

Ontology Driven Architecture

- Proposal by the W3C's task force on the Application of the Semantic Web in Software Engineering

•Editors:

Phil Tetlow, IBM, <philip.tetlow@uk.ibm.com>

Jeff Pan, University of Manchester, <pan@cs.man.ac.uk>

Daniel Oberle, Universität Karlsruhe, <oberle@fzi.de>

Evan Wallace, National Institute of Standards and Technology,
<ewallace@cme.nist.gov>

Elisa Kendall, Sandpiper Software, <ekendall@sandsoft.com>

•Additional Contributors and Special Thanks to:

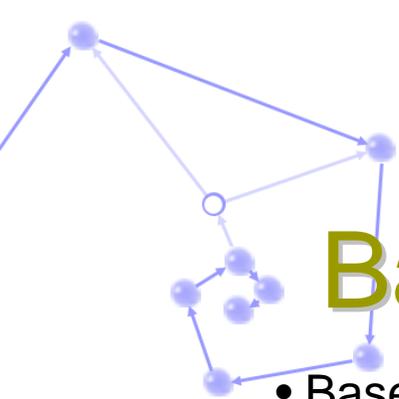
Andrea Zisman, City University London, <zisman@soi.city.ac.uk>

Alan Rector, Manchester University, <rector@cs.man.ac.uk>

Michael Goedicke, Universität Essen , <goedicke@s3.uni-essen.de>

Michael Uschold, Boeing, <michael.f.uschold@boeing.com>

Grady Booch, IBM, <gbooch@uk.ibm.com>



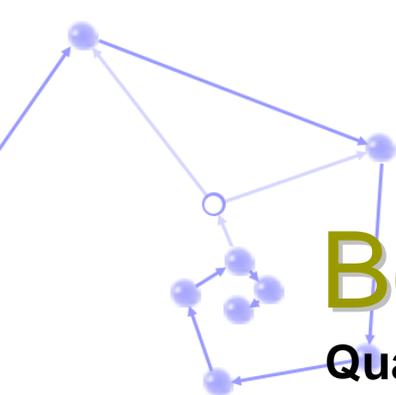
Background

- Based on similar work in the areas:
 - Automated Software Engineering
 - Model Driven Architecture
- So why is it different?
- The answer is simple:
 - ODA is aimed specifically at utilising Semantic Web technologies like RDF and OWL. Previous work in this area was not ‘Web friendly’
 - Furthermore previous approaches were aimed at ‘closed’ problem domains. The Web is not closed, but open, leading the potential for far greater capabilities in the area of component reuse and extensibility
 - Most models tend to be abstraction dependant as where ontologies are not necessarily so.



Premise

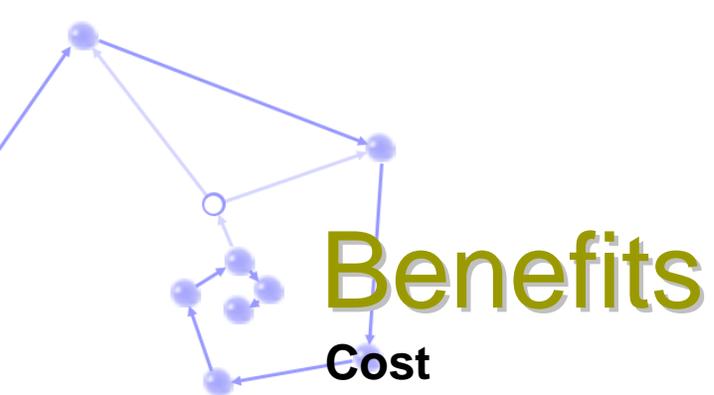
- All models are merely descriptions, be they in OWL, UML or even a programming language like Java
- The more formally meaningful the models and the earlier they can be introduced into the development lifecycle the better.
- If we have a model in one for formal description nomenclature it should be capable of translation into any other format. This leads to a natural progression of descriptions
 - a) Domain Knowledge (Underlying ontology(s))
 - b) Design Models
 - c) Implementation descriptions (deployable code)
- If different nomenclatures from this progression come from the same family of technologies (namely XML in this case), it should be possible to mix different descriptive languages to increase engineering capability – for example providing formal annotation in a UML model in the shape of an OWL ontology
- There are strong similarities between Object Oriented implementation approaches and the syntax of RDF and OWL



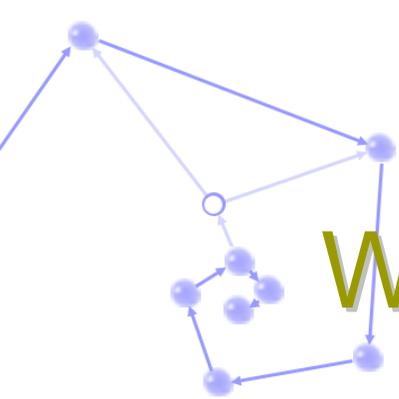
Benefits

Quality

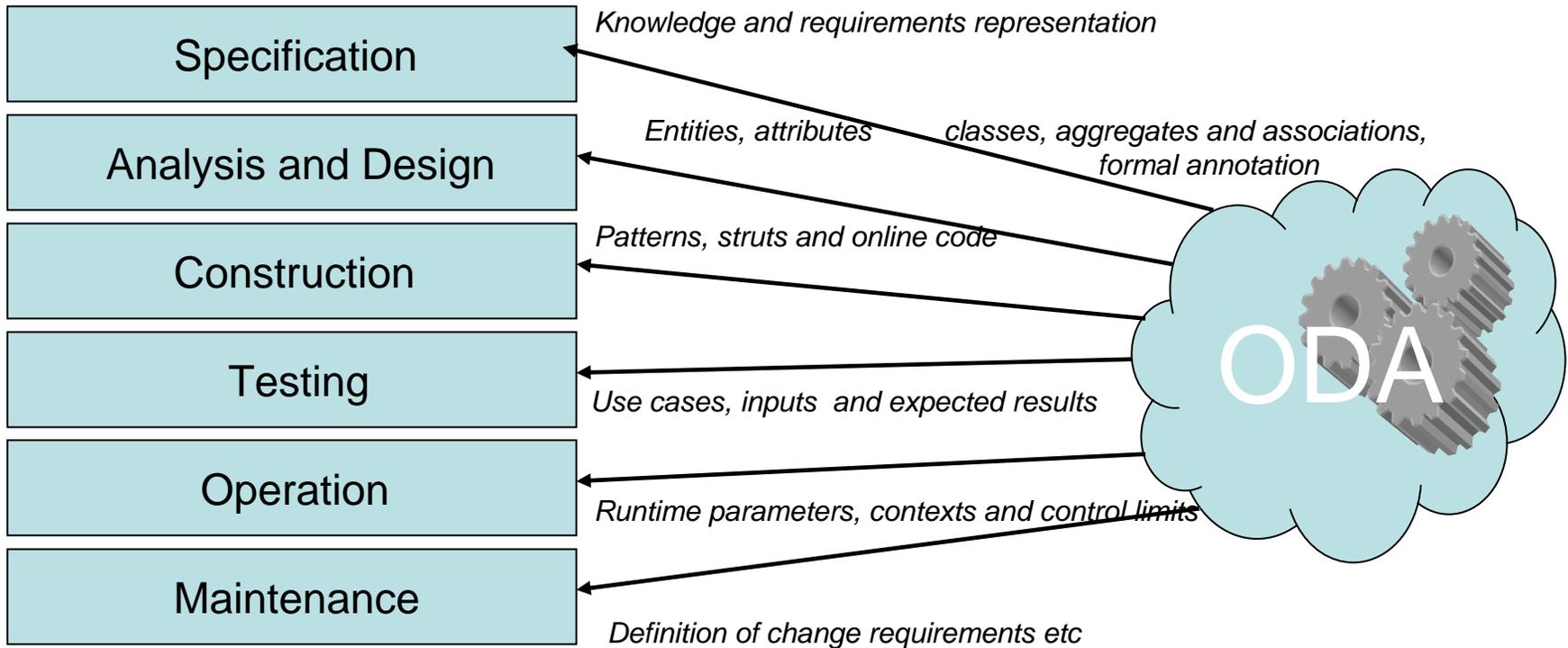
- Requirements conformance and consistency checking.
- Rigorous typing, categorisation and identification.
- Ease of formal specification.
- Communication of requirement and intent to domain experts and developers through:
 - The ability to capture, relate and manage models of systems and information at multiple levels and from multiple standpoints.
 - Increases in semantic expressivity through coverage of concepts not embodied in current se tools.
- Reductions in design ambiguity.
- Unified syntax for tooling.

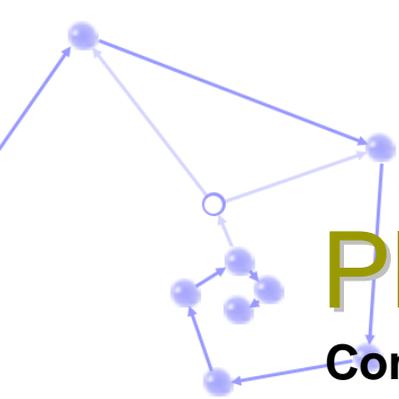


- Reduce maintenance overhead through increases in consistency.
- Increased potential for reuse, substitution and extension via accurate content discovery on the Semantic Web.



When and Where to Use





Please Feel Free to Contribute

Contact via

<http://www.w3.org/2001/sw/BestPractices/SE/ODA/>