TDT4252
Modelling of Information Systems
Advanced Course

Sobah Abbas Petersen
Adjunct Associate Professor
sap@idi.ntnu.no
Lecture Today

• EEML (Extended Enterprise Modelling Language): Goal and Process Modelling

• Model Evaluation: SEQUAL

  (Based on lecture notes from Spring 2010, by Prof. John Krogstie)

Based on the article:

A12: Krogstie, J. Using EEML for Combined Goal and Process Oriented Modeling: A Case Study

Additional reading:

From lecture on perspectives to conceptual modelling

- Structural
- Functional
- Behavioral
- **Rule-oriented**
- Object-oriented
- Social communication
- Actor/role-oriented
Overview

• Case: Trial-application for the State Education Loan Fund.
• Use of EEML (combined with formal rule system) in the case.
• High-level evaluation of trial results.
• Summary.
Case: Loan administration system

- State Education Loan Fund.
- This case was part of deciding a new overall architecture for their main application (PoC – Proof of concept).
- Want to improve automation of application handling.
- Must provide reasoning for the decisions being made.
Case (contd.)

• Goals for rule-modeling in the loan fund
  a) Support quick implementation of new rules, as these are changed regularly through the political process.
  b) Be able to analyze the consequences of proposed changes in the laws and regulations (with the politicians and department officials)
  c) Make it easier to maintain and evolve the rule base, including the more detailed internal rules.
  d) Support the education and training of the employees at the Loan Fund.

• a and c -> Blaze Advisor
High level architecture

Lecture 10: EEML

TDT4252, Spring 2011
Rule engine architecture

Forvaltning

Utviklere
setter opp regelarkitektur, integrasjon

Fagressurser
forvalter regelverket

Gjør det mulig for reglene å bli sentralt administrert:

- Hierarkisk organisering
- Arv / Gjenbruk
- Versjoner / korrigeringer
- Autorisasjon / sikkerhet

Struktur

Regelverk (repository)

Implementering

Deployment Manager

Java/J2EE .NET Mainframe

USS/COBOL

Gjør det mulig å:

- 24x7 tilgjengelighet
- Gjenbruk av regler
- Automatisk oppdatering
- Sentralisert kontroll
- Plattform uavhengig
- Arkitektur uavhengig
Adressing rule analysis in addition to rule execution - EEML
Goal modelling

• Goal
  – If context then deontic operator achieve state
    • e.g. If age of person is below 18 it is forbidden for that person to drive a car
  – Deontic operator: Necessitate, obligate, recommend, permit, discourage, forbid, contradict

• Goal connectors for more advance goal-hierarchies.

• Goal-relationships
  – goal deontic operator goal (argument)

• Goal applies to task/milestone/resourcerole/resource.

• Goal is action rule for task.

• Goal is precondition/decision rule/postcondition.

• Role/resource source of goal.
Goal hierarchies
Levels of rule-making in the State Loan Fund case

- Parliament
- Department
- Loan Fund Officials
- Programmers/ rule engineers
Goal hierarchy

- Everyone should be able to take a higher education
  - Obligate
  - Obligate

- Students should be able to study efficiently
  - Recommend

- Student financing must not be too expensive for the state
  - Obligate

- It should be possible to pay back student loans
  - Recommend
  - Recommend

- It should be possible to fully finance the studies
  - and

- The support is a mix of loan and stipend
  - Recommend
  - Recommend

- One need to ensure sufficient knowledge and skills in society
  - Obligate

- It should be possible to live wherever you want in Norway
  - and

- One need to ensure sufficient knowledge in all parts of the country
  - and
Goal Model and Source of rules
Control application for not paying interest
Different reasons for not paying interest

Control rights

- Exclude interest based on military service
- In prison
- Under full-time education
- Birth/adoption
- Under part-time education
- Student society work
- Care work
- On social security
- Low salary
- Student stipend
- Leonardo stipend
- Social security

Previous cases
Goals vs. processes
Detailed execution rule revisited
Model Evaluation revisited

• We have looked at several ways to evaluate models:
  – **Competency questions**: Can the model answer queries as though it has common sense about the enterprise? (Fox & Gruninger)
    • An example is available from: S. A. Petersen (2003) “Using Competency Questions to Evaluate an Agent-based Model for Virtual Enterprises”.
  – **Characteristics** of Enterprise Models (Fox & Gruninger).
  – **Principles** of Enterprise Models (Vernadat)
  – **SEQUAL** (Krogstie)

• Before we evaluate the EEML model using SEQUAL, we will revisit the other 3 ways to evaluate models.
A11: Competency Questions

• We should be able to query the model and obtain answers to support the organisation.

• Commonsense queries:
  – Ability to answer queries as though the model has a common sense understanding of the enterprise.
  – Such an understanding often represents knowledge about the enterprise acquired over a relative short time and does not require expert knowledge.
  – Examples of such knowledge: organisational structure, roles, goals and resources would enable the deduction of what resources a person might allocate based on his/her role in the organisation.
  – We refer to it as shallow-level processing: retrieval that requires a small number of deductions to answer the query.
    • Requires a set of rules of deduction, axioms.
## A11: Evaluating Enterprise Models (2)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional completeness</td>
<td>Can the DEM represent the information necessary for a function to perform its task?</td>
</tr>
<tr>
<td>Generality</td>
<td>To what extent is the DEM shared between the diverse activities in the enterprise, e.g. finance and manufacturing?</td>
</tr>
<tr>
<td>Efficiency</td>
<td>Does the DEM support efficient reasoning?</td>
</tr>
<tr>
<td>Perspicuity</td>
<td>Is the DEM easily understood by users so that it can be applied consistently and interpreted across the enterprise?</td>
</tr>
<tr>
<td>Precision granularity</td>
<td>Is there a core set? Does the representation support reasoning at various levels of abstraction and detail?</td>
</tr>
<tr>
<td>Minimality</td>
<td>Does the DEM contain the minimum number of objects necessary?</td>
</tr>
</tbody>
</table>
Enterprise Modelling Principles (1)

• Principle of separation of concerns:
  – It would be unrealistic to consider an enterprise as a whole. It must be analysed piece by piece.

• Principle of functional decomposition:
  – Major functions structured as sub-functions.

• Principle of Modularity:
  – To facilitate management of change.

• Principle of Genericity:
  – Important to define standard building blocks as generic classes to factor common descriptive attributes and behaviours.

• Principle of reusability:
  – To reduce modelling efforts and increase modularity.

Ref: Vernadat
Enterprise Modelling Principles (2)

• Principle of separation of functionality and behaviour:
  – Functionality: things to be done; behaviour: the way things are done.

• Principle of process and resource decoupling:
  – Consider separately the things being done (processes) and the agents performing them (resources).

• Principle of conformity:
  – Deals with syntax and semantics of the model to really accurately represent what it’s supposed to model.
Enterprise Modelling Principles (3)

• Principle of model visualisation:
  – To easily communicate models, e.g. simple graphical formalisms.

• Principal of simplicity vs. adequacy:
  – The modelling language to be rich enough to express what needs to be expressed, yet simple enough so that users can easily learn to use it.

• Principle of management of complexity:
  – Must permit the representation of systems of great complexity.

• Principle of rigor of representation:
  – The model must neither be ambiguous nor redundant.

• Principle of separation of data and control:
  – To support real-time systems. The process is operated not by the availability of data, but by events.

Ref: Vernadat
A12: High-level evaluation

• Based on piloting the approach
• Limited time/resources available
• Structured according to SEQUAL
  – Evaluation of rule language
  – Evaluation of rule-model
• First evaluation based on interviews with different stakeholders and models/documentation provided
• Results/interpretations structured and presented to stakeholders for comments
Evaluation framework – SEQUAL

Goals of Modelling G

Modeling Domain D

Model Externalization M

Language Extension L

Social actor explicit knowledge K

Social actor Interpretation I

Organizational

Physical

Empiric

Pragmatic (learning)

Pragmatic (human understanding)

Pragmatic (tool understanding)

Pragmatic (action)

Semantic

Perceived semantic

Social

TDT4252, Spring 2011
Evaluation framework – language quality

Goals of Modelling

Modeling Domain

Model Externalization

Technical actor interpretation

Social actor explicit knowledge

Modeller appropriateness

Organizational appropriateness

Social actor Interpretation

Comprehensibility appropriateness

Language Extension

Participant appropriateness

Domain appropriateness

Tool appropriateness

Lecture 10: EEML

TDT4252, Spring 2011
Basis for evaluation (1)

• **Domain D**: Primarily described through the laws and regulation for study financing. Although the domain seems to be fully externally given, in practice a large number of the resulting rules to follow are based on internal deliberations within the Loan Fund.

• **Rule modeler** (as a basis for K): The goals and rules were modeled in METIS and Blaze Advisor by professional rule designers and loan fund professionals in cooperation. For pre-defined changes the loan fund professionals could do this through the RMA.
Basis for evaluation (2)

• Rule interpreter (vs. I): Views of METIS models and rules in Blaze were to be understood by those involved in the modeling. All loan fund personal were to be able to understand the rule documentation. RMA-rules being easier to understand were to be available for all. Through rule execution, texts including the reasoning of the decision made were produced, which are meant to be understandable by everyone.

• Language used for process and goal modeling was EEML in METIS. Formal rules followed the syntax of the proprietary rule language SRL (Structured Rule Language) supported in Blaze Advisor.
Basis for evaluation (3)

• Model M:
  – Data model (as a basis for the database-application, but also as basis for data definitions used in the case processing system and rule engine).
  – Process and goal model in EEML.
  – Rule model (as a basis for the rule engine).

• The rule model can be looked upon as four interrelated models:
  – The laws and regulation as they are written in juridical terms. Here we look upon this as part of the domain.
  – Goals in METIS.
  – The rules as implemented in the rule engine (Blaze).
  – Some rules are made available through a web interface (RMA – Rule Maintenance Application).
Quality of language (1)

• **Domain appropriateness:** It was possible to express all the execution rules in the PoC formally in SRL. Informal (deontic) rule and relationships could be expressed in EEML.

• **Modeler appropriateness:** The loan fund professionals together with rule designers were able to express the rules in SRL. Loan fund professionals were also able to use the RMA. EEML goal hierarchies could be developed through detailed discussions.

• **Participant appropriateness:** EEML and SRL was only known by people from external companies, and it is found that specifically SRL represents a steep learning curve both for Loan Fund professionals as well as system developers internally.
Quality of language (2)

- **Comprehensibility appropriateness**: Those closely involved in the process appeared to understand the rules, especially since navigation was supported in the goal and process-models. Since the execution rules ended up as a mix of English keywords and Norwegian concepts used in the data model, they were hard to understand.

- **Tool appropriateness**: The tools were appropriate for enterprise modelling and rule execution, and other tests have been done supporting the scalability of the approach.

- **Organizational appropriateness**: A positive aspect here is that the language used (SRL) is according to an emerging standard (PRR). Expensive tools, with limited Norwegian expertise available.
Quality of model (1)

- **Physical quality**: The rules are primarily available through the tool, which limits the availability. It is also possible to generate HTML reports from the tools for wider availability, but it seems inappropriate for widespread dissemination. RMA includes standard authorization mechanisms, ensuring that only authorized personnel can change the rules.

- **Empirical quality**: Goal and process-model visualization was regarded as a useful way of getting an overview of the rule-base.

- **Syntactical quality**: The METIS model and SRL-rules implemented in the rule engine are syntactically correct.
Quality of model (2)

• **Semantic quality**: All the production rules are included, and rules as expressed in the underlying laws and regulations are included in the METIS model.

• **Pragmatic quality**: It was relatively easy to keep an overview of the implemented rules and how they were related to the laws and regulations. The METIS-model has made it easier to understand the underlying intention of the rules.

• **Social quality**: On some of the detailed rules, there were discussions on the appropriate interpretation of these. This does not apply to the rules and regulations themselves, but rather to how they should be follow up in practice in the Loan Fund.

• **Organizational quality**: The combined approach support all goals a to d (although not full support for b-simulation).
Combination EEML and SRL vs. Declarative modelling (1)

- **Problem-orientation**: The representation of business rules declaratively is independent of what they are used for and how they will be implemented. This is only partly the experience using the traditional production rule system in isolation (Blaze Advisor). The detailed expression of the rules in SRL is hampered by the needs of the implementation. A combination with a less formally defined rule language as we have illustrated with EEML is looked upon as beneficial instead of having to have different, not integrated representation.

- **Maintenance**: The benefits on this account is witnessed in the production-rule system, specifically with the added support of the RMA.

- **Knowledge enhancement**: The explicit rule-representation, and the possibility to quickly test their effect has proved beneficial. The possibility to also relate the rules to more high-level goals in the rule hierarchy enables an even broader debate on these issues.
Combination EEML and SRL vs. Declarative modelling (2)

• Possible to distinguish between rules of necessity and deontic rules: EEML rules can include deontic operators.

• In many goal and rule modeling languages it is not possible to specify who the rules apply to: EEML-rules can be explicitly related to organizational actors.

• Flat rule-bases: Development of a rule hierarchy is supported, and it is also possible to link the rules to a hierarchical process model.

• Link to other models of the organization used to understand and develop the information systems, such as data and process models: Provided in EEML.

• The source of the rule can be indicated.

• Support contradictory rules and goals: Possible in EEML. This is not shown in the case, but since the full range of deontic operators can be used between rules, it is possible to e.g. represent that the fulfillment of one rule forbids the fulfillment of another.
Summary

• We have looked at an example of an Enterprise Modelling language EEML, which combines Goal and Process Modelling.
• It combines Goal Modelling with a rule base.
• It uses Metis to define the modelling language and create the instantiated model.
• It used another tool Blaze Advisor to define the rules.
• We have revisited the different ways to evaluate Enterprise Models and Enterprise Modelling languages.
Next Lecture

• Introduction to Active Knowledge Modelling (AKM)