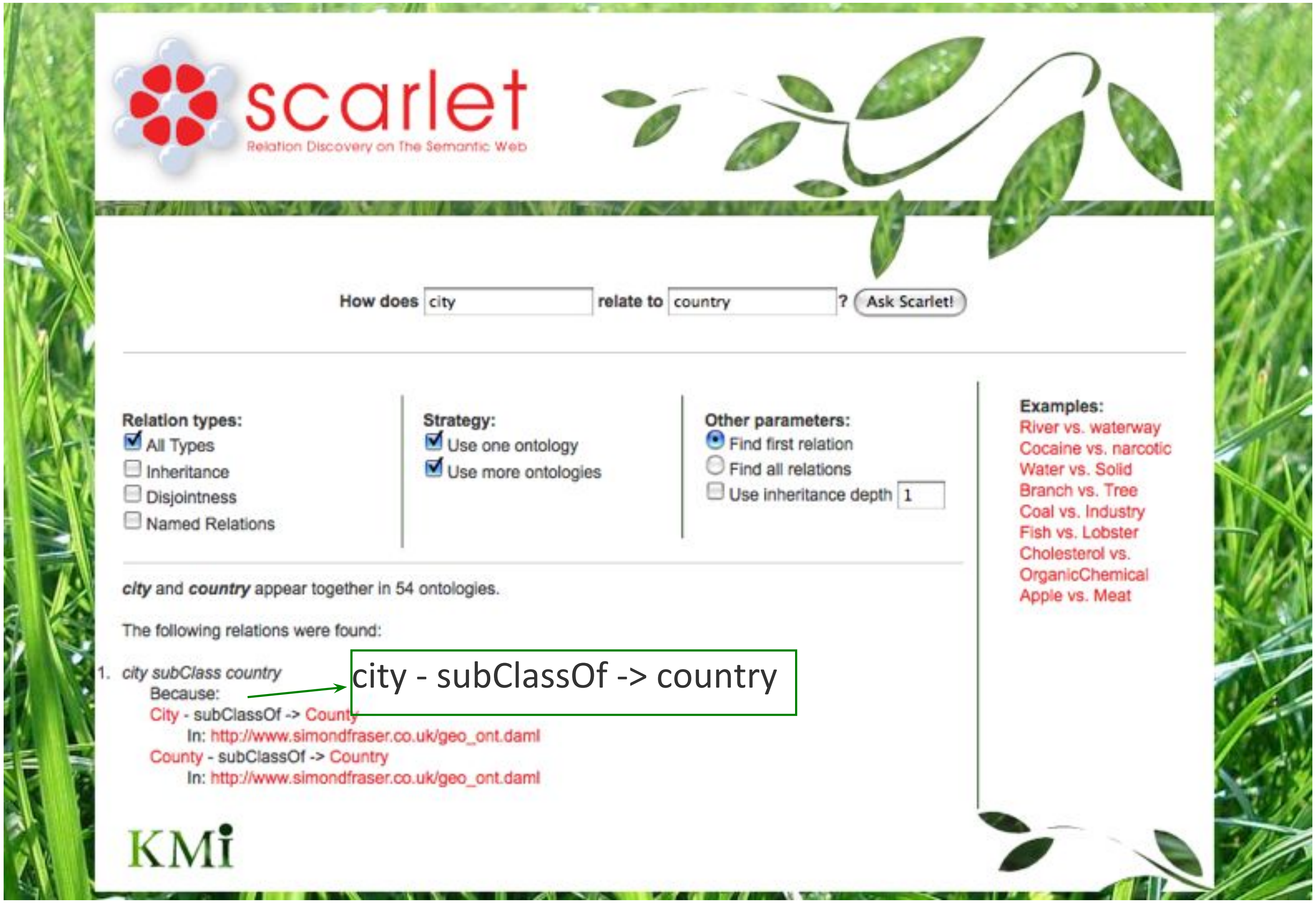
Ontology Design Patterns and XD

Eva Blomqvist

eva.blomqvist@liu.se

LINKÖPING UNIVERSITY



How does city relate to country ?

Relation types:
- [✓] All Types
- [ ] Inheritance
- [ ] Disjointness
- [ ] Named Relations

Strategy:
- [✓] Use one ontology
- [✓] Use more ontologies

Other parameters:
- [✓ Find first relation]
- Find all relations
- [ ] Use inheritance depth 1

Examples:
River vs. waterway
Cocaine vs. narcotic
Water vs. Solid
Branch vs. Tree
Coal vs. Industry
Fish vs. Lobster
Cholesterol vs. OrganicChemical
Apple vs. Meat

city and ***country*** appear together in 54 ontologies.

The following relations were found:

1. *city subClass country*
   Because:
   City - subClassOf -> County
   In: http://www.simondfraser.co.uk/geo_ont.daml
   County - subClassOf -> Country
   In: http://www.simondfraser.co.uk/geo_ont.daml

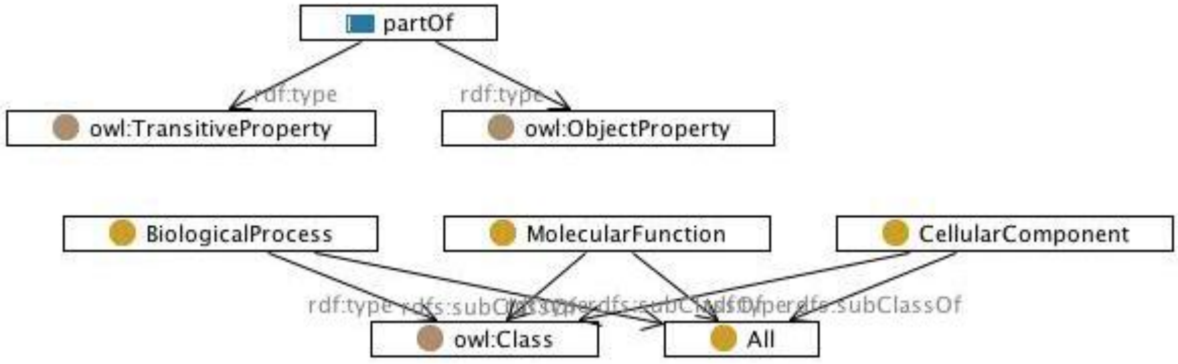
city - subClassOf -> country

What we can do with OWL

- … (maybe) we can check the consistency, classify, and query our knowledge base
- … but, remember the Scarlet example
  - City subClassOf Country
- Logical consistency is not the main problem
  - e.g. rdfs:subClassOf an be wrongly used and still we have consistency
- Why is OWL not enough?
  - OWL gives us logical language constructs, but does not give us any guidelines on how to use them in order to solve our tasks.
  - E.g. modeling something as an individual, a class, or an object property can be quite arbitrary

Solutions?

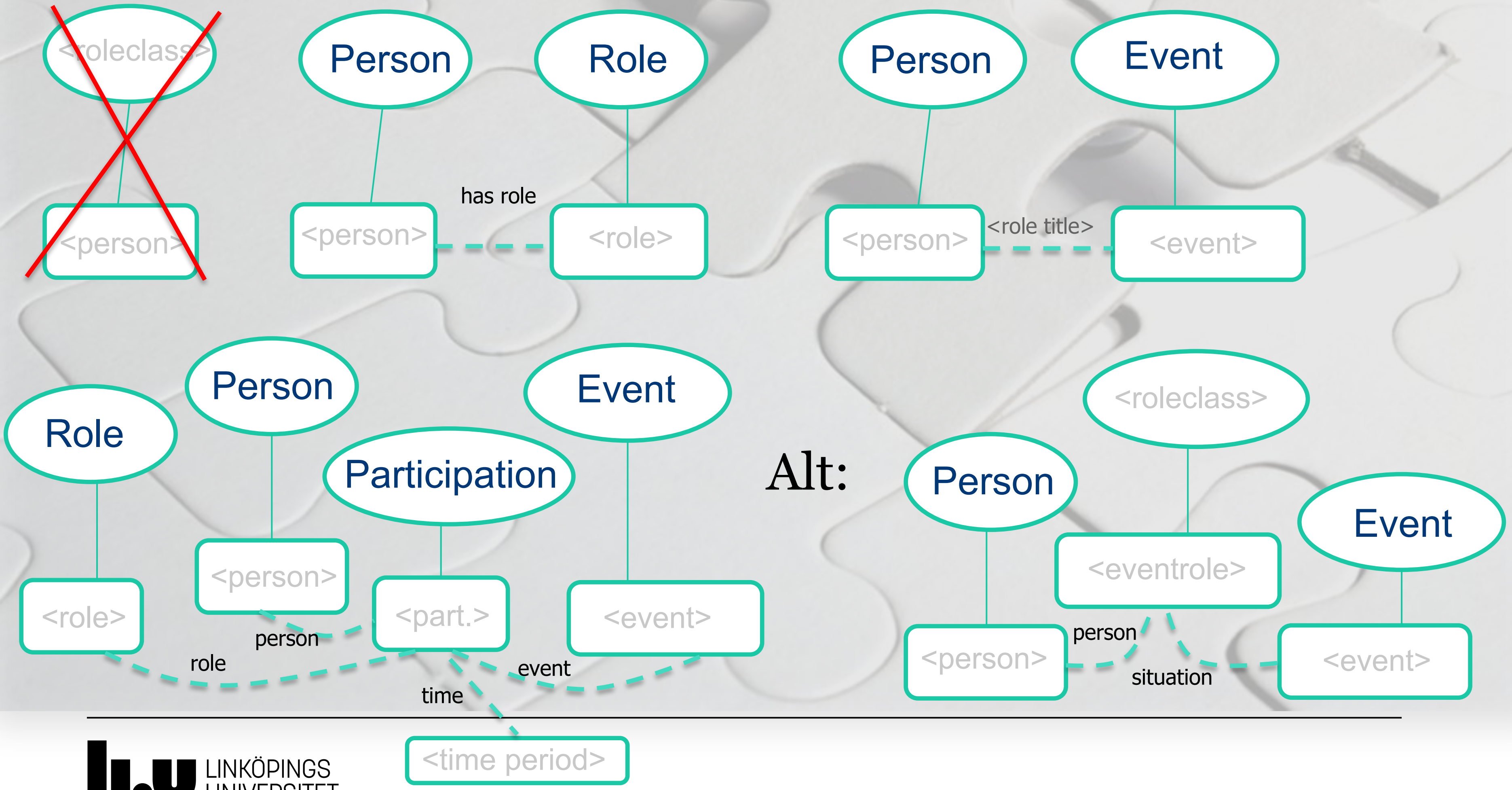
- OWL is not always enough for building a *good* ontology, and we cannot ask all web users neither to learn logic, or to study ontology design
- Reusable solutions
  - Top-level ontologies, standard ontologies etc.
  - **Ontology Design Patterns**
- … provided that tools have good usability ☺



Various types of ODPs

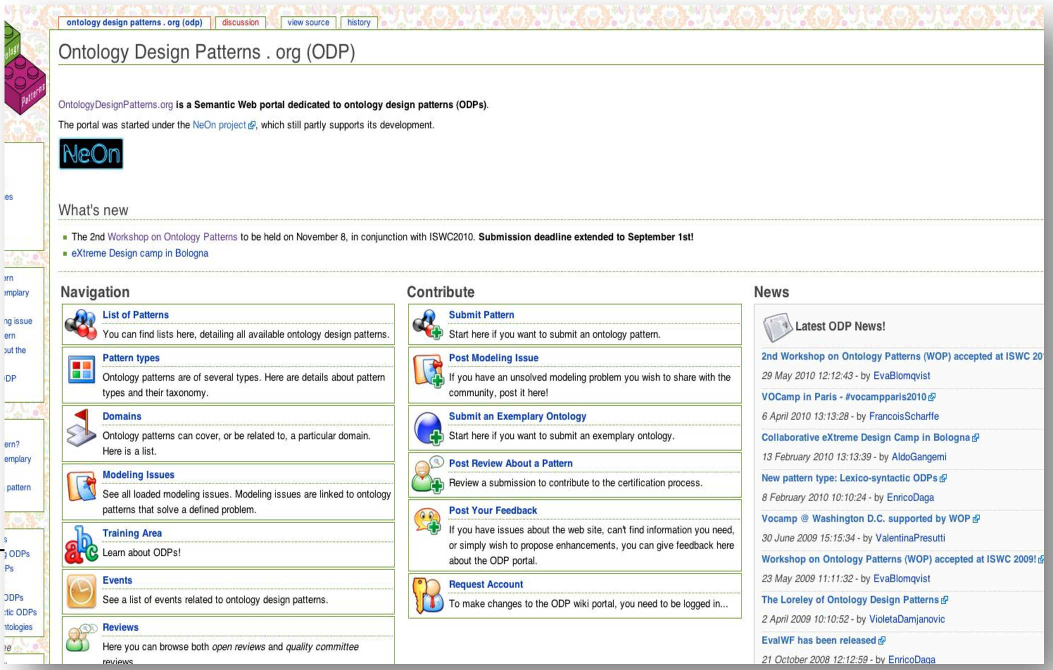
- Logical patterns – "workarounds" and shortcuts in modelling
  - Example: n-ary relations
- Content patterns – components with a non-empty signature, sometimes domain specific
  - Example: how to model roles
  - Can be used as "templates" or ideas for your own solution, or as components that are specialised
- Correspondence patterns, transformation patterns…

Example - Role patterns (ODP)



Catalogues of ODPs

- Content ODPs are collected and described in catalogues, books, papers…
- The ontologydesignpatterns.org initiative maintains a repository of ODP *proposals*



Ontology Design Patterns . org (ODP)

OntologyDesignPatterns.org **is a Semantic Web portal dedicated to ontology design patterns (ODPs)**.

The portal was started under the NeOn project, which still partly supports its development.

What's new

- The 2nd Workshop on Ontology Patterns to be held on November 8, in conjunction with ISWC2010. **Submission deadline extended to September 1st!**
- eXtreme Design camp in Bologna

Navigation

List of Patterns
You can find lists here, detailing all available ontology design patterns.

Pattern types
Ontology patterns are of several types. Here are details about pattern types and their taxonomy.

Domains
Ontology patterns can cover, or be related to, a particular domain. Here is a list.

Modeling Issues
See all loaded modeling issues. Modeling issues are linked to ontology patterns that solve a defined problem.

Training Area
Learn about ODPs!

Events
See a list of events related to ontology design patterns.

Reviews
Here you can browse both *open reviews* and *quality committee* reviews.

Contribute

Submit Pattern
Start here if you want to submit an ontology pattern.

Post Modeling Issue
If you have an unsolved modeling problem you wish to share with the community, post it here!

Submit an Exemplary Ontology
Start here if you want to submit an exemplary ontology.

Post Review About a Pattern
Review a submission to contribute to the certification process.

Post Your Feedback
If you have issues about the web site, can't find information you need, or simply wish to propose enhancements, you can give feedback here about the ODP portal.

Request Account
To make changes to the ODP wiki portal, you need to be logged in...

News

Latest ODP News!

2nd Workshop on Ontology Patterns (WOP) accepted at ISWC 20
29 May 2010 12:12:43 - by EvaBlomqvist

VOCamp in Paris - #vocampparis2010
6 April 2010 13:13:28 - by FrancoisScharffe

Collaborative eXtreme Design Camp in Bologna
13 February 2010 13:13:39 - by AldoGangemi

New pattern type: Lexico-syntactic ODPs
8 February 2010 10:10:24 - by EnricoDaga

Vocamp @ Washington D.C. supported by WOP
30 June 2009 15:15:34 - by ValentinaPresutti

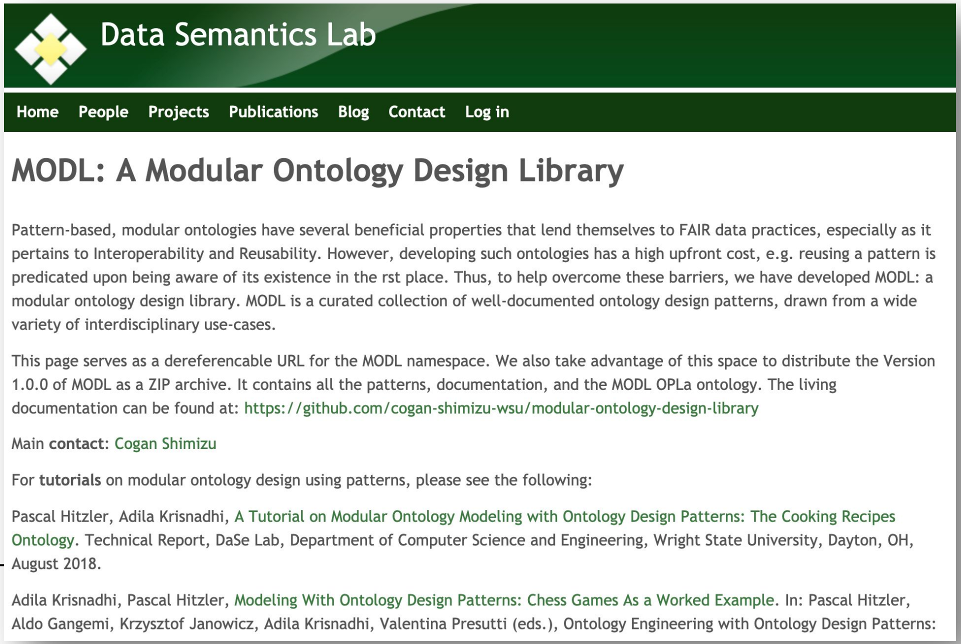
Workshop on Ontology Patterns (WOP) accepted at ISWC 2009!
23 May 2009 11:11:32 - by EvaBlomqvist

The Loreley of Ontology Design Patterns
2 April 2009 10:10:52 - by VioletaDamjanovic

EvalWF has been released
21 October 2008 12:12:59 - by EnricoDaga

Catalogues of ODPs

- A curated, but smaller repository of ODPs are available here: https://daselab.cs.ksu.edu//content/modl-modular-ontology-design-library



MODL: A Modular Ontology Design Library

Pattern-based, modular ontologies have several beneficial properties that lend themselves to FAIR data practices, especially as it pertains to Interoperability and Reusability. However, developing such ontologies has a high upfront cost, e.g. reusing a pattern is predicated upon being aware of its existence in the rst place. Thus, to help overcome these barriers, we have developed MODL: a modular ontology design library. MODL is a curated collection of well-documented ontology design patterns, drawn from a wide variety of interdisciplinary use-cases.

This page serves as a dereferencable URL for the MODL namespace. We also take advantage of this space to distribute the Version 1.0.0 of MODL as a ZIP archive. It contains all the patterns, documentation, and the MODL OPLa ontology. The living documentation can be found at: https://github.com/cogan-shimizu-wsu/modular-ontology-design-library

Main **contact**: Cogan Shimizu

For **tutorials** on modular ontology design using patterns, please see the following:

Pascal Hitzler, Adila Krisnadhi, A Tutorial on Modular Ontology Modeling with Ontology Design Patterns: The Cooking Recipes Ontology. Technical Report, DaSe Lab, Department of Computer Science and Engineering, Wright State University, Dayton, OH, August 2018.

Adila Krisnadhi, Pascal Hitzler, Modeling With Ontology Design Patterns: Chess Games As a Worked Example. In: Pascal Hitzler, Aldo Gangemi, Krzysztof Janowicz, Adila Krisnadhi, Valentina Presutti (eds.), Ontology Engineering with Ontology Design Patterns:

The eXtreme Design methodology

Ontology Engineering Methodologies

- Mostly focus has been on overall life-cycle and steps of the methodology – rather than *how* to actually perform them
- Few are focused on *reuse* and *getting something out fast*

- 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

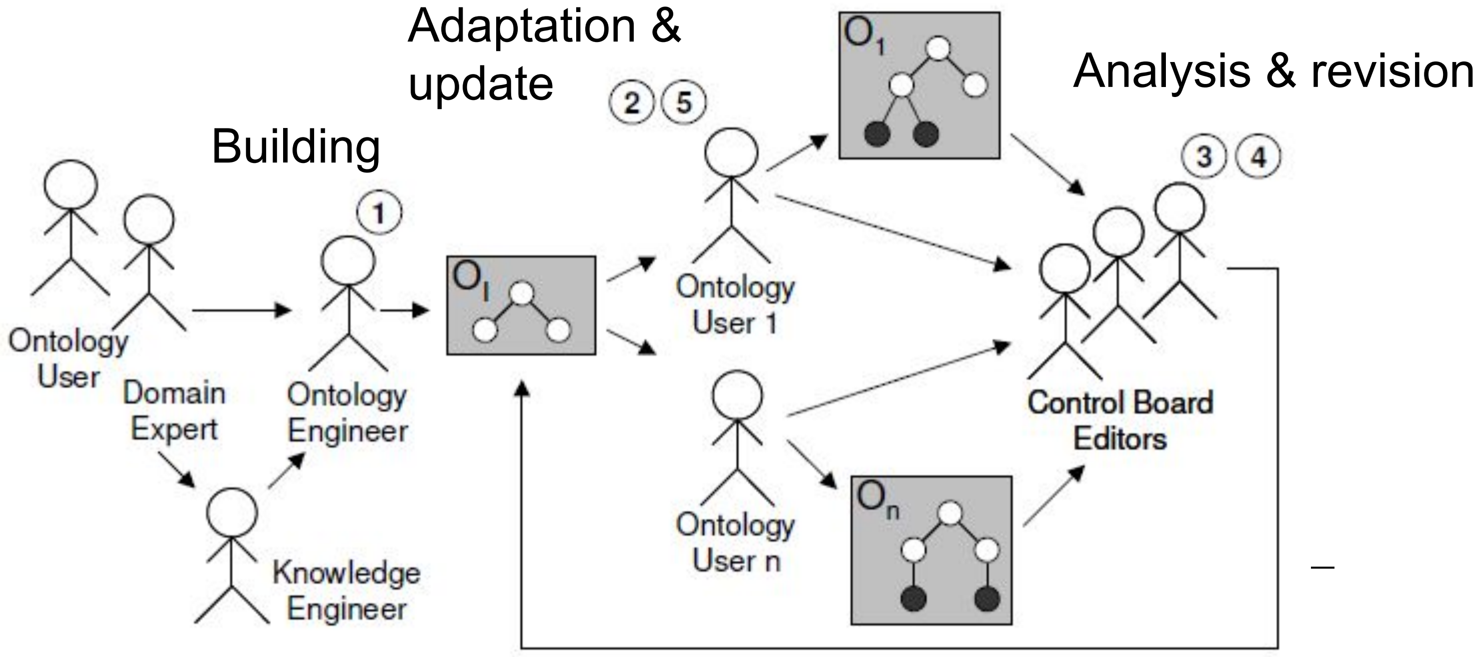
      (1) Domain an scope (2) Consider reuse (3) Enumerate terms
      (4) Develop class hierarchy (5) Define the properties
      (6) Define restrictions and constraints (7)Create instances

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

Example: DILIGENT (~2004)

- Based on theories for argumentation
- Intended for
  - Empowering domain experts in ontology engineering
  - Continuous and distributed construction and update



eXtreme Design – XD

- Provides a different perspective
  - Modular ontologies
  - "Rapid prototyping" - get something out fast

- **Probably you want to develop your own process and your own ODPs in the end!**

Why the name "XD"?

- Inspired by XP but with focus on modelling and design
- An agile methodology for web ontology design
- Developed as part of the NeOn methodology

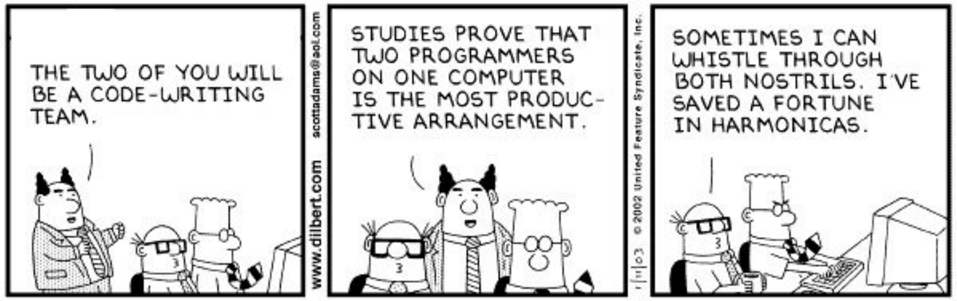
WE'RE GOING TO TRY SOMETHING CALLED EXTREME PROGRAMMING.
FIRST, PICK A PARTNER. THE TWO OF YOU WILL WORK AT ONE COMPUTER FOR FORTY HOURS A WEEK.
THE NEW SYSTEM IS A MINUTE OLD AND I ALREADY HATE EVERYONE.
www.dilbert.com
scottadams@aol.com
© 2002 United Feature Syndicate, Inc.
1/9/03

XD principles

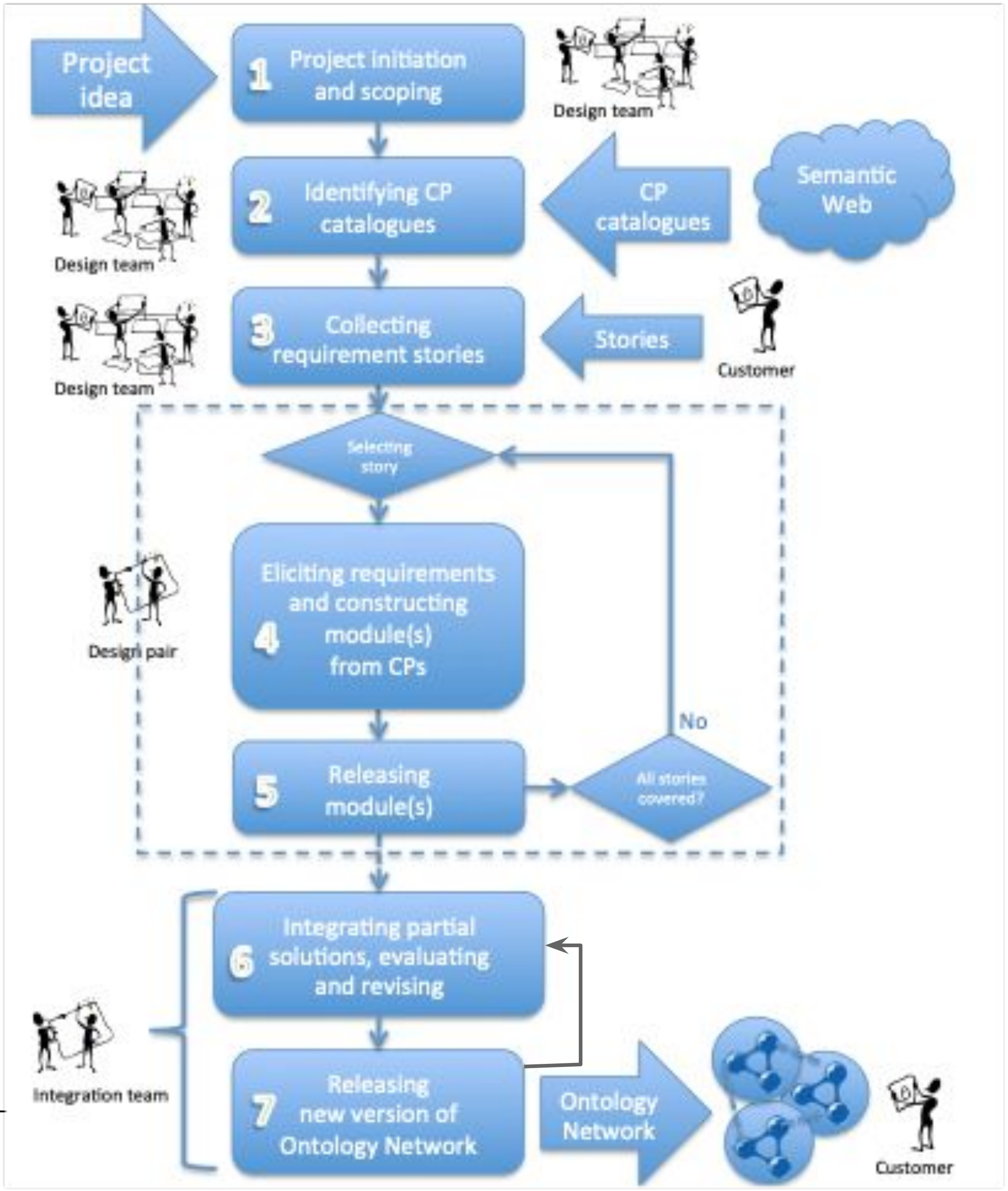
- Customer/domain expert/developer involvement and feedback
- "Customer" stories to derive CQs (+ restrictions/constraints, reasoning requirements)

I CAN'T GIVE YOU ALL OF THESE FEATURES IN THE FIRST VERSION.
AND EACH FEATURE NEEDS TO HAVE WHAT WE CALL A "USER STORY."
OKAY, HERE'S A STORY: YOU GIVE ME ALL OF MY FEATURES OR I'LL RUIN YOUR LIFE.
www.dilbert.com
scottadams@aol.com
© 2002 United Feature Syndicate, Inc.

- ODP reuse and modular design (ontology networks)
- Collaboration and integration
- Task-oriented design, verified by tests
- Pair design


THE TWO OF YOU WILL BE A CODE-WRITING TEAM.
STUDIES PROVE THAT TWO PROGRAMMERS ON ONE COMPUTER IS THE MOST PRODUCTIVE ARRANGEMENT.
SOMETIMES I CAN WHISTLE THROUGH BOTH NOSTRILS. I'VE SAVED A FORTUNE IN HARMONICAS.
www.dilbert.com
scottadams@aol.com
© 2002 United Feature Syndicate, Inc.

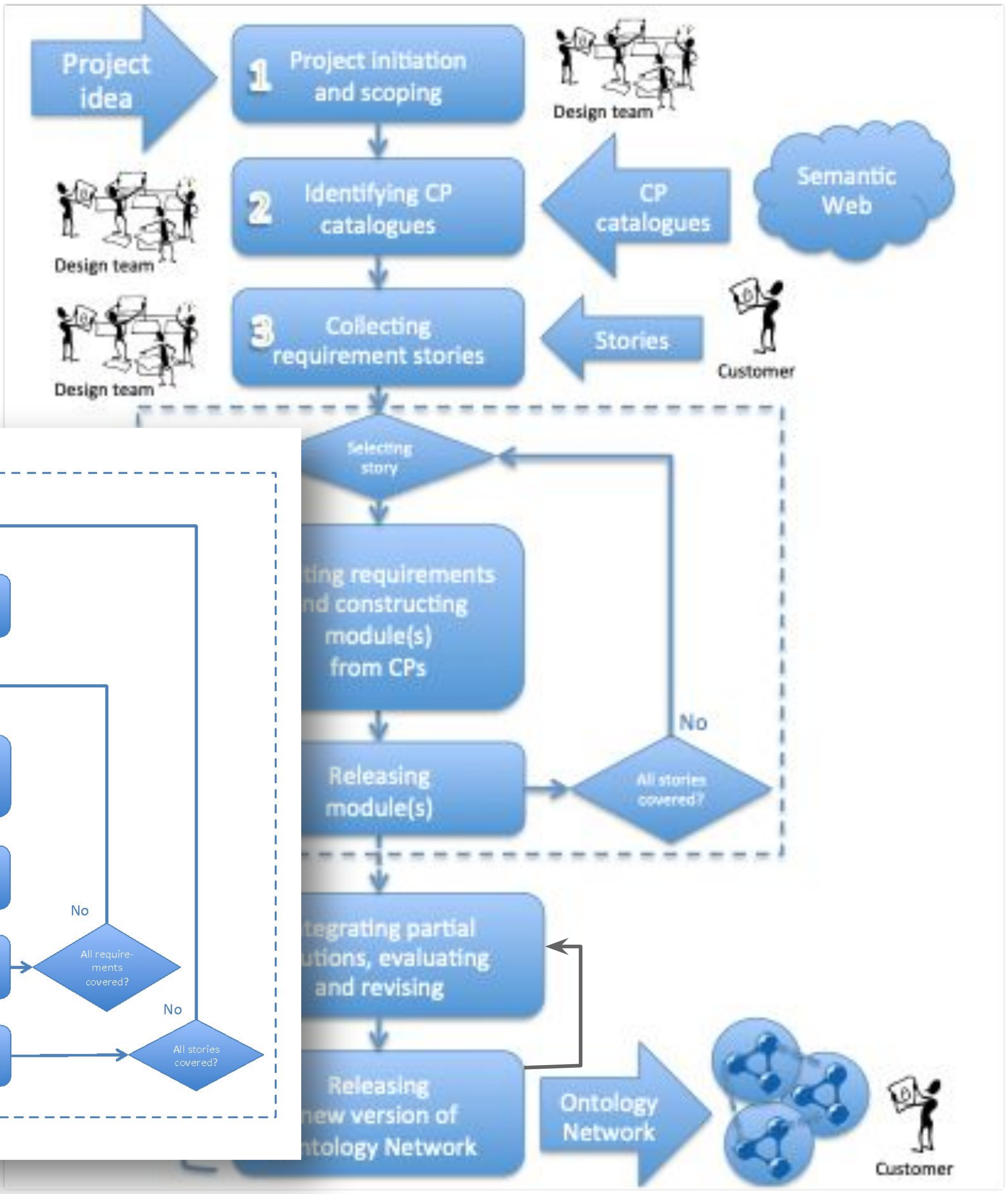
XD Overall Process

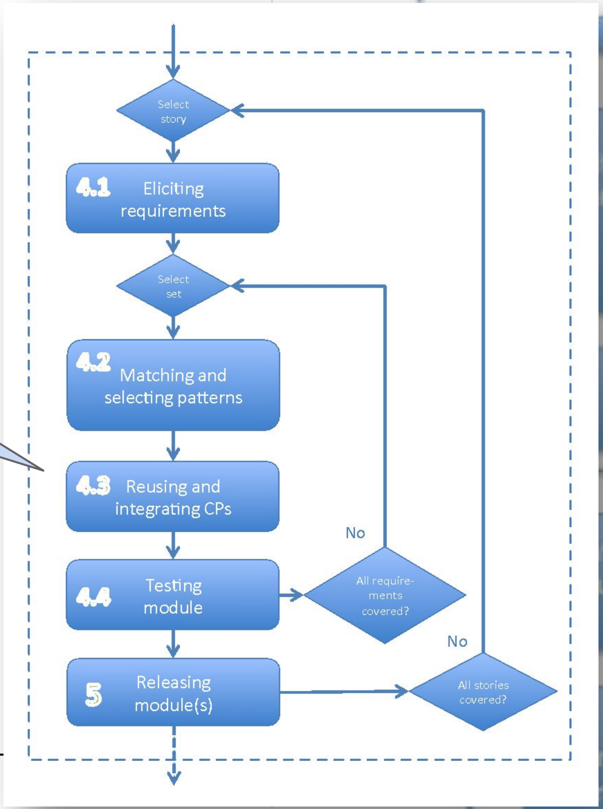


Project idea
1 Project initiation and scoping
Design team
2 Identifying CP catalogues
CP catalogues
Semantic Web
Design team
3 Collecting requirement stories
Stories
Customer
Design team
Selecting story
Design pair
4 Eliciting requirements and constructing module(s) from CPs
No
5 Releasing module(s)
All stories covered?
6 Integrating partial solutions, evaluating and revising
Integration team
7 Releasing new version of Ontology Network
Ontology Network
Customer

LINKÖPING UNIVERSITY

XD Modelling Iteration


Project idea
1 Project initiation and scoping
Design team
2 Identifying CP catalogues
CP catalogues
Semantic Web
Design team
3 Collecting requirement stories
Stories
Customer
Design team
Selecting story
ting requirements nd constructing module(s) from CPs
No
Releasing module(s)
All stories covered?
tegrating partial utions, evaluating and revising
Releasing new version of ntology Network
Ontology Network
Customer


Select story
4.1 Eliciting requirements
Select set
4.2 Matching and selecting patterns
4.3 Reusing and integrating CPs
No
4.4 Testing module
All require-ments covered?
No
5 Releasing module(s)
All stories covered?

OR modelling from scratch!

Things to note about XD

- Can be adapted to various settings
  - Pairs or individual development?
  - Roles of ontology engineers and other experts
  - Adapt the level of communication and control
- You quickly have a tangible result
  - "Rapid prototyping" of ontologies
- Integration step is crucial and may involve lots of refactoring unless you introduce more guidance

Blomqvist, E., Hammar, K., and Presutti, V.: Engineering Ontologies with Patterns – The eXtreme Design Methodology. In: *Ontology Engineering with Ontology Design Pattern – Foundations and Applications*, IOS Press, 2016.

www.liu.se

LINKÖPING
UNIVERSITY