Methodologies
- Exemplified through XD

Eva Blomqvist
eva.blomqvist@liu.se
Department of Computer and Information Science (IDA)
Linköpings universitet
Sweden

Slides partly by Valentina Presutti, STLab, ISTC-CNR, Italy
Ontology Engineering Methodologies

- Mostly focus has been on overall life-cycle and “model” of the methodology – rather than how to actually perform it
- Few are focused on reuse and the networked nature of web ontologies

- One of the most cited:
  - Ontology development 101 – Noy & McGuinnes (2001)
    - Pre-OWL methodology
    - Traditional in the sense
      - It doesn’t have a specific task focus
      - It is a waterfall like method
    - Although detailed in some steps, no details on requirements or testing etc.
    - Basic steps for modelling
      1. Domain an scope
      2. Consider reuse
      3. Enumerate terms
      4. Develop class hierarchy
      5. Define the properties
      6. Define restrictions and constraints
      7. Create instances
Example: METHONTOLOGY (~1997)

- Waterfall-like process consisting of (overlapping) phases
  1. Specification – document requirements, scope, level of formality etc.
  2. Knowledge Acquisition – gathering and studying sources of information
  3. Conceptualization – structure the terminology identified in 1, going from glossary to logical formulas
  4. Integration – find and select other ontologies to reuse
  5. Implementation – represent in formal language using tool
  6. Evaluation – verification and validation
  7. Documentation

- Based on theories for argumentation
- Intended for
  - Empowering domain experts in ontology engineering
  - Continuous and distributed construction and update
The NeOn Methodology (2006-2010)

- Seven scenarios for ontology engineering
eXtreme Design

“Rapid Prototyping” based on ODPs
Method and tool support

- eXtreme Design (XD)
  - a method for developing ontologies with Content Patterns
- XD tools
  - a tool that supports the XD method
  - released as both an Eclipse plugin and a NeOn Toolkit plugin
  - Can be used with the NeOn toolkit version 2.3.2 and older

http://neon-toolkit.org/wiki/Download/2.3.2
eXtreme Design

- XD is a general approach to ontology engineering
- Local Use Case (LUC)
  - represents the current modeling issue
- Generic Use Case (GUC)
  - represents a generic problem that is solved by the associated ODP
- GUC and LUC are represented in a compatible comparable way – Competency Questions?
- What ODP to reuse?
  - The one where LUC matches GUC
  - Note: GUC are often more abstract than LUC
Matching GUC and LUC

- A LUC can be completely or partly described exactly in terms of the GUC
  - Guc: Performing in a concert
  - Luc: John Coltrane performed in a concert in Japan in 1966
- A LUC is a more specific case of the GUC
  - Guc: Participating in an event
  - Luc: Mary attended a scientific conference
- A LUC can be described in terms of part of the GUC
  - Guc: Participating in an event held in a certain place at a certain time
  - Luc: Mary attended a conference in Italy
Why the name “XD”?

- Inspired by XP 😊 with focus on design
- An agile methodology for web ontology design
- It is part of the NeOn methodology
XD principles

- Customer involvement and feedback
- Customer stories to derive CQs (+ contextual statements, reasoning requirements)
- CP reuse and modular design (ontology networks)
- Collaboration and integration
- Task-oriented design
- Test-driven design
- Pair design
**XD Iteration**

1. **Project initiation and scoping**
   - Design team
   - CP catalogues

2. **Identifying CP catalogues**
   - Design team

3. **Collecting requirement stories**
   - Design pair

4. **Eliciting requirements and constructing module(s) from CPs**
   - Design pair

5. **Releasing module(s)**
   - All stories covered?

6. **Integrating partial solutions, evaluating and revising**
   - Integration team

7. **Releasing new version of Ontology Network**
   - Ontology Network

---

Department of Computer and Information Science (IDA)
Linköpings universitet, Sweden

October 5, 2011
XD – Method steps

- **Tasks 1-2: Project initiation and scoping, collecting resources**
  - Essential to understand the context and task of the ontology
  - Customer involvement – domain experts
  - Setup the project environment (collaboration support) + the resources to be used
  - Agree on general “rules”, e.g. naming conventions

- **Task 3 – Collect requirements stories**
  - Example scenarios (cf. the story of the exercise)
  - Should be short and “modular”

- **Select a story (each design pair!)**
XD – Method steps

- Task 4.1 – Transform the story into CQs
  - Divide & Conquer strategy in the large – each pair works on one story at the time
    - Stories are associated with priority values
    - Based also on design pair competencies
  - Derive requirements from the text
    - CQs, contextual statements and reasoning requirements
    - Check with customer representative & SW developers
  - Should correspond to actual queries/tasks that the user/system need to pose/perform

- Select a CQ (each pair iterates)
  - …or coherent set of CQs
  - Together with associated contextual statements & reasoning requirements
  - Divide & Conquer strategy in the small – pair iterates over CQs and creates module(s)
XD – Method steps

- Task 4.2 – Matching and selecting ODPs
  - How? Either only intellectually or with some tool support e.g. XD Selector
  - Can I describe my local problem in terms of the general problem of the ODP?
  - Does the ODP solve the same “design issue”? 
  - Partial match – Is it worth the overhead?
  - Several ODPs needed – Composition of ODPs
  - In case there is no matching ODP – consider to create one!
  - May exist several options
  - “Rule of thumb” – use the most (domain) specific one applicable
XD – Method steps

- Task 4.3 – Reuse and integrate selected Content ODPs
  - Specialize
  - Import
  - Extend
  - Integrate (compose)
Task 4.3. Test and Fix – Unit tests

1. SPARQL queries

Assume the following CQ:
What role did a certain person play in the production of a certain play during a certain time period?

```
WHERE {
  ?rolePlaying a :PlayingSituation .
  ?production :productionOfPlay ?play
  ?timeInterval :hasStartDate ?startTime .
  ?timeInterval :hasEndDate ?endTime .
}
```

2. Producing inferences

3. “Stress testing”
XD – Method steps

- **Task 5 – Release module**
  - Make sure the module is commented and ready
  - Post the module so that it is accessible by the other pairs
  - Post any new reusable Content ODP developed
  - Taken over by integration pair

- **Task 6 – Integrate, test and fix (integration team)**
  - Integrate with overall ontology so far
    - Alignment may be needed
    - Refactoring may be needed
  - Run all unit tests based on all included requirements

- **Task 7 – Release new version of the ontology**
  - Distribute
  - Generate documentation
  - Check customer satisfaction, i.e. evaluation of overall ontology including non-functional requirements
XD Summary

- XD is an agile method – start building small modules that solve a few requirements, then add more, we don’t decide on all the requirements at once
- Testing is essential – by figuring out the test you figure out how the model should work!
- Collaboration is essential
- Many problems are resolved in the integration phase – alignments or refactoring?
  - Need for good overall design policies
- You are about to experience the method!