TDT4252
Modelling of Information Systems
Advanced Course

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

• Evaluating Enterprise Models

Based on the following article:


Additional reading:

Krogstie, J. Using EEML for Combined Goal and Process Oriented Modeling: A Case Study
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: (A11) 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) – additional reading.
A10: Competency Questions (Fox & Gruninger)

• 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.
A10: 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.

Ref: Vernadat
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
A11: An example of using Competency Questions

• The paper is based on these ideas:
  • What is a Virtual Enterprise (VE)?
  • Agent-based Model
  • Competency Questions as an evaluation technique
• Examples
What is a Virtual Enterprise?

- A Virtual Enterprise (VE) has the following characteristics:
  - Goal-oriented
  - Autonomous entities that collaborate
  - Distributed and heterogeneous
  - Has a limited lifetime

A team of partners (human beings, organisations or software agent) that collaborate to achieve a specific goal.
What is an Agent?

• An agent can be defined as a computer system that has the following properties:
  – Autonomy
  – Social ability
  – Reactivity
  – Pro-activeness

• Goal-oriented

• Designed to operate in distributed environments

• Handle sophisticated interactions such as negotiations
Why Agents & Virtual Enterprises?

• Human beings or organisations can delegate responsibilities to agents.
• Agents can provide a flexible means of modelling the VE in terms of cooperative work.
• Agents are computational entities. Thus the resulting model can be easily and efficiently converted to the required computational support.
• Agents can handle sophisticated interactions such as coordination & negotiations.
## Agent-based Model: Contents

<table>
<thead>
<tr>
<th>Entity</th>
<th>Attributes</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal, e.g. Produce</td>
<td>goal(goal_name, product_area, deadline, max_cost).</td>
<td>achieved_by(goal, activity).</td>
</tr>
<tr>
<td>some pens</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity, e.g.</td>
<td>activity(name, start_date, completion_date).</td>
<td>performed_by(activity, role).</td>
</tr>
<tr>
<td>Design pens</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agent, e.g. Michael</td>
<td>agent(name, address, goals, skills, availability).</td>
<td>filled_by(role, agent).</td>
</tr>
<tr>
<td>Michael</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Requirements of the Model

• A model is no longer just an information repository. But, it must be able to:
  • Answer queries about what is explicitly represented in as well as implied by the model.
  • Provide decision-making support.
The Need for Evaluation of Models

- We need to know when the modelling is complete.
- We need to be able to judge the correctness of representation.
Evaluation Techniques

There is a need for an appropriate evaluation technique.
Competency Questions

• Proposed by Gruninger & Fox.
• **Competency of a model** is its ability to answer a set of questions.
• The requirements for a model can be formulated as a set of questions that the model is required to answer.
Formulating Competency Questions

• A challenge in using Competency Questions is the design of an appropriate set of questions.

• For our work, we have obtained questions based on:
  • Literature
  • Industrial case studies.
Case Studies

• **Company A: Small Consulting Company**, which maintains a database of highly qualified consultants in various fields and draws upon these resources to form project teams.

• **Company B: Student Project Groups**: A company that operates in the oil and offshore industry, on a global scale. The particular scenario that we analysed was the selection of several groups of students who will work together as teams during their summer holidays.

• **Company C: Building Construction Project**: Company C is hired by a customer to assist them to evaluate bids in the selection of a contractor for large scale projects in various domains.
Example 1: Do all agents contribute to the goals of the VE?

<table>
<thead>
<tr>
<th>Agent</th>
<th>Activity</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>role</td>
<td>achieved_by</td>
<td>activity, goal</td>
</tr>
<tr>
<td>filled_by</td>
<td>performed_by</td>
<td>role, agent</td>
</tr>
<tr>
<td>performed_by</td>
<td>is_assigned</td>
<td>agent, goal</td>
</tr>
</tbody>
</table>

For all agents:

if (is_assigned(agent, goal))
then (all agents contribute to the goals of the VE).

performs(role, activity, agent) :-
  filled_by(role, agent),
  performed_by(activity, role).

is_assigned(agent, goal) :-
  performs(role, activity, agent),
  achieved_by(activity, goal).

Lecture 9: Evaluating Enterprise Models
TDT4252, Spring 2012
Example 2: How do we know when VE formation is complete?

- VE formation is complete when all roles are filled by agents.
  - All goals are assigned activities, all activities have roles that perform them.

for all roles
  if  (performs(role, activity, agent))
  then  (VE formation is complete).
Example 3: What are the skills required to perform a particular activity?

performed_by(activity, role).
filled_by(role, agent).
requires(role, requirements(Skills, availability, price)).

if (performed_by(activity, role) and requires(role, requirements(Skills, availability, price)))
then (skills required for activity = Skills).
Evaluating Models

• The Competency Questions evaluate the competency of a model.
• For a complete evaluation, other characteristics of the model must also be evaluated.
• Fox et. al. proposed a set of characteristics to evaluate enterprise models.
### Considering other characteristics

<table>
<thead>
<tr>
<th><strong>Characteristic</strong></th>
<th><strong>Our Agent-based Model</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generality</strong></td>
<td>Can be applied to VEs in different domains.</td>
</tr>
<tr>
<td><strong>Transformability and efficiency</strong></td>
<td>Since model is described in formal logic, it can easily be transformed into a computer-based model that can be used for efficient reasoning.</td>
</tr>
<tr>
<td><strong>Perspicuity</strong></td>
<td>Since the model can be described in semi-formal syntax, the users can easily understand it.</td>
</tr>
<tr>
<td><strong>Granularity and scalability</strong></td>
<td>Can be enhanced to cover the complete lifecycle of the VE.</td>
</tr>
<tr>
<td><strong>Extensibility</strong></td>
<td>Currently, the model is very generic. It can be enhanced to include concepts such as ”change”, using rules.</td>
</tr>
</tbody>
</table>
Summary

• Presented an agent-based model to support the formation of VEs.
• Presented a few examples of how Competency Questions have been used to evaluate the model.
An example of an Evaluation Framework – SEQUAL

Note: Slides 28-36 are not pensum, but included as an example, based on additional reading.

Ref: Krogstie

Lecture 9: Evaluating Enterprise Models

TDT4252, Spring 2012
Evaluation framework – language quality

Goals of Modelling

Modeling Domain D

Model Externalization M

Technical actor interpretation I

Social actor explicit knowledge K

Modeller appropriateness

Organizational appropriateness

Comprehensibility appropriateness

Language Extension L

Participant appropriateness

Domain appropriateness

Tool appropriateness

Ref: Krogstie
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 followed 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).
Summary of Model Evaluation

- 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)
  - Characteristics of Enterprise Models (Fox & Gruninger).
  - Principles of Enterprise Models (Vernadat)
  - SEQUAL (Krogstie) – additional reading.
Next Lecture

• Competences for Enterprise Modelling.