INF5120 – Model-based System Development

Lecture 14

April 29th, 2013
Arne J. Berre, SINTEF ICT
Contents

- System Architecture models
- ADM – Architecture Driven Modernisation
- SBVR – OMG standard, semantics of business vocabulary and business rules
- MDI – Model Driven Interoperability
- Comparison Semantic mappings
- Conclusion & Further work
1 (14/1): Introduction – overview Enterprise Architecture with UML and BPMN and DSLs
2 (21/1): Service Innovation and Design, AT ONE method/workshop – myServiceFellow (Marika Lüders)
3: (28/1): Value Networks/VDML BPMN, vs. UML Activity diagrams - Oryx
5 (11/2): UML and Req.Modeling – Agile User stories versus Use cases
6 (18/2): Business-SoaML, Requirements Modeling, Goal Modeling, BMM, and Non Functional requirements
7 (25/2): Model driven engineering – Metamodels, DSL, UML Profiles etc. NAV on NFR
9 (11/3): Method Engineering, SW Process frameworks, SPEM/EPF, ISO 24744, FACESEM/ESSENCE (Brian Elvesæter)
10(18/3): Model driven engineering, transformation technologies,
11(8/4): MDE and DSL in practice, with ThingML example – Franck Fleurey
12(15/4): System Architecture and Information/Ontology modeling, UML, ISO 19103
13(22/4): UI Models, WebML and IFML, Process models (WebRatio, Cordys Cloud)
14(29/4): System realisation models, Entity-Process-Service-UI (Cordys, MagicDraw, JEE), BPMN and CMMN, SBVR and fact models, MDA-ADM, MDI
15(6/5): Platform models for the Cloud, with CloudML, Alessandro Rossini
16(13/5): Conclusion and Summary for INF5120 - Preparation for Exam

Exam: Monday June 3rd, 2013, (4 hours)
### INF5120 – Oblig plan - 2013

<table>
<thead>
<tr>
<th>Date</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (14/1)</td>
<td>Introduction</td>
</tr>
<tr>
<td>2 (21/1)</td>
<td>myServiceFellow/Service Design</td>
</tr>
<tr>
<td>3 (28/1)</td>
<td>Oryx</td>
</tr>
<tr>
<td>4 (4/2)</td>
<td>Balsamiq</td>
</tr>
<tr>
<td>5 (11/2)</td>
<td>Use cases 2.0</td>
</tr>
<tr>
<td>6 (18/2)</td>
<td>Oblig 1 – Group work</td>
</tr>
<tr>
<td>7 (25/2)</td>
<td>EMF and Eclipse – Group (3) presentation – Business Model, Business-SoaML</td>
</tr>
<tr>
<td>8 (4/3)</td>
<td>Oryx– Group (3) presentation - User stories/use cases, UI, NFR</td>
</tr>
<tr>
<td>9 (11/3)</td>
<td>Group presentation(s) on Oblig 1: User stories/use cases, UI/Balsamiq</td>
</tr>
<tr>
<td>10 (18/3)</td>
<td>Group presentation on Oblig 1 – NFR 1 – Parallel Start on Oblig 2 (individual) and Oblig 3 (group)</td>
</tr>
<tr>
<td></td>
<td>- 22/3 : Delivery of Oblig 1</td>
</tr>
<tr>
<td></td>
<td>EASTER</td>
</tr>
<tr>
<td>11(8/4)</td>
<td>Walk through of Oblig 1 – Questions on Oblig 2 and 3</td>
</tr>
<tr>
<td>13(22/4)</td>
<td>Group work, Oblig 2/3  – Group presentation/discussion</td>
</tr>
<tr>
<td>14(29/4)</td>
<td>Group work, Oblig 2/3  – Group presentation /discussion – delivery Oblig 2</td>
</tr>
<tr>
<td></td>
<td>11/5: Delivery of Oblig 3</td>
</tr>
<tr>
<td>16(13/5)</td>
<td>Walk trough of Oblig 3 - Preparation for Exam</td>
</tr>
<tr>
<td></td>
<td>Exam: Monday June 3rd, 2013, (4 hours)</td>
</tr>
</tbody>
</table>
Oblig 3 – Concierge Model-based System Architecture

- System Architecture for Concierge
- Basis for Cloud realisation
- MagicDraw Cameo Enterprise Architecture 17.0.3
- Use of Java, JEE 6.x
  - or use CORDYS Cloud platform
  - or use WebRatio model platform with WebML-IFML

- Information Model – UML Class model
- Service Model – SoaML model
- Process/Behaviour Model – BPMN/Seq.diagrams, …
- UI Model – (ref. also Balsamiq)
Model Driven Enterprise Architecture

- Arne.J.Berre@sintef.no

- OMG standards related to the Zachman framework
# Zachman with OMG standards

<table>
<thead>
<tr>
<th>Scope (Contexts)</th>
<th>Business (Concepts)</th>
<th>System (Logic)</th>
<th>Technology (Physics)</th>
<th>Component (Assemblies)</th>
<th>Operation (Instances)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data (What)</strong></td>
<td><strong>Function (How)</strong></td>
<td><strong>Network (Where)</strong></td>
<td><strong>People (Who)</strong></td>
<td><strong>Time (When)</strong></td>
<td><strong>Motivation (Why)</strong></td>
</tr>
<tr>
<td>List of things important to business</td>
<td>List of processes that the business performs</td>
<td>List of locations which the business operates</td>
<td>List of organizations important to the business</td>
<td>List of events/cycles important to the business</td>
<td>List of business goals/strategies</td>
</tr>
<tr>
<td>SBVR</td>
<td>VDM</td>
<td>VDM</td>
<td>OSM</td>
<td>DTFV</td>
<td>BMM</td>
</tr>
<tr>
<td><strong>Semantic Model</strong></td>
<td><strong>Business Process Model</strong></td>
<td><strong>Business Logistics System</strong></td>
<td><strong>Workflow Model</strong></td>
<td><strong>Master Schedule</strong></td>
<td><strong>Business Plan</strong></td>
</tr>
<tr>
<td>ODM, IMM (CWM)</td>
<td>BPMN, CMPM</td>
<td>BPMN, CMPM</td>
<td>OSM, BPMN, CMPM</td>
<td>BPMN, CMPM</td>
<td>SBVR</td>
</tr>
<tr>
<td><strong>Logical Data Model</strong></td>
<td><strong>Application Architecture</strong></td>
<td><strong>Distributed System Architecture</strong></td>
<td><strong>Human Interface Architecture</strong></td>
<td><strong>Control Structure</strong></td>
<td><strong>Rule Design</strong></td>
</tr>
<tr>
<td>ODM, IMM (CWM), UML</td>
<td>SoaML, UML</td>
<td>SoaML, UML</td>
<td>SoaML, UML</td>
<td>BPMN, CMPM</td>
<td>SBVR</td>
</tr>
<tr>
<td><strong>Physical Data Model</strong></td>
<td><strong>System Design</strong></td>
<td><strong>Technology Architecture</strong></td>
<td><strong>Presentation Architecture</strong></td>
<td><strong>Timing Definition</strong></td>
<td><strong>Rule Definition</strong></td>
</tr>
<tr>
<td>IMM (CWM), UML</td>
<td>SoaML, UML</td>
<td>SoaML, UML</td>
<td>SoaML, UML</td>
<td>DTFV</td>
<td>SBVR</td>
</tr>
<tr>
<td><strong>Data Definition</strong></td>
<td><strong>Program</strong></td>
<td><strong>Network Architecture</strong></td>
<td><strong>Security Architecture</strong></td>
<td><strong>Schedule</strong></td>
<td><strong>Strategy</strong></td>
</tr>
<tr>
<td>IMM (CWM), UML</td>
<td>UML</td>
<td>UML</td>
<td>UML</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
OMG standards coverage

Scope (Contexts)
- SBVR
- VDM
- OSM
- BMM

Business (Concepts)
- ODM
- BPMN
- SBVR

System (Logic)
- IMM (CWM)
- SoaML
- CMPM

Technology (Physics)
- UML
- DTFV

Component (Assemblies)
- Data Definition
- Program Architecture
- Security Architecture
- Rule Definition

Operation (Instances)
- Data
- Function
- Network
- Organization
- Schedule
- Strategy

List of things important to business
List of processes that the business performs
List of locations which the business operates
List of organizations important to the business
List of events/cycles important to the business
List of business goals/strategies

Telecom and Informatics
3-tier System Reference architecture

Application Layer:

User Service

Business Layer:

Service

Entity

Data Layer:

DataService

Legacy

UIC

DataService
MagicDraw
Cameo Enterprise Architecture

DoDAF, MODAF, and NAF with UPDM Compliance

Cameo Enterprise Architecture

No Magic has deep experience with DoDAF 2.0, MODAF, NAF 3 and the Defense Industry. Our Cameo Enterprise Architecture product, based on our core product MagicDraw, offers the most robust standards compliant DoDAF 2.0, MODAF and NAF 3 via a UPDM standardized solution. And what’s more, No Magic fully supports all architectural framework products ensuring you achieve mission results. No Magic also leads the industry in its integration of DIA requirements, ensuring that you achieve net-centric success. Meet your interoperability challenges with proven, tested No Magic solutions.

No Magic Specifically Meets DoDAF 2.0, MODAF, NAF 3 and UPDM Needs

Improved Mission Results - Your team will do a better job of mining available data, measuring and visualizing architecture and overall success factors resulting in improved mission results.

- Convey the knowledge faster and easier
- Easily represent and communicate complex architecture
- Reduce assumptions, misconceptions and risk

Program Accountability - Provide Program Manager accountability including the enablement of net-centric processes and architectures, flexibility and responsiveness.

- Meet standards and easily follow guidance
- Understand risk/cost
- Gaps are identified and eliminated

Testimonials

"One of the best, if not the best, object-oriented modeling tools I’ve used is MagicDraw.
Mark Laurence
Labcorp"

"MagicDraw is BY FAR the greatest modeling tool I have ever used.
Stan Butler
Deacisco"

"Thank you very much for your help! I must say that I have never experienced such excellent technical support.
Dr. Jim Stivel
CurtainView Training"
Download

- MagicDraw – Enterprise Architecture

- Download MagicDraw from NoMagic full EA version
  - Cameo Enterprise Architecture 17.0.3 FR Enterprise Edition
  - (Full license file until September 1, 2013 will be provided to each student)

- See many MagicDraw demonstrations/tutorials here:
  - [http://www.nomagic.com/support/demos](http://www.nomagic.com/support/demos)
  - Including demo of UI modeling and code generation!
JEE: Java Enterprise Edition

- Java for 3 tiers architectures
- Includes
  - Java Libraries (Mail, Protocols, etc.)
  - Tools, compilers, code generators,
  - Application Server
- JEE 6.x
References

- SiSaS Methodology Wiki
  - http://sisas.modelbased.net/

- JEE 6.0
  - First cup of JEE 6
    - http://docs.oracle.com/javaee/6/firstcup/doc/
  - Official JEE 6 tutorial
    - http://docs.oracle.com/javaee/6/tutorial/doc/
Cordys BOP 4.1 Fundamentals - Learning materials

Using the links below, you can go through all the training materials to learn more about the Cordys Platform. Although most modules can be used independently and have their own prerequisites listed, we advise you to work through the materials from top to bottom. Each module has a link to the course exercises ( ) in the section on Exercises.

Next to that, you will find a Presentations section in which you can find the presentation slides ( ) and a Theory section ( ) giving you the theory behind each presentation.
WebML models and concepts
- WebRatio model based development -
Preview of WebML concepts

- Site = Content + Composition + Navigation + Presentation
ADM – Architecture-Driven Modernization (Reverse MDA)

- Standards on Architecture-Driven Modernization
- Solutions Directory
  - [www.systemtransformation.com/transfor.htm](http://www.systemtransformation.com/transfor.htm)
- Modernization Process Framework & Articles
  - [www.systemtransformation.com/it_architecture_transformation.htm](http://www.systemtransformation.com/it_architecture_transformation.htm)
- Reference Book
  - Note: New book on case studies (Ulrich / Newcomb) in the works for 2010 from Elsevier / Morgan Kaufmann / OMG Press
Modernization horse shoe model
Modernization scenarios
ADM Framework

Enterprise Assessment: Planning Level analysis that is broad in scope and light on depth

- Enterprise Analysis Planning
- Technical Architecture Assessment
- Modernization Planning
- Business Architecture Assessment
- Data Architecture Assessment

Project-Level Assessment: Detailed analysis to support refactoring & transformation

- Objective Setting/Proposal Development
- Technical Assessment
- IT Infrastructure Assessment
- Integration/Testing Analysis
- Interim Support Plan
- Strategic Modernization Plan

Refactoring: Applies dramatic improvements to existing IT architecture

- Code Stabilization
- Application Staging
- Language Change upgrade
- Remodularization
- Middleware Enabling
- Validation
- Data Definition Standardization

Transformation: Redeploy essence of existing system within a new target architecture

Any combination of tasks/subtasks may be combined to create given project scenario.
ADM Standards in OMG

- Business Modeling Standards
- SwA Evidence
- ADM Pattern Recognition
- SW Metrics
- KDM
- ASTM

Existing Solution to Target Solution
KDM & ASTM

#1: ADM: Knowledge Discovery Meta-Model (KDM)
- Established metamodel that allows modernization tools to exchange application meta-data across applications, languages, platforms and environments
- Foundation for subsequent ADM standards
- KDM adopted in 2006 and will become ISO standard in 2009
- Deployed by several vendors and service companies

#2: ADM: Abstract Syntax Tree Meta-Model (ASTM)
- Builds upon KDM to represent software at very granular level
- ASTM supports automated system to system transformations, generally involving language and platform changes
- ASTM passed the architecture board in 2008 is being rolled out in 2009
Pattern Recognition and SMM

#3: ADM: Pattern Recognition
- Pattern Recognition will facilitate examination of structural meta-data with the intent of deriving patterns and anti-patterns about existing systems
- To be used to determine refactoring and transformation requirements and other alignment opportunities
- Analysis could be extended to business architecture
- Discussions proceeding in Fall of 2009

#4: ADM: Structured Metrics Metamodel (SMM)
- SMM is a general purpose metrics standard that supports multiple types of metric libraries
- Software Evidence, ADM Patterns Recognition, Sustainability Metamodel and Business Architecture all considering use of SMM
- Passed in 2008 and will be finalized in 2009
Patterns: From Analysis to Implementation

Analysis (Domain) Patterns

Architecture Patterns (Macro Architecture)

Design Patterns (Micro Architecture)

Domain Framework

Implementation

Idioms (Language dependent patterns)

( OO) Reusable Components
Visualisation, Refactoring and Transformation

#5: ADM: Visualization
- Focuses on ways to depict application meta-data stored within the KDM
- Work has not proceeded on this to date

#6: ADM: Refactoring
- Refactoring defines ways in which the KDM can be used to refactor applications
- Envisions metamodel to metamodel transformations
- There is no target date for this work

#7: ADM: Transformation
- Transformation defines mappings between the KDM / ASTM and transformed versions of those metamodels
- Defines mappings and transformations that may occur between existing applications and top down, target models
SINTEF projects

- REMICS (REuse and Migration of legacy applications to Interoperable Cloud Services);
  - started in 2010, SINTEF lead, two SMEs and other research partners. Focus on migration to SOA and cloud and developing a CloudML language

- NEFFICS (Networked Enterprise transformation and resource management in Future Internet enabled Innovation Clouds)
  - Started in 2010, SINTEF lead, using SaaS platforms combined with an advanced innovation management software platform
The REMICS approach

Knowledge REMICS KDM
Business Process and Rules Components
SoaML Implementation
UML, U2, TP

Knowledge Discovery
Reverse Engineering

Source code, binaries, documentation, users
Knowledge, configuration files, execution logs and traces

Legacy Artifacts

Model Driven Interoperability

Service mediation for adaptation
SOA and Cloud Computing Patterns applied
Legacy Components Replacement and Wrapping
Design by Service Composition

Target Architecture for Service Cloud platform
SoaML with REMICS extensions for Service Clouds
Links to Business Models

Forward MDA through PIM4Cloud
Model Transformation, Code Generation, Traceability

RESERVOIR, Joyant, Amazon, Google, Microsoft

Models@Runtime for application management
Model Checking, Model-based Testing for validation

Source Architecture

Recover

Validate, Control and Supervise

Service Cloud Implementation
Project concept

- Migration of legacy systems to service clouds
- Model-driven modernization approach

Diagram:
- Requirements engineering
- Model-based testing
- Model metrics
- Agile methodologies
- Models @runtime
- Model-driven Interoperability

Recovery → Migration
REMICS challenges regarding cloud

- Whether to have private or public or hybrid cloud. Business models and scalability concerns.

- Which cloud platform to choose?

- What does it mean for the architecture? Data handling (reporting system), authorization for access to data, maintenance and adaptation of the systems

- Developing a platform independent model for cloud that hides technologies (and related standardization activities such as PIM4Cloud, Interoperability for cloud)
REMICS Metamodel extensions

REMICS KDM Extension

BPMN \(\rightarrow\) SBVR \(\rightarrow\) BMM \(\rightarrow\) ODM

CIM

SoaML

PIM 4 Cloud \(\rightarrow\) OCL \(\rightarrow\) UML 2 Test Profile

PIM

REMICS PIM4ServiceClouds

PIM 4 Service Interoperability \(\rightarrow\) PIM 4 Models @ Runtime

PSM

UML Profiles for Web Services (XSD, WSDL, BPEL), OGSA and Service Cloud Platforms

Deployment Platforms

RESERVOIR, SUN, Joyant, Amazon, Google, Microsoft, Cordys
How does it work?

Recovery
- Code
- Reverse Engineering
- UML Model
- Componentization
- SoaML Model

Migration
- UML and SoaML models
- SOA and Cloud patterns application
- Service Mediation
- PIM4Cloud, deployment generation
- Service Cloud Application

Validation and Control
- UML, SoaML, PIM4Cloud models
- Model-based Testing
- Model metrication
- Models@Runtime
- Performance measurement
Involved tools

Recovery
- Netfective BluAge reverse engineering tool
- WUT Tale GUI recovery tool
- SOFTEAM Modelio reverse engineering tools
- SOFTEAM Architecture Componentization Tool for Modelio

Migration
- SOFTEAM Patterns Designer Tool – SOA and Cloud patterns application
- SINTEF Interoperability Framework
- WUT RedSeeds – application generation tools
- SINTEF, SOFTEAM PIM4Cloud modeling and deployment generation
- UT Desktop to Cloud Migration tool

Validation and Control
- FOKUS!MBT
- FOKUS!Metrino
- WUT RSL Testing Tool
- Open source libraries for Models@Runtime
- Framework for performance measurement of 3-tier applications
Tools involved

- BluAge Reverse Engineering tool
- WUT Tale GUI recovery tool
- Modelio, PIM4Cloud, Componentization and Patterns Designer
- SINTEF, PIM4Cloud DSL
- FOKUS!Metrino
- FOKUS!MBT
- WUT, RedSeeds
- UT Desktop to Cloud Migration tool
- RSL Testing Tool
- Models@Runtime
- Interoperability Framework
SBVR
Brief Introduction into
Semantics of Business Vocabulary and Business Rules (SBVR)

OMG presentation from Nikolai Mansourov
CTO, Hatha Systems
http://www.hathasystems.com
Agenda

- Introduction to SBVR
- Overview of SBVR Structure
- Community – context for meaning
- Business Vocabulary
- Business Rules
- Semantic Formulation
- What next?
EU-Rent Case Study

- EU-Rent is a (fictitious) car rental company, used to provide coherent examples in SBVR (and in this presentation)

- The business requirements are fairly simple:
  - EU-Rent operates in several countries; in each country it has local areas containing branches
  - EU-Rent rents cars to customers from branches; one-way rentals are allowed
  - Rentals may be booked in advance or “walk-in”
  - Cars are owned by local areas and stored at branches
  - Each car is of a given model; car models are grouped into car groups; all the cars in a car group have the same rental tariff
  - Cars are serviced at 5,000 mile intervals
  - EU-Rent notes “bad experiences” with drivers (police action, unpaid parking fines, cars damaged or not returned to EU-Rent branches, etc) and may bar drivers who cause them.
EU-Rent samples: vocabulary

car movement
Definition: planned movement of a rental car of a specified car group from a sending branch to a receiving branch.
Reference Scheme: movement-id
Description: A car movement meets the business requirement that a car of a given group has to be moved between branches (“we need to move a full-size car from the London City branch to the Heathrow Airport branch”). A specific car will be assigned to it at some time, not necessarily when the requirement is first identified.
Note: car movements play roles in both ‘rental’ and ‘car transfer’ and car movements are scheduled in these roles.

car movement has movement-id
Necessity: Each car movement has exactly one movement-id.

car movement has receiving branch
Necessity: Each car movement has exactly one receiving branch.

car movement has sending branch
Necessity: Each car movement has exactly one sending branch.

car movement specifies car group
Synonymous Form: car group is specified in car movement
Necessity: Each car movement specifies exactly one car group.
EU-Rent samples: vocabulary

Figure E.3 - Car Movements
EU-Rent samples: rule

E.2.2.2.3 Rule Set -- Charging / Billing / Payment Rules

It is permitted that a rental is open only if an estimated rental charge is provisionally charged to a credit card of the renter that is responsible for the rental.

- Guidance Type: operative business rule
- Description: While a renter has possession of a car, there is a provisional charge to EU-Rent against his credit card. This will be replaced by an actual charge at the end of the rental.
- Enforcement Level: Strict
- Supporting fact types: rental has rental charge
  estimated rental charge is provisionally charged to credit card
  renter has credit card
  rental has driver
What will SBVR do? (1)

Support specification of business rules:
- From an organization perspective - not an IT system perspective
- Using the vocabulary of the business - not the vocabulary of its IT models
- Regardless of whether the rules can be, or will be, automated
Transformations will be needed

SBVR Model

Business Rules

Shared Concepts

Transformations / mappings

End-to-end Workflow Model

Procedures for people in the business

IT System Model

IT Rules

Database Schema

IT Rules

Database Schema

applies to

Customer

Employee

Customers

Employees

Cars
What will SBVR do? (2)

SBVR realizes the ‘Business Rules Mantra’:

“Rules are built on Facts. Facts are built on Terms.”

Base Business Rules on Fact Types
Associate Concepts to define Fact Types
Define Concepts

Develop Vocabularies to represent them (starting with terms for the concepts)

... to describe businesses, not the IT systems that serve them
... in language understandable by business people
Preview: making a business rule

Start with a fact type, e.g.

rental has driver

Add a modal operator (from a limited set: “it is obligatory”, “it is necessary” …), e.g.

it is obligatory that rental has driver

Quantify and qualify:

Add quantifiers to roles in the fact type (“each”, “at least one”, “no more than N”, …)

it is obligatory that each rental has at least one driver
it is obligatory that each rental has no more than 4 drivers

Use additional fact types as qualifiers (“the location of the return branch of the rental …”)

Add conditions based on fact types (“if a rental return is more than 4 hours late …”)
Two kinds of Business Rule in SBVR

- Structural business rules:
  - Specify what an enterprise takes things to be
  - Cannot be broken (are “true by definition”)
  - e.g. local area is in exactly one operating country

- Operative business rules:
  - Guide what an enterprise must do
  - Can be broken, so need an enforcement regime:
    - Detection of violations
    - Remedial action to restore compliance
    - (perhaps) Application of sanctions
  - e.g. Each rental car that is assigned to a rental must be at the pick-up branch of the rental.
Semantic Formulation of a business rule

Each rental car must have exactly one vehicle identification number.

This means that it is necessary that each rental car has exactly one vehicle identification number.

The necessity claim 'exactly one' quantification means that it is universal.

Variable (rental car) and variable (vehicle identification number) are atomic formulations.

Rental car has vehicle identification number.
From Business Rule Statement to XML

1. Start with a business rule statement.
   - It is prohibited that a barred driver is a driver of a rental.

2. Identify symbols in vocabulary.
   - It is prohibited that a barred driver is a driver of a rental.

3. Parse according to language rules.

4. Restate as facts of logical formulation.
   - An obligation claim embeds a logical negation....

5. Represent facts of logical formulation as objects.

6. Write objects as XML.
   - <is-obligation-claim />
Business Rule – Parsed

It is prohibited that a barred driver is a driver of a rental.
Logical Formulation

It is prohibited that a **barred driver** *is a driver of* a **rental**.

- **obligation claim**
  - *embeds* a **logical formulation** that *is a logical negation*
  - *has a negand that is an existential quantification*
  - *introduces a variable*
  - *has the type* **barred driver**
  - *scopes over an existential quantification*
  - *introduces a variable*
  - *has the type* **rental**
  - *scopes over an atomic formulation*
  - *is based on the verb concept: 'rental has driver'*
  - *has a role binding*
  - *is of the fact type role that is 'rental' of 'rental has driver'*
  - *binds to the variable that has the type** **rental**
  - *has a role binding*
  - *is of a fact type role that is 'driver' of 'rental has driver'*
  - *binds to the variable that has the type** **barred driver**
XML (for Logical Formulation)

<is-obligation-claim obligation-claim="oc"/>
<modal-formulation-embeds-logical-formulation modal-formulation="oc" logical-formulation="n"/>
<logical-negation-has-negand logical-negation="n" negand="eq1"/>
<is-existential-quantification existential-quantification="eq1"/>
<quantification-introduces-variable quantification="eq1" variable="v2"/>
<variable-has-type variable="v1" type="bdt"/>
<quantification-scopes-over-logical-formulation quantification="eq1" logical-formulation="eq2"/>
<is-existential-quantification existential-quantification="eq2"/>
<quantification-introduces-variable quantification="eq2" variable="v2"/>
<variable-has-type variable="v2" type="rt"/>
<quantification-scopes-over-logical-formulation quantification="eq2" logical-formulation="af"/>
<is-atomic-formulation atomic-formulation="af"/>
Model Driven Interoperability
Introduction

- Organizations are collaborating with other organizations in order to meet their business objectives.

- For business optimization, organizations re-structure their business realizations by creating new constellations within an enterprise and across the organizational border that need to interoperate.

- Key issue: service network, who is to produce the service, who is to consume the service, business goals.

- It seems BMM and SoaML can combine these issues through:
  - Align goals with service-centric approach.
ATHENA Interoperability Framework (each system is described by enterprise models and different viewpoints, such as business, process, service, information)
Current MDA Interoperability Architecture

- CIM/EM models
  - Semantic annotation
- PIM System models
  - Semantic annotation
- PSM System models
  - Semantic annotation
- System
- Semantic annotation
- Technical mapping
- Interoperability execution
- IF
- IF
Architecture for semantic annotation and reconciliation
Contents

- Introduction
- Description of EMPOWER and MEMPOWER
  - EMPOWER Project
  - MEMPOWER Project
- Comparison Semantic mappings
- Conclusion & Further work
EMPOWER

- an innovative framework for interoperability between enterprise systems
- a flexible and extensible architecture
- a system environment
**Model Transformation Services** support the runtime lifting and lowering transformations among messages and ontologies based on the Model Map.

**Ontology Definition Meta-model** is a family of MOF meta-models, mappings between those meta-models, and a set of profiles that enable ontology modeling through the use of UML-based tools.

**Semantic Annotation Model editor** is used to relate different PIM models and ontology. It is used to annotate the SoaML model with Ontology.

**SoaML** describes the services models. The **Model Mapping** in the MEMPOWER includes transformations from models to ontology and ontology to models.
The EMPOWER Enterprise Interoperable Services Semantic Map

- Account Processing
- Inquiry Processing
- Quotation Processing
- Sales Order Processing
- Billing

Business Services Catalogue:
- Financial Accounting
- Management Accounting
- Procurement
- Warehouse Management
- Sales Order Management
- After Sales Support

Legacy System A (IT Assets Inventory)
Semantic Adaptation Architecture
PIM level use of Ontology mappings

![Diagram](image)
Ontology example
Address Ontology
Address in Source and UML
“Address” in the source and target transformation rules
“Address” transformations from source.xml and target.xmi
SAM editor realized in tree views
Ontology is represented as a structured and classified tree view. It shows the properties and relationships between those classes.

A simple example of class annotations on the PIM level

Interface of demo

Annotations
After annotating and exporting the model, you will get the file with an additional attribute. The annotations are displayed in red.

```xml
<soaml:Class name="POMessage" saName="PurchaseOrderMessage"
    stereotype="messageType">
</soaml:Class>
<soaml:Class name="Customer" saName="Customer"
    stereotype="DataType">
    <soaml:Attribute name="customerId" saName="hasCompanyRegNo"
        type="String" modifier="public" />
    <soaml:Attribute name="name" saName="hasComanyName"
        type="Name" modifier="public" />
    <soaml:Attribute name="address" saName="hasAddress"
        type="String" modifier="public" />
    <soaml:Attribute name="creditScore" type="Integer" modifier="public" />
</soaml:Class>
```
Semantic Mapping

1. Ontology-based mapping on the PSM-Level (EMPOWER)
2. Direct mapping on the PSM-Level
3. Ontology-based mapping on the PIM level (MEMPOWER)
4. Direct mapping on the PIM level

<table>
<thead>
<tr>
<th>Approach</th>
<th>Ontology-based PSM</th>
<th>Direct mapping PSM</th>
<th>Ontology-based PIM</th>
<th>Direct mapping PIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Address

Address in Target.xsd has only one element: Address
Address in Source.xsd is divided into three elements: Address, Place, and Province
Address in Ontology is divided into three elements: Address, Region, and Province
1. PSM: Ontology-based

- Annotation based on ontology on the PSM-level
  -- Annotate source.xml and target.xml using Ontology
2. PSM: Direct Mapping

- Mapping without ontology on the PSM-level
  -- Map between source.xml and target.xml (xsl:easy)
3. PIM: Ontology-based

1. Transformation From PSM level to PIM level

-- Generate sources.uml and target.uml from schemas (HyperModel Designer)

3.1)
3. PIM: Ontology-based

1. Transformation From PSM level to PIM level

   --Generate sources.uml and target.uml from schemas (HyperModel Designer)

2. Mapping Between Models based on ontology on the PIM level

   Step 1: Generate meta-models of models and ontology using EMF
3. PIM: Ontology-based

1. Transformation From PSM level to PIM level
   --Generate sources.uml and target.uml from schemas (HyperModel Designer)

2. Mapping Between Models based on ontology on the PIM level
   - Generate mapping rules from source to ontology, and ontology to target using ATL

```
rule Address2OnAddress
from
    sc: ABW!AddressInfo (sc.oclIsTypeOf(ABW.AddressInfo))
to
    t: Ontology!Address
        addressType <- sc.AddressType,
        addressCountryCode <- sc.CountryCode,
        addressPostCode <- sc.ZipCode,
        addressRegion <- sc.Place,
        addressSequenceNo <- sc.SequenceNo,
        address <- sc.Address,
        addressProvince <- sc.Province,
        addressUpdaterFlag <- sc.UpdateFlag,
        contactName <- sc.ContactName,
        contactPosition <- sc.ContactPosition,
        phone <- sc.Phone,
        postalCode <- sc.PostalCode,
        netInfo
helper context Ontology!Address def totalAddress : String =
    self.address.toString() + self.addressProvince.toString() + self.addressRegion.toString();
```
3. PIM: Ontology-based

1. Transformation From PSM level to PIM level
   -- Generate sources.uml and target.uml from schema (HyperModel Designer)

2. Mapping Between Models based on ontology on the PIM level

```xml
<PayMethod>DD</PayMethod>
<BankAccount>13101010958</BankAccount>
<Swift>45</Swift>
<ClearingCode>123</ClearingCode>
<In_ruleId>IK1</In_ruleId>
>Status>N</Status>
</PaymentInfo>
</Supplier/Customer>
<AddressInfo>
  <UpdateFlag>0</UpdateFlag>
  <AddressType>1</AddressType>
  <ContactName>CS15 kontakt person</ContactName>
  <ContactPosition>CS15</ContactPosition>
  <Address>CS15 address</Address>
  <Place>CS15 place</Place>
  <Province>CS15 Province</Province>
  <CountryCode>GB</CountryCode>
  <SequenceNo>0</SequenceNo>
</AddressInfo>
```

addressRegion="CS15 province" address="CS15 address" addressProvince="CS15 Province"
4. PIM: Direct Mapping

- Transformation Between Models without ontology on the PIM level

--Use Semaphore tool to map source to target
Conclusion

- Ontology-based mapping (S-O-T) VS Direct mapping (S-T) on the PIM level
  - $2N$ vs $N^2$

Mapping between each model and ontology will result in a linear growth of number of mappings.

Mapping between all model pairs will result in $N$-squared mappings.
## Conclusion

### Mapping PIM-Level VS PSM-Level

<table>
<thead>
<tr>
<th>Mapping</th>
<th>2N</th>
<th>N²</th>
<th>2N</th>
<th>N²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Ontology</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Platform Independent</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Multi-source documents Input</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Multi-target documents Output</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Conclusion & Further work

**Conclusion**
- Ontology-based semantic annotations reduces mapping times from N-squared to 2N, but cost is a standard ontology.
- Model Driven approach supports the interoperability independent from platform technologies, compared to a platform specific technical approach.

**Further work**
- Implement multiple industrial use cases with five scenarios for comparing EMPOWER and MEMPOWER.