Model-Driven Semantic Web Engineering

Dragan Gašević[1] and Gerd Wagner[2]

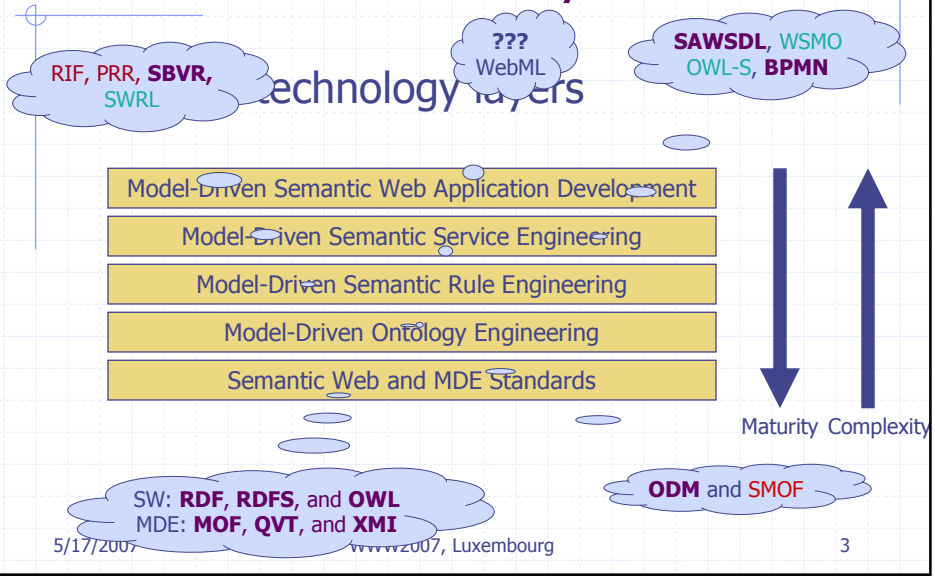
[1]Athabasca University, Canada
dgasevic@acm.org
[2]Brandenburg University of Technology at Cottbus, Germany
wagnerg@tu-cottbus.de

Outline

http://hydrogen.informatik.tu-cottbus.de/moodle/course/view.php?id=12

- Fundamentals
- Model-Driven Ontology Engineering
- Model-Driven Semantic Web Rule Engineering
- Model-Driven Semantic Web Service Engineering
- Model-Driven Semantic Web Application Development

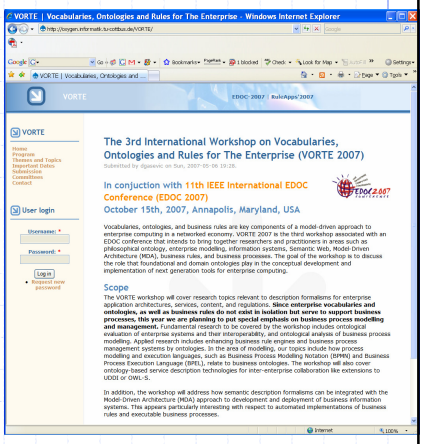
Summary



VORTE 2007

http://oxygen.informatik.tu-cottbus.de/VORTE/

- 3rd International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE 2007) @ EDOC2007, Annapolis, MD, USA
- Submission:
  - Full-papers: July 7, 2007
- **Information Systems**
  - Best papers: J. special issue
  - ISI-indexed



ATEM 2007

- 4th International Workshop on Language Engineering @ MoDELS2007, Nashville, TN, Sep 30 – Oct 5, 2007
- Submission:
  - http://planetmde.org/atem2007
  - Abstracts: June 13, 2007
  - Full-papers: June 20, 2007
- **IET Software (aka IEE Proceedings Software)**
  - ISI-indexed
  - Special issue on ***Language Engineering***
  - Submission due: June 1, 2007

Model-Driven Semantic Web Engineering

Dragan Gašević[1] and Gerd Wagner[2]

[1]Athabasca University, Canada
dgasevic@acm.org
[2]Brandenburg University of Technology at Cottbus, Germany
wagnerg@tu-cottbus.de

Fundamentals

Semantic Web &
Model Driven Engineering

Semantic Web

To create a universal medium for the exchange of data.

It is envisaged to smoothly interconnect
personal information management,
enterprise application integration, and the global sharing
of commercial, scientific and cultural data.
Facilities to put machine-understandable data on
the Web are quickly becoming a high priority
for many organizations, individuals and communities.

(Semantic Web Activity Statement, 2006)
http://www.w3.org/2001/sw/Activity

Modeling-Driven (Software) Engineering

Modeling-Driven Engineering addresses platform complexity and the inability of third-generation (programming) languages to alleviate this complexity and express domain concepts effectively.

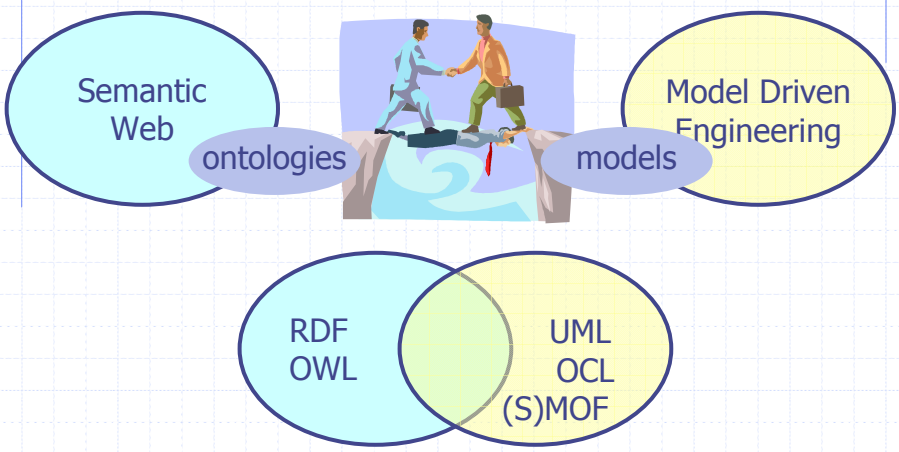
(Schmidt, 2006)

Modeling is the future …

And the promise here is that you write a lot less code, that you have a model of the business process …

(Bill Gates, 2004)

Bridging the SW and MDE



Some Differences

- MDE
  - models are abstractions/simplifications
  - prescriptive (specification) or descriptive
  - using a single author/designer perspective
- Semantic Web
  - intended for knowledge representation
  - everyone can say anything

Some Similarities

- Semantic Web and MDE
  - UML models
    - classes, properties (attributes), generalization (inheritance), …
  - ontologies
    - classes, properties, specialization (inheritance), …
  - model the real world!!!!
  - help to build the next generation of software

Semantic Web

What is an ontology?

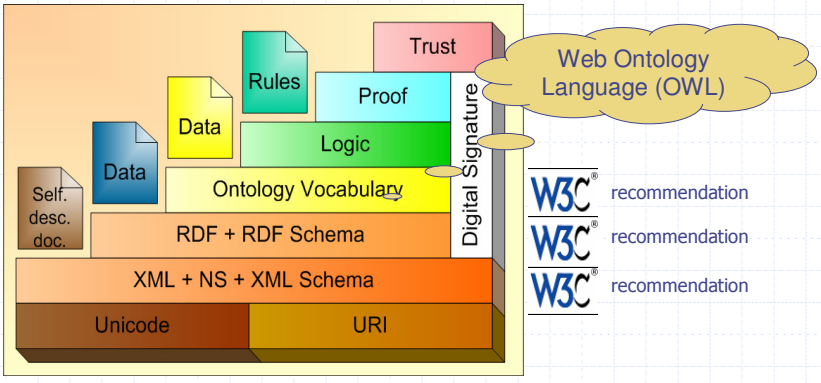
- Classic definitions
  (Gruber, 1993), (Guarino, 1994)
  - a specification of a conceptualization
  - a formal and declarative representation of some subject area

What is an ontology?

- Other definitions (Hendler, 2001)
  - a set of knowledge terms, including the vocabulary, the semantic interconnections, and some simple rules of inference and logic for some particular topic

Semantic Web "Layer Cake"

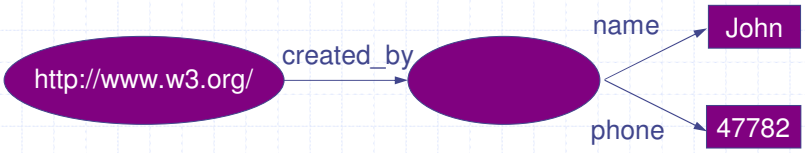
- **Ontologies** and rules



RDF

- Example

OBJECT	PROPERTY	VALUE
http://www.w3.org/	created_by	_:x
_:x	name	"John"
_:x	phone	"47782"



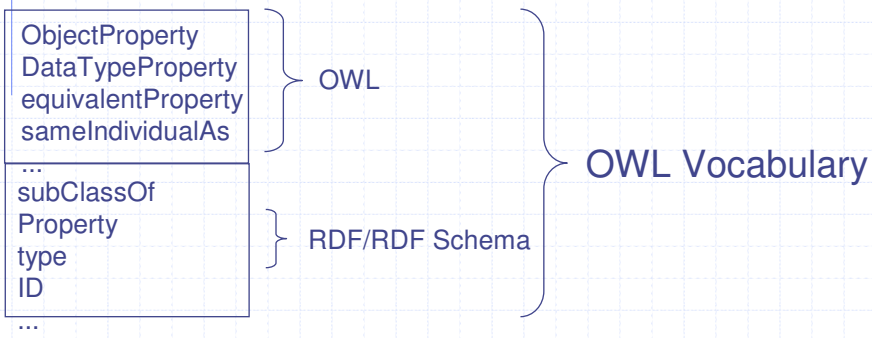
RDF Schema

- Allows to define a vocabulary (classes and properties)
- Example

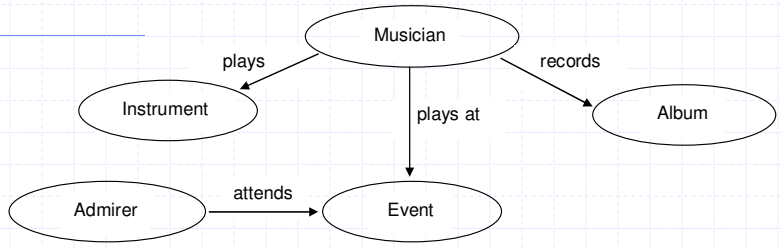
```
<rdfs:Class rdf:ID="Herbivore">
   <rdfs:subClassOf rdf:resource="#Animal"/>
</rdfs:Class>
```

Web Ontology Language

- OWL extends RDF and RDF Schema



OWL Example: Musician Ontology



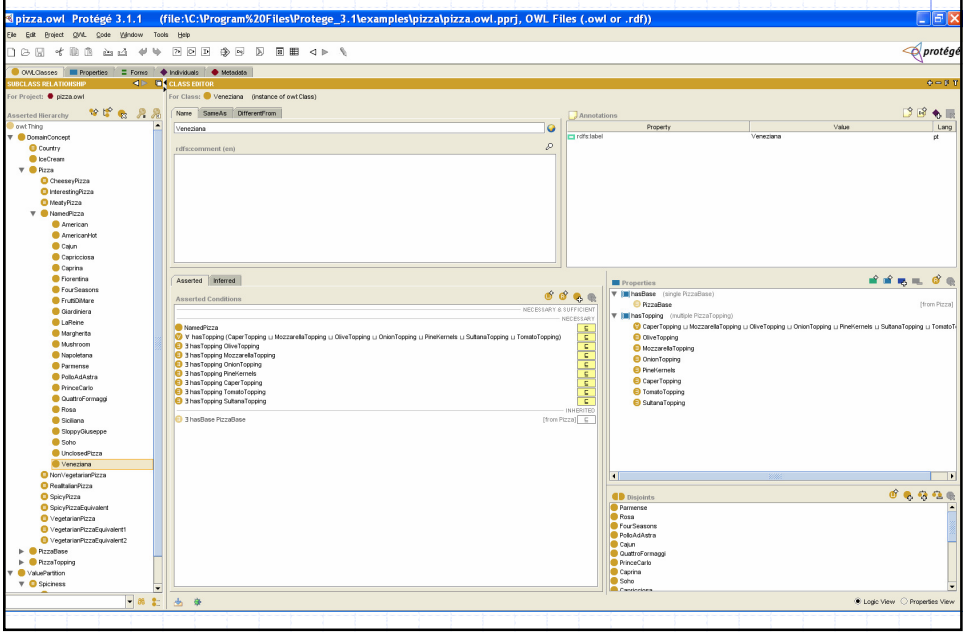

```
<owl:Class rdf:ID="Event"/>
<owl:Class rdf:ID="Album"/>
<owl:Class rdf:ID="Instrument"/>
<owl:Class rdf:ID="Musician"/>
<owl:Class rdf:ID="Admirer"/>
```

```
<owl:ObjectProperty rdf:ID="plays">
  <rdfs:domain rdf:resource="#Musician"/>
  <rdfs:range rdf:resource="#Instrument"/>
</owl:ObjectProperty>
```

Tools for Building Ontologies

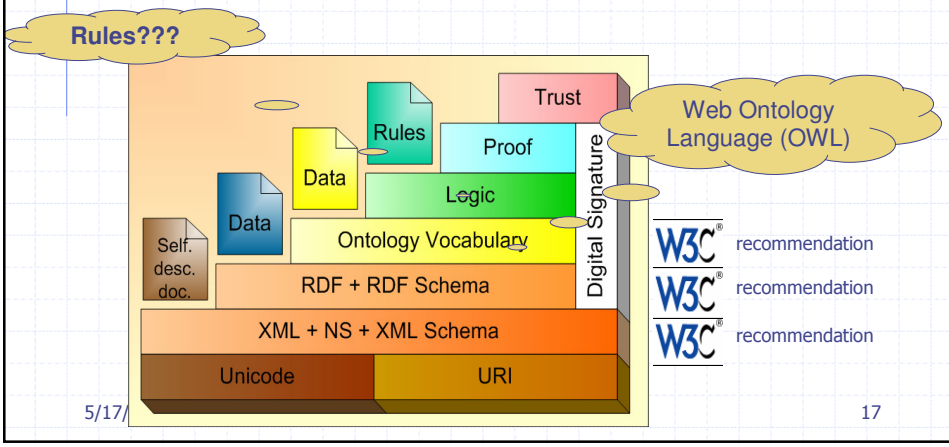
- Protégé
- OntoEdit
- OilEd
- Chimaera
- ...

Protégé



Semantic Web "Layer Cake"

- Ontologies and **Rules**



Semantic Web Rule Efforts

- Official W3C effort:
  Rule Interchange Format (RIF)
- Semantic Web reasoning layer over ontology languages
  - Over RDF/S: N3
  - Over OWL: Semantic Web Rule Language (SWRL)

W3C Rule Interchange Format (RIF)

- W3C initiative
- Identified ten use-cases to be supported. Example rules:
  - A buyer must provide credit card information together with delivery information (address, postal code, city, and country).
  - A wireless device can transmit on a 5 GHz band if no priority user is currently using that band.
  - If inspector believes vehicle is repairable then process as repair otherwise process as total loss.
- Related efforts
  - REWERSE Rule Interchange Format (R2ML)
  - RuleML

Semantic Web Rules

- There is no standard
- There is no consent whether this language should based on
  - Open-World Assumption
  - Closed-World Assumption (Negation-as-Failure)
- Semantic Web Rule Language (SWRL)
  - An extension of OWL

SWRL Example

A brother of a person's parent is the person's uncle.

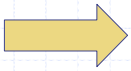
```
<ruleml:imp>
 <ruleml:_body>
  <swrlx:individualPropertyAtom  swrlx:property="hasParent">
   <ruleml:var>x1</ruleml:var>
   <ruleml:var>x2</ruleml:var>
  </swrlx:individualPropertyAtom>
  <swrlx:individualPropertyAtom  swrlx:property="hasBrother">
   <ruleml:var>x2</ruleml:var>
   <ruleml:var>x3</ruleml:var>
  </swrlx:individualPropertyAtom>
 </ruleml:_body>
 <ruleml:_head>
  <swrlx:individualPropertyAtom  swrlx:property="hasUncle">
   <ruleml:var>x1</ruleml:var>
   <ruleml:var>x3</ruleml:var>
  </swrlx:individualPropertyAtom>
 </ruleml:_head>
</ruleml:imp>
```

5/17/2007

Model-Driven Engineering

Model Driven Engineering

- Developing in parallel with Semantic Web
- Object Modeling Group effort
- The latest paradigm shift in software engineering (Bézivin, 2002)
  - from OO technology…
  - …to model technology



Model Driven Development

(Mellor et al, 2003)

Model-Driven Engineering

- [Favre, 2004]
  - Model engineering is the disciplined and rationalized production of models
  - MDE is a subset of system engineering in which the process heavily relies on the use of models and model engineering
  - Model Driven (Software) Development is the intersection between MDE and software engineering, that is, it is the subset of MDE which is concerned with software production

CIM – PIM – PSMs

Domain Model (CIM)

? + requirements definition

Design Model (PIM)

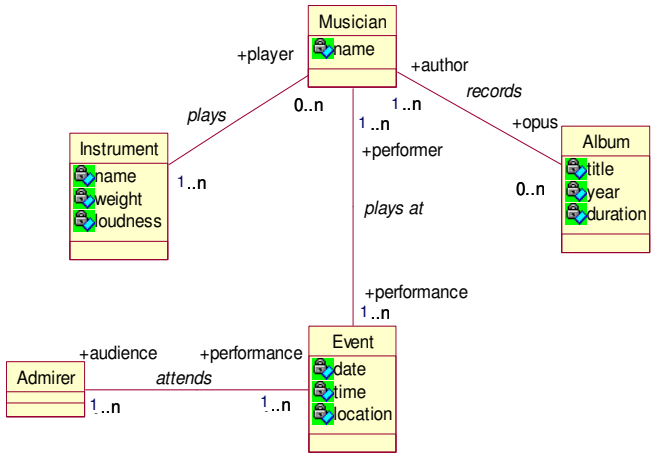
PSMs

Implementation model 1 (SQL)

Implementation model 2 (Java)

Implementation model 3 (XML)

Example of a Model



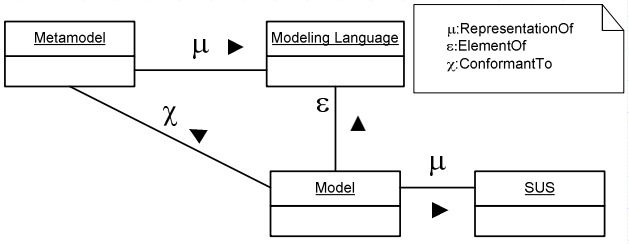
What is a metamodel?

A metamodel makes statements about what can be expressed in the valid models of a certain modeling language.

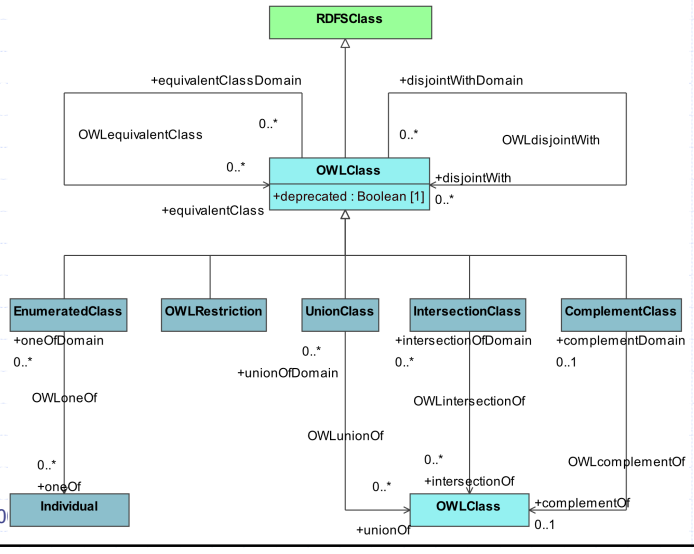
Seidewitz, 2003

What is a metamodel?

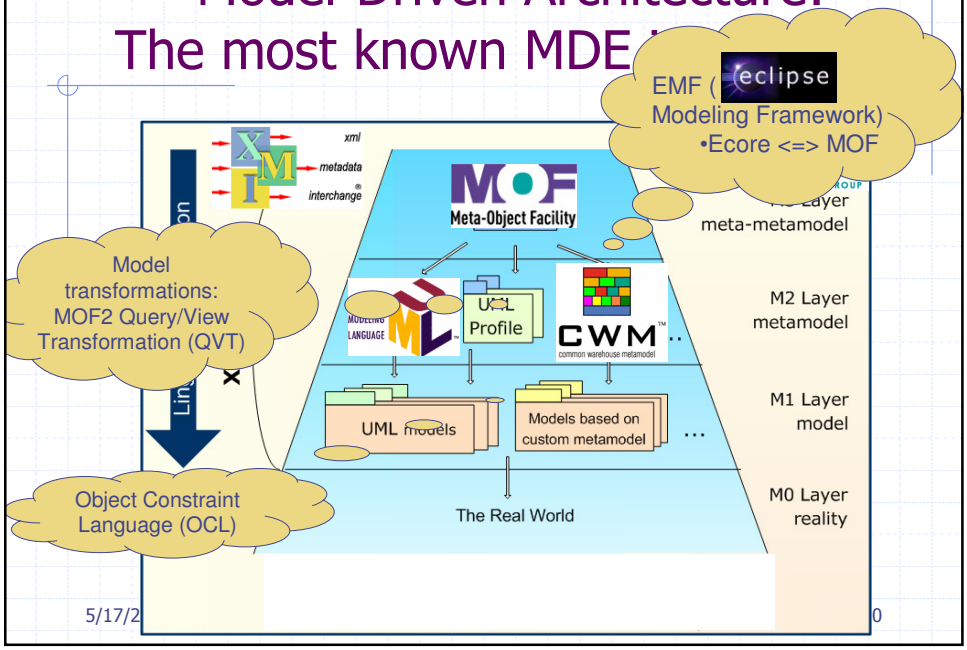
- In fact, a metamodel:
  - is a model of a modeling language, or
  - makes statements about what can be expressed in the valid models of a certain modeling language
- The correspondence between a model, a metamodel, a modeling language, a system under study



Example of a Metametamodel

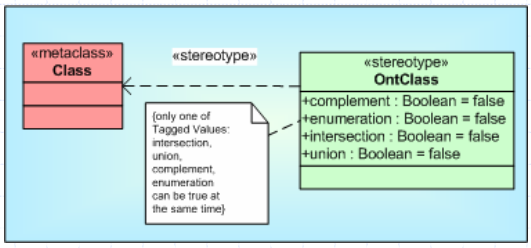


Model-Driven Architecture: The most known MDE

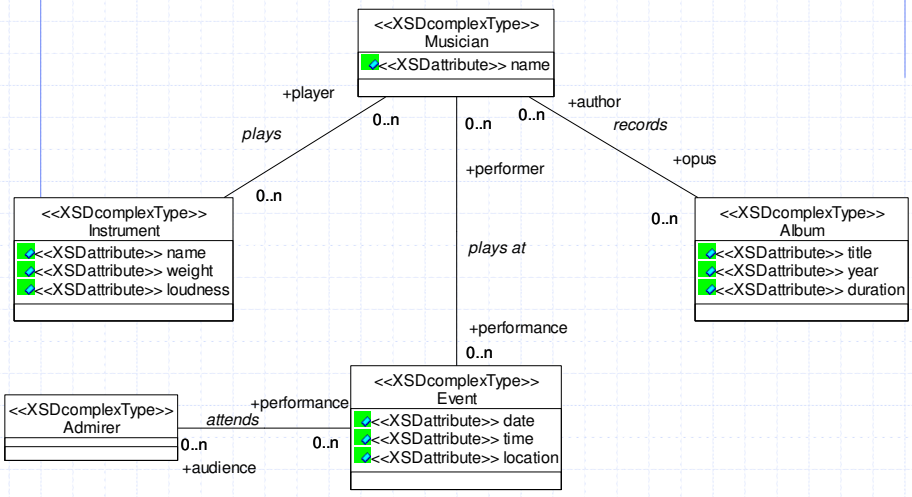


UML Profiles

- Extension mechanism
  - stereotypes, tagged values, and OCL constraints
  - UML2 improved support for profiles

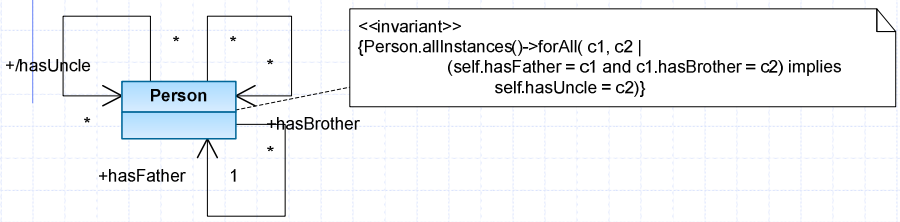


An Example of a UML Profile



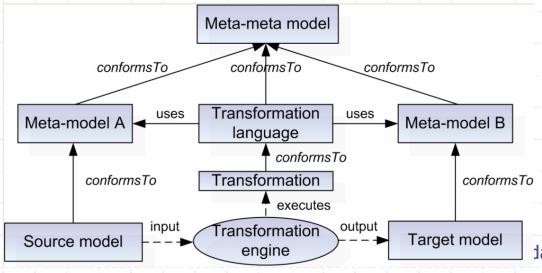
Object Constraint Language (OCL)

- Example



Model Transformations

- Model-to-Model Transformations
  - Query / View / Transformation (QVT)
  - Atlas Transformation Language (ATL)
- Technical Spaces
  - Model-to-Text and Text-To-Model
    - Textual Concrete Syntax (TCS)
  - Model-to-XML and XML-to-Model
    - ATL Injector and ATL Extractor



Why marriage?

Credit: Elisa Kendall

- Knowledge Representation supports *reasoning* about resources
  - Supports semantic alignment among differing vocabularies and nomenclatures
  - Enables consistency checking and model validation, business rule analysis
  - Allows us to ask questions over multiple resources that we could not answer previously
  - Enables policy-driven applications
- MOF/UML provides no help with reasoning
- KR is not focused on the mechanics of managing models or metadata
- Complementary technologies – despite some overlap

Ontologies and Software engineering

- An approach
  - Ontology Driven Architecture (ODA)
  - Trying to improve the state of the art in software engineering by using ontologies
  - W3C's effort
    - http://www.w3.org/2001/sw/BestPractices/SE/
    - Ontology Driven Architectures and Potential Uses of the Semantic Web in Software Engineering
    - A Semantic Web Primer for Object-Oriented Software Developers
  - Still, vague and unclear definition

Ontologies in Software engineering

- Happel, H.J. & Seedorf, S., "Applications of Ontologies in Software Engineering," *In Proceedings of the 2nd International Workshop on Semantic Web Enabled Software Engineering*, Athens, GA, USA, Nov 6, 2006.
- Start from the SE definition
  - application of a systematic, disciplined, quantifiable approach to **the development, operation, and maintenance** of software
  - captures software life-cycle

So, where do we go today?!

- The focus of the tutorial
  - How to integrate Semantic Web technologies into (model-driven) software engineering development process
  - How to use MDE principles to manage definitions of Semantic Web technologies
  - How to use MDE principles to develop Semantic service-oriented architectures
  - How to employ MDE principles to develop semantic service-oriented Web applications

Approach

- No mature/complete framework yet
  - Just initial steps

Model-Driven Ontology Engineering

Initial steps

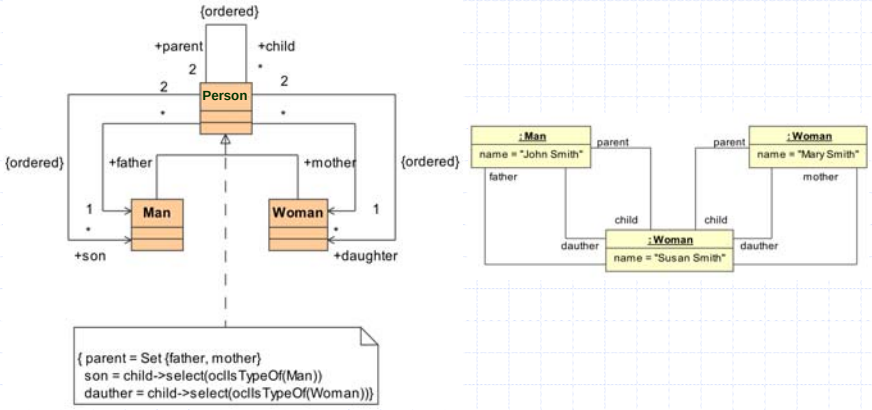
- [Cranefield, 2001]
  - UML class diagrams provide a static modeling capability that is well-suited for representing ontologies
  - UML object diagrams can be interpreted as declarative representations of knowledge
  - OCL for ontology constraints
  - advantage: using the same paradigm for modeling ontologies and knowledge

Cranefield's approach

- Technology requirements
  - XMI – for sharing UML models
  - RDF/XML – for sharing RDS(S) ontologies
  - UML tools that produce UML XMI
  - XSLT that transforms UML XMI to:
    - a set of Java classes and interfaces corresponding to those in the ontology
    - RDF & RDF Schema

Cranefield's approach

- Example: The family ontology



Cranefield's approach

- Transformation to RDF(S)
  - XSLT implementation
    - Classes to RDFS
    - Objects to RDF
  - Mapping problems
    - UML classes have different features – attributes, associations, and association classes
    - RDFS – fields or properties
    - RDFS properties are first-class objects

Cranefield's approach

- Transformation to RDFS
  - Some solutions
    - properties in RDFS have a class prefix (but, this has a problem with class inheritance)
    - upper limit for multiplicity greater than 1 $\Rightarrow$ RDFS bag
    - association ends with a UML "ordered" constraint $\Rightarrow$ RDFS sequences
    - …

Cranefield's approach

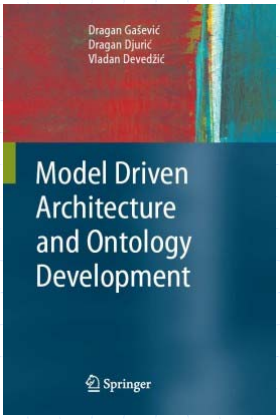
- Resulting RDFS for the family ontology

(excerpt)

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xml:lang="en"
    xmlns:rdfsx="http://nzdis.otago.ac.nz/2000/01/rdf-schema-extensions#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdfs:Class rdf:ID="http://nzdis.otago.ac.nz/0_1/family#Person"/>
    <rdf:Property ID="http://nzdis.otago.ac.nz/0_1/family#Person.name">
        <rdfs:domain rdf:resource="http://nzdis.otago.ac.nz/0_1/family#Person"/>
        <rdfs:range rdf:resource="rdfs:Literal"/>
    </rdf:Property>
    <rdf:Property ID="http://nzdis.otago.ac.nz/0_1/family#Person.parent">
        <rdfs:domain rdf:resource="http://nzdis.otago.ac.nz/0_1/family#Person"/>
        <rdfs:range rdf:resource="rdf:Bag"/>
    </rdf:Property>
    <rdf:Property ID="http://nzdis.otago.ac.nz/0_1/family#Person.father">
        <rdfs:domain rdf:resource="http://nzdis.otago.ac.nz/0_1/family#Person"/>
        <rdfs:range rdf:resource="http://nzdis.otago.ac.nz/0_1/family#Man"/>
    </rdf:Property>
```

```
<rdf:Property ID="http://nzdis.otago.ac.nz/0_1/family#Person.mother">
    <rdfs:domain rdf:resource="http://nzdis.otago.ac.nz/0_1/family#Person"/>
    <rdfs:range rdf:resource="http://nzdis.otago.ac.nz/0_1/family#Woman"/>
</rdf:Property>
```


Monograph



- D. Gašević
  D. Djurić
  V. Devedžić
  Model Driven Architecture and Ontology Development
  Springer, 2006
  ISBN: 3-540-32180-2
- http://www.modelingspaces.org

OMG's Request for Proposal (RFP)

- UML could be a means towards more rapid development of ontologies:
  - familiarity of users with UML
  - availability of UML tools
  - existence of many domain models in UML
  - similarity of those models to ontologies
  - using UML-based tools
    for developing ontologies can be practical

OMG's Request for Proposal (RFP)

This approach continues the Object
...nt Gr... ...l move to more
...man... would also
...hat... commonality and
...y, much
...er metamo... ...es have created
...with the developer and middleware
communities.

OMG Document: ad/2003-03-40

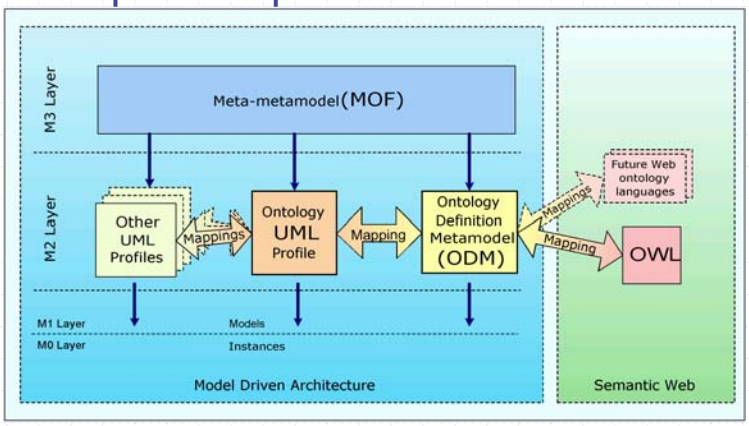
OMG's Request for Proposal (RFP)

This approach continues the Object Management Group's "gradual move to more complete semantic models." It would also create a link between the UML community and the emerging Semantic Web community, much as other metamodels and profiles have created links with the developer and middleware communities.

OMG Document: ad/2003-03-40

OMG's RFP: Ontology Definition Metamodel (ODM)

- Graphical representation of RFP



ODM Specification Requirements

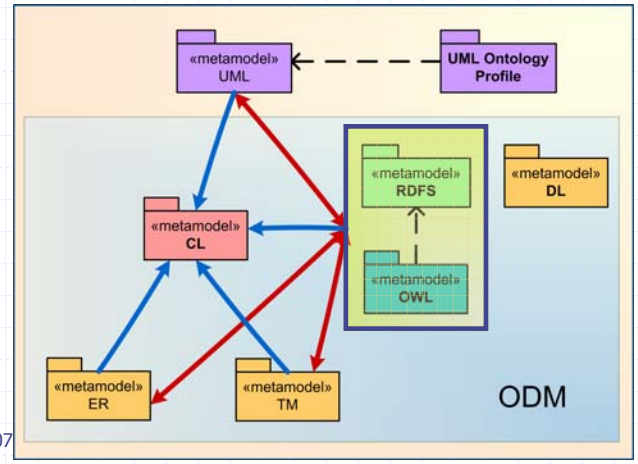
- Mandatory Requirements
  - define ODM using MOF2 Core that represents the semantics of ontologies, including but not necessarily limited to OWL ontologies
    - depict ODM using UML
  - a UML2 Profile extending the UML2 metamodel for ontology definition

ODM Specification Requirements

- Mandatory Requirements
  - forward and reverse engineering of logically equivalent ontologies between environments
    - iterative development of ontologies
  - a language mapping from ODM to OWL DL
    - this mapping should be two-way and bounded
  - an XMI Schema based on ODM

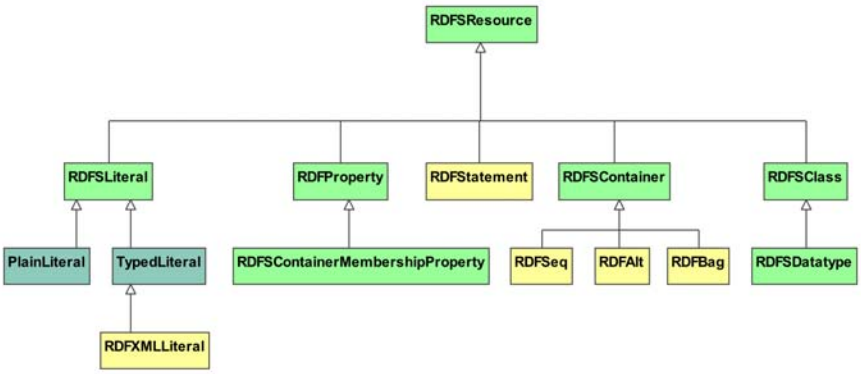
OMG ODM Current Proposal

- ODM Metamodels



OMG ODM Current Proposal – RDFS Metamodel

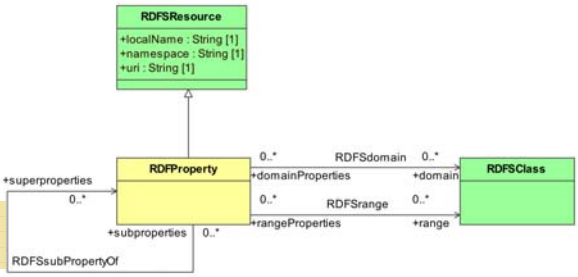
- RDFSResource – All things described by RDF are called resources



OMG ODM Current Proposal – RDFS Metamodel

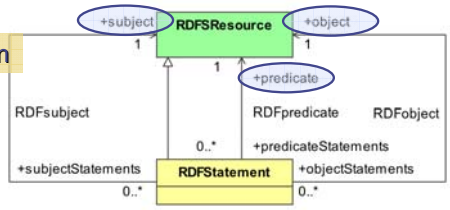
- Property and Statement

Property – Relates Resources to Classes



Bob Marley was Jamaican

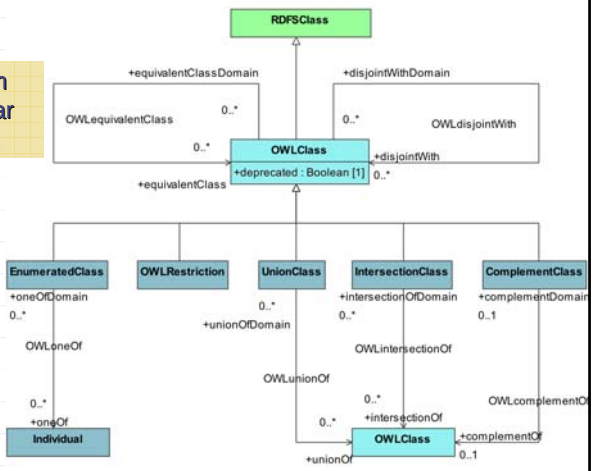
Statement – Connects concrete Resources



OMG ODM Current Proposal – OWL Metamodel

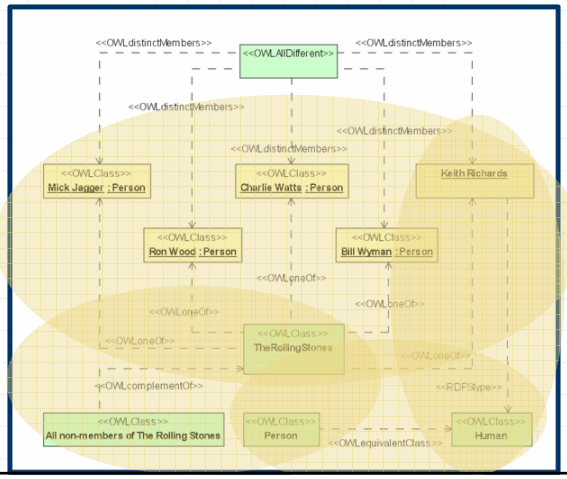
- Classes

Class – an abstraction mechanism for grouping Individuals with similar characteristics



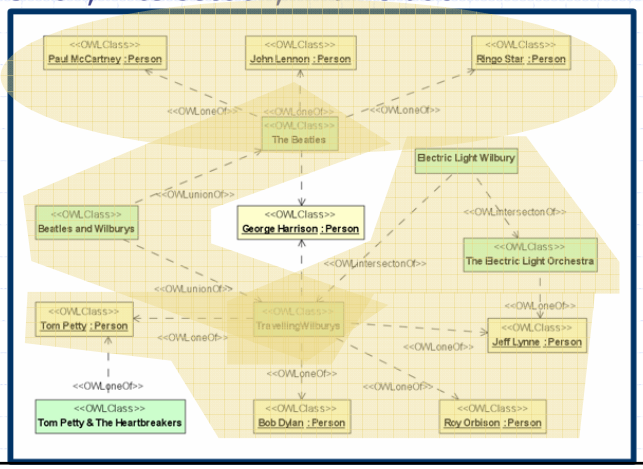
OMG OUP Current Proposal

- Ontology UML Profile (OUP) – OWL Classes



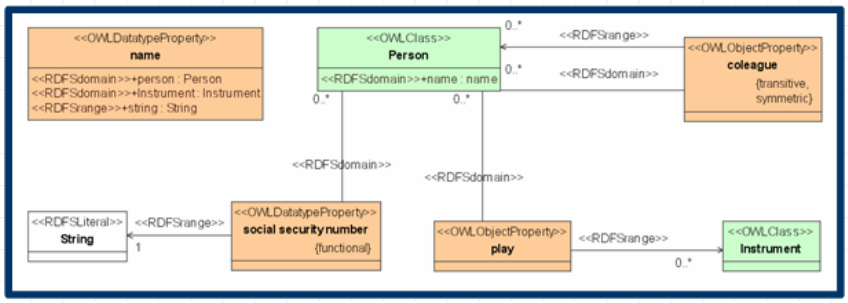
OMG OUP Current Proposal

- Ontology UML Profile (OUP) – OWL Classes
  - Union, Intersection, Enumeration



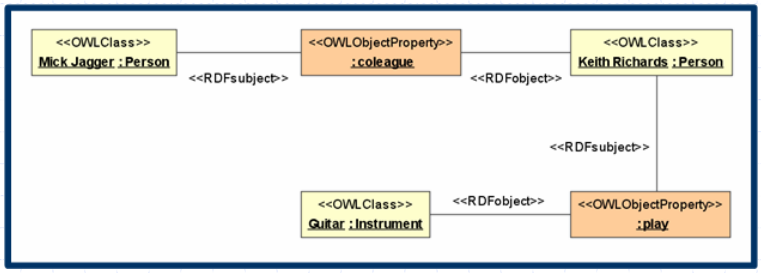
OMG OUP Current Proposal

- Ontology UML Profile (OUP) – OWL Properties



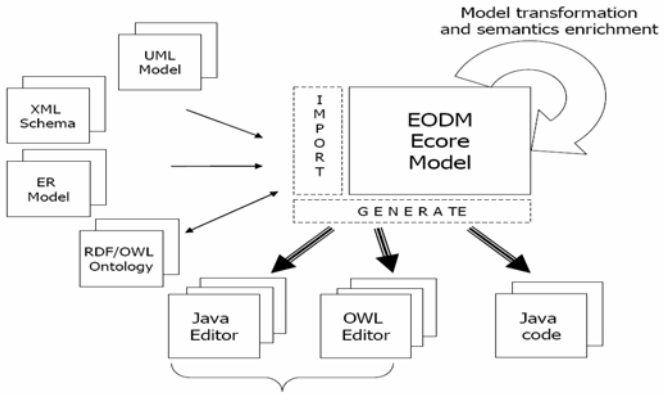
OMG OUP Current Proposal

- Ontology UML Profile (OUP) – OWL Statement



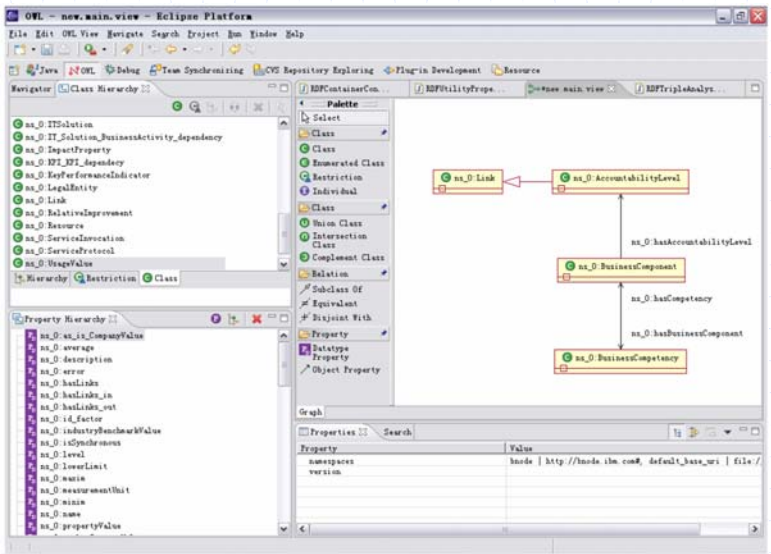
IBM's MDA-Based System for Ontology Engineering

- ODM-Based Ontology editor:
  EODM (EMF ODM)



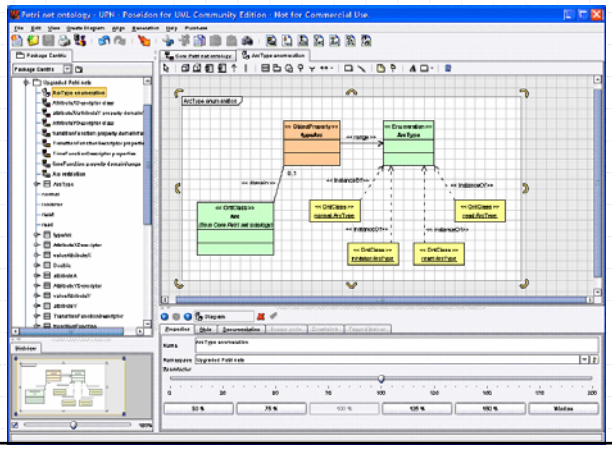
Editors are enhanced to handle RDFS/OWL ontology

IBM's MDA-Based System for Ontology Engineering



Ontology UML Profile

- Ontology UML Profile to OWL converter (Gašević et al, 2005)
  - http://www.sfu.ca/~dgasevic/projects/UMLtoOWL/



Model-Driven Ontology Engineering

- Summary
  - ODM is very close to be the official standard
    - Area for developing tools is mature
    - Initial solutions explored
    - More practical application is coming up
      - Medicine, Risk management, etc.
    - More integration with other relevant standards
      - Rules (production rules), SBVR, Service modeling, etc.
  - New areas of research
    - Unification of knowledge representation and modeling techniques
      - Ontologies and models are similar, buy originating from different communities [Atkinson, 2004]

Model-Driven Ontology Engineering

- Semantic MOF
  - OMG's recent RFP
  - ODM required an appendix to modify the metamodel for MOF implementation
  - Many people thought multiple types were supported

Model-Driven Semantic Web Rule Engineering

Model-Driven Semantic Web Rule Engineering

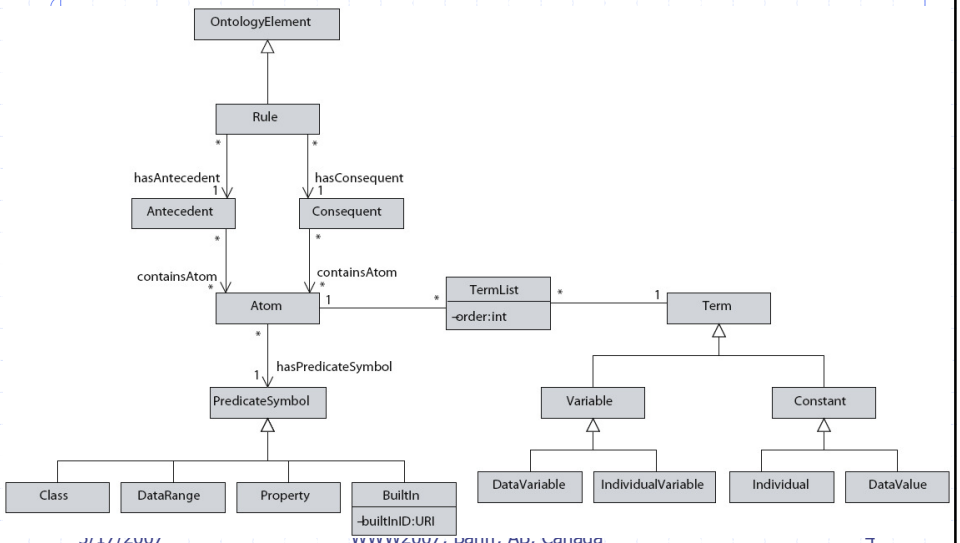
- Continuing efforts of the ODM initiative
  - Using MDE principles to define an abstract syntax (i.e., metamodel) of a Semantic Web rule language
  - Initial steps
    - Rule Definition Metamodel [Brockmans et al, 2006]
      - A metamodel for SWRL
    - Abstract syntax of RuleML [Wagner et al, 2004]

Rule Definition Metamodel (RDM) [Brockmans et al, 2006]

Basic idea:

- ODM is an abstract syntax for OWL
- RDM is an abstract syntax for SWRL
- SWRL is based on OWL
- Thus, RDM is based on ODM

Rule Definition Metamodel (RDM) [Brockmans et al, 2006]



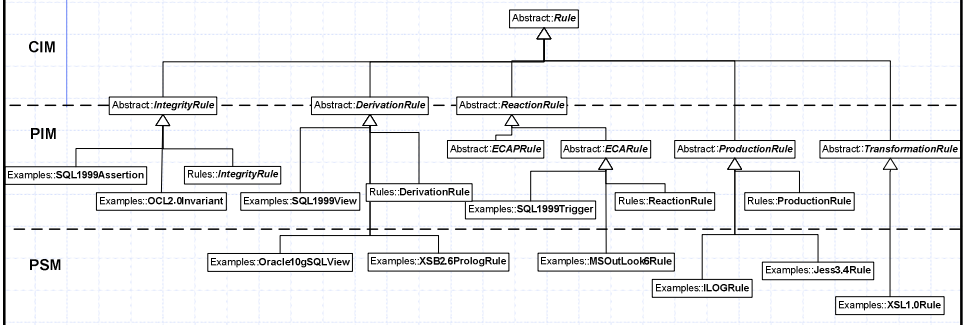
Rule Definition Metamodel Summary

- A good starting point for integrating SWRL and MDA
- Its authors did not develop model transformations or reported on its use
- It was based on non-standard ODM
- Does not satisfy all Semantic Web needs
  - Other types of rules, policies, services, and applications

REWERSE Rule Markup Language (R2ML)

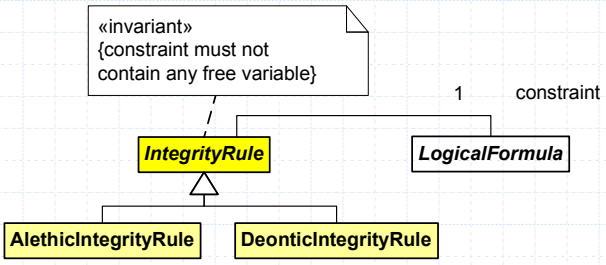
- http://rewerse.net/I1/
- Current version 0.5
- Addresses RIF requirements
- Organization:
  - MOF-based metamodel defining the abstract syntax
  - XML Schema as a concrete XML syntax
  - UML-based Rule Modeling Language (URML) as another concrete (visual) syntax
  - Transformations

R2ML – Rule Concepts



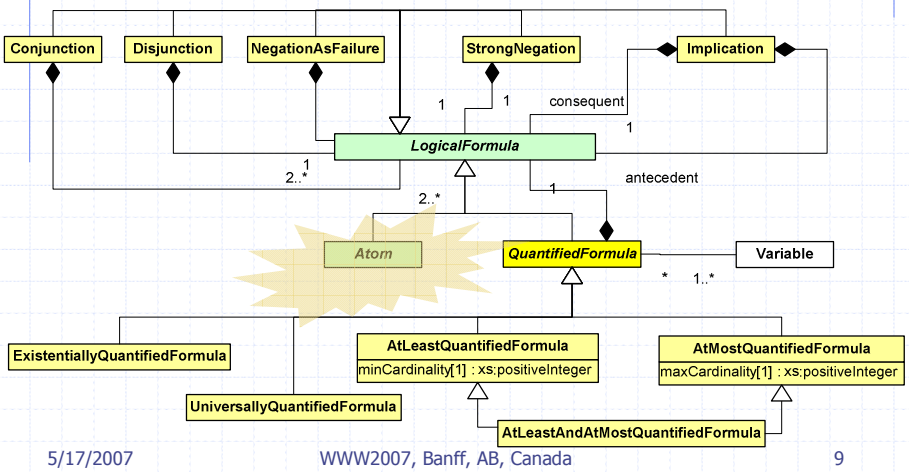
R2ML Integrity Rules

Example: The driver of a rental car must be at least 25 years old



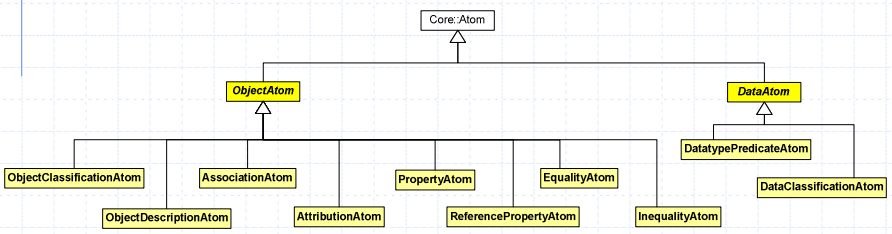
R2ML Integrity Rules

Concept of Logical Formula



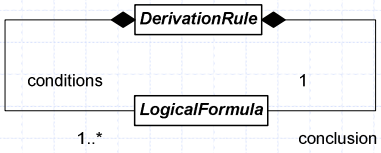
Model-Driven Semantic Web Rule Engineering

- R2ML integrity rules
  - Atoms



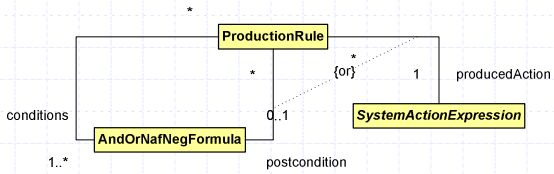
Model-Driven Semantic Web Rule Engineering

- R2ML derivation rules
  - Example: If male is not a husband then the male is a bachelor



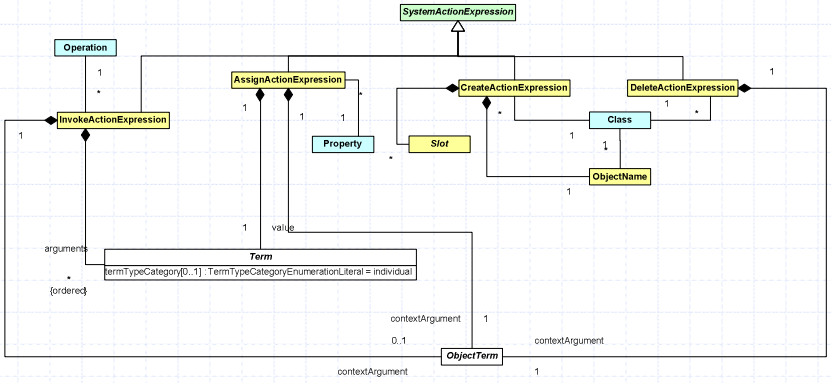
Model-Driven Semantic Web Rule Engineering

- R2ML production rules
  - Example: If customer has no items with type 'CD' in his shopping cart, then add CD link to customer page



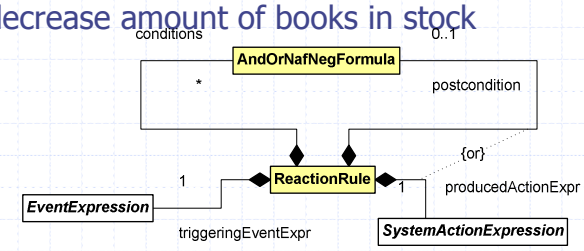
Model-Driven Semantic Web Rule Engineering

- R2ML production rules
  - System action expression



Model-Driven Semantic Web Rule Engineering

- R2ML reaction rules
  - Event-Condition-Action (ECA) rules
  - Example: On customer book request, if the book is available, then approve order and decrease amount of books in stock

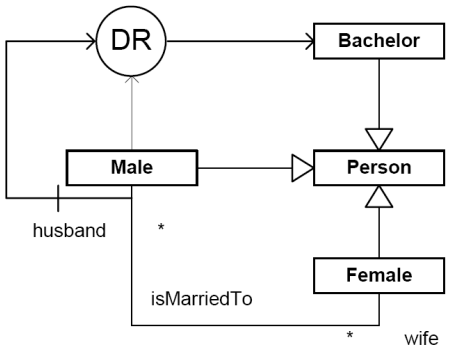


Model-Driven Semantic Web Rule Engineering

- UML-based Rule Language (URML)
  - An extension of UML metamodel
  - Defining rules on top of vocabulary definitions (UML classes)
  - Syntax for derivation, production and reaction rules
    - Integrity rules can be expressed with OCL
  - Developing rules using UML
  - Tool support – Strelka
    - A plug-in for Fujaba
    - Migration to Eclipse is an on going effort

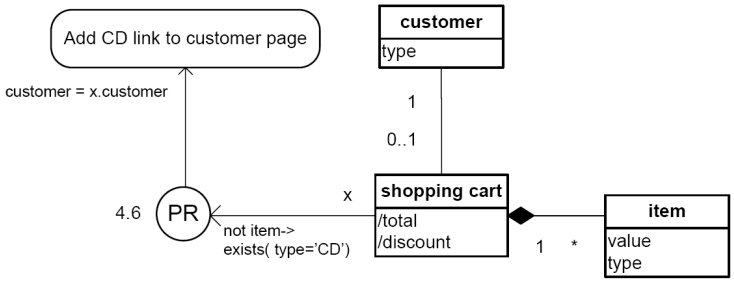
Model-Driven Semantic Web Rule Engineering

- URML derivation rules
  - Example: If male is not a husband then the male is a bachelor



Model-Driven Semantic Web Rule Engineering

- URML production rules
  - Example:
    If customer has no items with type 'CD'
    in his shopping cart, then
    add CD link to customer page



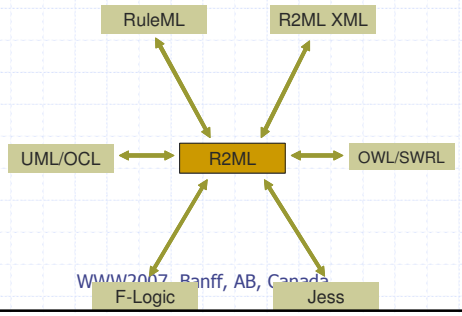
5/17/2

R2ML XML Schema

- Concrete syntax
  - R2ML metamodel has an XMI schema
    - verbose and hard to follow
  - Syntax to be used in R2ML applications
  - Defined as a regular XML schema
  - Vocabulary agnostic
    - any vocabulary can be referred by URI:
      OWL, RDFS, UML, XSD

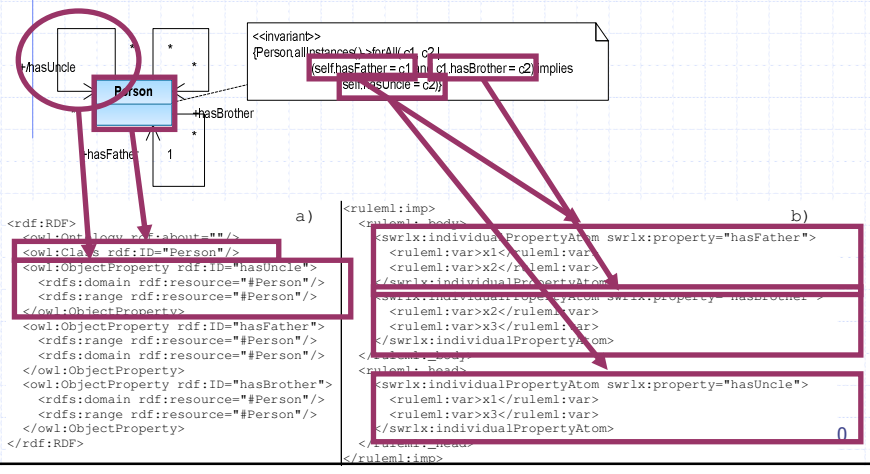
R2ML Transformations

- R2ML as a pivotal metamodel
  - Transformation reusability
    - Number of transformations: 2N instead of N(N-1)
  - ATLAS Transformation Language (ATL), XSLT, and Textual Concrete Syntax (TCS)



R2ML Transformations

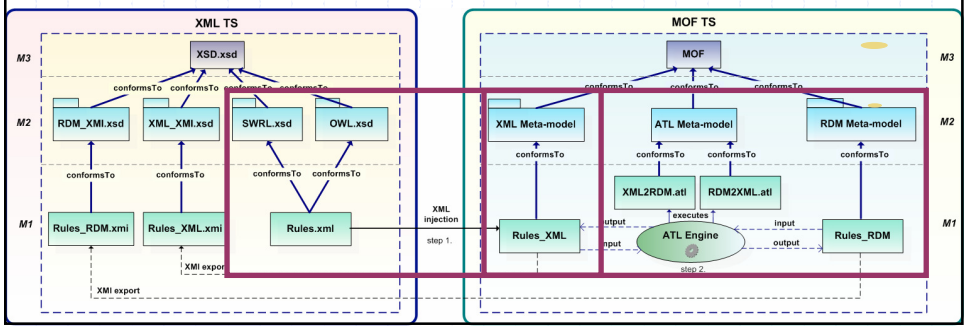
- Example: UML/OCL <-> OWL/SWRL



R2ML Transformations

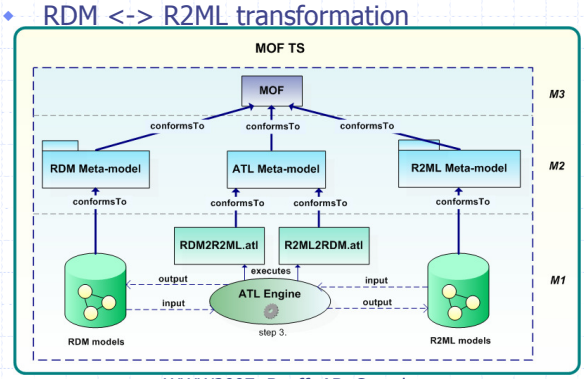
- R2ML metamodel <-> OWL/SWRL
  - Model transformation – ATL (not XSLT)
  - Preparation stage – 2 steps:
  - Injection (automatic) and XML<->RDM ATL

OWL/SWRL



R2ML Transformations

- R2ML metamodel <-> OWL/SWRL
  - main transformation – step 3
    - RDM <-> R2ML transformation



R2ML Transformations

- Transformations
  - http://oxygen.informatik.tu-cottbus.de/rewerse-i1/?q=node/15

R2ML	RuleML	Jess	F-Logic	F-Logic XML	Jena	KAoS	Rei	JBoss	SWRL	OCL
Derivation	⇔	⇒	⇒	⇔	⇒	⇔	⇔	⇒		
Integrity									⇔	⇔
Reaction										
Production	⇔	⇒			⇒					
Transformation Language	XSLT								QVT/ATL	

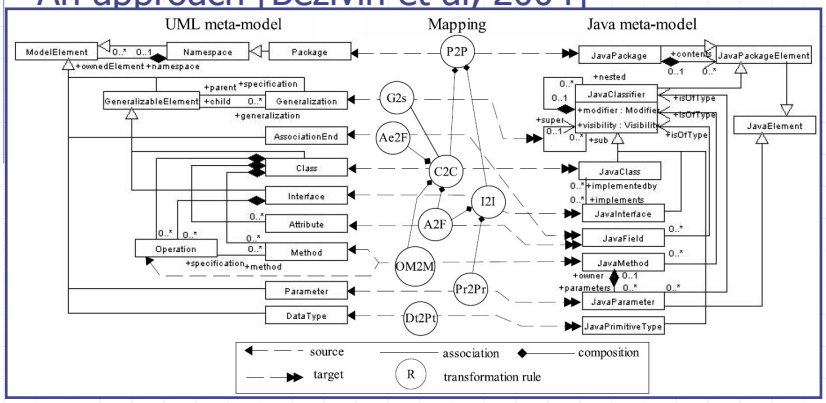
Model-Driven Semantic Web Rule Engineering

- Summary
  - Semantic Web rules is the area that requires a lot of research
    - Impacts the use of MDE principles and way back
      - RIF as a MOF-based metamodel
    - Efforts to use MDE for Semantic Web rules are promising
  - Connecting with relevant OMG's standards
    - UML, ODM, Production Rule Representation (PRR), and Semantics for Business Vocabularies and Rules (SBVR)
  - Connecting rule metamodels with policies, service choreographies, and applications
  - Defining the place of rules in software development methodologies
    - Service behavior or service description

Model-Driven Semantic Web Service Engineering

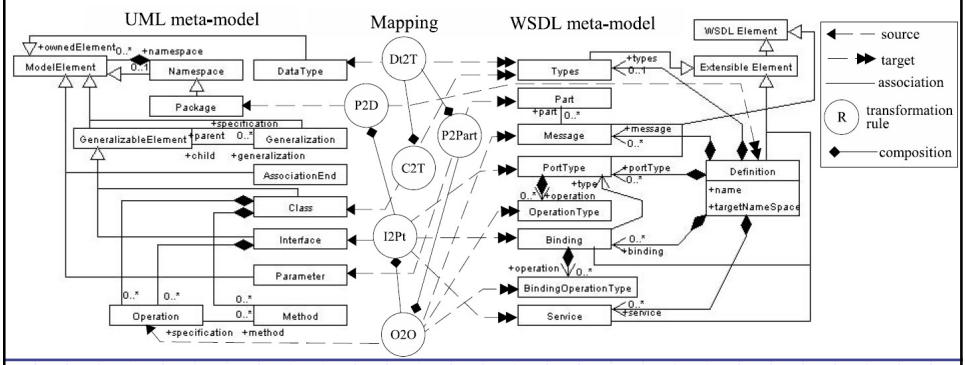
MDA for Web Services

An approach [Bezivin et al, 2004]



MDA for Web Services

An approach [Bezivin et al, 2004]



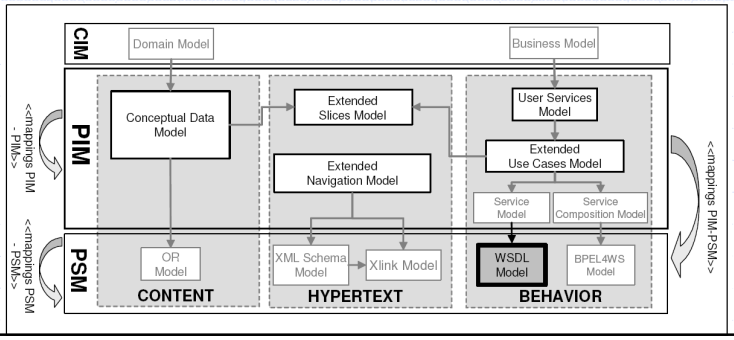
MDA for Web Services

An approach [Bezivin et al, 2004]

- Shortcomings
  - This approach does not provides two way transformations between PIMs and PSM
  - Translation of regular UML class models into WSDL is limited
    - Unless one defines some UML patterns for modeling Web services, OCL constrains or extend UML for Web services

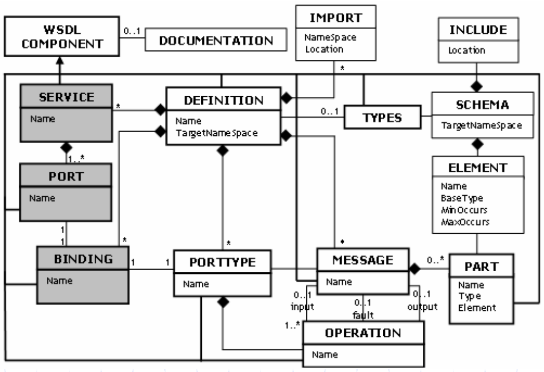
MIDAS-CASE

- Web information system development [Vara et al, 2005]
  - MIDAS-CASE tool supports the whole MDA model chain
    - it also supports defining CIMs, PIMs, and PSMs
  - A metamodel for WSDL and its corresponding UML profile for modeling of Web services (PSM)
  - Automatic generation of the respective WSDL description for Web services modeled

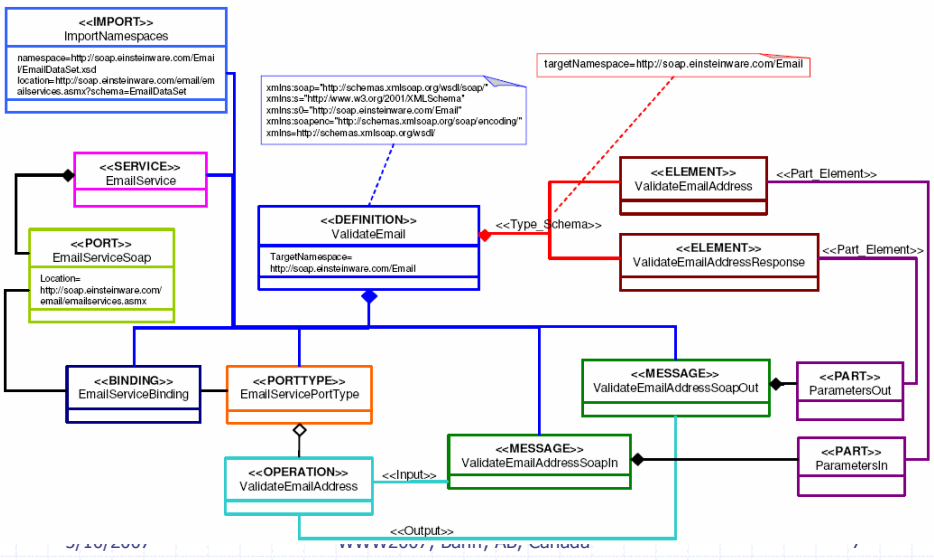


MIDAS-CASE

- Web information system development [Vara et al, 2005]



MIDAS-CASE



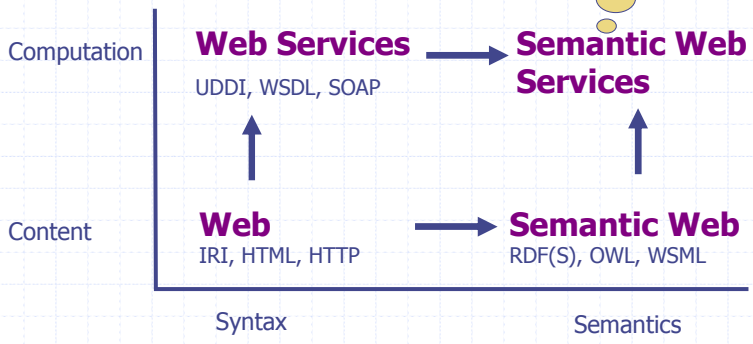
MIDAS-CASE

- Web information system development [Vara et al, 2005]
  - Some shortcomings
    - Generation of PSMs from PIMs is dependent on a definition of use cases and service compositions in which a service is used
    - PSM behavior is not modeled

Semantic Web Se

Automation of service discovery, composition, invocation, and monitoring

- Evolution of the Web into a Se infrastructure



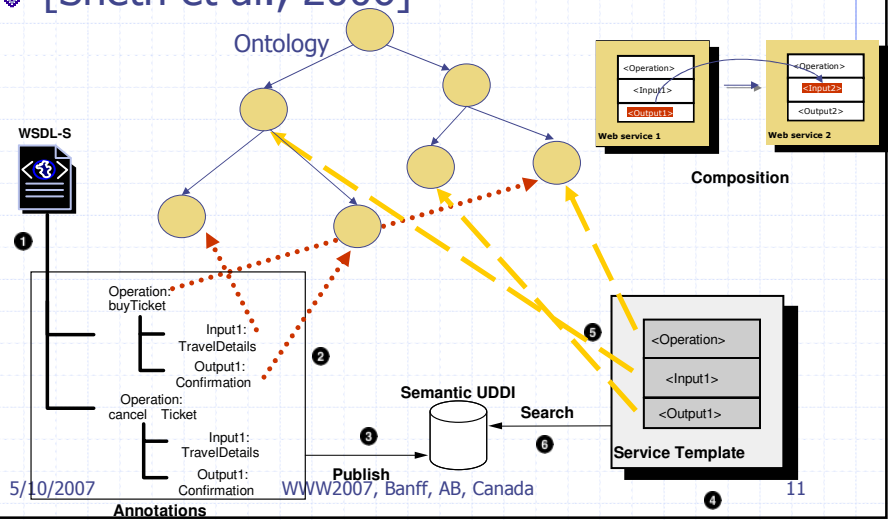
Semantic Web Services

- SWS descriptions languages
  - Semantic Annotations for WSDL and XML Schema (SAWSDL) — W3C® Candidate recommendation
    - Stems from Web Service Semantics (WSDL-S)
  - W3C Submissions
    - Ontology Web Language for Services (OWL-S)
    - Web Service Modeling Ontology (WSMO)
    - Semantic Web Service Ontology (SWSO)

WSDL-S : An Extension of WSDL

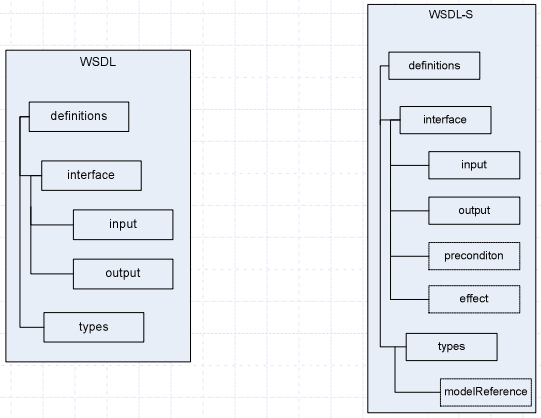
http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/WSDL-S-W3C-ppt.ppt

- [Sheth et al., 2006]



Semantic Web Services

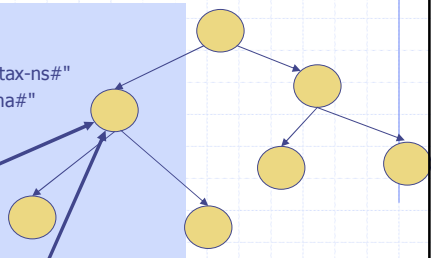
- WSDL-S extensions



SAWSDL

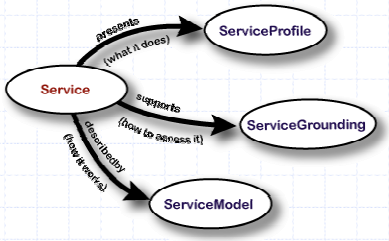
- **modelReference**
  - To specify the association between a WSDL or XML Schema component and a concept in a semantic model
  - To annotate XML Schema complex type definitions, simple type definitions, element declarations, and attribute declarations as well as WSDL interfaces, operations, and faults
- **liftingSchemaMapping** and **loweringSchemaMapping**
  - added to XML Schema element declarations, complex type definitions and simple type definitions for specifying mappings between semantic data and XML
  - mappings can be used during service invocation
- Tools
  - SAWSDL Editor (WSMO Studio)
    - http://www.ontotext.com/wsmostudio/demo/sawsdl.htm
  - Radiant (annotation tool) and Lumina (discovery and matching)
    - http://lsdis.cs.uga.edu/projects/meteor-s/SAWSDL/#anc0
  - SAWDL4J
    - http://knoesis.wright.edu/opensource/sawsdl4j/


```
<wsdl:description>
  <rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.w3.org/2002/ws/sawsdl
   /spec/ontology/purchaseorder#">
   <owl:Class rdf:ID="OrderRequest"/>
   <owl:ObjectProperty rdf:ID="has_items">
    <rdfs:domain rdf:resource="OrderRequest"/>
    <rdfs:range rdf:resource="Item"/>
   </owl:ObjectProperty>
   <owl:Class rdf:ID="Item"/>
  </rdf:RDF>
  <wsdl:types>
   <xs:element name="OrderRequest"
    sawsdl:modelReference="...#OrderRequest">
     ...
   </xs:element>
   ...
  </wsdl:types>
  <wsdl:interface name="Order">
   ...
  </wsdl:interface>
 </wsdl:description>
```



Ontology Web Language for Services (OWL-S)

- An OWL ontology for describing properties and capabilities of Web services

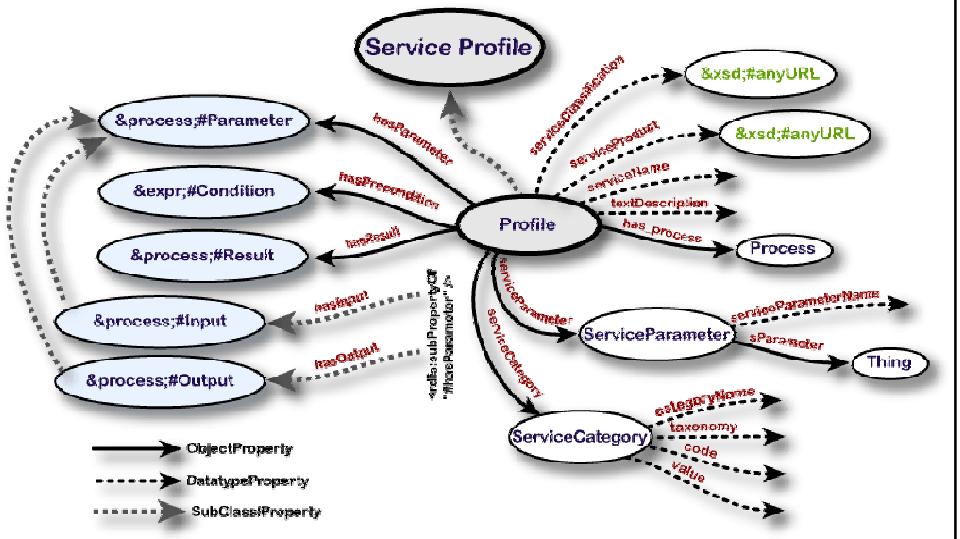


- W3C Submission http://www.w3.org/Submission/OWL-S/

OWL-S Service Profile

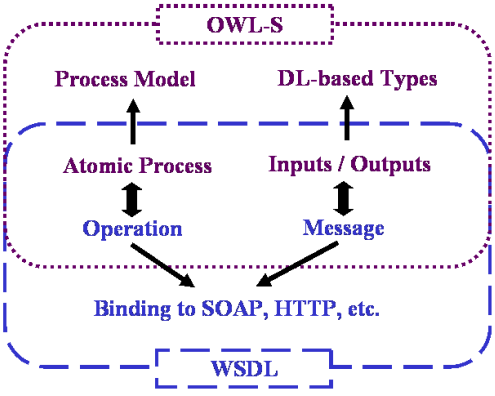
- "What does it do?"
- Populating service registries
- Automated service discovery and matchmaking

OWL-S Service Profile



OWL-S Service Grounding

- "How to access it?"
- Message formatting, transport mechanisms, protocols, serializations of types
- WSDL



OWL-S Tools

- OWL-S Editor
  - http://owlseditor.semwebcentral.org/
  - A Protégé plug-in
- IBM
  - Provides OWL-S API as part of the SNOBASE Semantic Web tool
  - http://www.alphaworks.ibm.com/tech/snobase

Web Service Modeling Ontology (WSMO)

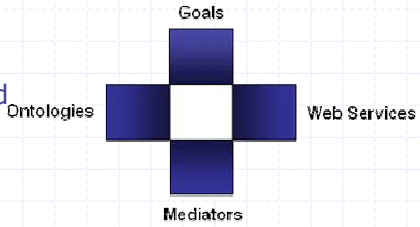
- A conceptual model for Semantic Web Services:
  - Ontology of core elements for Semantic Web Services
  - A formal description language for the conceptual elements (WSML)
    - Description Logics, Logic Programming, First-Order Logic, Frame Logic
    - No OWL or RDF(S)
  - Execution environment (WSMX)
- … derived from and based on the Web Service Modeling Framework WSMF
- Tutorials at http://www.wsmo.org/TR/d17/v0.2/

http://www.wsmo.org/TR/d17/v0.2/

WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Provide the formally specified terminology of the information used by all other components



Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

Connectors between components with mediation facilities for handling heterogeneities

Modeling OWL-S SWS [Timm & Gannod , 2005]

- Learning curve for OWL-S can be steep, providing a barrier to widespread adoption
- Developers to focus on creation of semantic web services and associated OWL-S specifications via the development of a standard UML model
- MDA approach facilitates creation of descriptions of semantic concepts while hiding the syntactic details associated with creating OWL-S definitions

Modeling OWL-S SWS [Timm & Gannod, 2005]

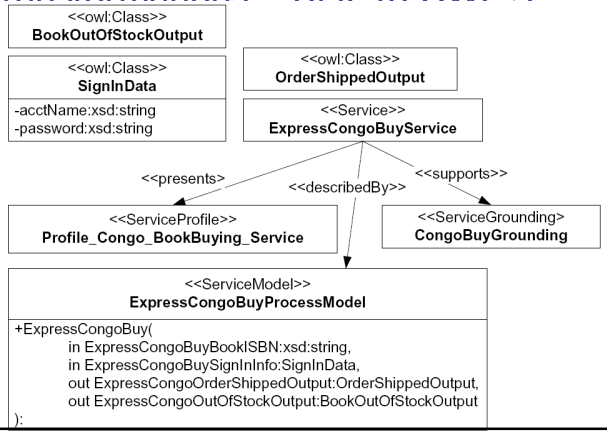
- UML Profile based on OWL-S and WSDL
- XSLT Transformations – UML to OWL-S

Example

Modeling OWL-S SWS [Timm & Gannod, 2005]

- UML Profile based on OWL-S and WSDL
- XSLT Transformations – UML to OWL-S

Example



Modeling OWL-S SWS [Timm & Gannod, 2005]

- UML Profile based on OWL-S and WSDL
- XSLT Transformations – UML to OWL-S

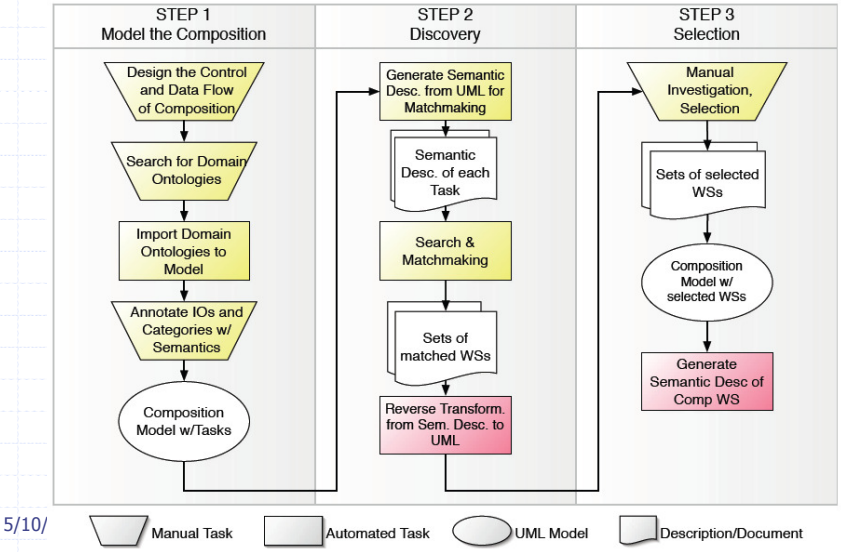
Example

- Shortcomings
  - It is still specific to OWL-S
  - No formal definition in terms of metamodeling
  - XSLT approach is not so reliable for MOF-based models
    - Model transformations are preferable
  - Does not use other relate MDA-based efforts (ODM)

Model-Driven SWS Engineering [Grønmo, Jaeger, & Hoff, 2005]

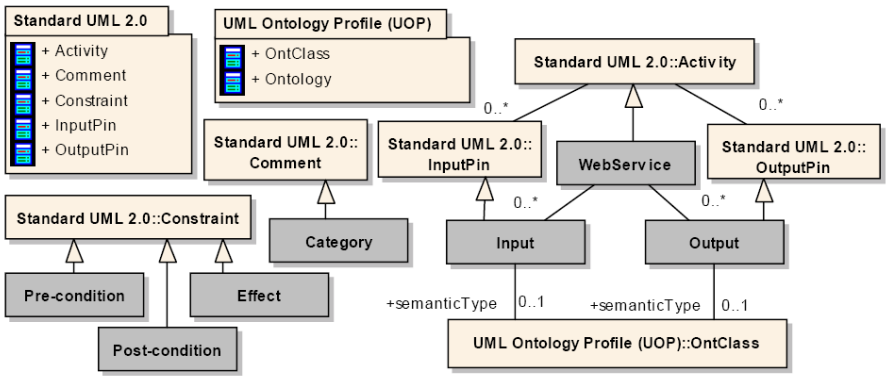
- OWL-S and WSMO are low-level and hard to use even for experienced Web service developers
- MDA increases reusability by independence of the lexical semantic Web service languages
- Models are easy to understand, interpret and specify for experienced modelers

Model-Driven SWS Engineering [Grønmo, Jaeger, & Hoff, 2005]



Model-Driven SWS Engineering [Grønmo, Jaeger, & Hoff, 2005]

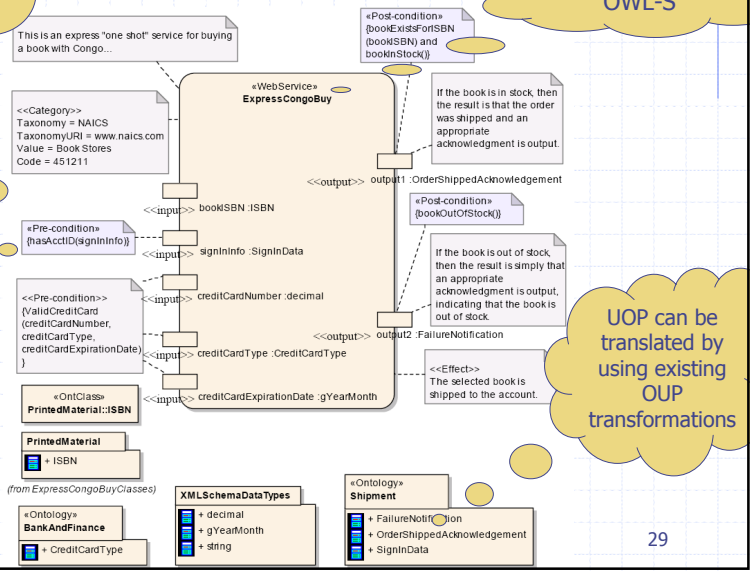
- UML Profile based on OWL-S and WSMO
- Transformations between UML and OWL-S



OCL constraints can be translated to SWRL using model transformations via R2ML

Driven SWS Engineering
o, Jaeger, & Hoff,

This is translated to OWL-S



UOP can be translated by using existing OUP transformations

Model-Driven SWS Engineering [Grønmo, Jaeger, & Hoff, 2005]

- Improved characteristics
  - Model Semantic Web languages analyzed
  - Both service descriptions are service compositions are supported
  - Relies on other relevant efforts (OUP)
  - Extensions are descried in a form of a metamodel (not tested though)
- Still, some shortcomings
  - Transformations are not done at the level of abstract syntax, but at the level of a concrete syntax (by using XSLT)
  - Service-oriented constructs are difficult to connect to business process models
    => Does not follow full MDA chain

UML-Based Rules for Web Services

- **Motivation**: There is still no high-level approach to modeling systems under study, which should be supported by Web services
  - Instead, developers mainly focus on platform specific and implementation details
- There is a need for automatic mechanisms for updating Web services based on the business process changes
  - This is due to the fact that business systems are highly-dynamic and may change quite often

UML-Based Rules for Web Services

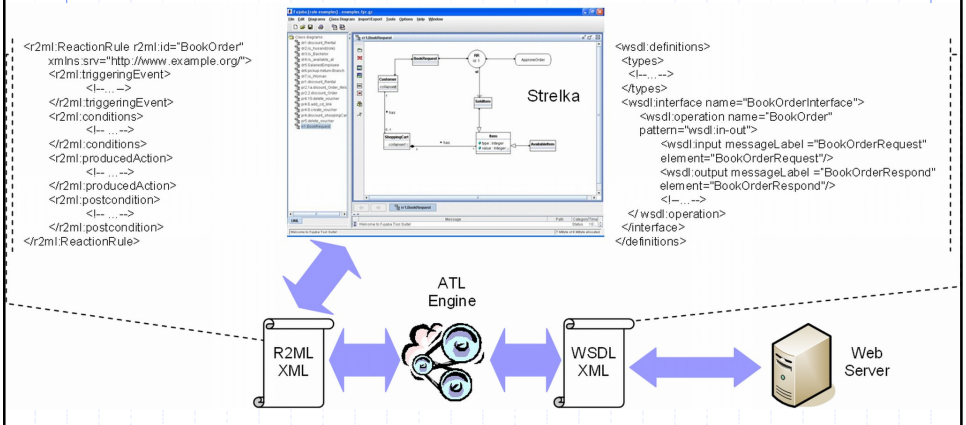
- Basic idea
  - Describe behavior of Web services by means of reaction rules
  - URML for modeling web services
  - Apply rule modeling techniques developed in REWERSE Working Group I1 for designing Web services

Advantages of Using Rules

- Business requirements are often captured in the form of rules in a natural language ("business rules"), formulated by business people
- The topic of rules validation and verification is well-studied
- Reaction rules
  - a flexible way to specify control flow and integrates events/actions from the real life
  - easier to maintain and integrate with other kinds of rules, used in business applications
    - integrity rules and derivation rules

Rule-based Web Services Modeling

- From modeling to the execution platform



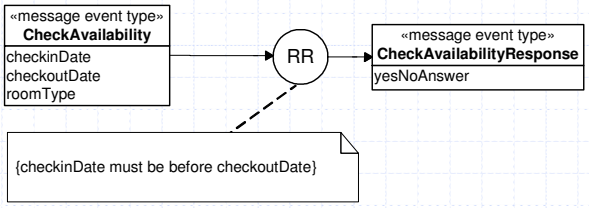
In-Out pattern: Fault Replaces Message

ON CheckAvailability[input](checkinDate, checkoutDate)
IF checkinDate < checkoutDate AND isAvailable(Room)
THEN DO CheckAvailabilityResponse[output]("YES")

ON CheckAvailability[input](checkinDate, checkoutDate)
IF NOT checkinDate < checkoutDate THEN
DO InvalidDataError[outfault](
"Check-in date is more than check-out date")

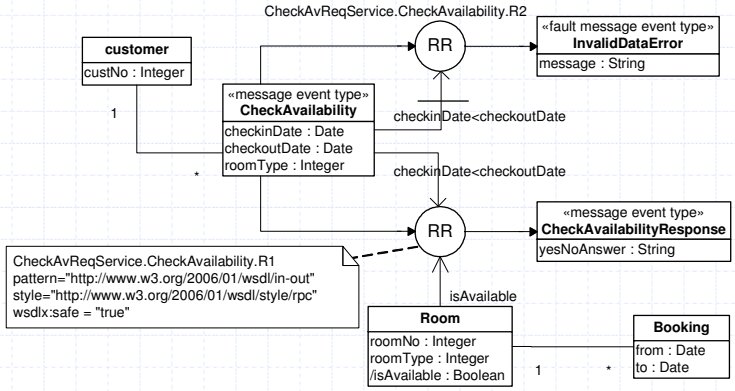
Representing MEPs with URML

- CIM of the In-Out MEP



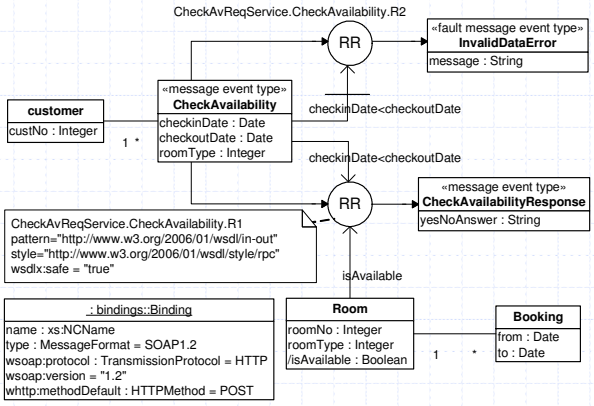
Representing MEPs with URML

- PIM of the In-Out MEP



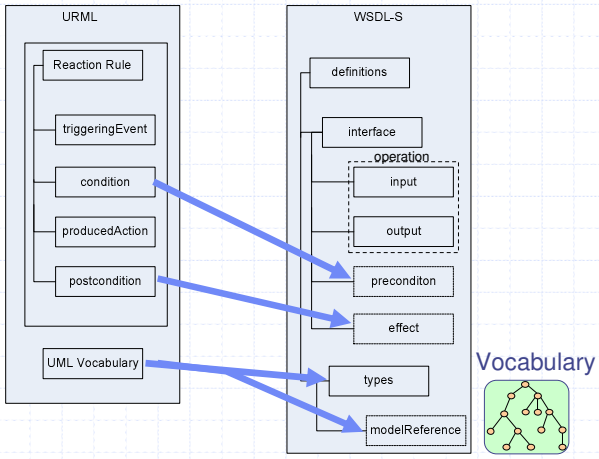
Representing MEPs with URML

- PSM of the In-Out MEP



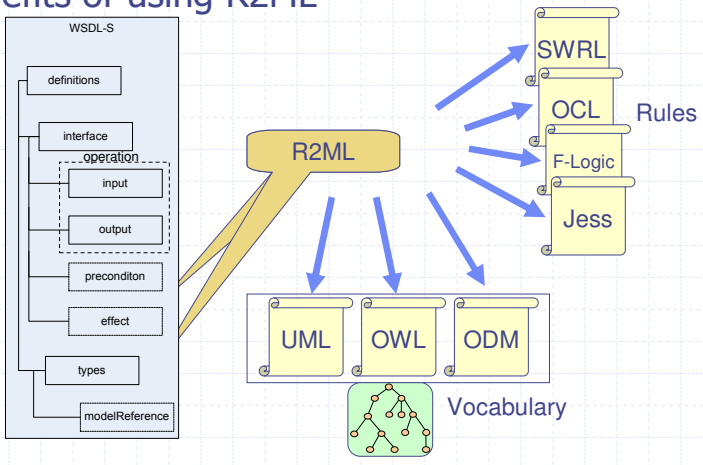
URML for Semantic Web Services

- URML vs. WSDL-S mappings



URML for Semantic Web Services

- Benefits of using R2ML



Model-Driven Semantic Web Services Engineering

- Summarizing
  - Several attempts to apply MDE principles to model (Semantic) Web services
    - Metamodels, UML Profiles, transformations
  - Though the current approaches are promising there are many research challenges to apply MDE for
    - Further use of rules for SWS
    - Non-functional characteristics of SWS
      - Security and QoS agreements

Model-Driven Semantic Web Services Engineering

- Summarizing
  - Though the current approaches are promising there are many research challenges to apply MDE for
    - Semantically annotated choreographies of SWS
    - Integration of policies and SWSs and SWS choreographies
    - Behaviors of Web services of SWS
    - Business process integration based on SWS
    - Portability on different (S)WS platforms
      - OWL-S, WSMO, SAWSDL and WSDL (with WS-CDL)

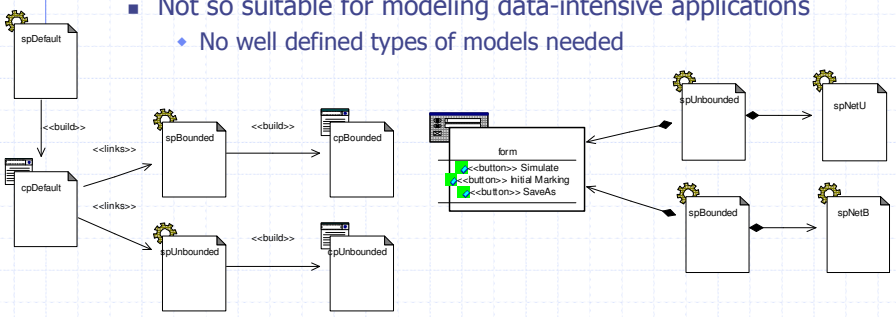
Model-Driven Semantic Web Application Development

Model-Driven Web Application Development

- Some relevant modeling methodologies
  - W2000 [Baresi et al, 2000]
  - OO-HMETHOD [Gomezet al, 2001]
  - UML-based Web Engineering (UWE) [Koch & Kraus, 2004]
  - Object-Oriented Hypermedia Design Model (OOHDM) [Rossi & Schwabe, 2006]
  - Web Site Design Method (WSDM) [De Troyer et al, 2005]
  - Object Oriented Web Solution (OOWS) [Pastor et al, 2006]
  - UML Profile for Web applications [Conallen, 2000]

Modeling Web Applications

- UML Profile for Web Applications [Conallen, 2000]
  - First step towards using the MDE principles
  - There is no formal metamodel definition
  - Not so suitable for modeling data-intensive applications
    - No well defined types of models needed

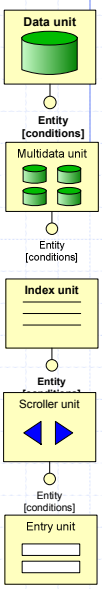


WebML: Modeling Data-Intensive Applications

- Web application design consists of
  - an information (structure) model
    - ER models
  - a hypertext UI model:
    - siteviews with areas and subareas
    - pages
    - page "units"
    - links
  - presentation style definitions

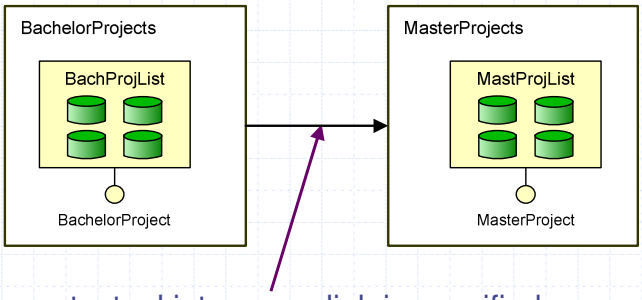
WebML: Basics

- Basic Page Units
  - A *data unit* presents information about a single object
  - A *multidata unit* presents information about a set of objects
  - An *index unit* allows to select an object from a list of objects
  - A *scroller unit* allows to browse an ordered set of objects
  - An *entry unit* allows to enter, query and update information about objects



WebML: Basics

- A Non-Contextual Link Example

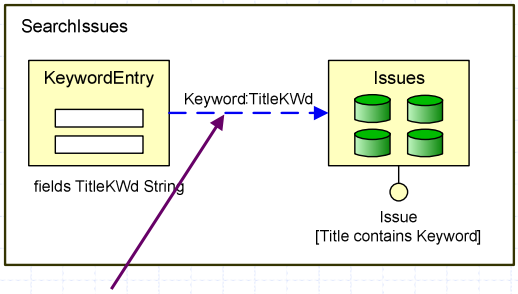


A non-contextual inter-page link is specified as:

```
link Bach2Mast1
(from BachelorProjects to MasterProjects)
```

WebML: Basics

- A Contextual Link between an Entry/Form Unit and a Multidata Unit




```
link KeywordEntry2Issues
(from KeywordEntry to Issues;
 parameters Keyword:TitleKWd)
```


WebML for Modeling Web Services

- Starting points
  - A hypertext model for describing Web interactions
    - Extension to define specific concepts in the model to represent Web service calls
  - Web service invocation is captured by a visual modeling language
    - relationships between invocations and data units, which provide their inputs and capture their outputs
  - Service-enabled Web applications can
    - automatically be derived from WebML diagrams and
    - be run on any platform providing the communication support required for Web service interactions

WebML for Modeling Web Services

- Specification language supports [Manolescu et al, 2005]
  - Workflow patterns
  - Exchange of messages with Web services in both synchronous and asynchronous manner, considered from the perspective of the end-user
    - synchronous is currently the most used
    - asynchronous the most promising in terms of future development of service-enabled Web applications
  - Duality - the ability to represent both:
    - application calls to Web services
    - deployment of applicative functions in the form of Web services

WebML for Modeling Web Services

- WebML hypertext specification extension for Web services
  - Operation categories that involve one message
    - *one-way* operation
      - initiated by the client of the service
      - consists of an input message
    - *notification* operation
      - initiated by the service
      - consists of an output message sent to the client

WebML for Modeling Web Services

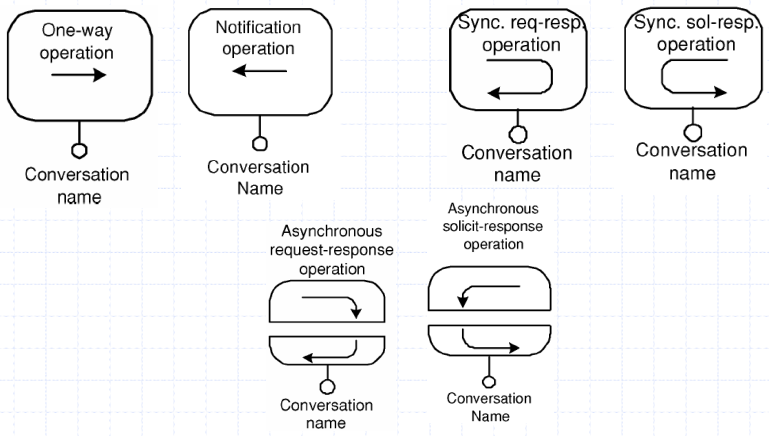
- WebML extension for Web services
  - Operation categories that involve a message exchange
    - *request-response* operation
      - initiated by the client
      - has one input message, followed by one output message
    - *solicit-response* operation
      - initiated by the service
      - has one output message directed to a client, followed by one input message returned from the client

WebML for Modeling Web Services

- WebML extension for Web services
  - Data marshaling and unmarshaling
    - Conversion between the ER representation and XML and between different XML representations

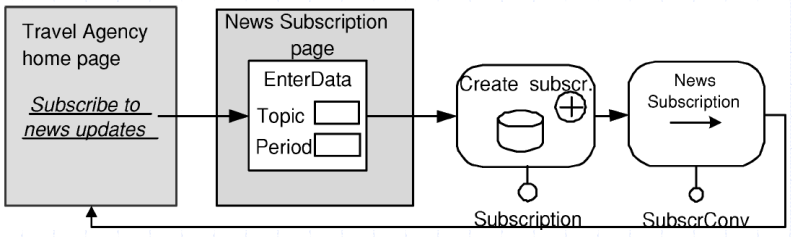
WebML for Modeling Web Services

- New WebML primitives – messages



WebML for Modeling Web Services

- Example: One-way operation in WebML

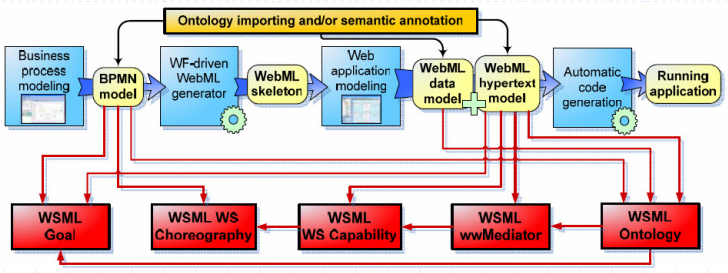


WebML for Modeling Web Services

- Implementation
  - Tools
    - WebRatio, http://www.webratio.com/
      - XSLT-based transformations
      - Different platforms for actions (Java and C#) and pages (JSP and ASP.NET)

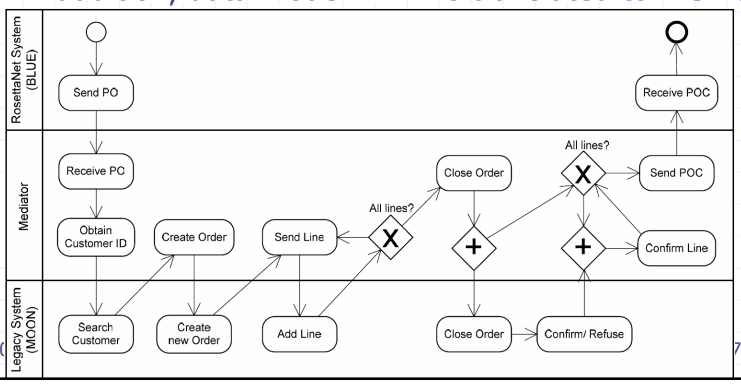
Modeling Semantic Web Service Applications

- Extension of the WebML approach [Brambilla et al, 2006]



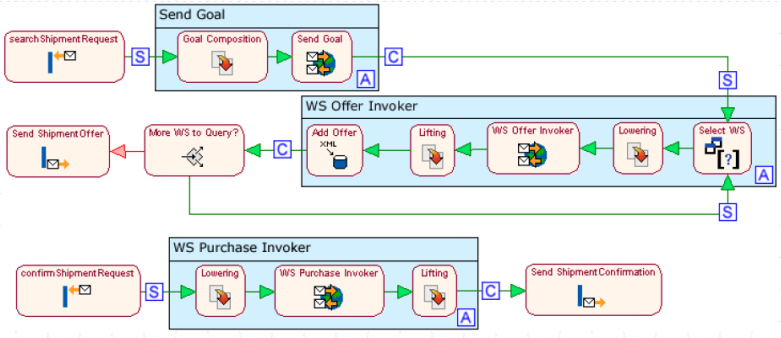
Modeling Semantic Web Service Applications

- Business Process Modeling Notation (BPMN)
  - Computation-independent model of choreography
  - Translation to hypertext model
  - In addition, data model in ER is translated to WSMO



Modeling Semantic Web Service Applications

- Semantic Web service in WebML
  - Extraction of WSMO Semantic Web Services



WebML for Modeling Web Services

- Open research challenges
  - WebML is not based on MOF or ECore technologies
    - It is not in the same technical space as other MDE technologies
    - The use of standards relevant
  - Metamodel for WebML and model transformations
    - Attempts: metamodel [Schauerhuber et al, 2006] and UML2 Profile [Moreno et al, 2006]
  - There is no connections with other relevant MDE efforts such as ODM or UML profiles for ontology/rule modeling
  - WebML is fully based on E-R models and databases
    - Other types of information models
    - Only supports WSML, but not OWL
  - Rules to be considered
    - Preconditions, postconditions, effects, and assumptions

WebML for Modeling Web Services

- Open research challenges
  - Security, QoS and Policies
    - General open challenges of SWS
    - Relevant standards as well WS-Trust, WS-Federation, XACML etc.
  - WSMO is only supported of SWS approaches for application development
    - SAWSDL and OWL-S
    - Other (semantically annotated) choreography languages WS-CDL
  - Error handling for Semantic Web services in application Web applications
  - Modeling Message Exchange Patterns (MEPS) besides workflow patterns
  - Development of services depended on their use in a specific application might not be sufficient