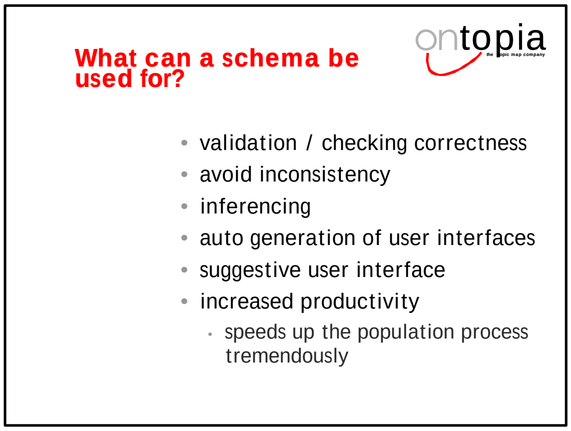
Delivery

- Presented to end-users
- Use of topic maps in real-world applications
- The ”productification” of a topic map
- Examples:
  - Web-based navigation
  - TM-driven applications
  - Search engines
- The responsibility of the *topic map publisher*



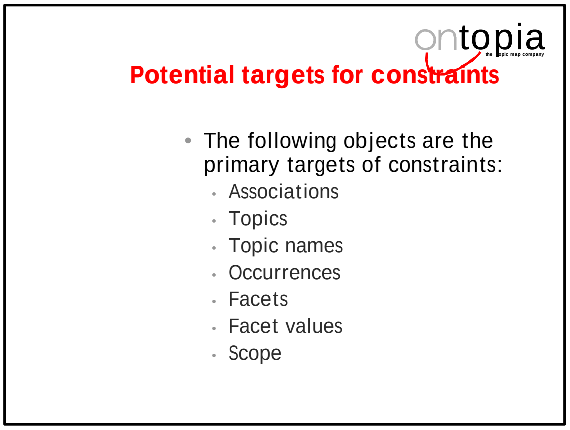
The life-cycle roles (a summary)

- Designer
  - designs the topic map schema (ontology and constraints)
- Editor
  - populates topic maps
- Publisher
  - publishes topic maps



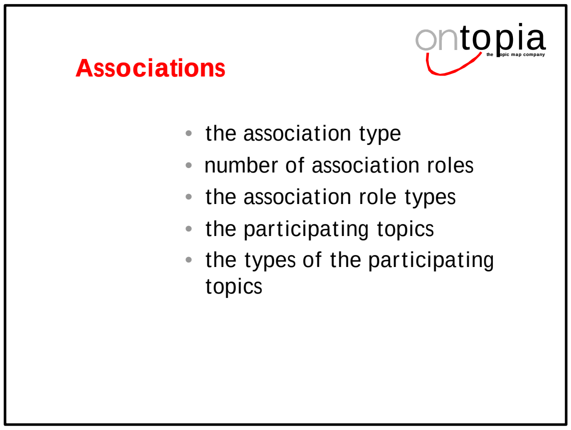
What can a schema be used for?

- validation / checking correctness
- avoid inconsistency
- inferencing
- auto generation of user interfaces
- suggestive user interface
- increased productivity
  - speeds up the population process tremendously



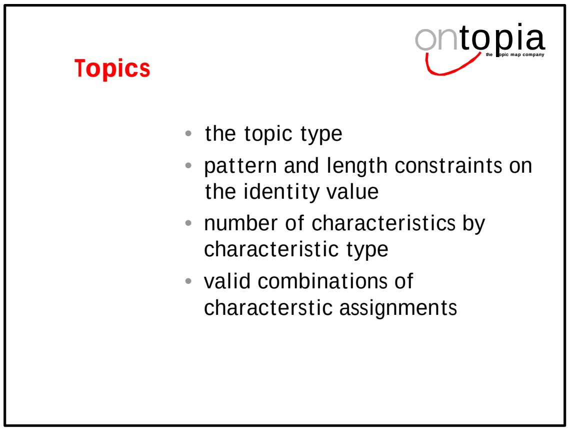
Potential targets for constraints

- The following objects are the primary targets of constraints:
  - Associations
  - Topics
  - Topic names
  - Occurrences
  - Facets
  - Facet values
  - Scope



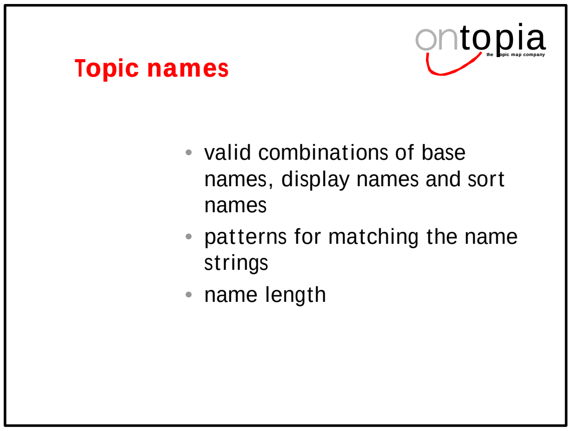
Associations

- the association type
- number of association roles
- the association role types
- the participating topics
- the types of the participating topics



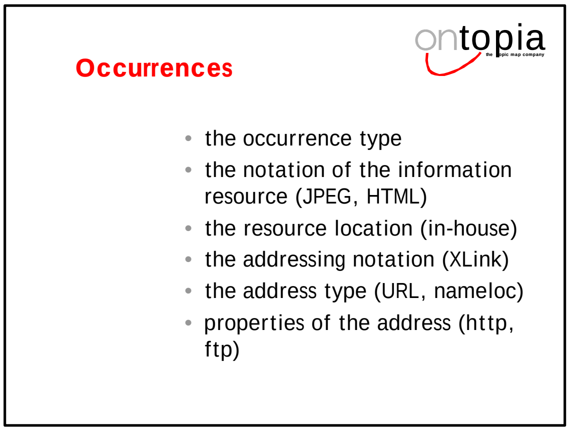
Topics

- the topic type
- pattern and length constraints on the identity value
- number of characteristics by characteristic type
- valid combinations of characterstic assignments



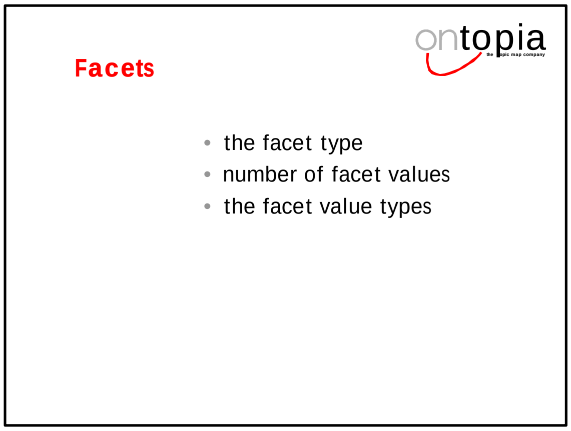
Topic names

- valid combinations of base names, display names and sort names
- patterns for matching the name strings
- name length



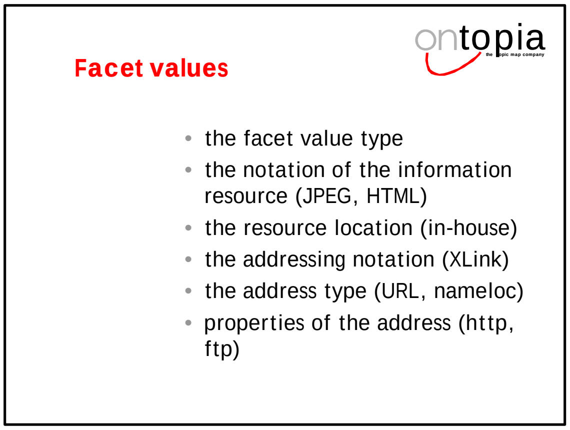
Occurrences

- the occurrence type
- the notation of the information resource (JPEG, HTML)
- the resource location (in-house)
- the addressing notation (XLink)
- the address type (URL, nameloc)
- properties of the address (http, ftp)



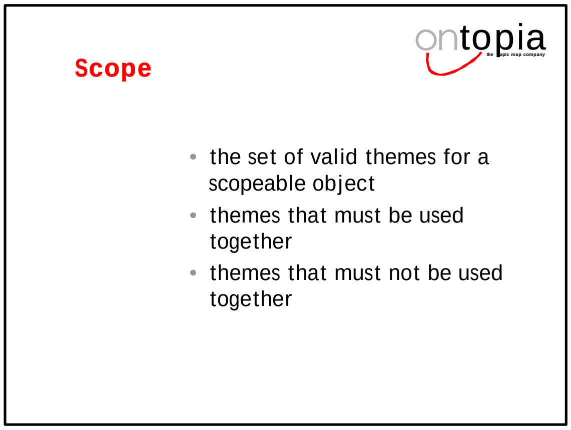
Facets

- the facet type
- number of facet values
- the facet value types



Facet values

- the facet value type
- the notation of the information resource (JPEG, HTML)
- the resource location (in-house)
- the addressing notation (XLink)
- the address type (URL, nameloc)
- properties of the address (http, ftp)



Scope

- the set of valid themes for a scopeable object
- themes that must be used together
- themes that must not be used together

How to describe constraints

- two basic requirements for the use of constraints:
  - predicates – tells whether constraints are satisfied or not
  - suggestive – gives a list of valid values for a property based on information from the constraints
- combination of declarative language and programming language most powerful
- hard / impossible to create a complete constraint language

A standardized constraint language

- Are sorely needed for standardization
- This is the most important missing part as the standard stands today
- Interchangeable between topic map systems
- Self documentation

Contestants?

- Topic Map Object Model API
- Generic grove / property set constraint language
- Schemas defined using topic maps
- Conceptual graphs
- EXPRESS
- The Object Constraint Language (OCL)

Sample syntax

- Association constraint described using topic map constructs:
  - <assoc type=”contains” scope=”schema”>
    <assocrl type=”container” href=”country”/>
    <assocrl type=”containee” href=”*city*”/>
    </assoc>
- a.k.a topic map template
- 80/20 solution

Demo-time

- A simple schema-driven application for creating associations
- Both able to make inferences (autocompletion) and be suggestive.
- Written on top of the Atlas topic map engine from Ontopia

Conclusion

- Important part of a topic map system
  - increase productivity
  - avoid inconsistency
- A standardized schema language would be preferable
  - would make schemas interchangeable and self-documenting
- An (explicit) abstract data model would probably have to be defined first
  - preferably a property set