Semantic modelling

Service-Oriented Architecture

Jeremy Gibbons

Contents

1 The semantic web vision

- from the human web to the semantic web
- standardisation, representation, formalisation
- smart discovery, query, integration, interaction...
- machine-processable information
- supporting software agents

1.1 A scenario

The entertainment system was belting out the Beatles' *We Can Work It Out* when the phone rang. When Pete answered, his phone turned the sound down by sending a message to all the other **local** devices that had a **volume control**. His sister, Lucy, was on the line from the doctor's office: "Mom needs to see a specialist and then has to have a series of physical therapy sessions. Biweekly or something. I'm going to have my agent set up the appointments." Pete immediately agreed to share the chauffeuring.

At the doctor's office, Lucy instructed her Semantic Web agent through her handheld Web browser. The agent promptly retrieved information about Mom's **prescribed treatment** from the doctor's agent, looked up several lists of **providers**, and checked for the ones **in-plan** for Mom's insurance within a **20-mile radius** of her **home** and with a rating of **excellent** or **very good** on **trusted rating services**. It then began trying to find a match between available **appointment times** (supplied by the agents of individual providers through their Web sites) and Pete's and Lucy's busy schedules.

In a few minutes the agent presented them with a plan. Pete didn't like it—University Hospital was all the way across town from Mom's place, and he'd be driving back in the middle of rush hour. He set his own agent to redo the search with stricter preferences about **location** and **time**. Lucy's agent, having **complete trust** in Pete's agent in the context of the present task, automatically assisted by supplying access certificates and shortcuts to the data it had already processed.

Almost instantly the new plan was presented: a much closer clinic and earlier times—but there were two warning notes. First, Pete would have to reschedule a couple of his **less important** appointments. He checked what they were—not a problem. The other was something about the insurance company's list failing to include this provider under **physical therapists**: "Service type and insurance plan status securely verified by other means," the agent reassured him. "(Details?)"

Lucy registered her assent at about the same moment Pete was muttering, "Spare me the details," and it was all set. (Of course, Pete couldn't resist the details and later that night had his agent explain how it had found that provider even though it wasn't on the proper list.)

(http://www.sciam.com/article.cfm?id=the-semantic-web)

1.2 Doesn't XML give you semantic representations?

As enchanting as it is to contemplate the apparent 'semantic' clarity, flexibility, and extensibility of XML vis-á-vis HTML (e.g., how wonderfully perspicuous XML <bookTitle> seems when compared to HTML <i>), we must reckon with the cold fact that XML does not of itself enable blind interchange or information reuse. XML may help humans predict what information might lie 'between the tags' in the case of <trunk></trunk>, but XML can only help. For an XML processor, <trunk> and <i> and <bookTitle> are all equally (and totally) meaningless. Yes, meaningless.

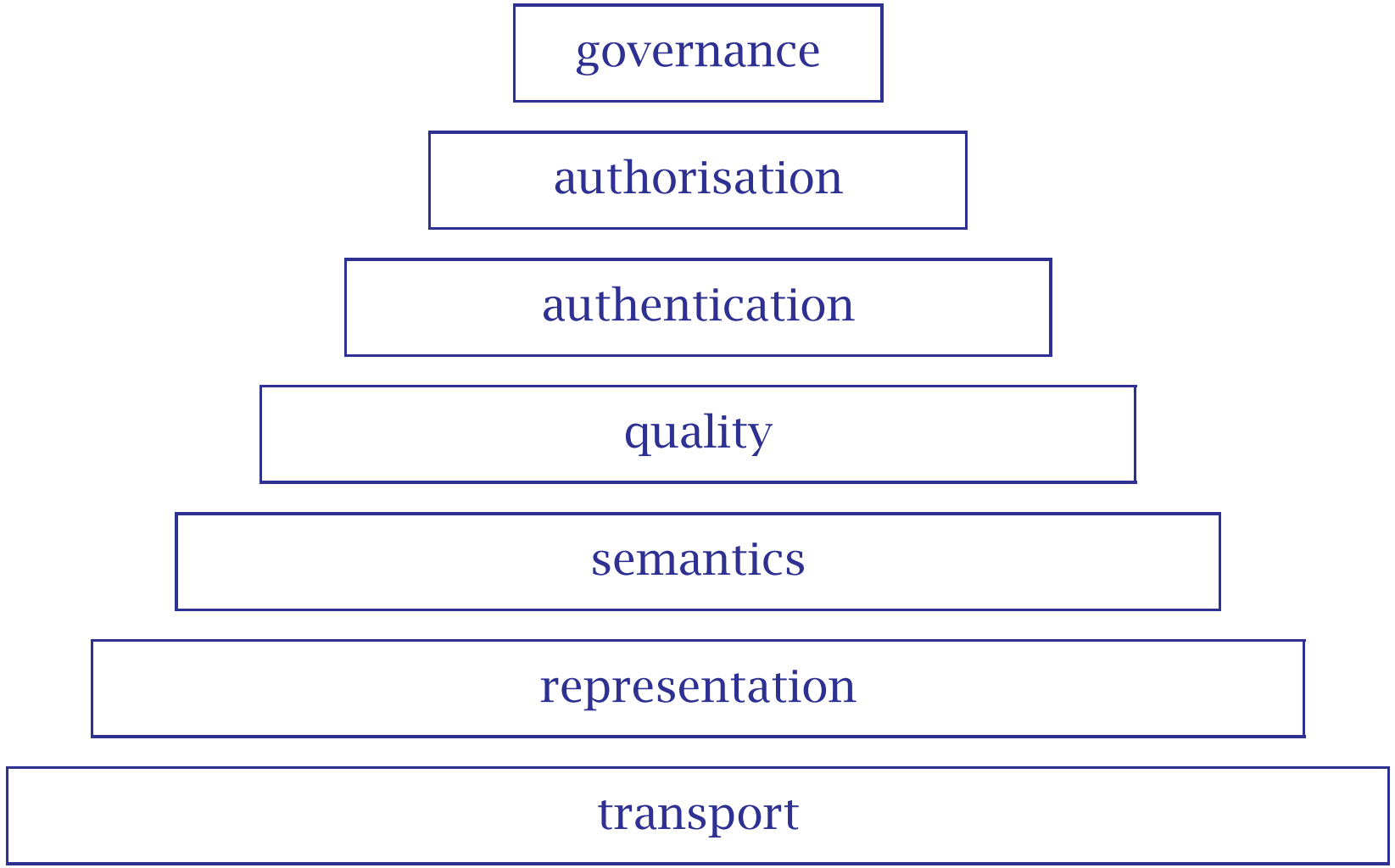
(Robin Cover, *XML and Semantic Transparency*, http://xml.coverpages.org/xmlAndSemantics.html)

1.3 What's needed

Not really science fiction; rather, engineering and technology adoption.

- explicit *metadata* recording meaning
- *ontologies* recording shared understanding
- logical *inference* of knowledge, decisions, explanations
- trusted *agents*

1.4 Interoperability



(Due to Paul Davidson.)

1.5 Knowledge representation

- *vocabulary*

  list of terms, possibly definitions and synonyms

- *taxonomy*

  broader/narrower relationships

- *thesaurus*

  as taxonomy, but additionally ‘see also’ and ‘use instead’

- *ontology*

  arbitrary relationships

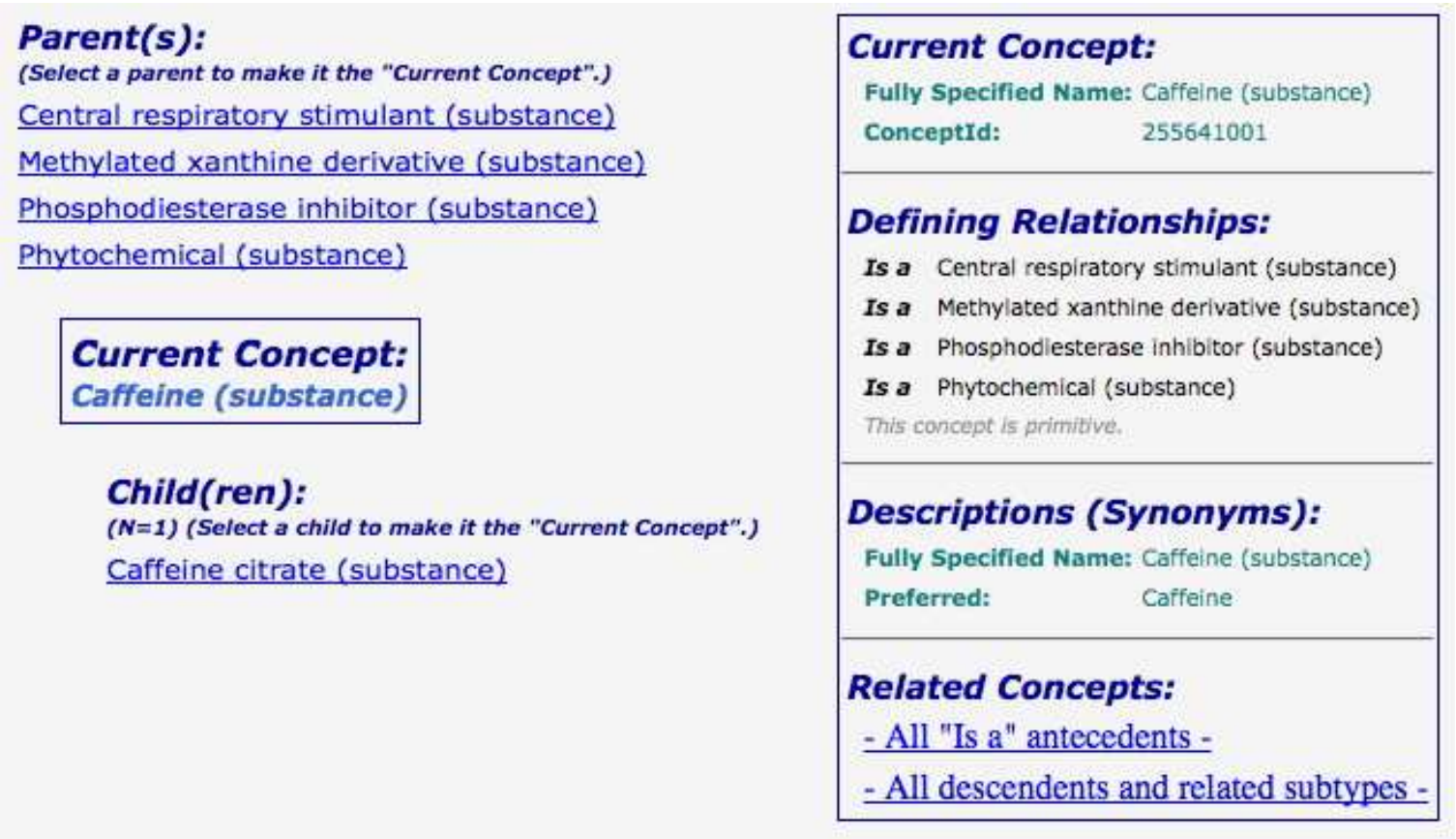
1.6 Vocabulary

US credit rating systems:

- *A. M. Best*: A++, A+, A, A–, B++, B+, B–, C++, C+, C–, D, E, F
- *Moody's*: Aaa, Aa1, Aa2, Aa3, A1, A2, A3, Baa1, Baa2, Baa3, Ba1, Ba2, Ba3, B1, B2, B3, Caa, Ca, C
- *Std & Poors*: AAA, AA+, AA, AA–, A+, A, A–, BBB+, BBB, BBB–, BB+, BB, BB-, B+, B, B–, CCC, R
- *Weiss*: A+, A, A–, B+, B, B–, C+, C, C–, D+, D, D–, E+, E, E–, F

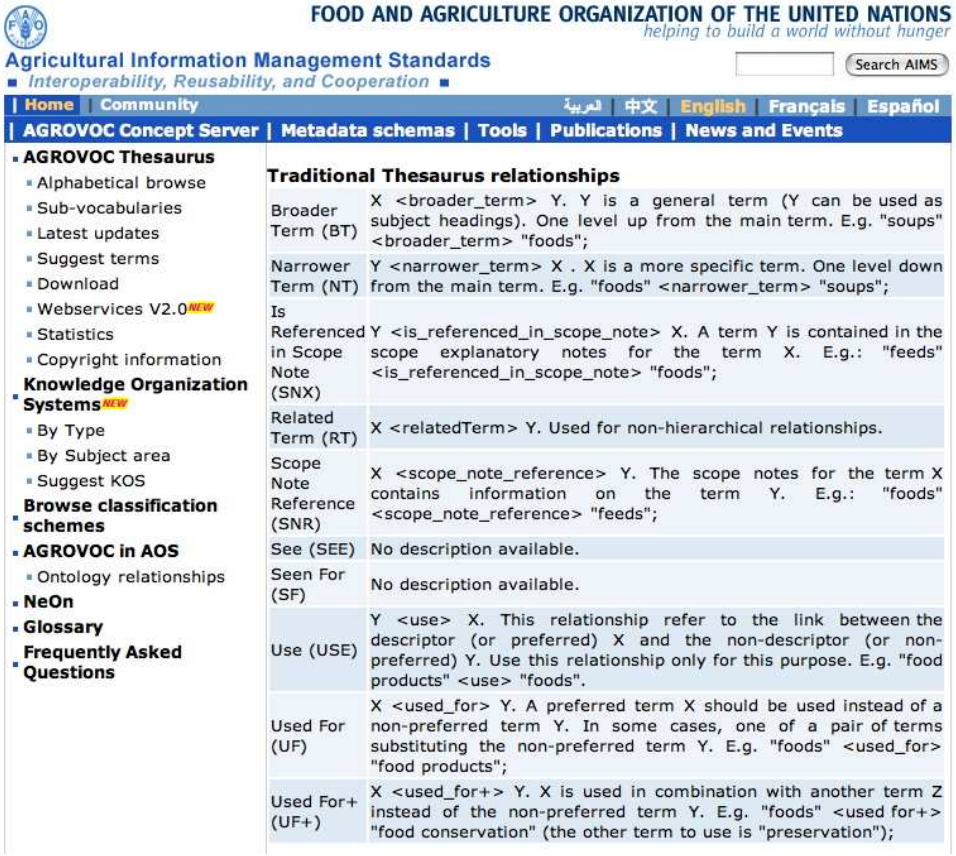
No direct mappings between scales. (Are they even monotonic?)

1.7 Taxonomy



SNOMED CT (Systemized Nomenclature of Medicine – Clinical Terms)
(http://terminology.vetmed.vt.edu/SCT/menu.cfm)

1.8 Thesaurus



FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS
helping to build a world without hunger

Agricultural Information Management Standards
Interoperability, Reusability, and Cooperation

Search AIMS

Home | Community | العربية | 中文 | English | Français | Español

AGROVOC Concept Server | Metadata schemas | Tools | Publications | News and Events

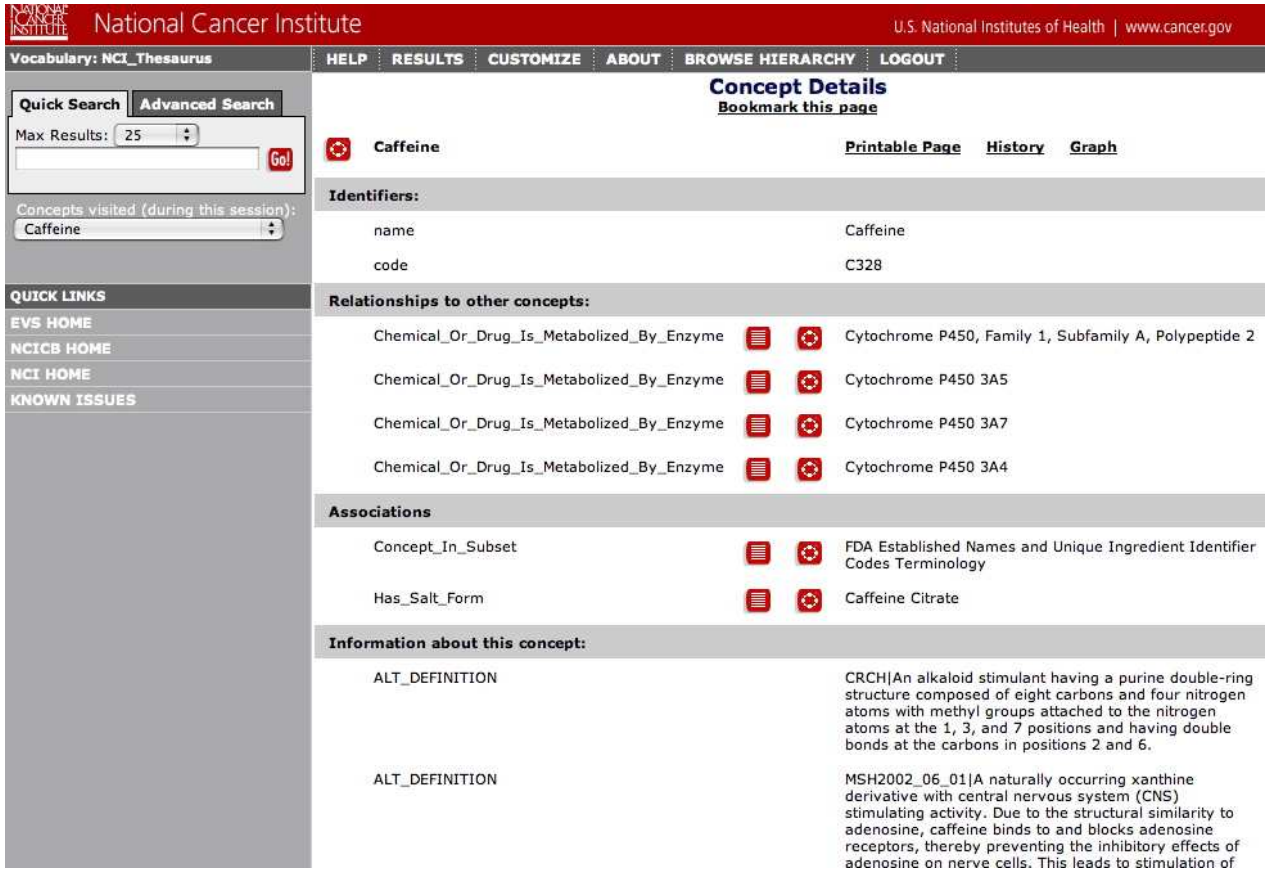
- **AGROVOC Thesaurus**
  - Alphabetical browse
  - Sub-vocabularies
  - Latest updates
  - Suggest terms
  - Download
  - Webservices V2.0 NEW
  - Statistics
  - Copyright information
- **Knowledge Organization Systems** NEW
  - By Type
  - By Subject area
  - Suggest KOS
- **Browse classification schemes**
- **AGROVOC in AOS**
  - Ontology relationships
- **NeOn**
- **Glossary**
- **Frequently Asked Questions**

Traditional Thesaurus relationships

Broader Term (BT)	X <broader_term> Y. Y is a general term (Y can be used as subject headings). One level up from the main term. E.g. "soups" <broader_term> "foods";
Narrower Term (NT)	Y <narrower_term> X . X is a more specific term. One level down from the main term. E.g. "foods" <narrower_term> "soups";
Is Referenced in Scope Note (SNX)	Y <is_referenced_in_scope_note> X. A term Y is contained in the scope explanatory notes for the term X. E.g.: "feeds" <is_referenced_in_scope_note> "foods";
Related Term (RT)	X <relatedTerm> Y. Used for non-hierarchical relationships.
Scope Note Reference (SNR)	X <scope_note_reference> Y. The scope notes for the term X contains information on the term Y. E.g.: "foods" <scope_note_reference> "feeds";
See (SEE)	No description available.
Seen For (SF)	No description available.
Use (USE)	Y <use> X. This relationship refer to the link between the descriptor (or preferred) X and the non-descriptor (or non-preferred) Y. Use this relationship only for this purpose. E.g. "food products" <use> "foods".
Used For (UF)	X <used_for> Y. A preferred term X should be used instead of a non-preferred term Y. In some cases, one of a pair of terms substituting the non-preferred term Y. E.g. "foods" <used_for> "food products";
Used For+ (UF+)	X <used_for+> Y. X is used in combination with another term Z instead of the non-preferred term Y. E.g. "foods" <used for+> "food conservation" (the other term to use is "preservation");

Agrovoc Thesaurus
(http://www.fao.org/aims/cs_relationships.htm)

1.9 Ontology



US National Cancer Institute Thesaurus
(http://nciterms.nci.nih.gov/)

1.10 Isn't this strong AI all over again?

The concept of machine-understandable documents does not imply some magical artificial intelligence which allows machines to comprehend human mumblings. It only indicates a machine's ability to solve a well-defined problem by performing well-defined operations on existing well-defined data. Instead of asking machines to understand people's language, it involves asking people to make the extra effort.

(Tim Berners-Lee, *What the Semantic Web Can Represent*, http://www.w3.org/DesignIssues/RDFnot.html)

2 Resource Description Framework (RDF)

- relations between concepts
- *triples* as statements
  - *resource*, the subject

    a URI; unambiguous (homonyms resolved)
  - *property*, the verb

    a binary predicate, also a named resource
  - *value*, the object

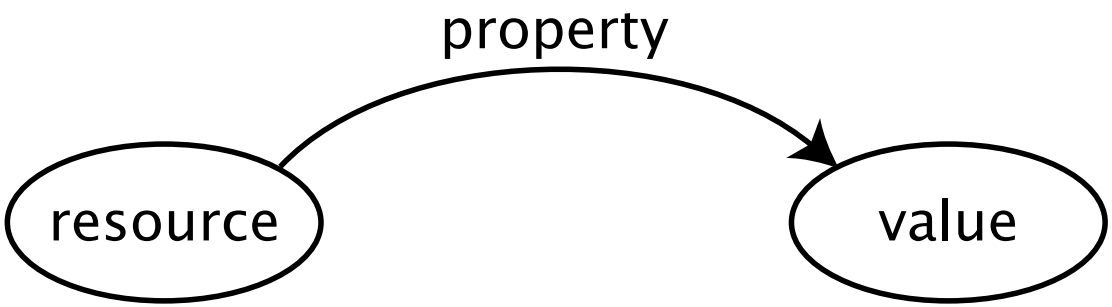
    either a literal or another resource
- *property*(*resource*, *value*)

2.1 Real things

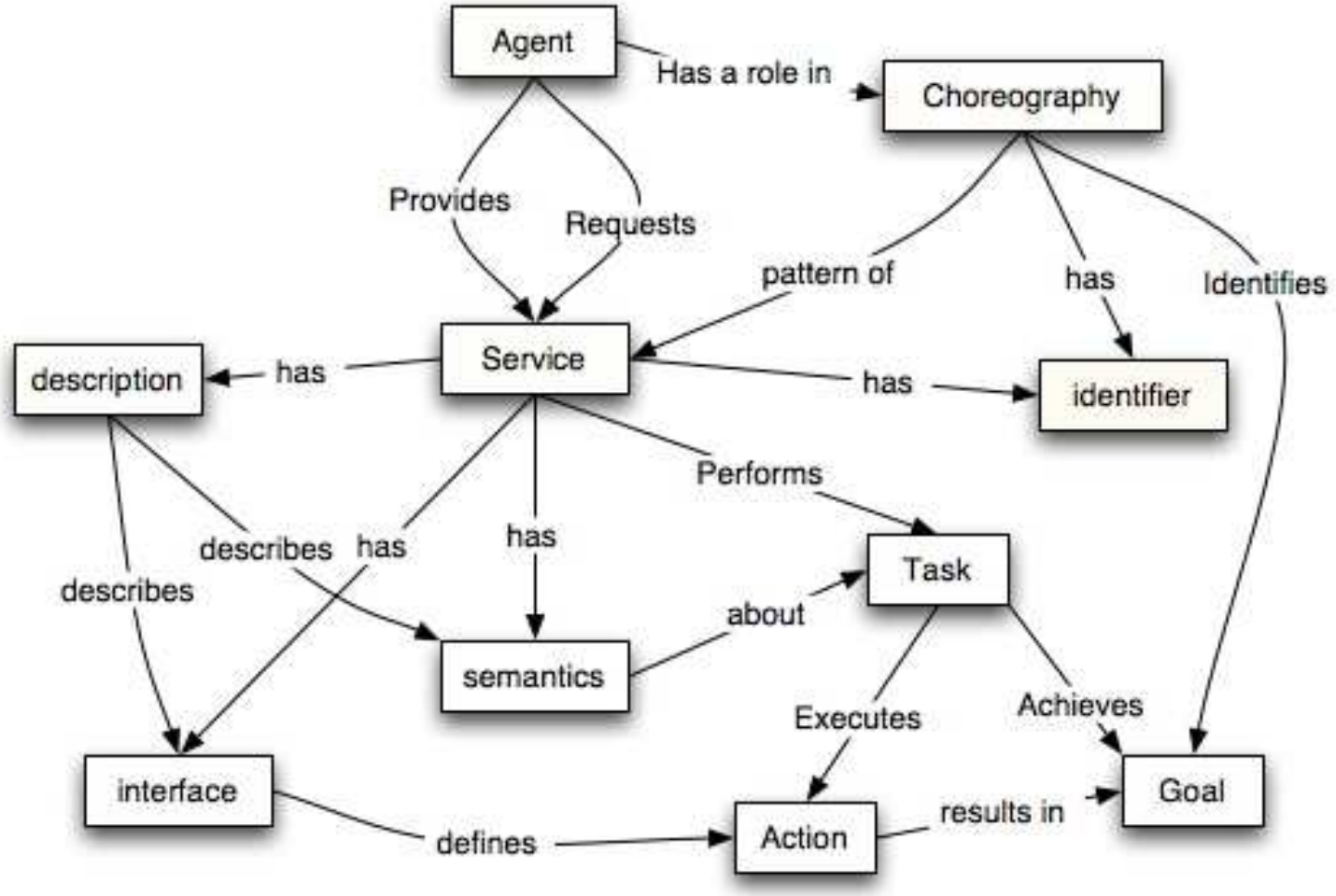
- apply power of URIs to relationships too
- don't just say 'colour', say `urn:pantone://solid.coated/`
- model real things, not just documents
  - XML: car registration document contains single license number field
  - RDF: car has unique license number
- data plus metadata

2.2 Graphical representation

- directed graph
- subjects and objects as vertices
- properties as edges

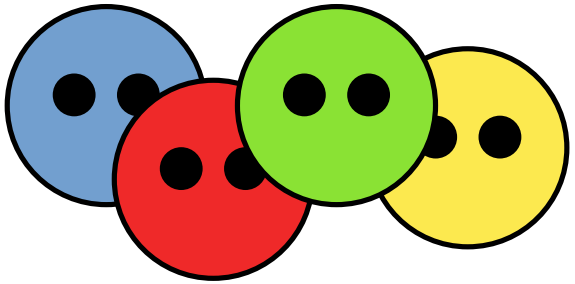


2.3 Eg Service-oriented WS Architecture Model



2.4 Friend-of-a-friend (FOAF)

- ontology of people, activities, relationships
- social networks without lock-in: *social semantic web*
- based on RDF (and OWL)
- eg find the people you and your friend both know; recent publications by your co-authors
- `http://www.foaf-project.org/`



2.5 Why is plain XML not enough?

- even if you know what entities mean, relationships are not clear
- consider

```
<subject name="SOA"> <lecturer>Jeremy</lecturer> </subject>

<lecturer name="Jeremy"> <subject>SOA</subject> </lecturer>

<lecturing>
  <lecturer>Jeremy</lecturer>
  <subject>SOA</subject>
</lecturing>
```

- each perfectly good, but combination is incompatible
- no standard representation, so no standard queries

2.6 XML is enough!

An XML Schema for RDF triples.

```
<?xml version="1.0" encoding="UTF-16"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:sep="http://www.softeng.ox.ac.uk/rdf-ns">
  <rdf:Description
    rdf:about="http://www.softeng.ox.ac.uk/subjects/SOA.html">
    <sep:lecturer
      rdf:resource="http://web.comlab.ox.ac.uk/Jeremy.Gibbons/"/>
  </rdf:Description>
</rdf>
```

2.7 RDF in XML

- namespaces for definition, not just for disambiguation
- `rdf:about` attribute is typically a predefined subject
- subject may be *defined* here:

```
<rdf:Description rdf:ID="#SOA">
  <sep:title>Service-Oriented Architecture</sep:title>
</rdf:Description>
```

- `<rdf:Description>` entity may contain multiple *property elements*, read conjunctively

```
<rdf:Description
  rdf:about="http://www.softeng.ox.ac.uk/subjects/SOA.html">
  <sep:lecturer
    rdf:resource="http://web.comlab.ox.ac.uk/Jeremy.Gibbons/"/>
  <sep:title>Service-Oriented Architecture</sep:title>
</rdf:Description>
```

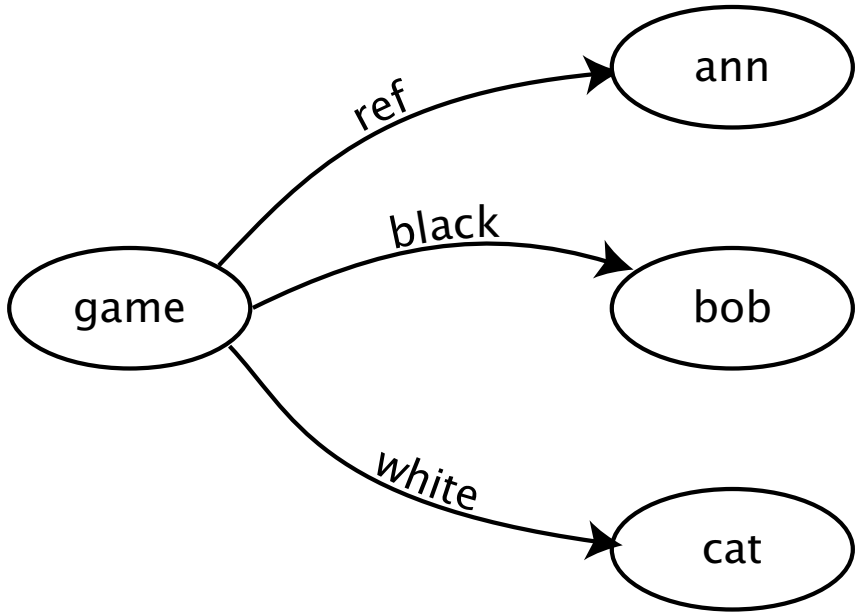
- literal values may be typed

```
<rdf:Description
  rdf:about="http://www.softeng.ox.ac.uk/subjects/SOA.html">
  <sep:cats rdf:datatype="&xsd;integer">15</sep:cats>
</rdf:Description>
```

- other niceties too (more types, abbreviations, collections)

2.8 Non-binary predicates

- RDF triples only support binary predicate
- sometimes want ternary or higher-arity
- eg *referees*(*ann*, *bob*, *cat*): "Ann referees game between Bob and Cat"
- represent with three binary relationships
- introduce auxilliary resource *game*
- now use *referee*(*game*, *ann*), *white*(*game*, *bob*), *black*(*game*, *cat*)



2.9 Reification

- statements about other statements — *meta-statements*
- eg provenance, trust, belief

```
<rdf:Statement rdf:ID="#SOA-lecturer">
  <rdf:subject
    rdf:resource="http://www.softeng.ox.ac.uk/subjects/SOA.html"/>
  <rdf:predicate rdf:resource="&sep;lecturer"/>
  <rdf:object
    rdf:resource="http://web.comlab.ox.ac.uk/Jeremy.Gibbons/"/>
</rdf:Statement>

<rdf:Description rdf:about="#SOA-lecturer">
  <sep:authorized>November 2007</sep:authorized>
</rdf:Description>
```

2.10 RDF Schema (RDFS)

- a type system for RDF
- restrictions on validity
- classification hierarchies

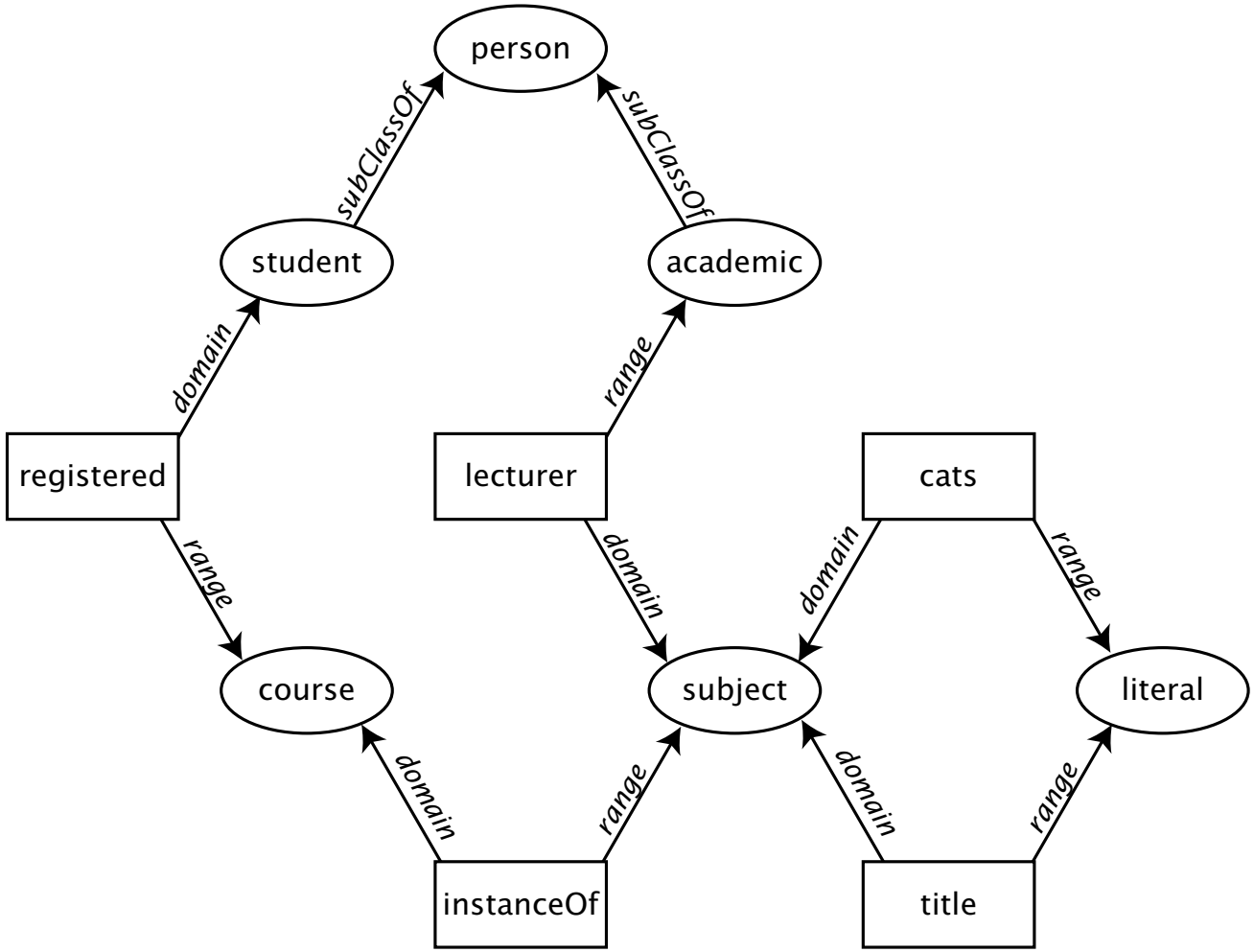
- RDF Schema model is itself about binary relationships
- so is presented in RDF
- (and so there's an RDF Schema for RDF and for RDF Schema...)

2.11 RDFS core classes and properties

- rdfs:Resource
- rdfs:Class
- rdfs:Literal
- rdfs:Property
- rdfs:Statement

- rdfs:subClassOf
- rdfs:subPropertyOf
- rdfs:domain
- rdfs:range

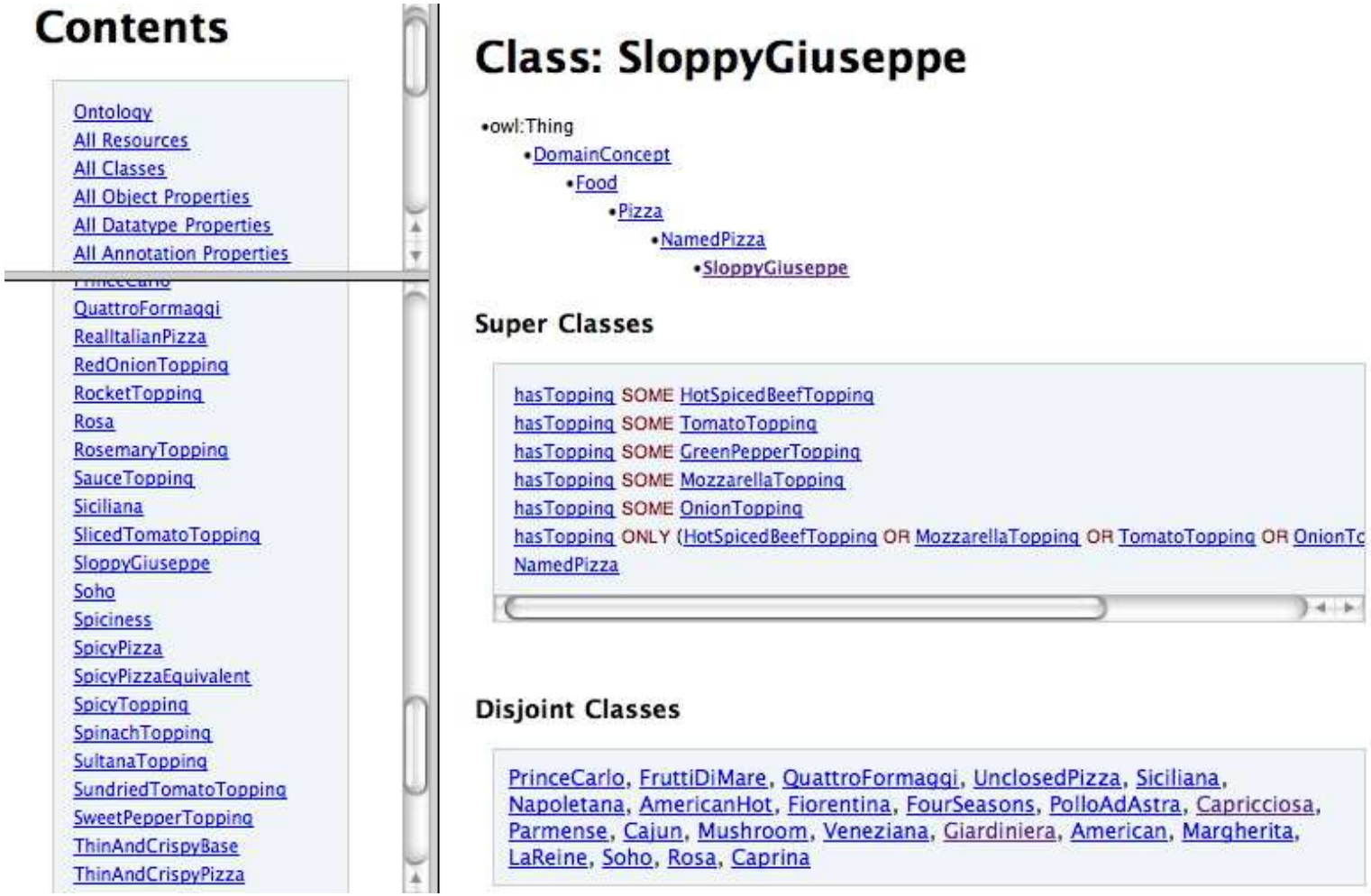
2.12 RDFS example



3 Web Ontology Language (OWL)

- "an ontology is a formal, explicit specification of a shared conceptualization" (Gruber, 1993)
- in particular, providing *support for reasoning* about that conceptualization
  - consistency-checking
  - completion
  - validation and verification
  - inference
- application of logic-programming/theorem-proving technology

3.1 Pizza



http://www.co-ode.org/ontologies/pizza/

3.2 Why isn't RDF enough?

- some reasoning possible with RDF documents: typed classification hierarchy
- but much other expressivity missing
  - locally scoped properties

    animals eat food, but herbivores eat only plants
  - boolean combination of classes

    intersection, union, complement, disjointness
  - cardinality restriction

    person has two parents; course has exactly one lecturer
  - characteristics of properties

    transitivity, uniqueness, converse

3.3 The birth of OWL

- DARPA Agent Markup Language Ontology (DAML-ONT) (`http://www.daml.org/2000/10/daml-ont.html`)
- Ontology Inference Layer (OIL) (`http://www.ontoknowledge.org/oil/`)
- both based on RDF and RDFS, and ideas from *description logic*
- joint initiative DAML+OIL
- the basis for W3C OWL

3.4 Three species of OWL

- OWL Full

  fully upwards compatible with RDF; but undecidable

- OWL DL

  drop metamodelling to allow efficient reasoning support; based on description logic, a decidable fragment of first-order predicate logic

- OWL Lite

  excludes enumerations, disjointness, arbitrary cardinality; easier to learn and to implement

3.5 Conservative extension

- every legal OWL Lite ontology is legal OWL DL
- every legal OWL DL ontology is legal OWL Full
- every legal RDF document is also legal OWL Full

- every valid OWL Lite conclusion is valid with OWL DL
- every valid OWL DL conclusion is valid with OWL Full
- every valid RDF/RDFS conclusion is valid with OWL Full

3.6 OWL example

- taken from OWL Guide
  http://www.w3.org/TR/2003/PR-owl-guide-20031215/wine.rdf
- three root classes:

```
<owl:Class rdf:ID="Winery"/>
<owl:Class rdf:ID="Region"/>
<owl:Class rdf:ID="ConsumableThing"/>
```

Wine is consumable

```
<owl:Class rdf:ID="PotableLiquid">
  <rdfs:subClassOf rdf:resource="#ConsumableThing" />
  ...
</owl:Class>

<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid"/>
  <rdfs:label xml:lang="en">wine</rdfs:label>
  <rdfs:label xml:lang="fr">vin</rdfs:label>
  ...
</owl:Class>
```

Things and types

```
<Region rdf:ID="MarlboroughRegion" />
```

or equivalently

```
<owl:Thing rdf:ID="MarlboroughRegion" />
<owl:Thing rdf:about="#MarlboroughRegion">
   <rdf:type rdf:resource="#Region"/>
</owl:Thing>
```

Subclasses and instances

```
<owl:Class rdf:ID="WineGrape">
  <rdfs:subClassOf rdf:resource="&food;Grape" />
</owl:Class>

<WineGrape rdf:ID="SauvignonBlancGrape" />
```

Properties

```
<owl:ObjectProperty rdf:ID="madeFromGrape">
  <rdfs:domain rdf:resource="#Wine"/>
  <rdfs:range rdf:resource="#WineGrape"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="course">
  <rdfs:domain rdf:resource="#Meal" />
  <rdfs:range rdf:resource="#MealCourse" />
</owl:ObjectProperty>
```

Restriction

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid" />
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMaker" />
      <owl:allValuesFrom rdf:resource="#Winery" />
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

Maker of a wine must be a winery. Analogous `owl:someValuesFrom` would say at least one of the `hasMaker` relations must point to a winery.

Inference

Given

```
<owl:Thing rdf:ID="CloudyBaySB">
  <madeFromGrape rdf:resource="#SauvignonBlancGrape" />
</owl:Thing>
```

we can infer that `CloudyBaySB` is a wine because the domain of `madeFromGrape` is `Wine`.

3.7 Semantic web services

- services should be semantically described
- service interfaces published in RDF
- ‘smart’ directory searches using OWL inference
- supporting automatic service discovery, invocation, composition

3.8 OWL for Services (OWL-S)

- three aspects of service description
- *service profile*

  yellow pages entry: inputs, outputs, preconditions, side-effects (IOPEs)

- *service model*

  computer-interpretable description of service behaviour (process)

- *service grounding*

  message format, transport mechanisms, serializations

- `http://www.w3.org/Submission/OWL-S/`

4 ‘Semantic frameworks’

- *metadata-based, model-driven* development
- *terminology services*

  collections of defined terms, and relationships
- *metadata registries*

  observations, measurements, properties: terms, intended purposes, possible values
- *model repositories*

  database schemas, service descriptions, forms, queries, reports; entries linked to metadata elements

4.1 CancerGrid

- semantic frameworks for cancer clinical trials
- partners at Oxford, Cambridge, London, Birmingham, Belfast
- funded by UK Medical Research Council, subsequently by EPSRC and Microsoft
- much more widely applicable than cancer clinical trials!

4.2 Electronic government

- *joined-up government* requires large-scale system integration and data sharing
- depends on computable representations of semantics of data
- implicit or informal representations suffice for straightforward problems, small homogeneous communites, and short time-scales
- but electronic government is characterised by complex problems, heterogeneous communities, and long-running initiatives
- a more formal approach to data semantics is required

Index

Contents

Service-Oriented Architecture

Monday	Tuesday	Wednesday	Thursday	Friday
Introduction Components	REST	Composition	Architecture	Engineering
coffee	coffee	coffee	coffee	coffee
Components	REST	Composition	Architecture	Conclusion
lunch	lunch	lunch	lunch	lunch
Web Services	Qualities	Objects	Semantic Web	
tea	tea	tea	tea	
Web Services	Qualities	Objects	Semantic Web	