Using EEML for Combined Goal and Process Oriented Modeling: A Case Study

John Krogstie

IDI, NTNU,Trondheim, Norway
krogstie@idi.ntnu.no

Abstract. In organizations, goals and rules on different levels ranging from visions, to strategies, tactics, and operational goals have been expressed for a long time. In the information systems field, the interest on goals and rules has come from two directions. A) Business goals for use in requirements specification. B) Rule-based (expert) systems, focusing on automation of rule-execution. Using the modeling language EEML we are able to bridge these usage areas, and link business and executable rules with more traditional process modeling. The paper presents the use of this technique through a case study with the Norwegian State Loan fund. The results from the case are evaluated using SEQUAL, a semiotic quality framework for the evaluation of models, modeling languages and modeling environments. The result from the evaluation is promising in addressing the diverse needs of goal, rule and process modeling in analysis and design of information systems.

1 Introduction

The use of process modeling has a long tradition. The range of languages available spans simple flowcharting techniques, languages initially used as part of requirements engineering such as UML, dedicated business-oriented modeling languages such as Event-driven Process Chains, and also formalized and academically studied languages such as Petri nets and their dialects. The interest for process modeling in IS and BPM practice has increased over the past few years [3]. With this increased interest it has also become apparent that there is a lot of relevant business knowledge that is not captured in the process model. In conceptual modeling one thus early looked at combining process and data models, whereas combining process modeling, data modeling and rule modeling goes at least back to BNM [21]. Here, and more precisely stated in Tempora [12], it was emphasized the need to focus on modeling of business rules and goal modeling not only alone, but in combination with process modeling, proving the underlying argumentation for why the process is as modeled in the first place.

System analysis and modeling on the business levels usually is using the same kind of approaches as used for enterprise modeling. According to [2,16,22], enterprise models, including enterprise process and goal models may be usefully utilized in the following different areas:

1. Human-sense making and communication: The main purpose of modeling is to make sense of aspects of an enterprise and communicate with other people
2. Computer-assisted analysis: The main purpose of modeling is to gain knowledge about the enterprise through simulation or deduction
3. Business Process Management in the meaning of maintaining a corporate memory e.g. as part of the quality system of the enterprise
4. Model deployment and activation: The main purpose of modeling is to integrate the model in an information system and thereby have the model to actively take part in the work performed by the organization. Models can be activated in three ways:
   a) *Through people* guided by process 'maps', where the system offers no active support or enforcement
   b) *Automatically*, where the system plays an active role in enforcing the 'script', as in most workflow engines
   c) *Interactively*, where the computer and the users co-operate in interpreting the model in the situations that arises
5. The model is a basis and gives the context for a traditional system development project, without being implemented directly

The focus in this article is on the use of combined goal and process modeling in a way where all these areas of use is covered, with an emphasis on the first three areas, and the usage of a language called EEML in this respect.

In the next section, we present the case study. EEML including models from the case is presented in section 3. The experiences from using EEML on the case study are evaluated in section 4. Section 5 discusses briefly related work, before finalizing the article with ideas for further work.

2 Case Study

In Norway, one attempt to make it possible for everyone with the necessary skills and competence to afford pursuing higher education. In connection to this, a specific organization within the public sector, the State Education Loan Fund is established to manage student financing. This involves accepting application for student financing, evaluating these, and ensuring loans are paid back according to the regulations. Whereas the Parliament decides the overall laws for the area, the relevant department (currently named the ‘Knowledge Department’) provides more detailed regulations. The guidelines for how to follow-up of the laws and regulations are further detailed by experts in the Loan Fund to be followed in case processing (and to be used in the automated systems partly performing the case processing). The way the regulations as interpreted by the loan fund is implemented in the application can be looked upon as a fourth rule-model, which one needs to make sure adheres to the other levels.

Several goals for the representation of rules have been identified in the loan fund:

1. Must be possible to implement new rules rapidly, as these are changed regularly (both externally through the political process, and internally).

2. Must be possible to analyze the *consequences* of proposed changes in the laws and regulations (and discuss this with the politicians and department officials).
3. Make it easier to maintain and evolve the rule base, including the more detailed internal rules.
4. Support the education and training of the employees at the Loan Fund.

One envisaged in the Loan Fund that areas 1 and 3 could be supported by a rule engine, and Blaze Advisor www.fairisaac.com/rules was chosen for this purpose. To also address areas 2 and 4 with one approach, we tried in addition to production rules in the SRL-language supported by Blaze Advisor to use the EEML goal modeling and process modeling.

The architecture of the approach is based on having the administrative process-oriented case-processing system to be in charge of the overall workflow, calling the rule engine on a case by case basis to evaluate the applications. It is possible to call the rule engine with incomplete data, in which case it gives an overview of the lacking data. The case processing system then have to support getting the missing data, either from internal or external data sources, before calling the rule engine again.

3 EEML – Extended Enterprise Modeling Language

We have over the last seven years been developing a model-driven approach to be able to quickly support the development of model-driven solutions primarily supporting interactive model activation (enterprise modeling area 4c above) [9]. Our main approach to achieve this is the use of model-generated work-places (MGWP [11]), is a user platform that provides the graphical front-end for human users to interact with software services supporting their day-to-day professional activities.

Although originally geared towards generation of process support environment, it is also possible to use EEML for more general enterprise modeling, focusing on usage areas 1 and 2 from the introduction. In connection to the case study, we have also extended the language to support the modeling of formal rules in the SRL-languages in an integrated manner, enabling automatic rule execution in a rule engine.

The EEML-language is divided into 4 sub-languages, with well-defined links across these languages:

- Process modeling
- Data modeling
- Resource modeling
- Goal modeling

Process modeling supports the modeling of process logic which is mainly expressed through nested structures of *tasks* and *decision points*. The sequencing of the tasks is expressed by the *flow* relation between decision points. Each task has minimum an input port and an output port being decision points for modeling process logic, *Resource roles* are used to connect resources of various kinds (persons, organizations, information, material objects, software tools and manual tools) to the tasks. In addition, data modeling (using UML class diagrams), goal modeling and competency modeling (skill requirements and skills possessed) can be integrated with the process models.

A number of resource types and construct related to resources can be defined:

- Person
- Organization
- Information object/material object
- Software tool/Manual tool
- Skill
- Physical location

A number of relationships exist between resources and resource roles to support e.g. organizational modeling such as

- Resourcerole Is filled by resourcerole|resource
- Resourcerole is candidate for resourcerole|resource
- Resourcerole|resource communicates with resourcerole|resource
- Resourcerole|resource has supervision over resourcerole|resource
- Resourcerole|resource provide support to resourcerole|resource
- Organization has members

Goals in the goal modeling is, inspired by [8] represented as

If context then modality achieve state

Where the modalities possible are: Necessitate, obligate, recommend, permit, discourage, forbid, and contradict.

There are a number of relationships between goals and other modeling constructs:

- Goal applies to task/milestone/resourcerole/resource
- Goal is action rule for task
- Goal is precondition/decision rule/postcondition for task
- Role/resource is the source of a goal

Finally, goals can be related in means end hierarchies in the format

- goal modality goal (argument)

where modality can be chosen from the same list as for the used for rules above. A model 'G1 obligatory G2' can be read as that to achieve G1 it is obligatory to achieve G2. 'G1 forbidden G2' on the other hand indicates that if you achieve G2, you are forbidden to achieve G1 (this is parallel to the positive and negative contributions of goals included in [15]). Also as in [15], we provide a way to model and/or graphs in the goal hierarchy.

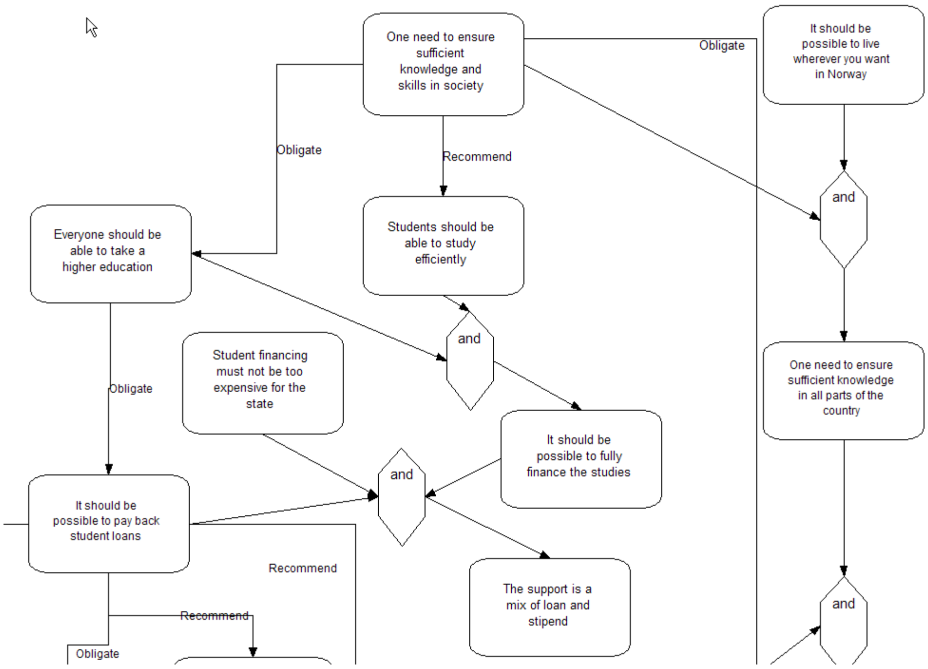

Fig. 1. Top-level goal-model for the case study

In figure 1, we have included parts of the overall goal-model for this case (note that since the models were originally in Norwegian, we have here remodeled only parts of this. In this and the model-examples below note that what is shown are *views* of a complete model covering only some aspects and relationships between the aspects. The views are developed using the METIS modeling tool, based on the complete model developed in the same tool).

The top-level goals are taken from the law on study financing (Including the need to ensure sufficient knowledge and skills in society, and because of this, that everyone should be able to pursue a higher education). All goals and rules can be expressed both informally and formally, and all kind of rules can be expressed. A modality can be added to each rule (indicating if the rule is a rule of necessity, or a deontic rule i.e. an obligation, recommendations etc.). For executable rules, a formal expression of the rule can be included, as illustrated below. The relationships between rules are also often deontic. An example is on the top left of the model, where it can be read that to ensure sufficient knowledge and skills in the society, it is obligatory that everyone is able to take a higher education. Note that although you will find rules at this level in the laws and regulations, the relationships between rules are not represented explicitly anywhere and have appeared through detailed discussions with different stakeholders. Relationships between rules can also be more complex, i.e. it is the combination (and) of that one want everyone to be able to study, and that they should be able to study efficiently (so-called full-time students not having to work on the side to finance the studying) that obligates the need for the study financing arrangements.

The model further illustrates some of the ways of using the hierarchy-mechanisms to model the more detailed relationships between sets of goals with other sets of goals.

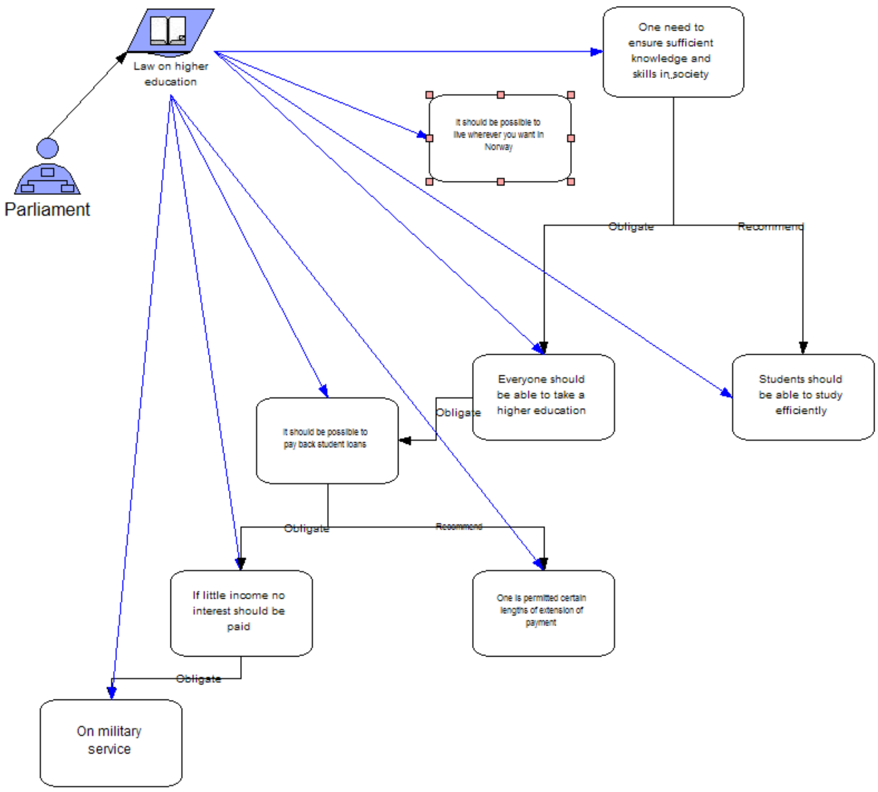

Fig. 2. Part of goal model for loan fund case, indicating the source of the rule

Fig.2 illustrates the link for organizational sources to the goals. Whereas the law goes down to the level in the goal-hierarchy where it is stated that 'if little income, no interest should be paid', and that one of the situations providing little income that is accepted is related to people serving military service, the detailing of this (e.g. that you need not pay interest if you are serving military duty), is taken from the departmental regulations which provide the details for the laws. In Fig. 3, we have further decomposed this rule, into the rules used to enforce this in the Loan Fund. As we see from the model the rules are developed at different levels (laws, regulations, internal loan fund practice).

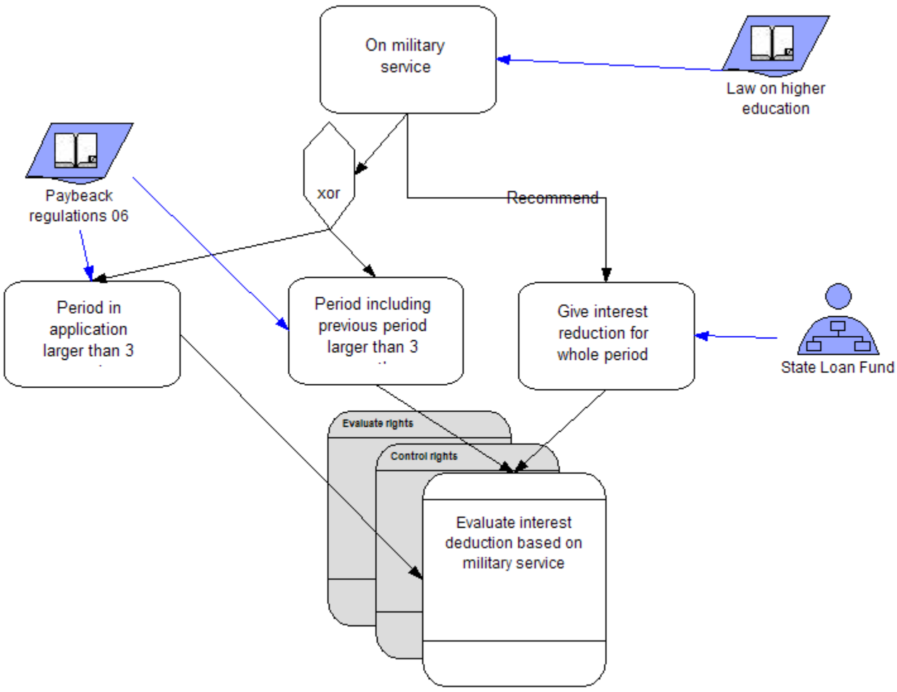

Fig. 3. Implementation of rules, linked to the process model and resource model

The relationship between these rules and the evaluation task is as 'action rules', using the formal representation of the rule (in this case in SRL to be able to include the rules directly in Blaze Advisor). For instance the rule 'Period in application larger than three months' is expressed as can be seen in Fig. 4. The formal description follows the SRL-syntax. The task also maps to the same task in the Blaze Advisor rule-flow, but the overall-process model includes both these tasks and the other tasks in the case processing system in an integrated manner. The links to the data model is not shown.

4 Evaluation

We will here evaluate the combined modeling approach relative the goals identified in the case study. We will first argue for the selection of evaluation framework, before presenting the actual evaluation.

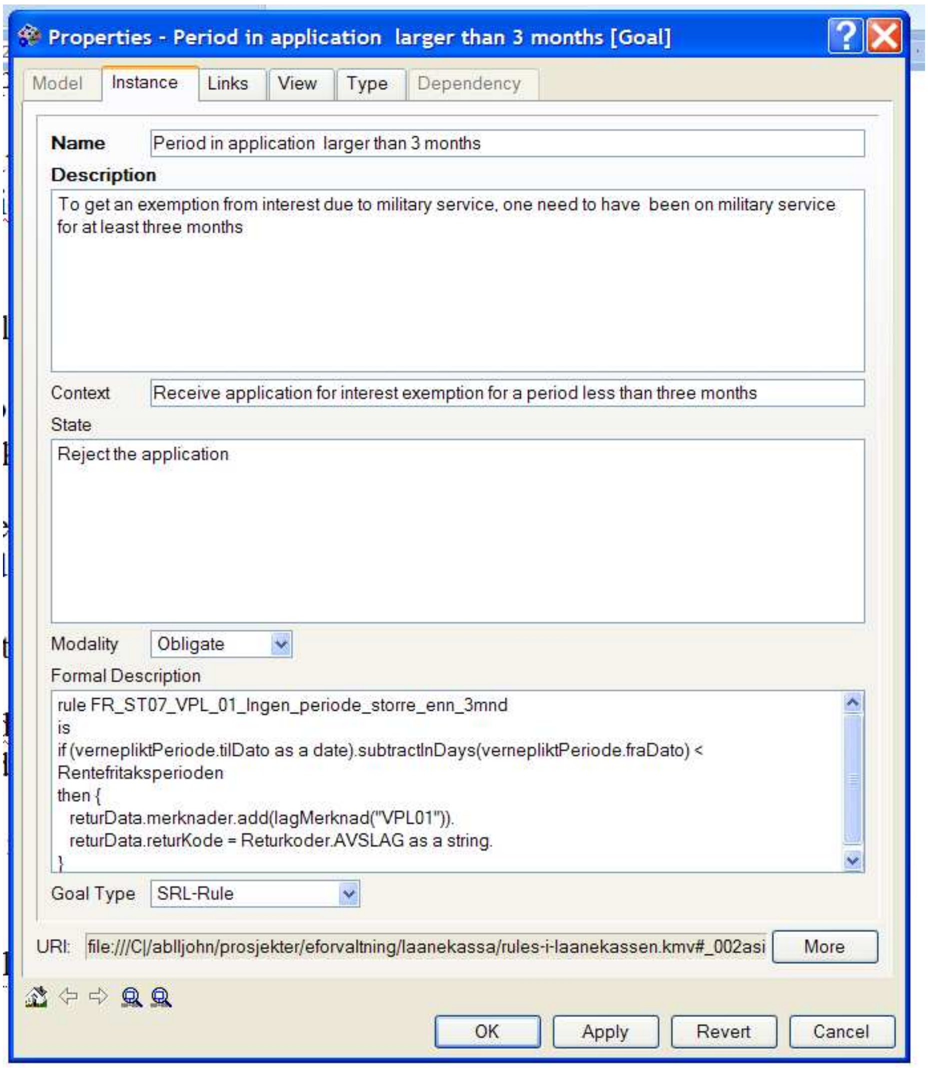

Fig. 4. Representation of individual rule

4.1 Evaluation Framework

There are a number of approaches and frameworks available for evaluating modeling approaches (including models, modeling languages, and modeling tools). Early proposals for quality goals for conceptual models and requirement specifications as summarized by Davis [4] included many useful aspects, but unfortunately in the form of unsystematic lists. They are also often restricted in the kind of models they regard (e.g. requirements specifications [4]) or the modeling language (e.g. ER-models [14] or process models [5, 20]. Few have specifically targeted goal-modeling. Another limitation of many approaches for evaluating modeling languages, is that they focus almost entirely on the expressiveness of the language (e.g. relative to some ontology, such as Bunge-Wand-Weber [23]. We have earlier developed a more comprehensive and generic framework for evaluating modeling approaches, called SEQUAL [10]. SEQUAL has the following unique properties:

- It distinguishes between goals and means by separating what you are trying to achieve from how to achieve it.
- It can be used for evaluation of models and modeling languages in general, but can also be easily extended for the evaluation of particular types of models.
- It is based on linguistic and semiotic concepts. In particular, the core of the framework including the discussion on syntax, semantics, and pragmatics is parallel to the use of these terms in the semiotic theory of Morris (see e.g. [17] for an introduction).
- It is based on a constructivistic world-view, recognizing that models are usually created as part of a dialogue between the participants involved in modeling, whose knowledge of the modeling domain and potentially the domain itself changes as modeling takes place.

Quality has been defined referring to the correspondence between statements belonging to the following sets:

- ***G,*** the goals of the modeling task.
- ***L,*** the language extension, i.e., the set of all statements that are possible to make according to the graphemes, vocabulary, and syntax of the modeling languages used.
- ***D***, the domain, i.e., the set of all statements that can be stated about the situation at hand.
- ***M***, the externalized model itself.
- ***K***, the relevant explicit knowledge of those being involved in modeling.
- ***I***, the social actor interpretation, i.e., the set of all statements that the audience thinks that an externalized model consists of.
- ***T***, the technical actor interpretation, i.e., the statements in the model as 'interpreted' by modeling tools.

The main quality types are described briefly below:

- Physical quality: The basic quality goal is that the externalized model ***M*** is available.
- Empirical quality deals with predictable error frequencies when a model ***M*** is read or written by different users
- Syntactic quality is the correspondence between the model ***M*** and the language extension ***L***.

- Semantic quality is the correspondence between the model ***M*** and the domain ***D***. This includes validity and completeness.
- Perceived semantic quality is the similar correspondence between the social actor interpretation ***I*** of a model and his or hers current knowledge ***K*** of the domain ***D***.
- Pragmatic quality is the correspondence between the model ***M*** and the actor interpretation ***(I*** and ***T)*** and application of it. We differentiate between social pragmatic quality (to what extent people understand and are able to use the models) and technical pragmatic quality (to what extent tools can be made that interpret the models). In addition, we include under pragmatic quality the extent that the participants after interpreting the model learn based on the model (increase ***K*** and that the audience after interpreting the model and learning from it are able to change the domain ***D.***
- The goal defined for social quality is agreement among audience members' interpretations.
- The organizational quality of the model relates to that all statements in the model ***M*** contribute to fulfilling the goals of modeling ***G*** (organizational goal validity), and that all the goals of modeling ***G*** are addressed through the model ***M*** (organizational goal completeness).

Language quality relates the modeling language used to the other sets. Six quality areas for language quality are identified.

- Domain appropriateness. This relates the language and the domain. Ideally, the conceptual basis must be powerful enough to express anything in the domain, not having what [23] terms construct deficit. On the other hand, you should not be able to express things that are not in the domain, i.e. what is termed construct excess [23]. Domain appropriateness is primarily a mean to achieve semantic quality.
- Participant appropriateness relates the social actors' explicit knowledge to the language. Do the participants have the necessary knowledge of the modeling language to understand the models created in the language. Participant appropriateness is primarily a mean to achieve pragmatic quality.
- Modeler appropriateness: This area relates the language extension to the participant knowledge. The goal is that there are no statements in the explicit knowledge of the modeler that cannot be expressed in the language. Modeler appropriateness is primarily a mean to achieve semantic quality.
- Comprehensibility appropriateness relates the language to the social actor interpretation. The goal is that the participants in the modeling effort using the language understand all the possible statements of the language. Comprehensibility appropriateness is primarily a mean to achieve empirical and pragmatic quality.
- Tool appropriateness relates the language to the technical audience interpretations. For tool interpretation, it is especially important that the language lend itself to automatic reasoning. This requires formality (i.e. both formal syntax and semantics being operational and/or logical), but formality is not necessarily enough, since the reasoning must also be efficient to be of practical use. This is covered by what we term analyzability (to exploit any mathematical semantics) and executability (to exploit any operational semantics). Different aspects of tool appropriateness are

means to achieve syntactic, semantic and pragmatic quality (through formal syntax, mathematical semantics, and operational semantics).
- Organizational appropriateness relates the language to standards and other organizational needs within the organizational context of modeling. These are means to support organizational quality.

4.2 Evaluation Results

In connection to the case study the sets in SEQUAL can be described as follows:

Model M: The model underlying the total system can be divided in three

- 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 model (relevant parts of the EEML process model as a basis for the case processing system)
- Goal and rule model (both for analysis and for the executable rules in the rule engine)

As indicated above, the rule model can be looked upon as four interrelated models:

1. The laws and regulation as they are written in juridical terms. Here we look upon this as part of the domain since they are unchangeable by the Loan Fund (see below).
2. Rule documentation through the rules in the goal hierarchy
3. The production rules as implemented in the rule engine. Not all rules are implemented. All implementable rules in the goal model match 1:1 to rules in Blaze Advisor, but also more technically oriented rules are included in the rule engine.
4. Some of the executable rules are made available through a web interface (called RMA – Rule Maintenance Application) so they can be changed by domain experts in the loan fund directly.

Finally, the implementation in Blaze Advisor can be looked upon as three models:

- Rule-flows, a simple decomposable process modeling notations to illustrate the implementation structure of rule sets. Where relevant, this matches parts of the EEML process model.
- Rule model: per rule/rule set (rules are put in rule sets that are evaluated together).
- Data model internally in Blaze Advisor that needs to be consistent with the data model in the case processing system.

Domain *I*: As indicated above, this is primarily described through the laws and regulation for study financing. Whereas the Parliament decides the overall laws for the area, the knowledge department produces more detailed regulations. The guidelines for how to follow-up of the laws and regulations are further detailed by experts in the Loan Fund. Also other relevant laws and regulations are part of the domain. Although the domain might appear to be fully externally given, in practice a large number of the detailed rules to follow are based on internal deliberations within the Loan Fund, thus there is a need to support quick changes, and not only the yearly revisions coming from parliament.

In connection to the audience of the model, two main roles are identified: Rule modeler and rule interpreter

Rule modeler (as a basis for *K*): These are of two types; Enterprise modelers using METIS and modeler of the detailed executable rules modeled in Blaze Advisor by professional rule designers (both in cooperation with loan fund professionals). For defined changes the loan fund professionals could do this through the RMA.

Rule interpreter (vs. *I*): Both the enterprise model and the detailed rules in Blaze were to be understood by those involved in the modeling. All loan fund personal were to be able to understand the goal-model. 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 (with extension *L*) used for enterprise and rule modeling was EEML extended with the proprietary rule language SRL (Structured Rule Language) found in Blaze Advisor.

Tool: METIS (which can store and parse the EEML-models) and Blaze Advisor (which can execute the SRL rules).

Based on SEQUAL, we looked at the following areas in the evaluation relative to the goals for the representation of rules described in section 2

- Quality of the rule modeling language
- Quality of the developed rule model

4.2.1 Quality of the rule modeling language

- Domain appropriateness: It was possible to express all the execution rules in the selected area of the case formally in SRL. In some exceptional cases one had to implement parts of these in functions being programmed procedurally. Likewise it was possible to express the high-level rules, and relationships between rules in EEML. More generally, one can evaluate the language relative to emergent standards for rule languages. In connection to this there are a number of initiatives particularly within OMG and W3C
  - OMG's PRR – Production Rule Representation [17] – The standard is focused on the management of production rule sets e.g. the kinds of rules that execute in Blaze Advisor, JRules etc, with a first version release last year.
  - W3C's RIF– Rule Interchange Format [24] - this standard has a very large number of companies involved and is trying to decide how much detail about

the rules to manage in the interchange format. This is being coordinated with PRR. This would allow the interchange format for PRR to be RIF.

– OMG's SBVR – Semantics of Business Vocabularies and Rules [18] - this standard was finalized in the end of 2007. It is a standard designed to manage source rules and is very thorough/complex.

Whereas e.g. Blaze Advisor will probably support PRR rules, one does not support many aspects of SBVR rules including support of deontic operators in SRL. These notions are supported in the goal modeling of EEML, although a detailed evaluation of EEML relative to SBVR is yet to be done.

- Modeler appropriateness: The loan fund professionals were together with rule designers able to express the rules in SRL. EEML rules were also possible to express, although the goal modeling demanded a lot of discussion as for filling in the connections between rules on the different levels. Loan fund professionals were also able to use the RMA for rule maintenance.
- Participant appropriateness: EEML was in the start only known my modeling experts. Likewise, SRL was only known by rule designers in external companies, and it is found that especially SRL represents a steep learning curve both for Loan Fund professionals as well as system developers.
- Comprehensibility appropriateness: Those closely involved in the process appeared to understand the rules, especially since navigation was supported through 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 are somewhat hard to comprehend.
- Tool appropriateness: METIS was appropriate for handling integrated goal, resource and process models. Blaze Advisor was appropriate for rule execution, and other tests have been done supporting the scalability of the approach based on the possibility of executing the rules in the rule-base in different ways.
- Organizational appropriateness: EEML is a non-standard language. A positive aspect with SRL is that the language used is according to an emerging standard (PRR). On the other hand, both are supported by relatively expensive tools, with relatively little local (loan fund and Norwegian) expertise available. The last thing in particular relates to SRL.

4.2.2 Quality of the Rule Model

- Physical quality: The SRL-rules are primarily available through the tools, which limits the availability. Both SRL-rules and the METIS models could be made available in a web-interface, but when it comes to the SRL-model the web-report is not appropriate for widespread dissemination. RMA includes standard authorization mechanisms, ensuring that only authorized personnel can change the rules.
- Empirical quality: The goal model and process model visualization provided a useful way of getting an overview of the rule-base, being an improvement of just having a 'flat' rule-base.

- Syntactical quality: The METIS model and the rules implemented in the rule engine were syntactically correct
- Semantic quality: All the production rules were included in the SRL-model, and the rules as expressed in the underlying laws and regulations are included in the METIS-model. As described above, there is also a link between these representations, although it is currently not automatically supported.
- Pragmatic quality: It is relatively easy to keep an overview of the implemented rules and how they are related to the laws and regulations (through expressed meta-data). The introduction of the EEML goal-modeling 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 did not apply to the rules and regulations itself, but rather to how they should be follow up in practice in the Loan Fund.
- Organizational quality: The combined approach appears to support all stated goals, although better support could be envisaged for the second area. On the other hand, having the rules implemented in this way, makes it possibly to simulate different scenarios. The rule engine support that only certain rules are enforced at a certain time, thus one can easily simulate the effects of new rules (or alternatively, see how a case or a number of cases would be handled with a previous set of rules).

5 Related Work

Several advantages have been experienced with a declarative, rule-based approach to information systems modeling [8]:

- Problem-orientation
- Maintenance
- Knowledge enhancement

On the other hand, several problems have been observed when using a simple rule-format such as the one provided in most rule engines.

- Every statement must be either true or false, there is nothing in between.
- It is usually not possible to distinguish between rules of necessity and deontic rules [25].
- In many goal and rule modeling languages it is not possible to specify who the rules apply to, and who is the source of the rule.
- Formal rule languages have the advantage of eliminating ambiguity. However, this does not mean that rule based models are easy to understand. There are two problems with the comprehension of such models, both the comprehension of single rules due to the formal notation, and the comprehension of the whole rule-base. Whereas the traditional operational models (e.g. process models) have decomposition and modularization facilities which make it possible to view a system at various levels of abstraction and to navigate in a hierarchical structure, rule models are usually flat. With many rules such a model soon becomes difficult

to grasp, even if each rule should be understandable in itself. They are also seldom linked to other models of the organization used to understand and develop the information systems, such as data and process models.

- A general problem is that a set of rules is either consistent or inconsistent. On the other hand, human organizations may often have more or less contradictory rules, and one have to be able to deal with this.

Goals and rules have thus been used for knowledge representation in a wide variety of applications. An early example was the so-called expert-systems, which received great interest in the eighties. Unfortunately, these systems did not scale sufficiently well for large-scale general industrial applications. Lately, these approaches has reappeared and are in fact now able to deal with the processing of large databases (e.g. experiences with tools like Blaze Advisor, which is an extension of the Nexpert Object system that goes back to the late eighties have shown this. See http://www.brcommunity.org for an overview of current industrial solutions on this marked). Although being an improvement as for efficiency, they still have limited internal structuring among rules, and few explicit links to the other models underlying large industrial information systems. They seldom differentiate between deontic rules and rules of necessity, although this might be changing after the development of the OMG SBVR-standard which supports the modeling of deontic rules [18]. On the other hand, since the way of representing deontic notions in SBVR is not executable, it is possible that these aspects will be ignored by vendors of rule-based solutions such as Blaze Advisor since these largely focus on the execution of formal rules.

On the other hand, high-level rules *are* the focus for goal-oriented modeling in the field of requirements specification. Over the last 15 years, a large number of these approaches have been developed, as summarized in [6]. They focus on different parts of requirements specification work, including

- Understanding the current organizational situation
- Understanding the need for change
- Providing the deliberation context of the RE process
- Relating business goals to functional and non-functional system components
- Validating system specifications against stakeholder goals

The existing approaches do not bridge the areas of requirements specification and rule-based systems. Few differentiate between deontic rules and rules of necessity. A notable contribution of these techniques, are the structuring of goals and rules in hierarchies and networks, such as exemplified in e.g. [15]. Some of the approaches also link rules to other models, but with limited support of following up these links in the running system. An early example of such an approach was Tempora [12]. A way to combine the process data and rule-models in these languages as a basis for generating prototypes where developed [7]. In addition to linking the different models, one had the possibility of relating rules in rule hierarchies [13]. EKD, another successor of Tempora is reported in [1]. Although similar to our approach, the goal and goal-hierarchy in EKD do not include deontic notions. Also the organizational modeling is less expressive in EKD than EEML. Linking rules to actor-models is exemplified in techniques such as I* [26]. This technique on the other hand do not relate to standard process and data-models, nor to production rules.

6 Conclusion and Future Work

As discussed above several advantages have been experienced with a declarative, rule-based approach to information systems modeling:

- 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. The expression of the rules in SRL is to some extent hampered by the need 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, specifically for the communication with the stakeholders with a non-technical background.
- Maintenance: The benefits on this account is witnesses in the production-rule system, specifically with the added support of the RMA, although for many of the perceived needed changes to the rules one need rule designer expertise.
- Knowledge enhancement: The explicit rule-representation, and the possibility to quickly test their effect have proved beneficially in this area. The possibility to also relate the rules to more high-level goals in the rule hierarchy enables an even broader debate on the appropriateness of existing rules.

As for the identified limitations, many of these can be addressed by using EEML together with the formal rule language including:

- EEML rules can be partly fulfilled (although we are not supporting fuzzy logic reasoning).
- EEML rules include deontic operators.
- EEML-rules can be explicitly related to organizational actors.
- Rule hierarchies are supported, and it is also possible to link the rules to a hierarchical process model.
- EEML provide links to other models of the organization used to understand and develop the information systems, such as data and process models.
- It is possible to support contradictory rules and goals in EEML.

Thus in addition to be instrumental for the development of a rule-based system for rule automation, the combined approach can also support process-modeling and goal-oriented modeling as part of requirements specification work.

Further work is planned to be done on this approach in particular on the combined informal and formal modeling of legislation related to public service case processing systems. We would also like to get more experience on the approach in other domains, especially in networked organization where the goal structures emerges and change much quicker than in the public sector. As SBVR now is standardized, we will also look at aligning the goal-modeling part of EEML to this.

References

1. Bubenko, J.A., Persson, A. ,Stirna, J.(2001) HyperKnowledge deliverable D3: EKD User Guide., Dept. of Computer and Systems Science, KTH and Stockholm University, Sweden.: Kista. Available at http://people.dsv.su.se/~js/ekd_user_guide.html
2. Curtis, B., Kellner, M., Over, J. (1992) Process Modeling, Communication of the ACM, (35:9), September, pp. 75-90.
3. Davies, I., P. Green, M. Rosemann, M. Indulska et al. (2006) How do Practitioners Use Conceptual Modeling in Practice?, Data & Knowledge Engineering (58) 3, pp. 358-380.
4. Davis, A. M. Overmeyer, S. Jordan, K. Caruso, J. Dandashi, F. Dinh, A. Kincaid, G. Ledeboer, G. Reynolds, P. Sitaram, P. Ta, A. & Theofanos, M. (1993) Identifying and measuring quality in a software requirements specification. In Proceedings of the First International Software Metrics Symposium pages 141-152.
5. Hepp, M. and D. Roman. (2007) An Ontology Framework for Semantic Business Process Management. 8th International Conference Wirtschaftsinformatik, Karlsruhe, Germany, 2007, pp. 423-440 2.
6. Kavakli, E., & Loucopoulos, P. (2005). Goal Modeling in Requirements Engineering: Analysis and critique of current methods in Information Modeling Methods and Methodologies, In J. Krogstie, K. Siau, and T. Halpin, (Eds.) Idea Group Publishing
7. Krogstie, J., Seltveit, A. H., McBrien, P., & Owens, R. (1991). Information Systems Development Using a Combination of Process and Rule Based Approaches. in Proceedings of the Third International Conference on Advanced Information Systems Engineering (CAiSE'91). Trondheim, Norway: Springer-Verlag
8. Krogstie, J. and G. Sindre, Utilizing deontic operators in information systems specifications. Requirement Engineering Journal, 1996. 1: p. 210-237.
9. Krogstie, J., & Jørgensen, H. D. (2004). Interactive Models for Supporting Networked Organisations. in 16th Conference on advanced Information Systems Engineering. Riga, Latvia: Springer Verlag.
10. Krogstie, J., Sindre, G., & Jørgensen, H. D. (2006). Process models representing knowledge for action: A revised quality framework. European Journal of Information Systems 15 91-102
11. Krogstie, J. (2007) Modelling of the People, by the People, for the People. I: Conceptual Modelling in Information Systems Engineering. Berlin: Springer Verlag
12. Loucopoulos, P., McBrien, P., Schumacker, F. ,Theodoulidis, B., Kopanas, V., & Wangler, B. (1991) Integrating database technology, rule-based systems and temporal reasoning for effective information systems: the TEMPORA paradigm, Journal of Information Systems 1: 129-152.
13. McBrien, P., Seltveit, A. H., & Wangler, B. (1994). Rule Based Specification of Information Systems, International Conference on Information Systems and Management of Data, Madras, India.
14. Moody, D. L. & Shanks, G. G. (1994) What makes a good data model? Evaluating the quality of entity relationship models, in Proceedings of the 13th International Conference on the Entity-Relationship Approach (ER’94), pages 94-111, Manchester, England.
15. Mylopoulos, Chung, and Yu (1999) : “From Object-oriented to Goal-oriented Requirements Analysis”. Communications of the ACM, January
16. Nysetvold, Anna Gunhild; Krogstie, John. Assessing Business Process Modeling Languages Using a Generic Quality Framework. Advanced Topics in Database Research 2006;5:79-93

17.Nöth, W. (1990) Handbook of Semiotics Indiana University Press
18.OMG (2003) PRR – Production Rule Representation- Request for Proposal, Retrieved January 1 2006 from http://www.omg.org/cgi-bin/doc?br/2003-9-3
19.OMG (2006).Semantics of Business Vocabulary and Rules Interim Specification. Retrieved January 1 2006 from http://www.omg.org/cgi-bin/doc?dtc/06/03/02
20.Sedera, W., Rosemann, M. & Doebeli, G. (2003) A Process Modelling Success Model: Insights From A Case Study. 11th European Conference on Information Systems, Naples, Italy
21.Sølvberg, A., Kung. C.H.: On structural and behaviour modelling of reality. In Steele and Meersman, editors, Database Semantics. North-Holland, 1986
22.Vernadat, F. (1996) Enterprise Modeling and Integration. Chapman and Hall
23.Wand, Y., & Weber, R. (1993). On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. Journal of Information Systems 3(4), 217-237.
24.W3C (2005) RIF – Rule Interchange Format, Retrieved January 1 2006 from http://www.w3.org/2005/rules/
25.Wieringa, R. (1989) Three roles of conceptual models in information systems design and use. I E. Falkenberg and P. Lindgren (Eds.) Information Systems Concepts: An In-Depth Analysis pages 31-51 North-Holland, 1989
26.Yu, E. "Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering" (Proc. RE'97)