Basics

Semantic Modelling
IT Architectures and Information modelling

• IT Architecture
  • Helps manage complexity in modern software systems
  • Supports agile processes
  • Supports reuse of assets
  • Reduces costs
  • Helps focus on core assets

• Decomposition
  • Functions decomposed into services
  • Business processes
  • Information provided at abstract level
Information modelling

• Purpose of information modelling is to:
  • Create a representation of real-world concepts and meaningful relationships between them
  • Provide a high level understanding of data by abstracting it further away from physical aspect of data storage
  • Represent a user’s perspective of the data

• Choosing a modelling language is trade-off between
  • Formality/Informality: is the meaning of the modelling language same regardless of audience?
  • Commonality and variability: how to manage things that are common and things that are variable?
  • Expressivity: how detailed in the model?
How expressive should a model be?

FIGURE 2.4
Different expressivity of models of a water molecule.
Moving up the Knowledge Continuum

Data

Facts
What (signals, symbols, measurements)

+ Structure =

Information

Descriptions
Who, Where, When (statements, structure, state, content)

+ Context =

Knowledge

Meaning
How (routines, processes, truth, beliefs)

Ontology

Wisdom

Understanding
Why (ideas, judgments, decisions, rationalization)

Semantic Query

?
Data modelling languages

• Many different types of data models
  • Entity-relationship (ER) model: traditional modelling technique associated with software analysis and design
  • Relational model: suitable for relational databases and SQL queries. Can be seen as a restricted ER-model
  • Object model: suitable for OO analysis and design.
  • Semantic data model
Semantic data model

- Semantic data modelling is a method for representing data enriched with semantic information in the form of data values and relationships.
- A semantic data model is more complex and expressive than the traditional data models.
- Besides representing intensional structures like the traditional models, a SDM can also express implicit or derived knowledge from the explicit information, which is not possible with the traditional models.
Why semantic models?

- **Reuse and interoperability**: Semantic models can be shared among applications and on the web.
- **Flexibility**: Semantic models can operate in an open environment in which classes can be defined dynamically.
- **Consistency and Quality Checking** across models.
- **Reasoning**: Semantic models are supported by automated reasoning tools.
- Semantic model creates *ontologies*.
- Semantic models are supported by a number of *technologies*. 
Subject–Predicate–Object expressions

• Semantics are expressed as triples of Subject-Predicate-Object
• RDF (Resource Description Framework) defines statements in this format
## Comparison with relational models

<table>
<thead>
<tr>
<th>Traditional data modelling (Relational databases):</th>
<th>Semantic data modelling:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concerned with structure of data</td>
<td>Concerned with the meaning of data (relationships)</td>
</tr>
<tr>
<td>Inflexible (any changes would require changing entire tables and queries)</td>
<td>Very flexible (new data can be added without affecting the existing data and queries)</td>
</tr>
<tr>
<td>Well suited for large but simple data</td>
<td>Well suited for fewer but complex data</td>
</tr>
<tr>
<td>No automated inferencing possible</td>
<td>Automated inferencing possible using semantic reasoners</td>
</tr>
<tr>
<td>Record-oriented modelling</td>
<td>User-oriented modelling (user’s view of the real-world concepts)</td>
</tr>
</tbody>
</table>
## Relational Database tables

### Person table

<table>
<thead>
<tr>
<th>ID</th>
<th>First Name</th>
<th>Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Sandra</td>
<td>Ferreira</td>
</tr>
<tr>
<td>P2</td>
<td>Steve</td>
<td>Barrett</td>
</tr>
<tr>
<td>P3</td>
<td>Mia</td>
<td>Shaw</td>
</tr>
</tbody>
</table>

### Country table

<table>
<thead>
<tr>
<th>ID</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Australia</td>
</tr>
<tr>
<td>C2</td>
<td>France</td>
</tr>
<tr>
<td>C3</td>
<td>Greece</td>
</tr>
</tbody>
</table>

### Capital city table

<table>
<thead>
<tr>
<th>ID</th>
<th>Capital City</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC1</td>
<td>Athens</td>
</tr>
<tr>
<td>CC2</td>
<td>Canberra</td>
</tr>
<tr>
<td>CC3</td>
<td>Paris</td>
</tr>
</tbody>
</table>

### isBornIn table

<table>
<thead>
<tr>
<th>Person ID</th>
<th>Country ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>C3</td>
</tr>
<tr>
<td>P2</td>
<td>C1</td>
</tr>
<tr>
<td>P3</td>
<td>C2</td>
</tr>
</tbody>
</table>

### hasCapital table

<table>
<thead>
<tr>
<th>Country ID</th>
<th>Capital City ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>CC2</td>
</tr>
<tr>
<td>C2</td>
<td>CC3</td>
</tr>
<tr>
<td>C3</td>
<td>CC1</td>
</tr>
</tbody>
</table>
Example of Triples

- **Obj.**
  - Greece
  - Australia
  - France

- **Subj.**
  - “Sandra Ferreira” isBornIn Greece
  - “Steve Barrett” isBornIn Australia
  - “Mia Shaw” isBornIn France

- **Obj.**
  - Canberra
  - Paris
  - Athens

- **Subj.**
  - Australia hasCapital Canberra
  - France hasCapital Paris
  - Greece hasCapital Athens

Relationships:
- isBornIn
- hasCapital
Linked Triples

isBornIn

“Sandra Ferreira”

Greece

hasCapital

Athens

isBornIn

“Steve Barrett”

Australia

hasCapital

Canberra

isBornIn

“Mia Shaw”

France

hasCapital

Paris
Triples representation

- A triple is a (3 tuple) an abstract representation in the form of <subject> <object> <predicate>
- The format of such representation is called RDF (Resource Description Framework)
- Triples can be encoded using text (e.g. XML, Turtle) and exchanged between different parties
- Triples from different files can be easily merged together
Technology stack overview

- Semantic Web provides a number of modelling languages that differ in their level of expressivity
  - RDF—The Resource Description Framework: provides a mechanism for allowing anyone to make a basic statement about anything and layering these statements into a single model.
  - RDFS—The RDF Schema language: is a language with the expressivity to describe the basic notions of commonality and variability familiar from object languages and other class systems—namely classes, subclasses, and properties.
  - OWL – Ontology Web Language: brings the expressivity of logic to the Semantic Web. It allows modelers to express detailed constraints between classes, entities, and properties
  - SPARQL is the query language associated with Semantic Web
The Semantic Web Technology Stack (not a piece of cake...)

- Most apps use only a subset of the stack
- Querying allows fine-grained data access
- Standardized information exchange is key
- Formats are necessary, but not too important
- The Semantic Web is based on the Web
- Linked Data uses a small selection of technologies

• There are a number of editors that assist users in defining semantic models

• Protégé
  • A free, open-source ontology editor and framework for building semantic models
  • http://protege.stanford.edu/

• Top Braid Composer
  • comes in multiple editions with free trials

• Jalapeno
  • part of the Capsicum Methodology for assisting enterprises in conducting business analysis
Conclusion

• Semantic data modelling is an approach for representing data models using richer types of relationships.

• At its most basic level, a semantic data model represents information in the form of triples.

• More complex representations can be built on top of this representation.

• There are a number of technologies and standards that support semantic models: RDF, RDFS, OWL, SPARQL.

• Semantic models underpin the Semantic Web.

• There are a number of editors that assist users in defining semantic models: Protégé, Top Braid, Jalapeno.