Information Interchange on the Semantic Web

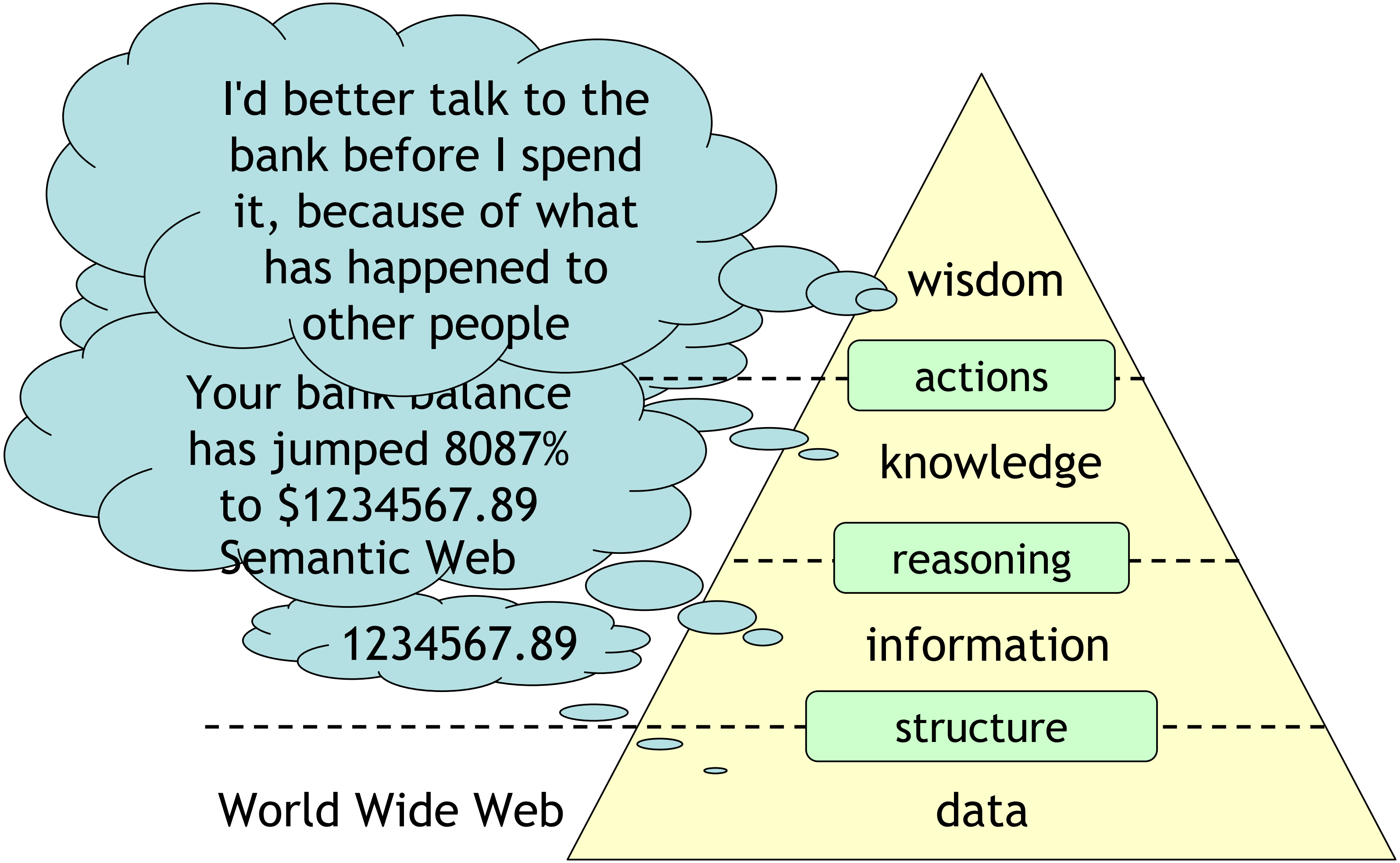
an interactive talk

by Piotr Kaminski, University of Victoria

www.ideanest.com

The Semantic Web

- Getting machines to *understand* the contents of the web



I'd better talk to the bank before I spend it, because of what has happened to other people

Your bank balance has jumped 8087% to $1234567.89

Semantic Web

1234567.89

World Wide Web

wisdom

actions

knowledge

reasoning

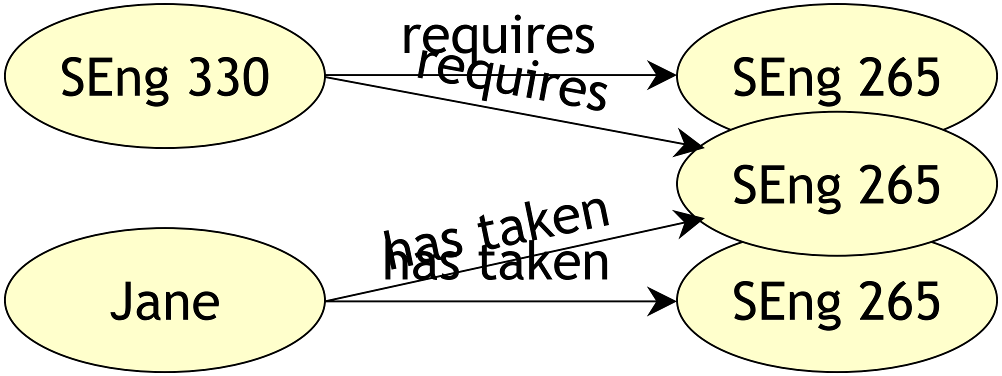
information

structure

data

Web Contents

- Why is the World Wide Web valuable?
- The Semantic Web:
  - contains statements about things
  - must be able to combine statements that talk about the same thing



Uniform Resource Identifiers

- A *resource* is

  anything that has an identity

- A *URI* identifies a single resource
  - decentralized, uniform syntax (RFC 2396)
  - URL (Locator) and URN (Name) are legacy terms

- If you go fetch a URI, do you retrieve its resource?

- What does http://www.engr.uvic.ca/~seng330/ identify?

Ontologies

- An *ontology* is a dictionary:

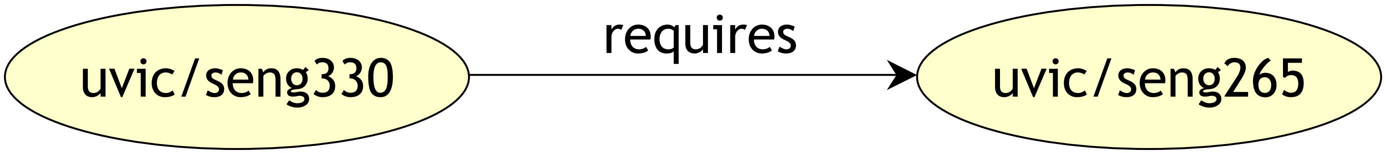
Identifier	Thing
http://www.ideanest.com/uvic	UVic itself
http://www.ideanest.com/uvic/seng330	SEng 330 course
http://www.ideanest.com/requires	one course having another as prereq

  - resources identified well defined and agreed upon
  - may also include relationships between resources, inference rules
  - can be general (upper ontology) or specialized
  - still decentralized!

Resource Descr. Framework

- A Semantic Web model proposal:
  - simple set of subject—predicate—object triples
  - many notations:

Graphs:



prefix: http://www.ideanest.com/

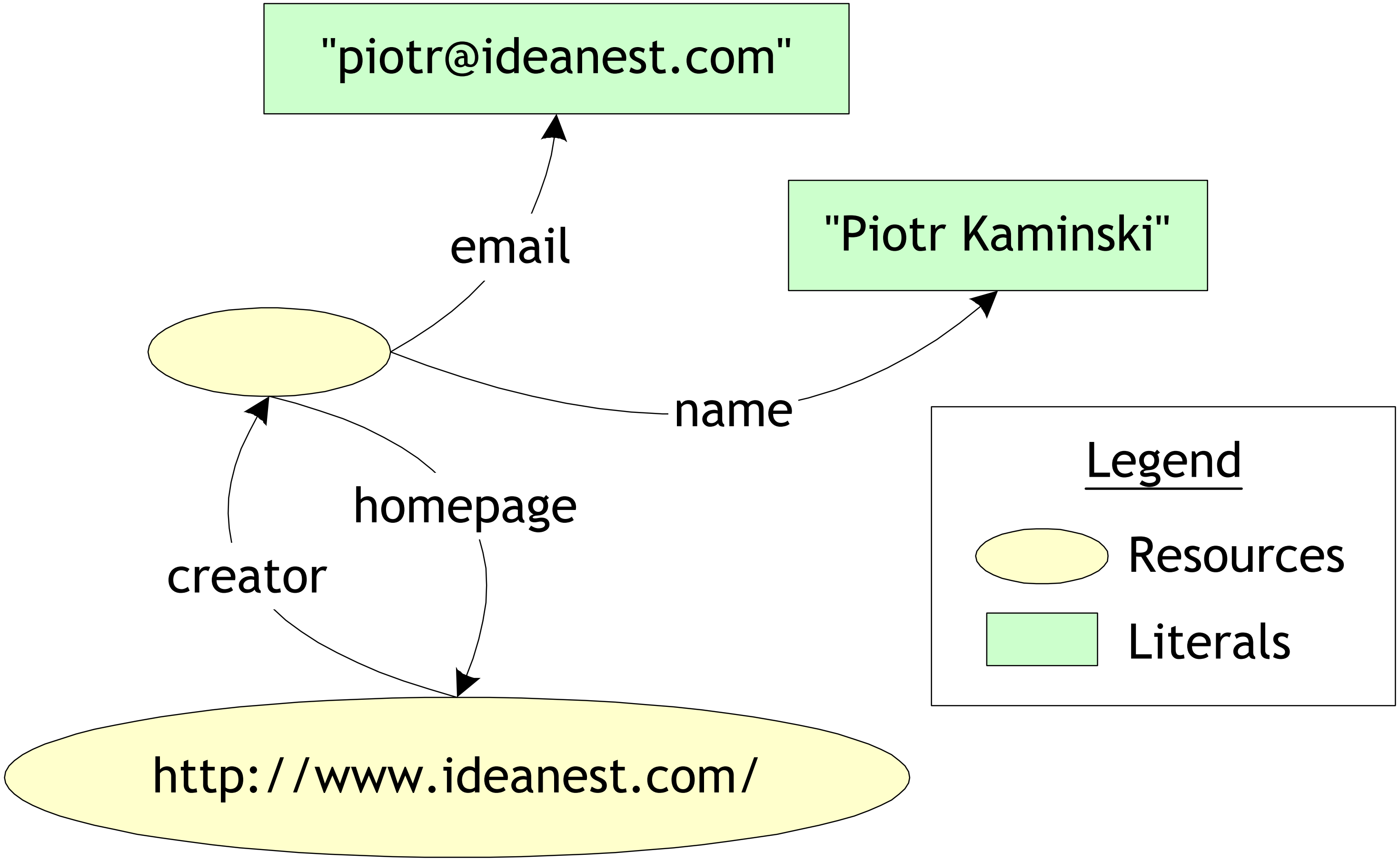
Notation3:

```
@prefix in: http://www.ideanest.com/
in:uvic/seng330 in:requires in:uvic/seng265
```

XML:

```
<rdf:RDF   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:in="http://www.ideanest.com/">
 <rdf:description rdf:about="http://www.ideanest.com/uvic/seng330">
  <in:requires rdf:resource="http://www.ideanest.com/uvic/seng265"/>
</rdf:description></rdf:RDF>
```

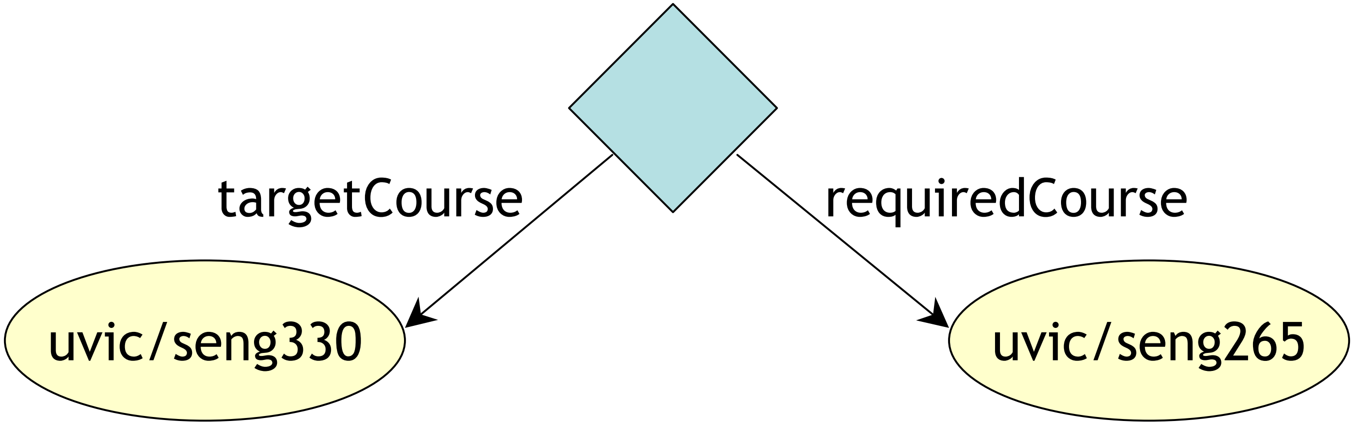
RDF Example



- literals are primitive values
- anonymous *bnodes* are identified by properties

Topic Maps

- Another Semantic Web model proposal:
  - set of topics and associations
  - a *topic* represents a *subject resource*
  - an *association* relates a set of topics together
    - each topic plays one or more *roles*



prefix: http://www.ideanest.com/

Identifying Subjects

- A topic can identify its subject by stating:
  - the primitive value

    "http://www.uvic.ca/" = just a string

  - the identifier of the resource

    http://www.uvic.ca/ = UVic home page / web site

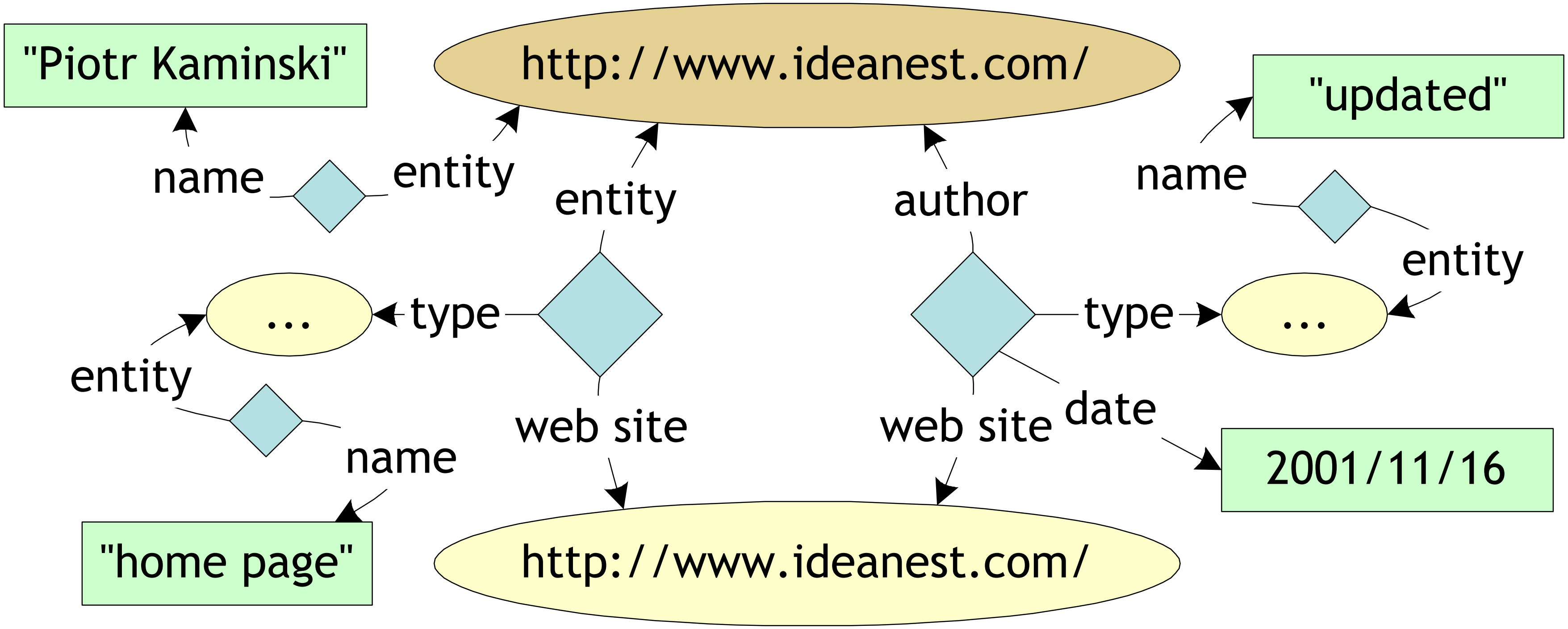
  - the identifier of a resource that indicates the subject

    http://www.uvic.ca/ = UVic itself

    - can also be done using newly proposed URI scheme *tdb*:

    tdb:http://www.uvic.ca/ = thing described by…

Topic Map Example

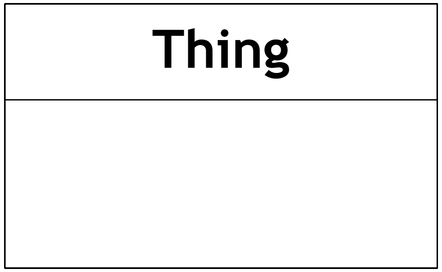


- n-ary associations are allowed (and encouraged)

- What's the obvious mapping to RDF?

Modeling the World

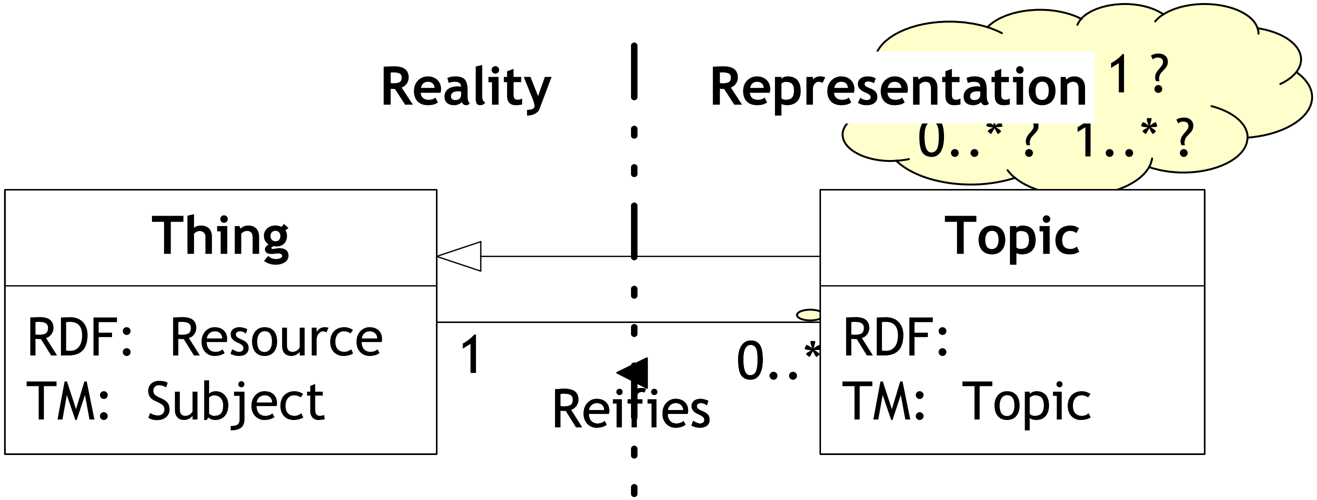
- The one point everybody agrees on:



Thing

…though not on its name

- Our software is part of the domain, so we explicitly show software artifacts:



Reality

Representation

Thing

RDF: Resource

TM: Subject

Topic

RDF:

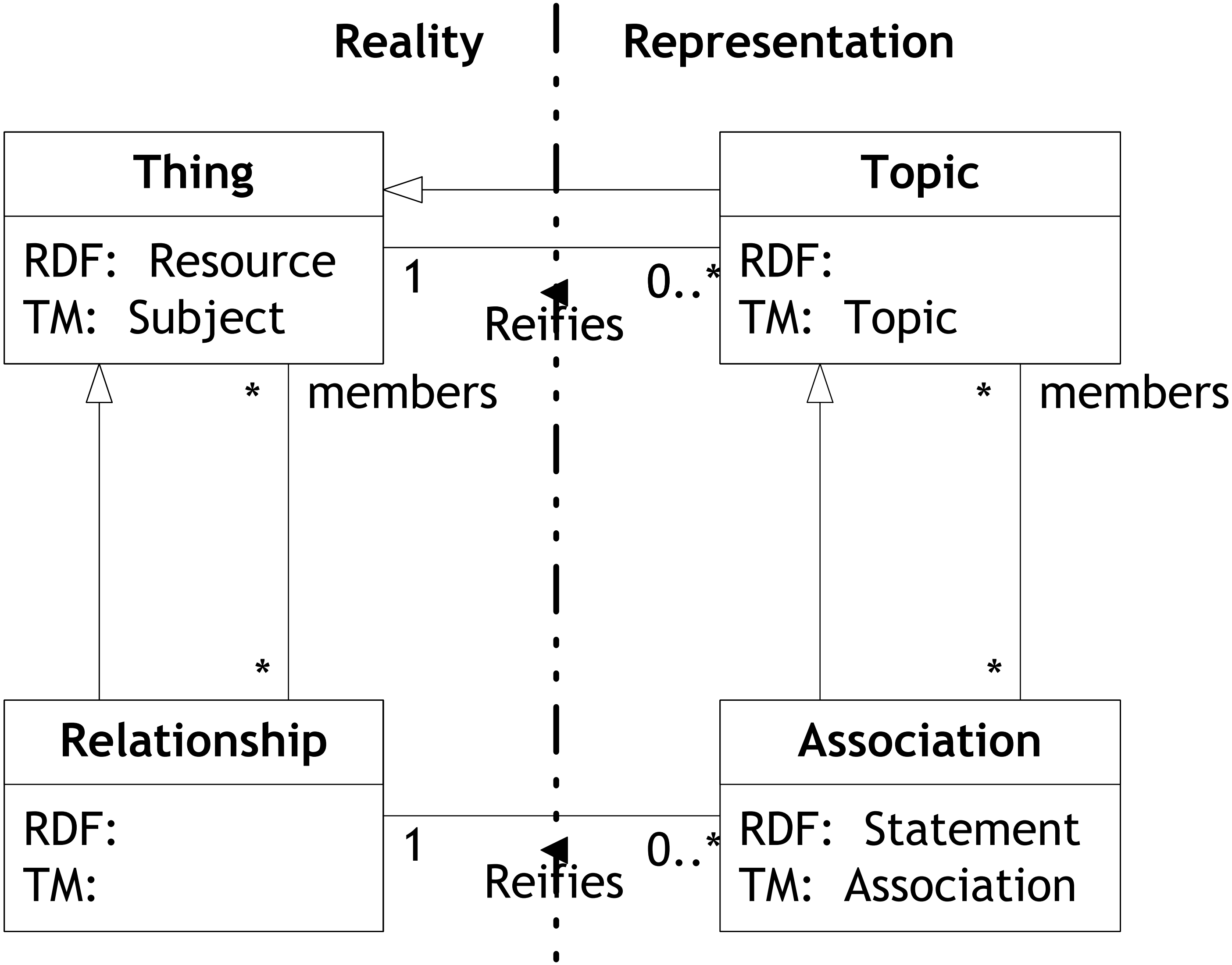
TM: Topic

1

0..*

Reifies

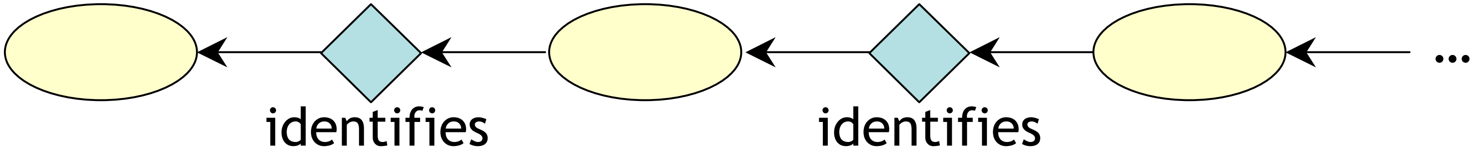
Relating Things



Identifying Things

- Identification is a complex problem:
  - may be indirect
  - may be scoped
  - might need to be reasoned about

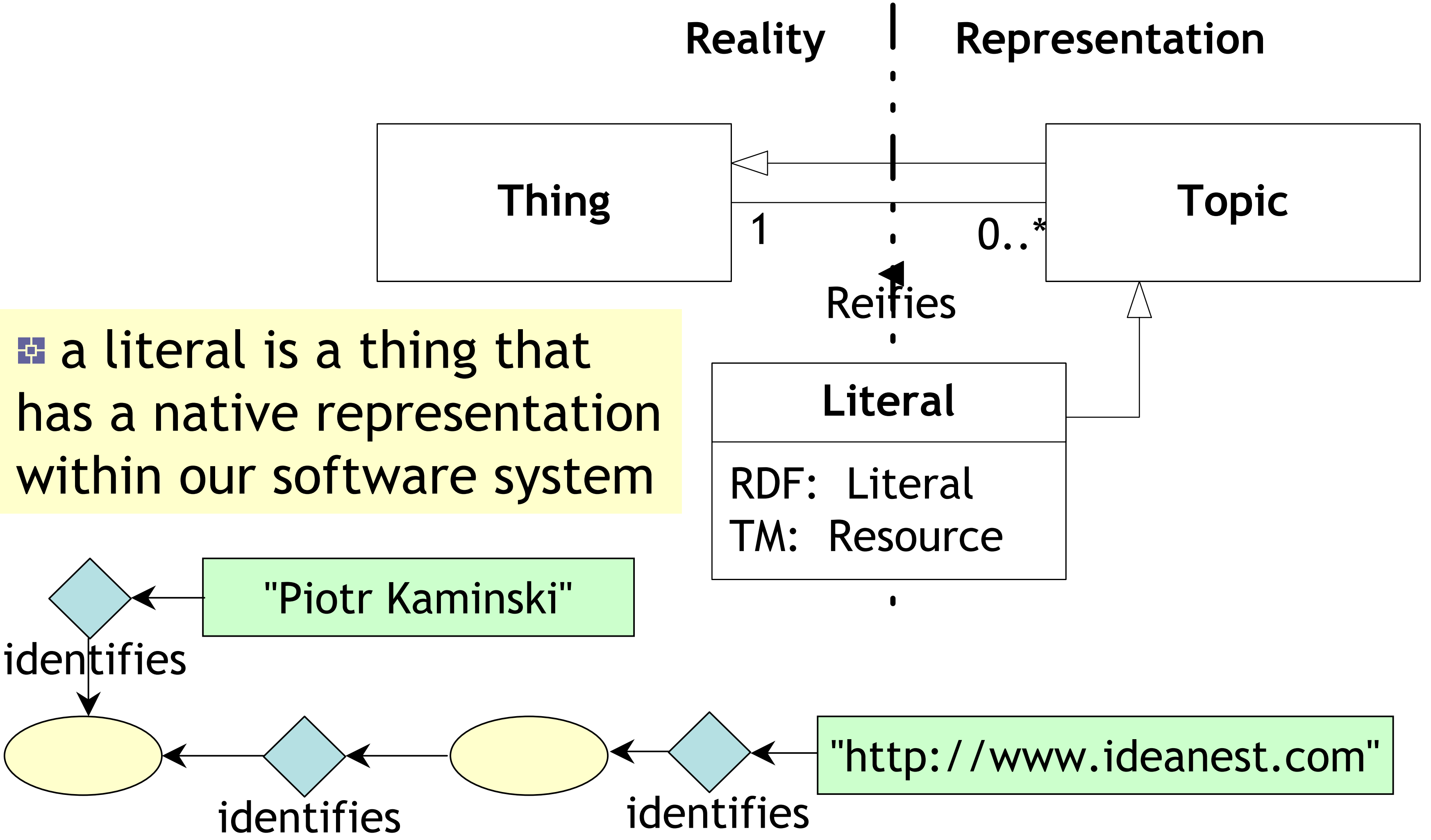
- Let "identifies" be a normal relationship between Things



  - define it in a standard upper ontology

Literals

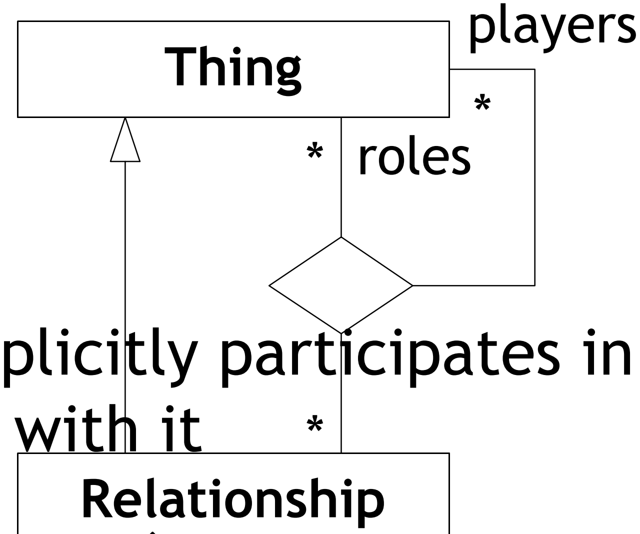
- We must bootstrap the identification chains:



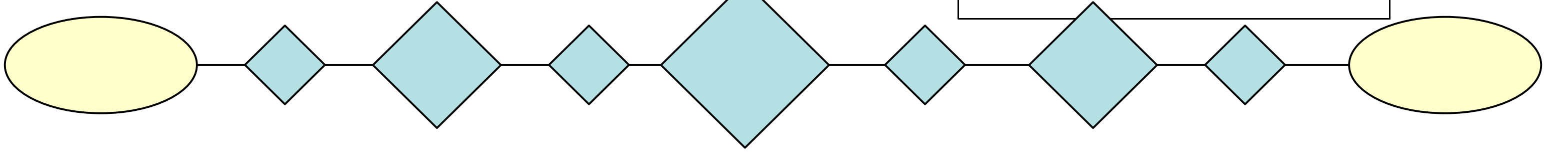
- a literal is a thing that has a native representation within our software system

Recursive Relationships

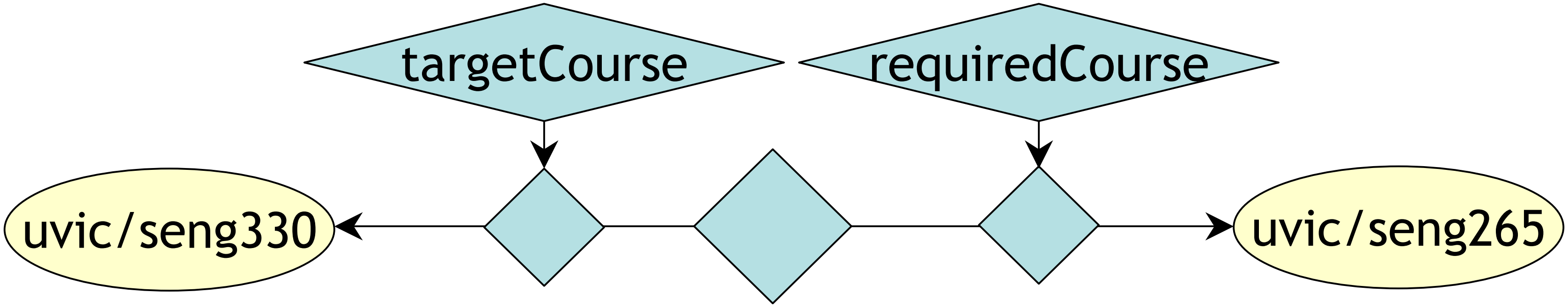
- How to model Topic Map roles?
  - could make it a primitive in the model…



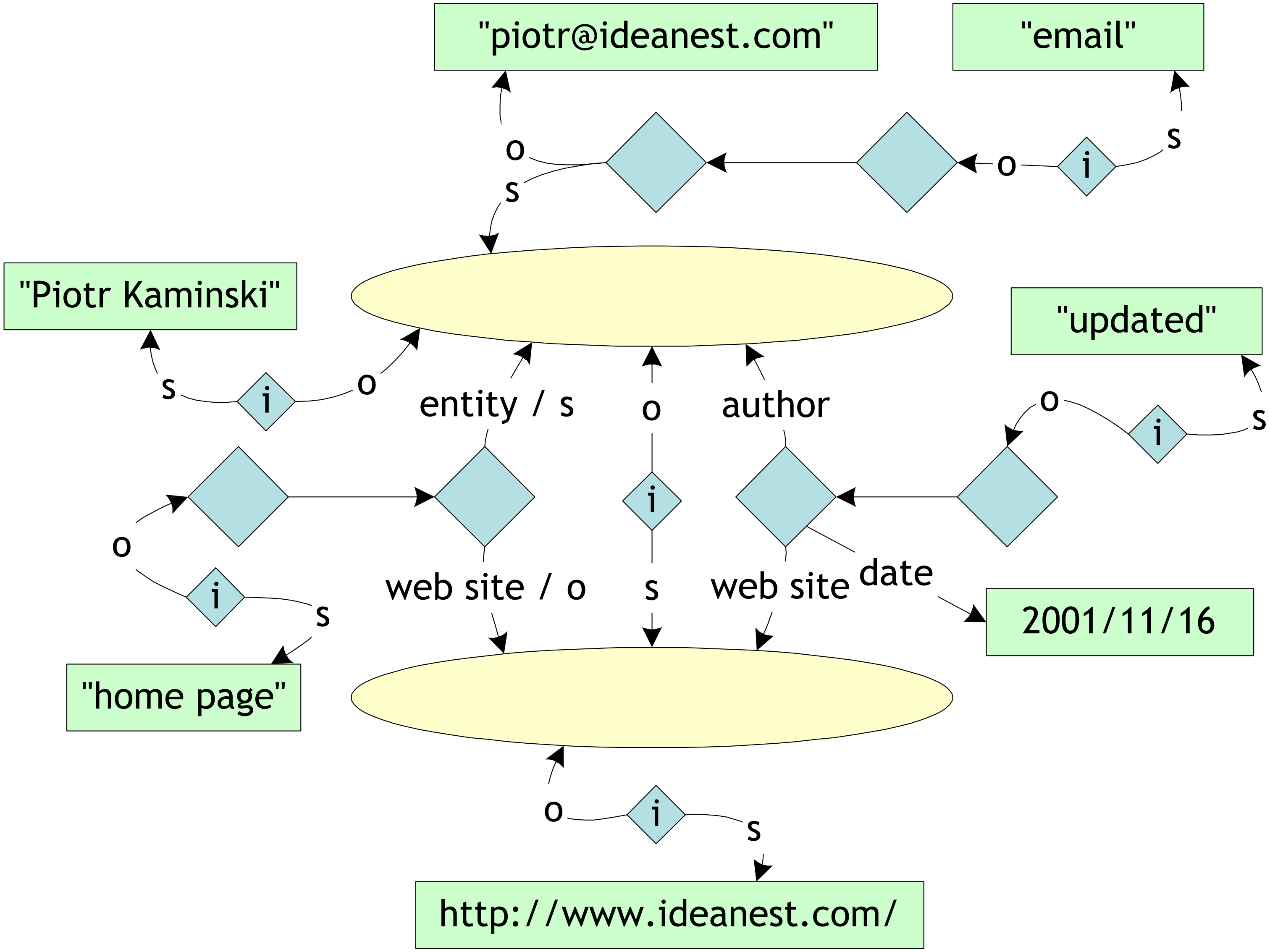
- Each member of a relationship implicitly participates in a binary "member of" relationship with it



  - only one level necessary for assigning roles



Integrating Models



Last Thoughts

- We've only scratched the surface:
  - type hierarchies, classes vs. properties
  - collections, ordering of members
  - contexts / scopes
  - reflection / reification
- Why not XML?
  - no universal interpretation
  - no universal representation for arbitrary graphs
- Project in progress:

Braque

www.ideanest.com