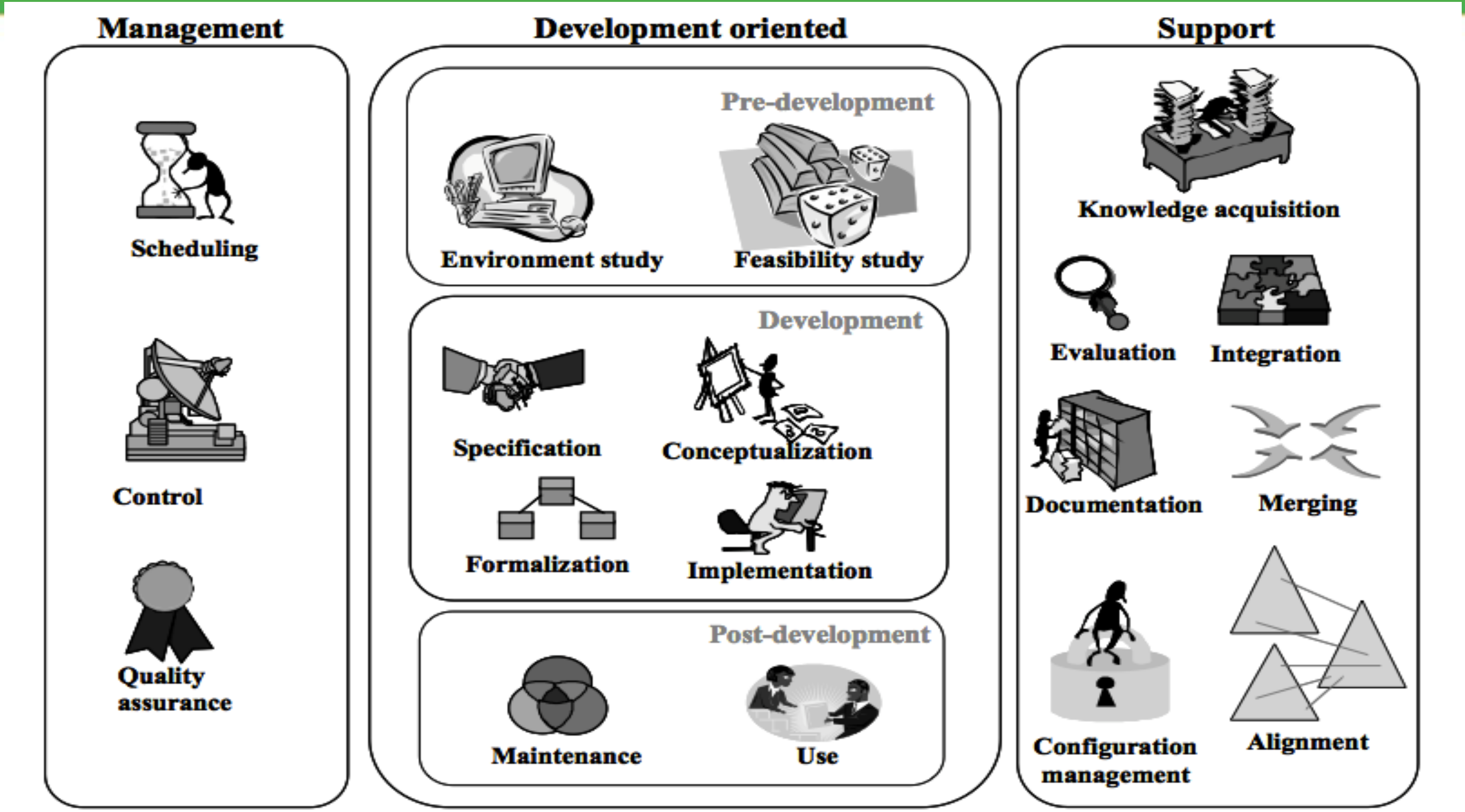
UNSW
SYDNEY

Building an Ontology

Sponsored by:

CAPSICUM
Business Architects

Activities in an Ontology Development Process[1]



Development Oriented Activities

- **Specification** - Identifying the specifics of the ontology such as the purpose of the ontology, the requirements it must meet, and who will be the end users of the ontology. Competency questions and requirement representations can be used to identify the purpose and requirements of the ontology.
- **Conceptualization** - Knowledge from the target domain is captured in the form of concepts and structured as a model that can be manipulated by the domain experts as required.
- **Formalization** – Concepts are formalized into a semi-computable model
- **Implementation** - model is represented in a formal ontology representation language

Principles of ontology development

- Know what you want
  - Listing the requirements
    Anticipate future requirements
- Say what you mean, mean what you say
  - The model must capture the intentions
  - Model should be able to answer direct questions
  - Model should be able to “guess” answers through inferencing

Method to Capture Requirements[2]

1. Identifying the **motivating scenarios** that represent where the ontology will be used and what it is expected to answer.
2. Motivating scenarios are transformed into **Competency Questions** (CQs), which are expressed informally in natural language. CQs should be developed in a stratified manner wherein simple questions can be composed to answer the more complex ones whereas complex questions can be decomposed to answer simple ones.
3. The CQs are used to determine the scope and to extract the ontology concepts, relationships, and axioms that represent the knowledge to be modelled in the ontology.
4. The terminology is formalized using first-order logic wherein the concepts are defined or expressed using axioms and constraints are applied to the axioms such that the competency questions can be answered.

Identifying the motivating scenarios

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?

Example Use Cases and Respective CQs

Use Case Number	Generic use case	Identified Association type	Generic competency questions
U1	The presence of some clinical condition by itself or when in association with one or more other comorbid conditions is diagnostic for some third clinical condition.	Diagnostic	• Which clinical conditions are diagnostic of some condition x? • Which conditions when present together are considered as diagnostic for some condition x?
U2	Some clinical condition or clinical procedure is a complicating factor for another leading to the development of some new condition associated with the latter.	Complicating	• Which factors can cause complications in some condition x? • What complications does a clinical condition cause in another? • What complications does some clinical procedure cause in a condition x?

Use Case Number	Domain-specific use cases (medical and oral health)	Competency questions
U1	Some systemic condition and oral condition together are diagnostic of another systemic condition.	• Which systemic and oral conditions are together considered diagnostic for some systemic condition x? • What are the oral manifestations of some systemic condition x?
U2	Some oral condition complicates a systemic condition and causes the development of another systemic condition.	• How does some oral condition x affect some systemic condition y?

Using class hierarchy to infer information

- Answered using RDF:
  - Who is Ziva’s professor? (answer: Joe)
- Answered using RDFS:
  - Who is Joe? (answer: Professor, specifically an associate professor)
  - Who is Ziva? (Student, specifically a post graduate student)

Who is Joe? Who is Ziva?

- Given information:

```
ex:Joe rdf:type ex:AssociateProfessor .
ex:AssociateProfessor rdfs:subClassOf
ex:Professor .
ex:Ziva rdf:type ex:PostgraduateStudent .
ex:PostgraduateStudent rdfs:subClassOf
ex:Student .
```

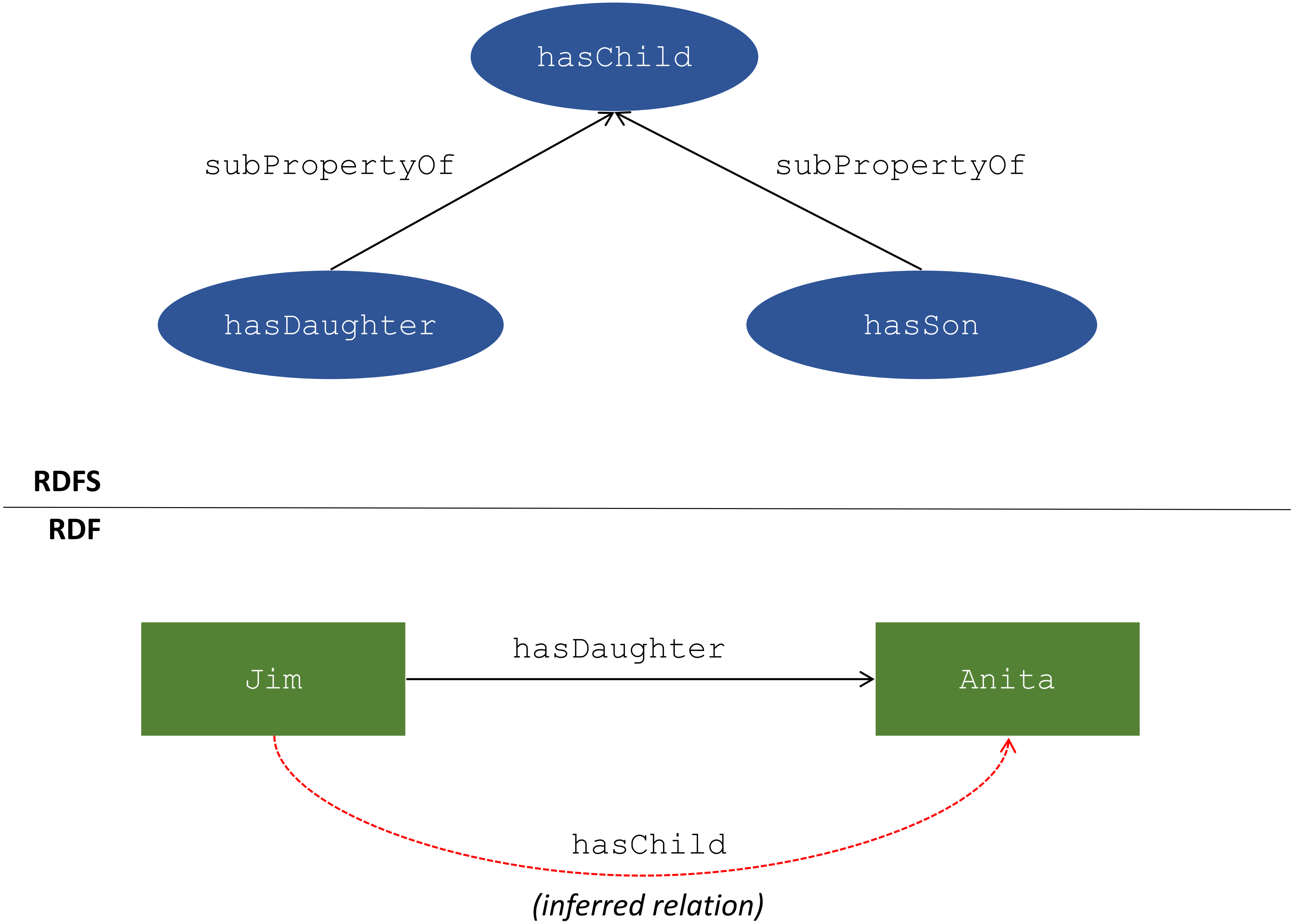
- Inferred information:

```
ex:Joe rdf:type ex:Professor .
ex:Ziva rdf:type ex:Student .
```

The subClassOf relation is transitive in nature

Using property hierachies

- Representing information using RDF and RDFS:
  - RDF:
    - Jim has a daughter named Anita
  - RDFS:
    - The property `hasChild` is further classified into subproperties namely `hasDaughter` and `hasSon`



hasChild

subPropertyOf

subPropertyOf

hasDaughter

hasSon

RDFS

RDF

Jim

hasDaughter

Anita

hasChild

(inferred relation)

CQs

- Answered using RDF:
  - Who is Jim’s daughter? (answer: Anita)
  - Does Jim have daughters? (answer: Yes (Boolean))
- Answered using RDFS:
  - Who is Jim’s child? (answer: Anita)
  - Does Jim have children? (answer: Yes (Boolean))

Who is Jim's child? Does Jim have children?

- Given information:

  ex:Jim ex:hasDaughter ex:Anita

  ex:hasDaughter rdfs:subPropertyOf ex:hasChild

- Inferred information:

  ex:Jim ex:hasChild ex:Anita

The subPropertyOf relation is transitive in nature

- Using RDF alone can cause loss of information or incomplete query answering.
- As seen in this example, the person querying will have to specifically know that Jim has a daughter in order to obtain the right answer.
- This is often not the case in real-life situations where the knowledge of the person questioning is more likely to be incomplete.

Modelling for reuse

- Important of insightful names
  - Use meaningful names for resources
  - Use annotations such as rdfs:label, rdfs:comment, rdfs:seeAlso for extra information
- CamelCase is the name given to the style of naming
  - Start class names with capital letters e.g. owl:Restriction and owl:Class.
  - Start property names with lowercase letters e.g. rdfs:subClassOf and owl:inverseOf
- Keeping track of classes and individuals
  - Resolving ambiguities in every day's language e.g. a poem can be a class or an individual

Reusing Non-ontological Resources for Ontology Building [3]

- Following resources can be used at ontology design phase
  - Glossaries
  - Thesauri
  - Lexicons
  - Classification Schemes
  - Taxonomies
  - O**ntology Design Patterns (ODP)** – Modelling solutions for solving recurrent ontology design problems [5]**,** best practices and experiences of ontology modelling along with knowledge about good solutions

Common Modelling Errors

- What is an error ?
  - Hard to define in a context where there is the AAA slogan (Anybody can say Anything about Any topic)
  - This is when we can say that they do not accomplish the desired goals of sharing information about a structured domain with other stakeholders
- Anti-patterns [8]
  - Common pitfalls of beginning modellers
  - Rampant classism AP: model everything as a class
  - Exclusivity AP: getting membership between classes and subclasses wrong
  - Objectification AP: assuming semantic model same as object model

Ontology Design Patterns (ODP) Reuse

- ODP search – Possible ODPs are searched for in the repositories based on the modeling problem(s) at hand, using generic and domain specific use cases
- ODP selection – The most relevant ODP or ODPs are identified and matched against the actual use case behind the modelling task
- ODP adaptation – The selected ODP may be the 'best-fit' but may not be applicable as-is for the modelling problem. Hence, the ODP must be customized by way of instantiation or extension or composition of more than one ODP
- ODP integration – The ODP is finally integrated into the ontology model

- Online ODP catalogues with example patterns and related CQs:
  - http://www.gong.manchester.ac.uk/odp/html/
  - http://ontologydesignpatterns.org/wiki/Main_Page

Benefits of using ODPs

1. Reduction in design mistakes
2. Increased knowledge about design practices
3. Identifying implicit requirements
4. Improved quality of the resulting ontology.
5. Offer ways to increase the expressivity of the language by providing alternative ways of modelling the same situation

References

- [1] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, “Methodologies and methods for building ontologies,” Chapter 3, *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, pp. 107-197, 2004.
- [2] M. Uschold, and M. King, “Towards a methodology for building ontologies,” in Workshop on Basic Ontological Issues in Knowledge Sharing held in conjunction with IJCAI-95, 1995.
- [3] M. C. Suárez-Figueroa, “NeOn Methodology for building ontology networks: specification, scheduling and reuse,” Doctoral thesis, Artificial Intelligence, Universidad Politécnica De Madrid, 2010.
- [4] Tejal Manojkumar Shah, Designing and conceptualising ontology patterns for modelling cross-domain health information, Ph.D. Thesis, University of New South Wales, 2016
- [5] V. Presutti, E. Daga, A. Gangemi et al., “eXtreme design with content ontology design patterns,” in Proceedings of the 2009 International Conference on Ontology Patterns-Volume 516, 2009, pp. 83-97.
- [6] E. Blomqvist, “Semi-automatic ontology construction based on patterns,” Dissertation, Department of Computer and Information Science, Linköping University Institute of Technology, 2009.
- [7] Natalya F. Noy, Deborah L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology
- [8] Allemang, Dean, and James Hendler. ”Semantic web for the working ontologist: effective modeling in RDFS and OWL”. Elsevier, (2011).