Graphical Notation for Topic Maps (GTM)

2005.11.12

Jaeho Lee
University of Seoul
jaeho@uos.ac.kr

Outline

- Motivation
- Requirements for GTM
    - Goals, Scope, Constraints, and Issues
- Survey on existing approaches
    - VisualScript for Topic Maps
        - Steve Pepper and Graham Moore
    - Conceptual Modeling of Topic Maps with ORM versus UML
        - Are D. Gulbrandsen
    - A simple UML-based notation for Topic Maps
        - Kal Ahmed, Techquila, “Beyond PSIs: Topic Map design patterns”
- Discussion
    - Evaluation Criteria
    - Work Items

Introduction

- GTM is for human
  - Communication between humans
  - Modeling by human
  - Presentation for human
  - Creation by human
    - Automatic transformation from CTM or XTM
- XTM remains as the basis for interchange syntax.
  - GTM is yet another representation, especially for human.

Uses of GTM

- Easy-to-read examples
  - For papers and presentations
  - Educational purpose
- Graphical Modeling Language
  - Conceptualization
  - Communication
  - Analysis
  - Verification
- Possible use case for TM Inference Language
  - Graph-based Navigation
  - Graph-based Query Language (GTMQL?)
  - Application of graph theory

Why GTM?

- Visual representation of TMDM
  - Providing Communication medium for human
  - Providing modeling tools for human
  - Providing visual presentation and illustration for human
  - Providing a foundation for analysis and verification
- A homogenous and coherent representation
  - CTM-equivalent GTM
  - TMCL and TMQL (and also GTMQL)

Requirements (1/3)

- GTM should be simple and intuitive
  - to read and write
  - to understand
  - to learn and teach
  - to implement
- GTM should be compatible with CTM, syntactically and semantically
- GTM specification must define deserialization of GTM to CTM, and to TMDM
- CTM must fully represent TMDM.
  - CTM must support embedded markup is implied by this statement.

Requirements (2/3)

- GTM should have a way to abstract up to see the overall picture and abstract down to see the details.

- GTM should be modular
  - to compose big top maps from pieces
  - to drop out parts of the topic maps
  - to replace topics and merge topics
  - to be able to reify a part of top maps

Requirements (3/3)

- Transformation between GTM and CTM (or XTM) should be easy and unambiguous to implement.
- GTM has to support all character encodings.
- GTM needs to have an escape syntax for Unicode characters.
- GTM must be syntactically and semantically aligned with TMCL and TMQL.

GTM Features

- Minimal Specification for user flexibility
  - Example
    - Topic: Polygon
      - Different line colors, fill colors and # of sides to represent different kind of topics
    - Association: Line
      - Dotted line, colored line, thickness, arrow
    - Occurrence: Round Shapes
  - Users can specify legends

- Compositional Shape
  - Topic and topic type
  - Topic and occurrence
  - Reification

Issues

- Philosophy
- Purpose
  - Modeling (top-down) / Domain mapping (bottom-up)
  - Presentation
  - Analysis and Reasoning using well known graph theory
- Scope
  - Pure modeling
  - CTM Equivalent
  - TMCL, TMQL
- Compatibility
  - Build up on existing modeling methodologies such as UML and RDF

Survey on Existing Graphical Notations

- VisualScript for Topic Maps
  - Steve Pepper and Graham Moore
- Conceptual Modeling of Topic Maps with ORM versus UML
  - Are D. Gulbrandsen
- A simple UML-based notation for Topic Maps
  - Beyond PSIs: Topic Map design patterns
  - Kal Ahmed, Techquila

VisualScript for Topic Maps

- Basis of a graphical tool for defining topic map ontologies rather than topic map instances
- Brief Description of Topic Map Model
  - Basic Components
    - Topics
      - subjects of discourse
    - Associations
      - relationships between topics
      - inherently multidirectional
    - Occurrences
      - relationships between topics and information resources
    - (Roles)
      - The role played by a topic in an association
  - Basic Component Types ("typing topics")
    - Topic types (TT)
    - Association types (AT)
    - Occurrence types (OT)
    - Role types (RT)

A simple topic map

```
<?xml version="1.0"?>
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/"
          xmlns="http://www.w3.org/1999/xlink">
<topic id="xzyyz">
  <baseName>
    <baseNameString>Redmond Computers Inc.</baseNameString>
  </baseName>
  <occurrence>
    <resourceRef xlink:href="http://www.redmondcomputers.com/"/> </occurrence>
  <occurrence>
    <resourceData>1977</resourceData>
  </occurrence>
</topic>
</topicMap>
```

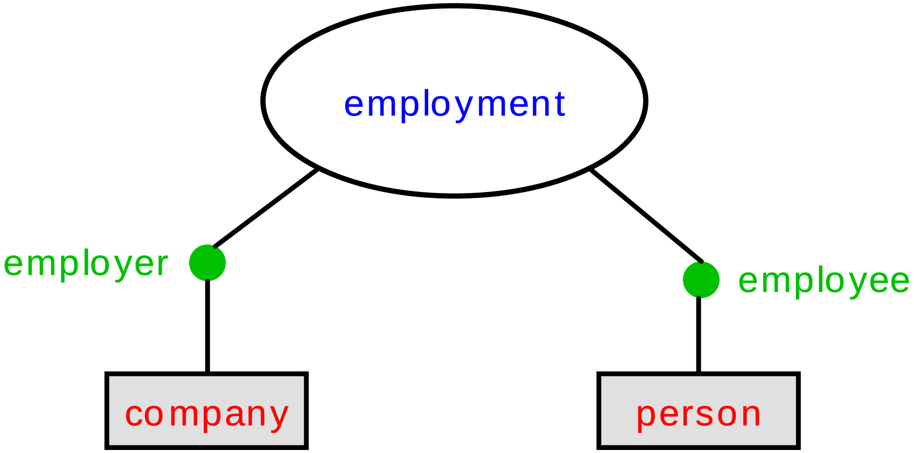
A topic map with types

```
<topicMap>
<topic id="company">
      <baseName><baseNameString>Company</baseNameString></baseName> </topic>
<topic id="homepage">
   <baseName><baseNameString>Home  Page</baseNameString></baseName> </topic>
<topic id="year-established">
   <baseName> <baseNameString>Year Established</baseNameString> </baseName> </topic>
<topic id="xzyyz">
    <instanceOf> <topicRef xlink:href="#company"/> </instanceOf>
   <baseName> <baseNameString>Redmond Computers Inc.</baseNameString> </baseName>
   <occurrence>
      <instanceOf> <topicRef xlink:href="#homepage"/> </instanceOf>
     <resourceRef xlink:href="http://www.redmondcomputers.com/"/> </occurrence>
   <occurrence>
      <instanceOf> <topicRef xlink:href="#year-established"/> </instanceOf>
     <resourceData>1977</resourceData> </occurrence> </topic>

</topicMap>
```

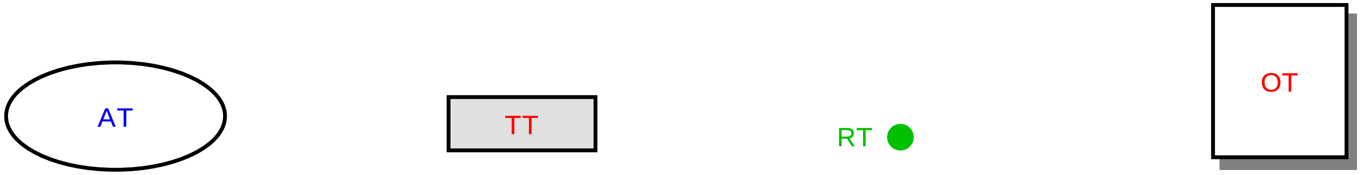
VisualScript for Topic Maps

- A topic map ontology
  - a set of typing topics
  - constraints on the ways in which their instances may be combined
- Association templates.
  - The graph structure with a set of interconnected nodes.



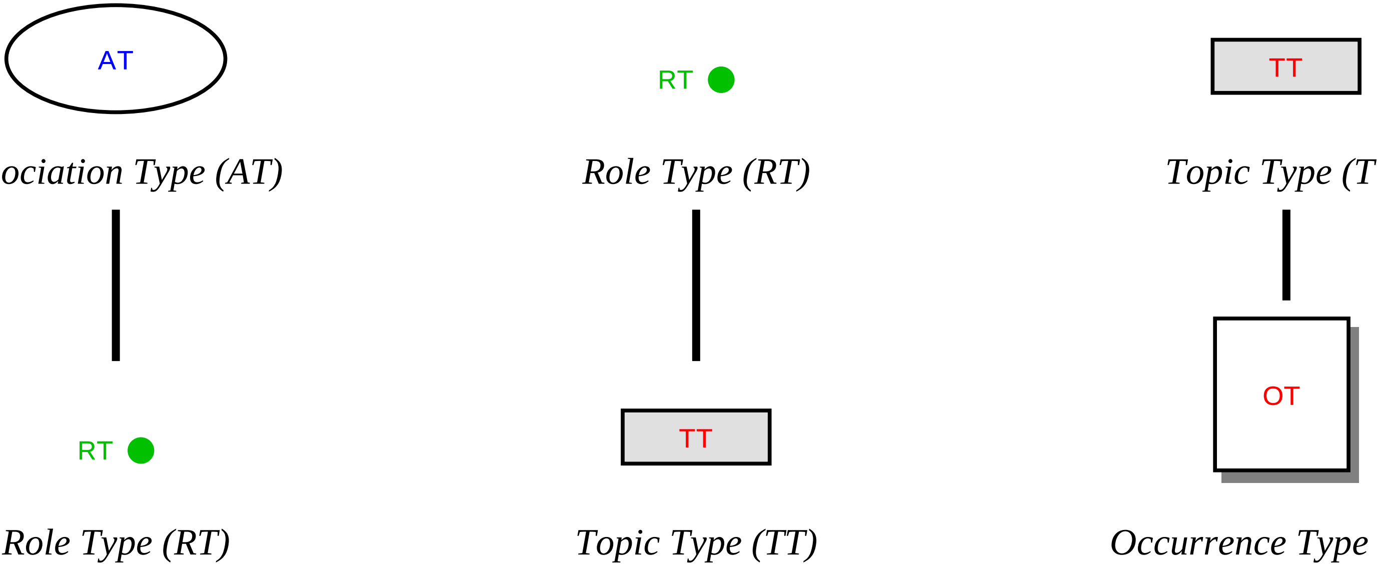
VisualScript for Topic Maps

- Basic Building Blocks for Topic Map Ontologies



Association Type (AT) *Topic Type (TT)* *Role Type (RT)* *Occurrence Type (OT):*

- Connectors

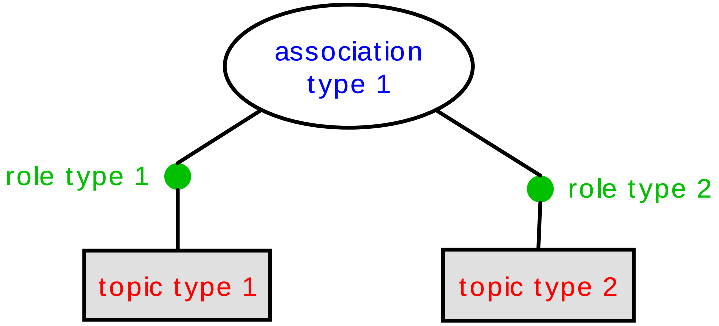


Association Type (AT) *Role Type (RT)* *Topic Type (TT)*

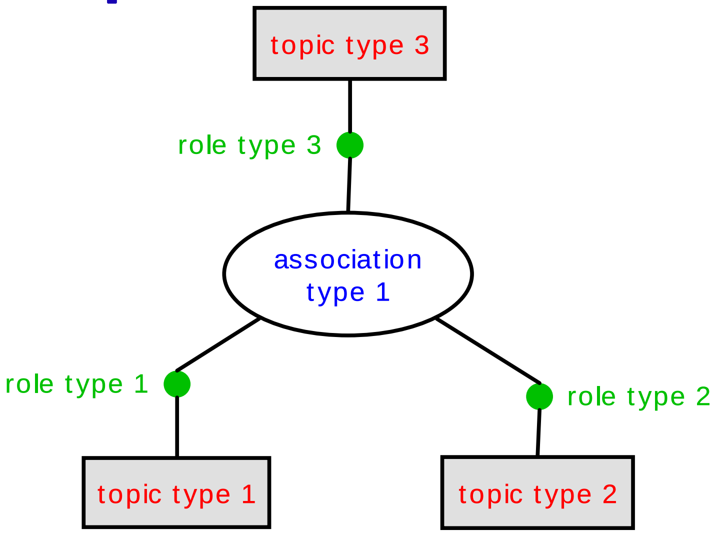
Role Type (RT) *Topic Type (TT)* *Occurrence Type (OT):*

VisualScript for Topic Maps

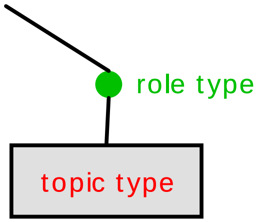
- **Templates**



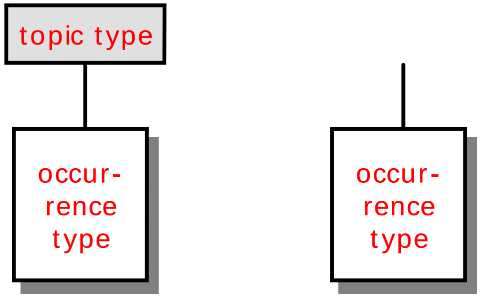
Binary association template



Ternary association template



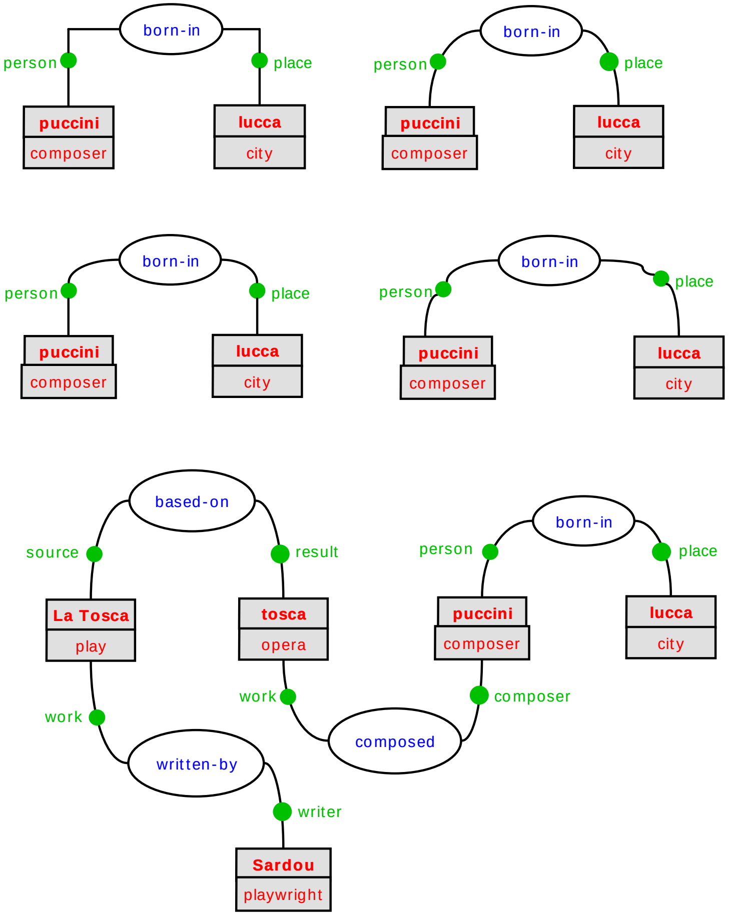
Role template



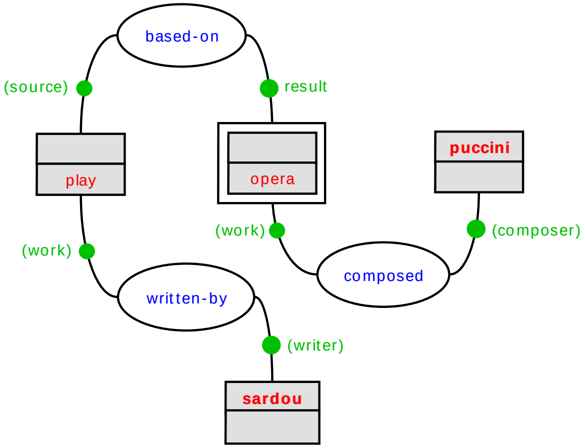
Occurrence templates

VisualScript for Topic Maps

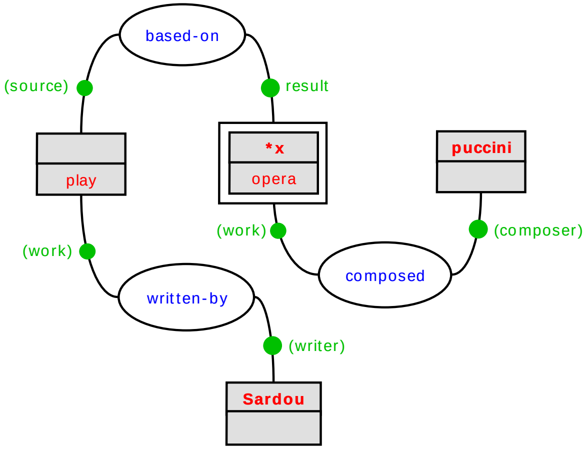
- Example



Representing the query:
"Which operas written by Puccini were based on plays by Sardou"



Representing the query:
"Which operas written by Puccini were based on plays by Sardou"

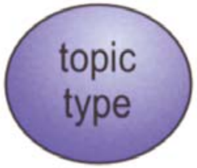


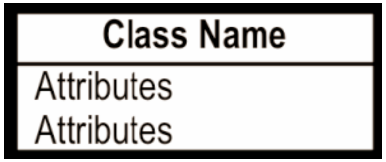
Object Role Modeling (ORM) versus UML

- Topic Maps Constructs
  - Topics, Associations and Occurrences (TAO)
  - Topic Names (Basic Names and Variant Names), Types, Scope, and Association role
- An ORM model is essentially a connected network of object types and relationship types
  - ORM classifies objects into entities (non-lexical objects) and values (lexical objects)

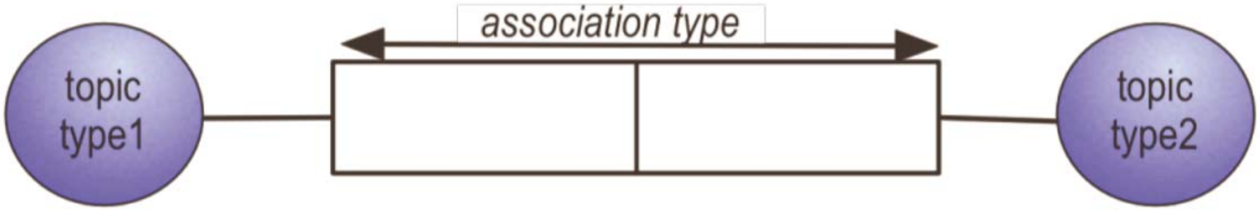
Object Role Modeling (ORM) versus UML

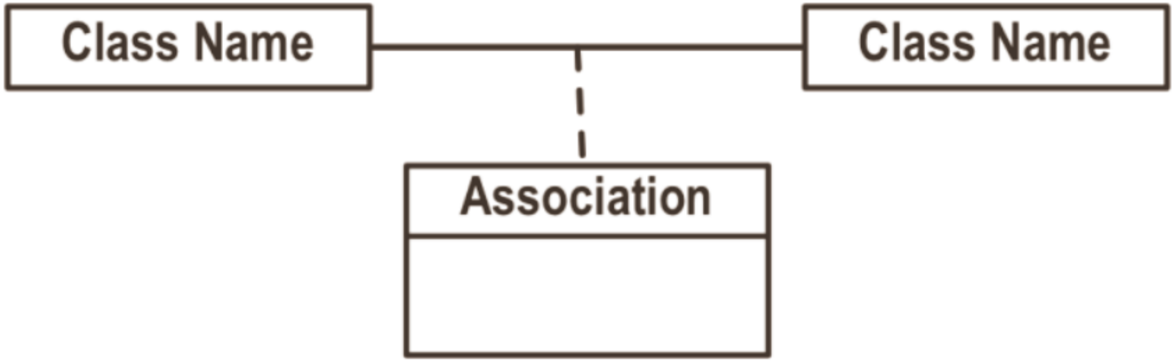
- Topic: An entity type versus a class





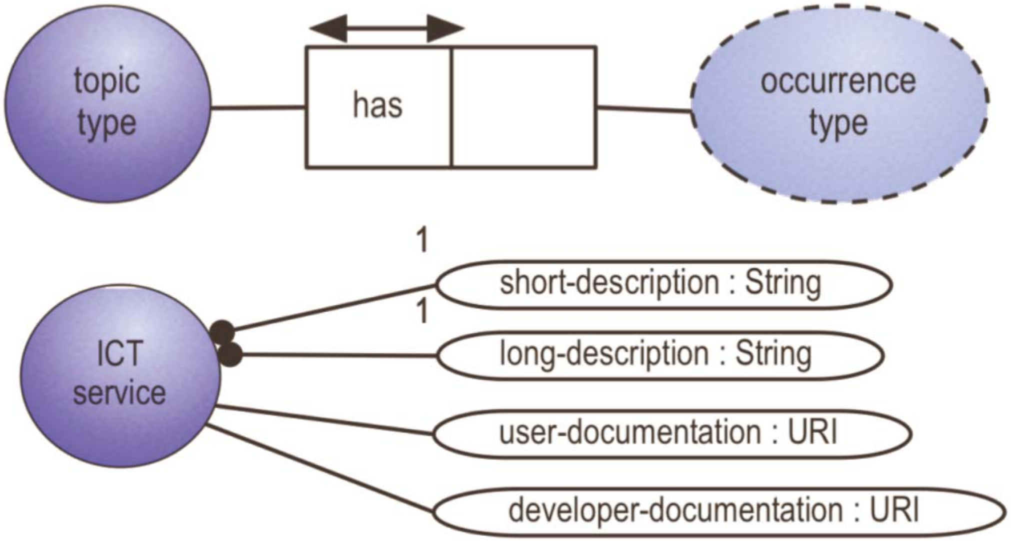
- Association: a relationship type versus an association

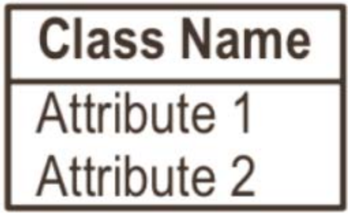




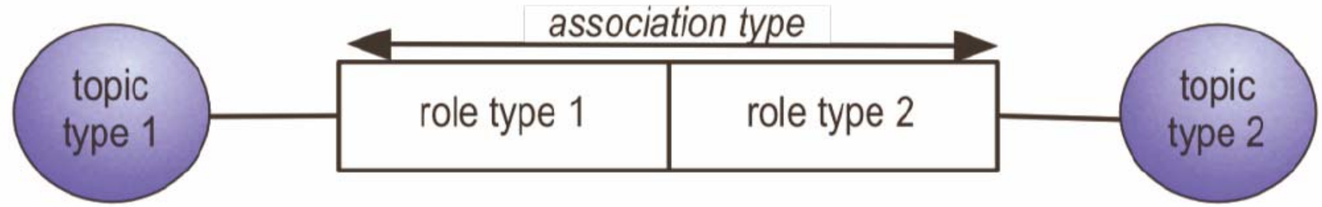
Object Role Modeling (ORM) versus UML

- Occurrence: a value type versus an attribute


topic type
has
occurrence type
1
short-description : String
1
ICT service
long-description : String
user-documentation : URI
developer-documentation : URI



- Association roles

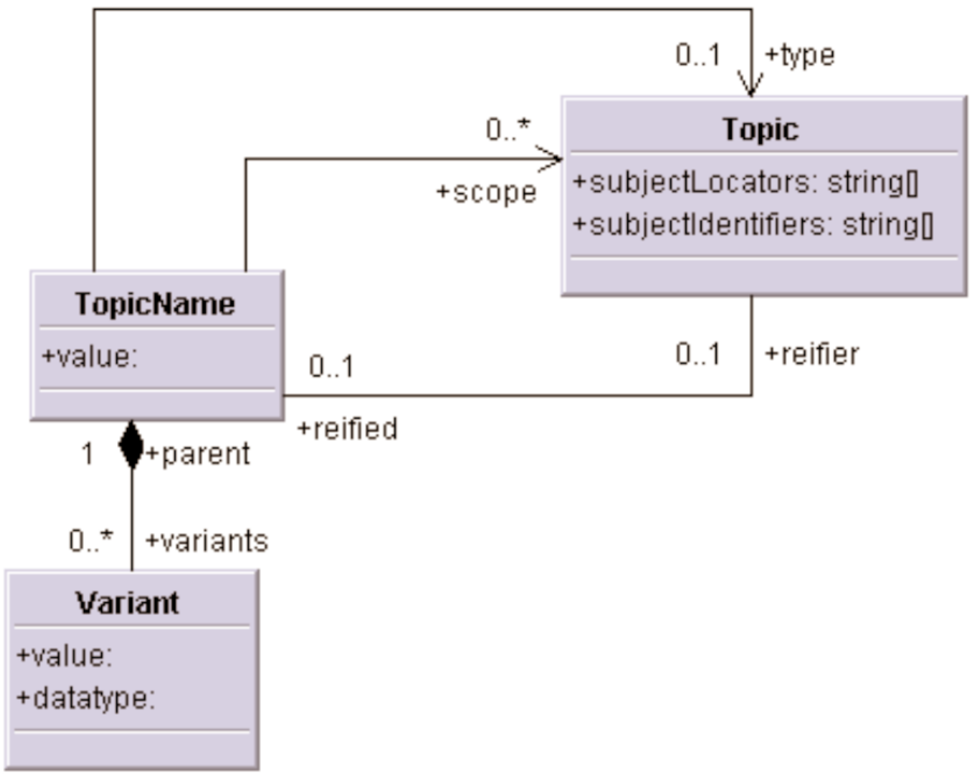


Object Role Modeling (ORM) versus UML

- Topic names
  - a specialized kind of association between one topic and potentially several string values that represent different types of names for the topic within different scopes (contexts).
  - A name consists of basename and may have several variant names.
  - The name may also have one type and the scope may consist of several themes.
    - The standard says that a basename essentially is a specialized kind of occurrence
    - If we model a simple name with one type and without variants, we may model it as an occurrence.
    - A name with variants or several types will have to be modeled as a relationship (sentence).

Object Role Modeling (ORM) versus UML

- Topic names



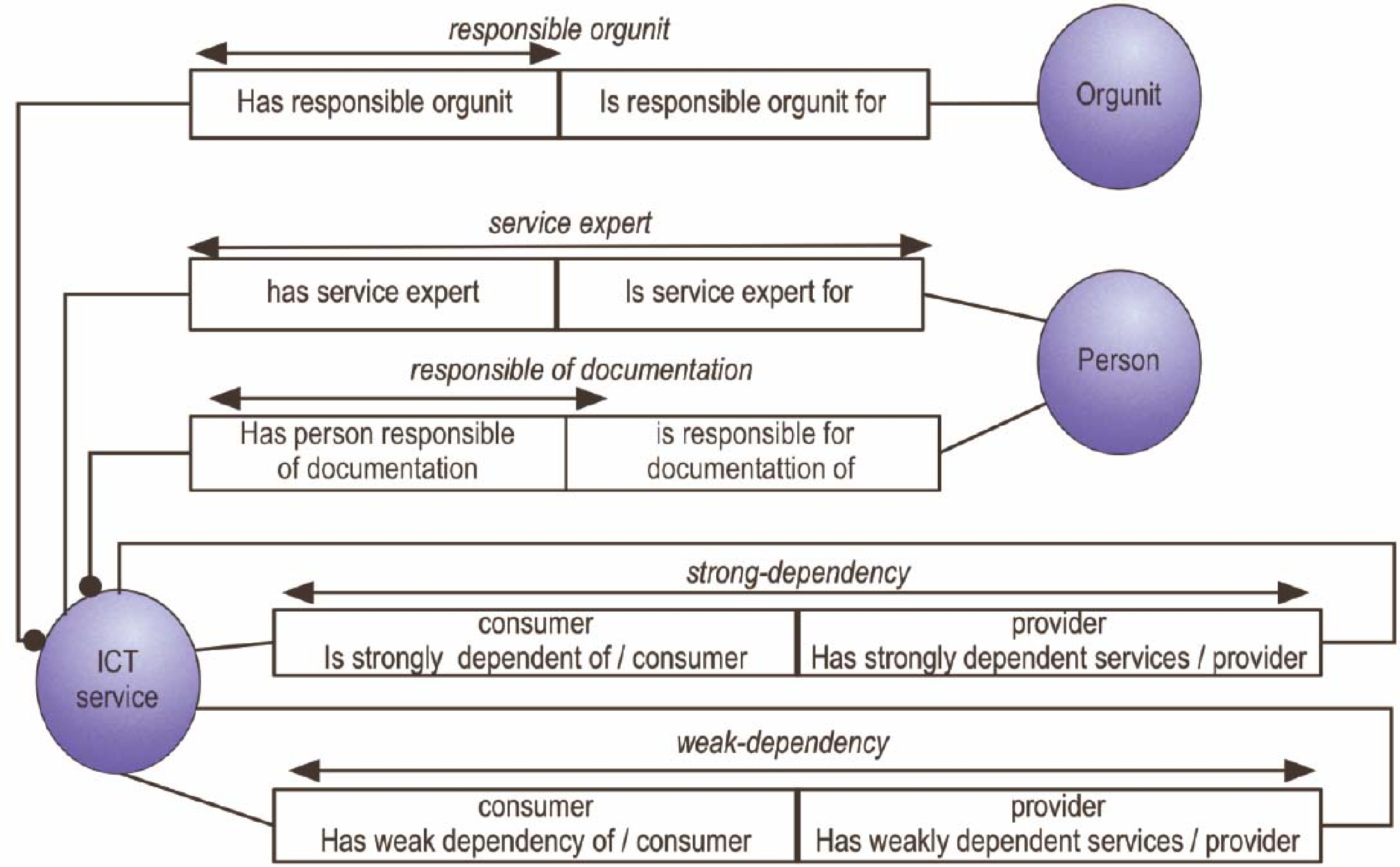
- Themes and scopes
  - Topics are used as themes (a scope may consists of many themes, - a scope is a set of themes).
  - It is easiest to model scope as a textual notation.

Object Role Modeling (ORM) versus UML

- Evaluation Criteria
  - Expressibility
    - how much of the application domain we can model
  - Clarity
    - how easy it is to understand and use
  - Semantic stability
    - How easy it is to change an attribute into a class in UML, or an occurrence changing into a topic in Topic Maps model
  - Semantic relevance
  - Validation mechanisms
    - ways in which domain experts can check whether the model matches the "real world"
  - Abstraction mechanisms
    - that allow unwanted details to be removed from immediate consideration.

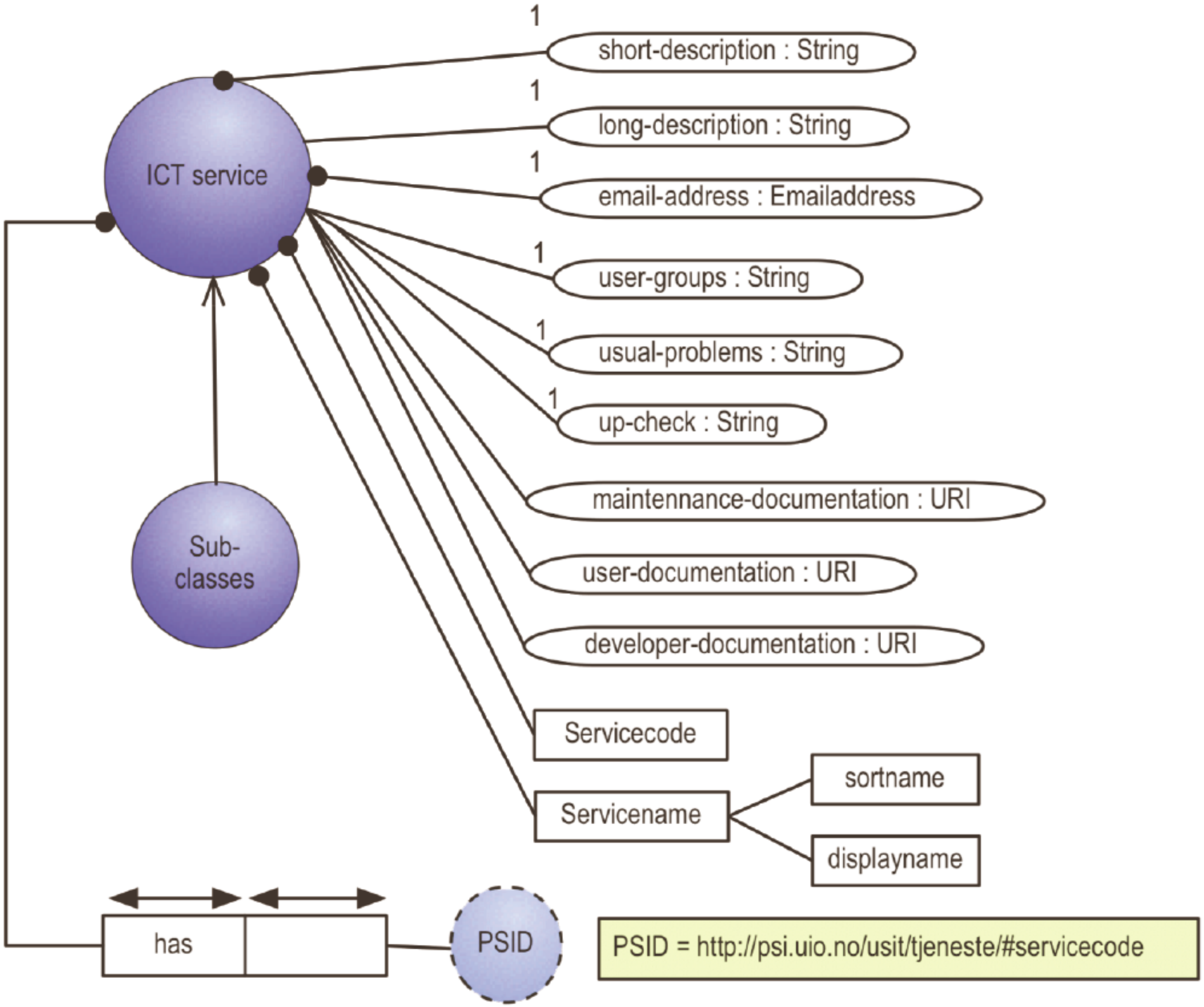
Object Role Modeling (ORM)

- Example


responsible orgunit
Has responsible orgunit
Is responsible orgunit for
Orgunit
service expert
has service expert
Is service expert for
Person
responsible of documentation
Has person responsible of documentation
is responsible for documentattion of
strong-dependency
consumer Is strongly dependent of / consumer
provider Has strongly dependent services / provider
ICT service
weak-dependency
consumer Has weak dependency of / consumer
provider Has weakly dependent services / provider

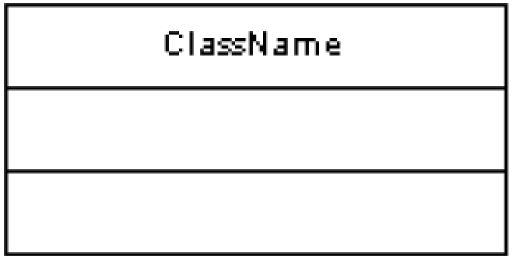
Object Role Modeling (ORM)

- Example


ICT service
1
short-description : String
1
long-description : String
1
email-address : Emailaddress
1
user-groups : String
1
usual-problems : String
1
up-check : String
maintennance-documentation : URI
user-documentation : URI
developer-documentation : URI
Servicecode
Servicename
sortname
displayname
Sub-classes
has
PSID
PSID = http://psi.uio.no/usit/tjeneste/#servicecode

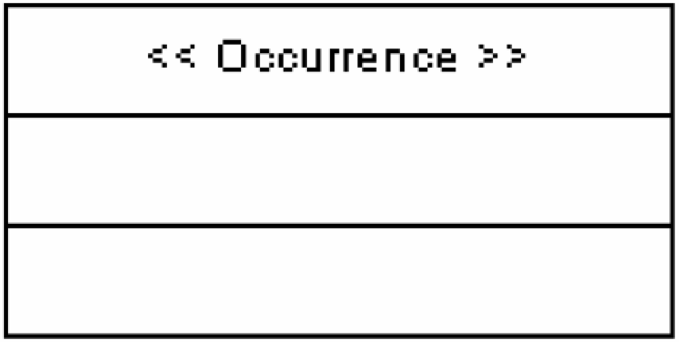
A simple UML-based notation for Topic Maps

- Beyond PSIs: Topic Map design patterns, Extreme Markup Languages 2003
- Topics
  - Topics are represented by the 3-segment box notation used to represent a class.
  - The name in the top segment of the box gives the name of the topic type.
  - Unlike a UML class diagram, in the topic map diagram there is not necessarily a direct correlation between the name given to the topic class and the name as it is represented in a topic map.

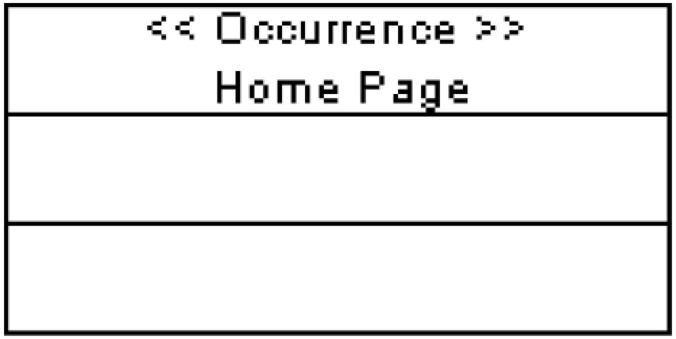


A simple UML-based notation for Topic Maps

- Occurrences and BaseNames
  - Occurrences and BaseNames are represented with the same class notation as topics.
  - As with the Topic notation, the name of the class relates directly to the topic used to define the class of occurrences or names that the instance belongs to.
  - To distinguish the three different uses of the class notation, we use a class stereotype.

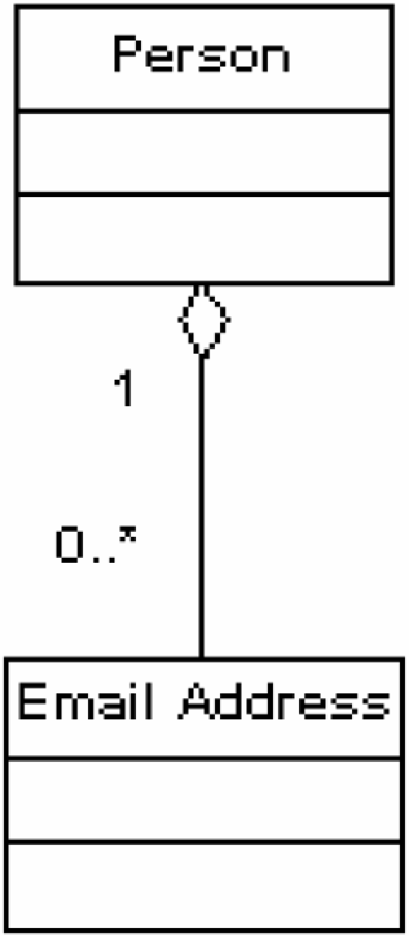


An untyped Occurrence



An typed Occurrence

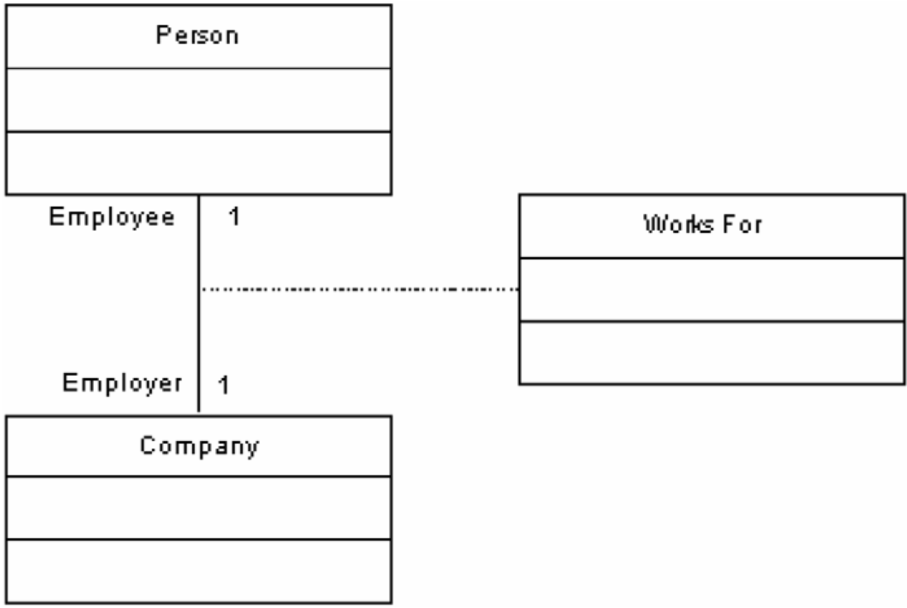
A simple UML-based notation for Topic Maps



A topic with Occurrence

A simple UML-based notation for Topic Maps

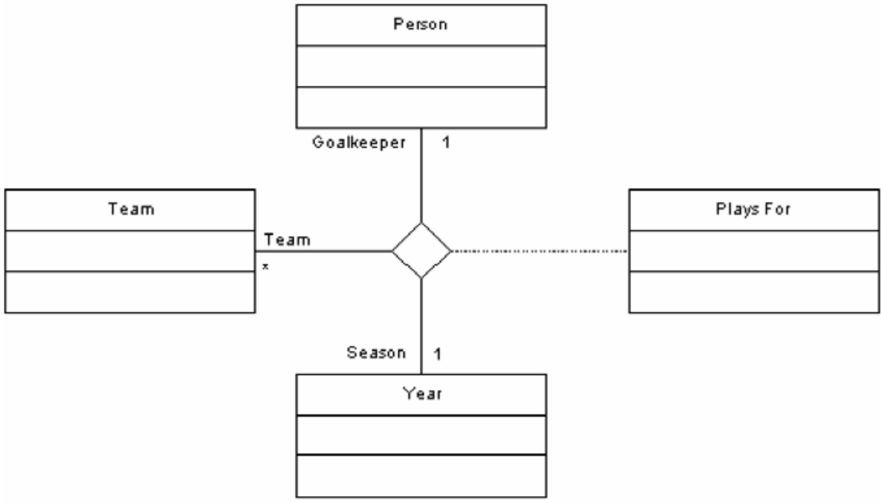
- Associations and association roles
  - The class notation can also be used to represent classes of association by additionally making use of the UML notation for association classes.
  - The association itself is represented as a solid line joining the role-playing classes.
  - The roles played are annotated against the line along with any cardinality constraints imposed by the association class.
  - The association class is represented by a three segment box connected to the association by a dotted line.



A typed binary association

A simple UML-based notation for Topic Maps

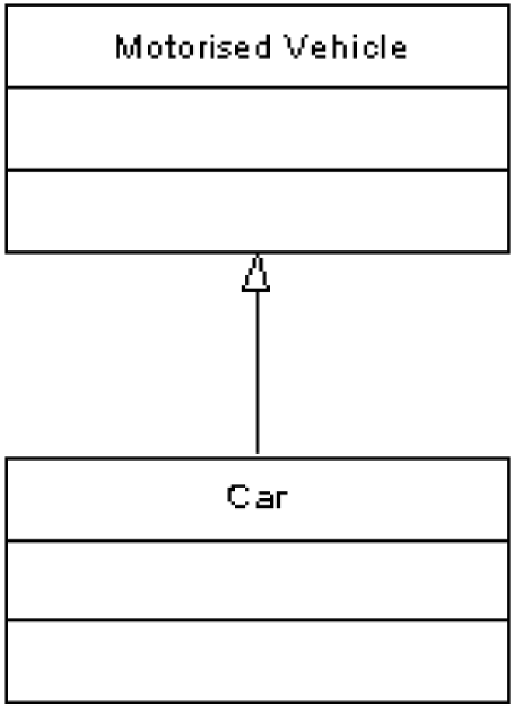
- Associations and association roles
  - N-ary associations are also supported by UML class diagram notation.
  - In this case, rather than having a single solid line, the association is represented by an open diamond shape with the role-playing classes connected to the diamond by solid lines.
  - The class of the association is represented with a dotted line from the diamond to the association class.



A typed 3-ary association

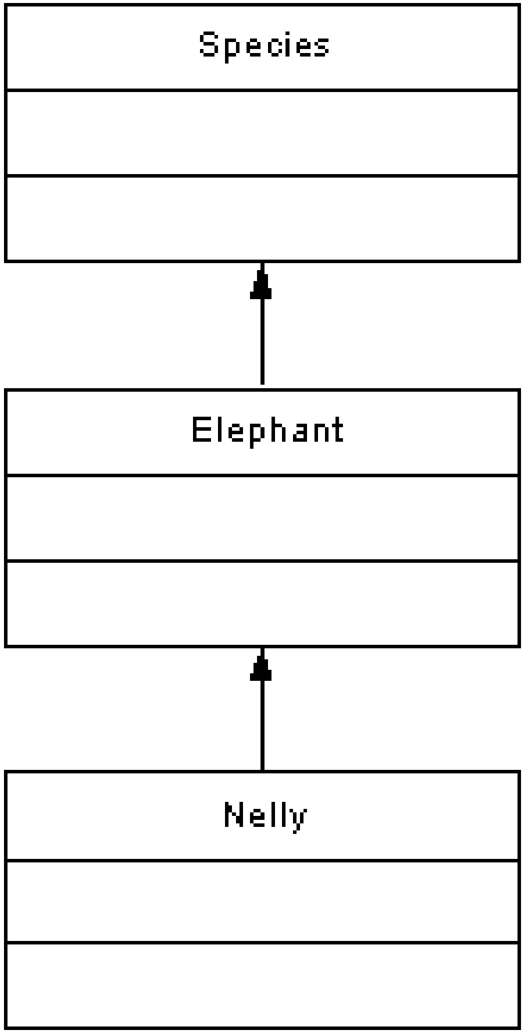
A simple UML-based notation for Topic Maps

- Subclass-superclass associations
  - Although in a topic map, the subclass-superclass relationship is defined through the use of associations, it makes sense in the diagrammatic notation to make use of the standard UML notation as a shortcut.
  - Rather than having a single solid line, the association is represented by an open diamond shape with the role-playing classes connected to the diamond by solid lines.



A simple UML-based notation for Topic Maps

- Instances in Topic Map class diagrams
  - Because in a topic map it is possible for a class to also be an instance, it may sometimes be necessary to create a UML diagram which would not be possible to translate directly into most OO programming languages.

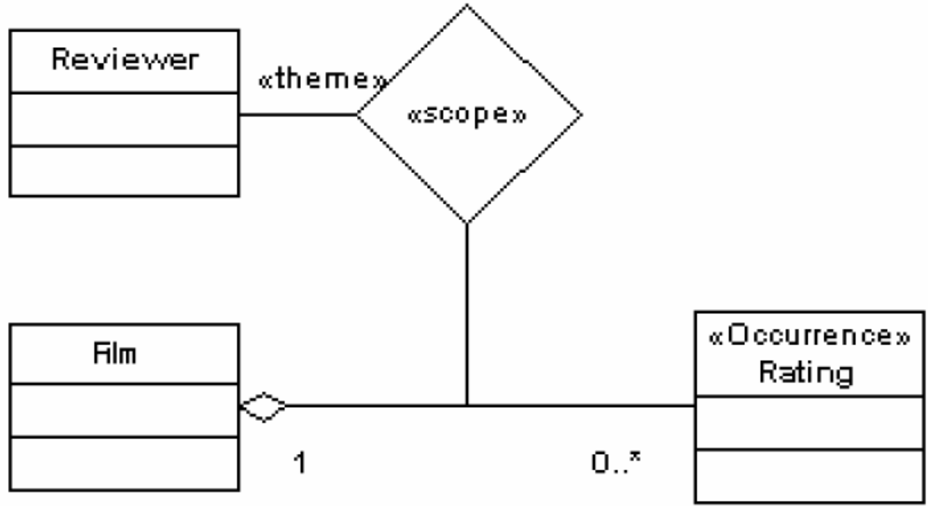


A simple UML-based notation for Topic Maps

- Scope
  - The topic map class diagram notation must support the specification of scope applied to topic names, occurrences and associations.

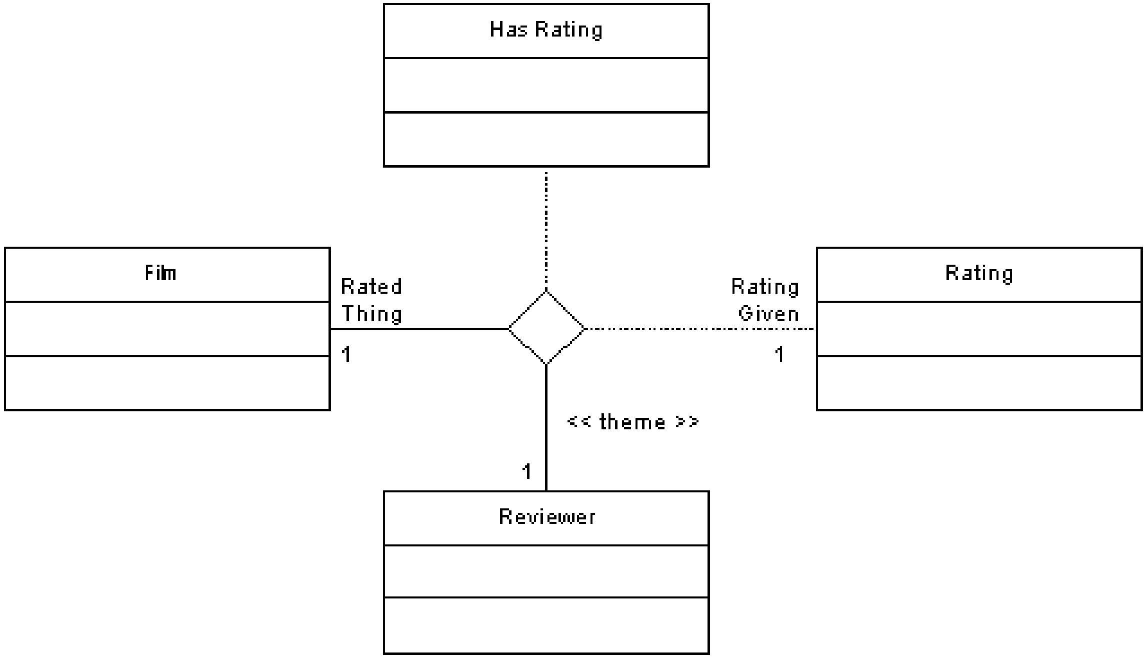
A simple UML-based notation for Topic Maps

- A scoped Occurrence
  - This requirement is met by defining a stereotype named "scope" which can be applied to a UML association and a stereotype "theme" which can be applied to a UML association link
  - The stereotyped Association can then be applied to the aggregation links used between a Topic class and a BaseName or Occurrence class; and the AssociationEnds with the stereotype "theme" can be used to apply the themes.



A simple UML-based notation for Topic Maps

- A scope association
  - The topic map diagramming notation just adds the extra theme-sterotyped association ends directly to the association being scoped and does away with the scope stereotype on the association.



Discussion

- …

Developing GTM in Parallel with CTM

- GTM and CTM as equivalent notations of each other
- GTM needs to developed in accordance with CTM, TMQL, and TMCL

Evaluation Criteria

- Conciseness of the language
- Easy to use
- Compactness
- Being part of a coherent language family

- Expressibility
- Clarity
- Validation mechanisms
- Abstraction mechanisms
- Modularity / Compositionality
- Understandability
- Implementation effort

What's next?

- NP?
- Editors?