

DEPARTMENT OF COMPUTER SCIENCE
COLLEGE OF SCIENCES
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529-0247

**A DECISION MAKING METHODOLOGY IN SUPPORT OF THE
BUSINESS RULES LIFECYCLE**

By

Dr. Chris Wild, Principal Investigator
And Ms. Daniela Rosca
Department of Computer Science

FINAL REPORT

For the period ending December 31, 1996

Prepared for

NASA Langley Research Center
Attn.: Mr. R. Todd Lacks
LaRC Grant Officer
Mail Stop 126
Hampton, VA 23681-0001

And

NASA Langley Research Center
Attn.: Dr. Dave E. Eckhardt
Technical Officer
Mail Stop 152D
Hampton, VA 23681-0001

Under

NASA Contract NAG1-1426
ODURF Project No. 126581

Submitted by

Old Dominion University
800 W. 46th Street
Norfolk, VA 23508

October 1998

A Decision Making Methodology in Support of the Business Rules Lifecycle

Daniela Rosca

Chris Wild

Department of Computer Science

Old Dominion University

Norfolk, VA 23529-0162

January 16, 1997

Abstract

The business rules that underlie an enterprise emerge as a new category of system requirements that represent decisions about how to run the business, and which are characterized by their business-orientation and their propensity for change. In this report, we introduce a decision making methodology which addresses several aspects of the business rules lifecycle: acquisition, deployment and evolution. We describe a meta-model for representing business rules in terms of an enterprise model, and also a decision support submodel for reasoning about and deriving the rules. The possibility for lifecycle automated assistance is demonstrated in terms of the automatic extraction of business rules from the decision structure. A system based on the metamodel has been implemented, including the extraction algorithm.

This is the final report for Daniela Rosca's PhD fellowship. It describes the work we have done over the past year, current research and the list of publications associated with her thesis topic.

1 Past work on fellowship

During the academic year 1995-1996 Ms. Rosca's accomplishments were divided among the work on her thesis and professional activities for the

scientific community. The work done on her thesis produced the following results:

- designed the BRADES (Business Rules Acquisition Deployment and Evolution System) methodology for decision support of business rules elicitation, deployment and evolution at the enterprise level.
- defined a meta-model for representing business rules in terms of an enterprise model and also a decision support model for reasoning about and deriving the rules.
- demonstrated the possibility for lifecycle automated assistance in terms of the automatic extraction of business rules from the decision structures.
- exemplified the importance and usefulness of decision support during business rules deployment.
- developed tool support for the methodology.

Also, during the Summer 1995 Ms. Rosca was a visiting research scientist at GTE Laboratories, Waltham, MA. There she investigated the suitability of the BRADES methodology in the context of in-house business process reengineering projects that required the acquisition and documentation of business rules. Also, she designed and developed modules of a prototype that implements the methodology.

Her professional activities during this period of time can be summarized as follows:

- Co-organizer of The Third Workshop on Artificial Intelligence and Software Engineering held in conjunction with IJCAI'95, Montreal, Canada, 1995.
- Session chair:
 - * The Software Engineering and Knowledge Engineering Conference (SEKE'96), "Code Reengineering" session, Lake Tahoe, 1996.
 - * The Third Workshop on Artificial Intelligence and Software Engineering, "Reuse" session, Montreal, Canada, 1995.

- Program Committee member for:
 - * The Knowledge Based Software Engineering Conference (KBSE'96),1996
 - * The Third Workshop on Artificial Intelligence and Software Engineering,1995
- Reviewer for the Journal of Automated Software Engineering.

1.1 A Metamodel for the Business Rules Environment

The methodology supporting the business rules lifecycle consists of

- a modeling framework (metamodel),
- a prescription of activities for populating the models, and
- techniques for using the information for requirements analysis, including continuous lifecycle support.

This section describes the metamodel, in preparation for describing the methodology activities in the next section. The metamodel described here has been implemented in our experimental requirements modeling/analysis environment for supporting the methodology. The environment consists of three submodels: the *Enterprise Model*, the *Business Rules*, and the *Decision Space*. The *Enterprise Model* represents the world to which the business rules apply. It defines the domain concepts about which the rules are expressed. The *Business Rules* submodel represents the business rules themselves. The *Decision Space* submodel offers information about the enterprise objectives that comprise the origin of business rules and captures the reasoning leading to the selection and ultimate generation of the business rules. (see Figure 1).

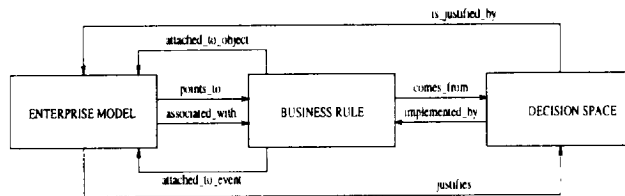


Figure 1: The Business Rules Environment

1.1.1 Enterprise Model

For the representation of the *Enterprise Model* we have chosen the paradigm of the LiveModel modeling environment [4]. In LiveModel, an enterprise is represented in terms of “objects” and “processes” (see [5]). Objects are represented by a set of Object Diagrams that are essentially Entity-Relationship diagrams (the Object diagram for a fragment of the LAS example can be seen in Figure 2). The business processes are represented by a set of Event Diagrams which define the sequence of operations for process execution (the corresponding Event Diagram for a fragment of the LAS example can be seen in Figure 3.) These Event Diagrams model a hierarchy of business processes, decomposing each operation in a diagram, if necessary, into a more detailed diagram. The Event Diagrams are executable specifications of a process as soon as: 1) input and output variables to operations are specified; 2) trigger rules are created to define branching and control conditions; 3) procedures to define operations are written. LiveModel allows the attachment of rules to event diagrams, with a particular operational semantics based on those of the Object and Process diagrams.

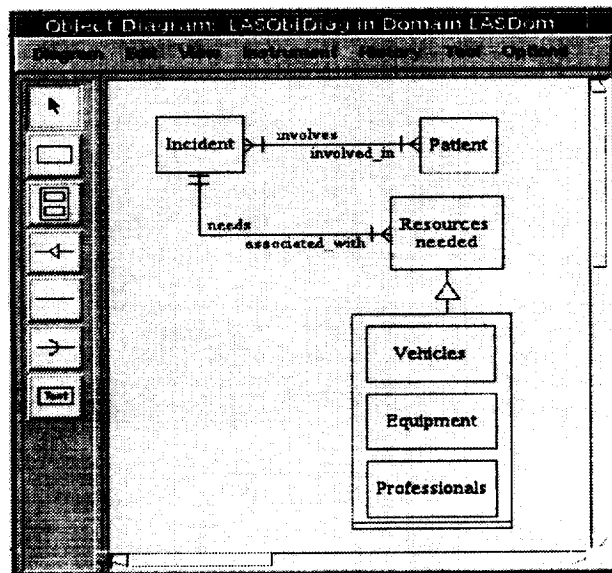


Figure 2: An Object Diagram for a fragment of the LAS example

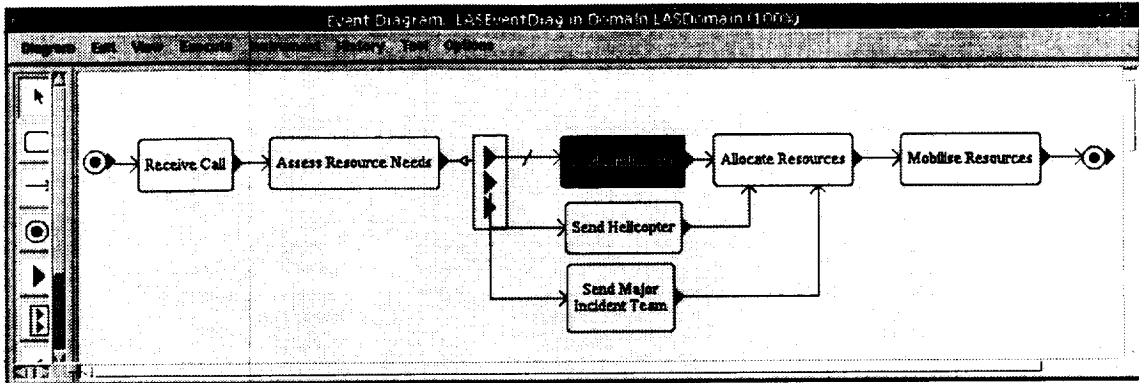


Figure 3: An Event Diagram for a fragment of the LAS example

1.1.2 Decision Space Submodel

The Decision Space submodel is shown in figure 4. It represents the primitives of an issue-based decision support model. We can interpret our decision support model as follows. Both functional and non-functional requirements generate *issues* that need to be solved. These issues are refined during the deliberation process. In order to solve an issue different alternative solutions are considered for evaluation. The *alternatives* are evaluated against a set of *criteria* in order to decide which gives the best solution. A decision involves assessing the degree to which each alternative meets the entire set of criteria and choosing that alternate which best satisfies this set. *Arguments* and counterarguments based on various *assumptions* are recorded to document the evaluation of the alternatives or the creation of new issues that may follow after making a decision. The best alternative solution is reflected in the resulting artifact, which in our case is represented by *DSS Business Rules*, a set of business rules in decision support system (DSS) format. All of the information content of the above primitives can be retrieved from the *decision matrix* associated with a specific issue. A more detailed description of this model and related work on decision support structures is given in [7]. Here we show an augmented model with the links to other submodels of the business rules environment: the *Enterprise Model* and the *Business Rules*.

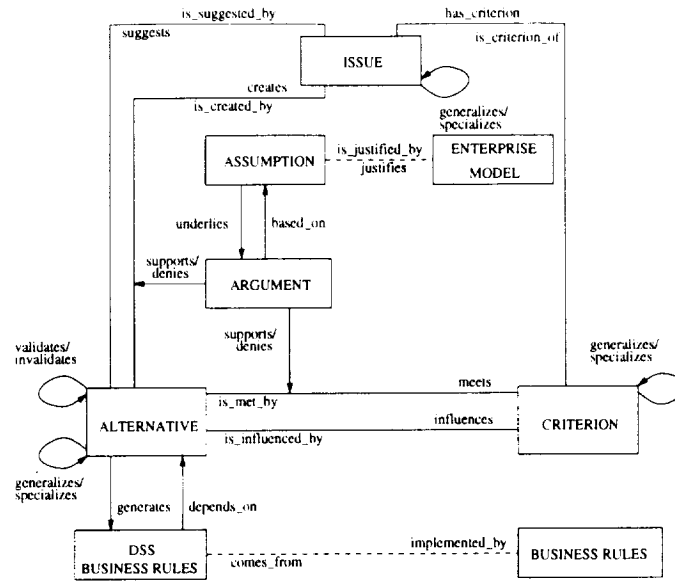


Figure 4: Decision Space Submodel

1.1.3 Business Rules Submodel

Business rules take the form of event-condition-action (ECA) rules, which we adopted from [3]. The ontology of the Enterprise Model, when examined at a more detailed level, contains events, conditions and actions. Whether a process/object enterprise model is used, as in this paper, or the extended SOS model described in [2] is used, ECA rules provide a convenient “assembly language” into which most kinds of rules can be translated. Since the ECA rules have a well-defined operational semantics, it has been straightforward to build an interpreter for them.

For the simplest form of an ECA rule

WHEN *event*
 IF *condition*
 DO *action*

when the event occurs, if at that time the condition is found to hold, then the action is initiated.

The events, conditions, and actions are formulated as expressions on the objects in the Enterprise Model.

ECA rules are more generally applicable than they might first appear. As discussed in [1], where similar rules are used, judicious interpretations of

special cases (such as default meanings for omitting one of the components of the rule) allow ECA rules to express several of the business rules types in the taxonomy. Additionally, an ECA rule can be used to express non-operational semantics, such as the situated enterprise objectives expressed at the *criteria level* in section 1.3.2.

1.1.4 Intermodel Relationships

Requirements analysis can be done by analyzing interrelationships between the submodels. Based on the links between these submodels the following types of analysis can be performed:

Business Rules → Enterprise Model:

Which process component(s) does a business rule define/constrain/govern? Which (event/action) operations operationalize a business rule? What object types are referred by a business rule? This information can be used for an impact/sensitivity analysis when a rule changes.

Enterprise Model → Business Rules:

In which business rules does a specific object type participate? This information can be used for impact/sensitivity analysis when the status of an object changes.

What business rules define/constrain/govern a specific process component? This information can be used for business processes improvement.

Business Rules → Decision Space:

Where does the rule come from? This links a business rule to the issue that has generated it. Thus one can have a comprehensive picture of the business rule rationale by looking at the alternatives, criteria, arguments and assumptions that have been stated during the deliberation of that business rule.

Decision Space → Business Rules:

What business rules address a specific issue? This information allows an impact/sensitivity analysis when factors like Government regulations, company policies, etc. change. It is also a useful source of information for a reuse process.

Decision Space → Enterprise Model:

What object types/attributes are addressed by a decision/issue? This can be useful for an impact/sensitivity analysis when a decision is changed.

Enterprise Model → Decision Space:

What decisions/issues involve this object or attribute? What decisions are affected when an object changes?

1.2 Methodology

The methodology we propose spans all phases of the business rules lifecycle: acquisition, deployment, change in response to changes in internal or external influences and change based on evaluation of the degree of requirements satisfaction.

1.2.1 Business Rules Acquisition

We see three major steps in the acquisition of business rules: the initial analysis, the analysis and generation of business rules in different areas of expertise, and final analysis (see Figure 5). During the initial analysis, brainstorming sessions take place for deliberating which are the goals, policies and constraints of the business that need to be modeled. As a result of these deliberations, initial versions of the enterprise model and decision space are sketched and also a first set of business rules that specify how the business should be run is defined. Because they define the goals of the enterprise, these are *strategic business rules* (in the next section, we will find these rules at the *criteria level* rules) that express very high level decisions. These rules need to be refined in order to become operational.

The first step in the refinement of business rules is the analysis and rule generation in different areas of expertise. In this phase business analysis is carried out by separate groups of people, with different areas of expertise, for refining the understanding of business entities, processes and business rules. These activities imply more detailed discussions on the ways of achieving the goals, policies and constraints of the business. They can be complemented with interviews with domain experts and/or reading existing documentation and information related to the subject of analysis. As the understanding of the enterprise objectives becomes clearer the Enterprise Model and the Decision Space are updated.

Based on the entities and processes stated in the Enterprise Model, on the decision structures captured in the Decision Space, and on statistical data from the enterprise's way of doing business, business rules (decision support system level rules, or DSS rules) can be automatically extracted

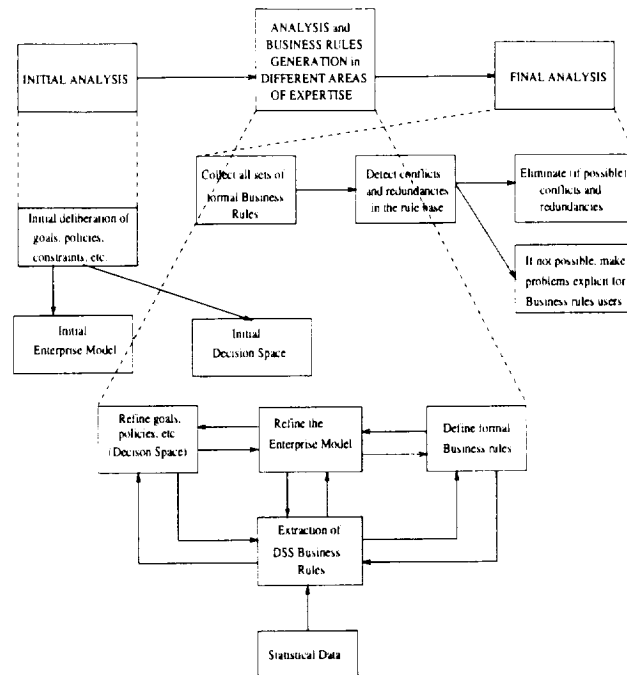


Figure 5: Business Rules Acquisition

following an algorithm described in detail in the next section. These rules (called *arguments*, respective *assumptions level rules*) are more concrete than the strategic rules. They underly the structure of operational rules in ECA format that are expressed in a formal rule language, like the one used in the Livemodel tool. At the end of this step, a formal business rule will be defined for each alternative solution in the Decision Space and will be available for deployment. For process simulations, these rules can be attached to operation triggers in process diagrams, like the ones defined in Livemodel. See the slash mark on the Figure 3 for an example of rules attachment to an event diagram and Figure 6 for an example of stimulus/response (trigger) rule implemented in Livemodel (the example has been oversimplified for presentation purposes).

There can be multiple iterations on each operation of this step until a stable set of business rules, as well as a clear and comprehensive Enterprise Model for each specific area of expertise, are obtained.

During the final analysis of the business rules acquisition all of the existing sets of business rules, Decision Spaces, and parts of the Enterprise Model are put together, leading to the detection of redundancies and conflicts. The

```

Call Debug View
Trigger Rule: SendAmbulanceRule
DURING LASEventDiag
WHEN Assess Resource Needs with
?Incident and ?Patient is
finished yielding ?Resources
needed
SO THAT ((Patient has a
Breathing 18 and
Patient has a Pulse equal to
high) or Incident has a distance
Less than 22)
THEN DO Send Ambulance

```

Figure 6: Example of stimulus/response (trigger) rule in the LAS case study

detection is facilitated by the types of analyses discussed in section 1.1. These redundancies and conflicts are either eliminated, or if not possible, made explicit to the designers, developers and users of the information system that will underlie the business and that will incorporate these business rules, or to the users of people oriented business rules.

1.2.2 Business Rules Deployment

After the business rules are defined and integrated into the enterprise process model they become operational. Therefore, whenever a new case is run through a process model inside the enterprise, there are a couple of situations that can arise in the application of business rules (see Figure 7):

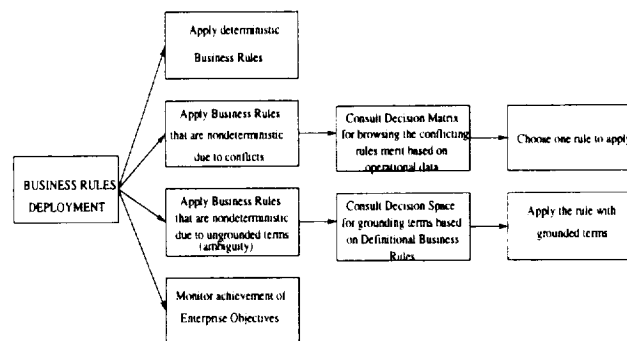


Figure 7: Business Rules Deployment

1. The situation is deterministic, e.g. characterized by a single business rule and the data referenced by the rule are known with certainty. Therefore that rule can be automatically applied by the underlying information system, either by people or by machine.

2. The situation is characterized by multiple, conflicting business rules. In these cases we can show the decision matrix associated with those rules and let the user browse through it, analyze the information contained in the decision structures and assess the merit of each alternative associated with each rule. The user can choose one of the proposed rules or apply his own judgment and select another rule. This way the decision of which rule is the best has been shifted from the analysis time to operation time, when concrete data about a case is available.

3. If the applicable rule(s) is(are) ambiguous, e.g. they contain ungrounded terms whose grounding couldn't be done with certitude at analysis time we can show the interpretation of these terms using the links among the business rules, the Decision Space and the Enterprise Model. This way the definitional business rules attached to various attributes of the entities in the business can be made available for consulting. The user can choose one of the legal values of an ungrounded term based on the definitional business rules or can disagree with those rules and choose a value according to his own judgment. This approach permits development to move forward even when requirements are not fully understood.

4. For evaluating how well the enterprise objectives are achieved we propose instrumenting the system with monitoring to check whether the assumptions underlying the business rules are valid. This information is fed back to the system for updating the business rules, enterprise model and decision space.

1.2.3 Business Rules Evolution

There are several possibilities for improving the business rules based on the information captured in the methodology framework. Data obtained through monitoring of the operational system can be used to study the validity of assumptions recorded in the Decision Space, leading to changed rules. New sources of information, both inside and outside of the enterprise, may arise. New solutions may be chosen by users for resolving conflicting or ambiguous situations. For example, by studying the Decision Space one can detect that some criteria, alternatives or arguments could be added/eliminated, or that their current weights were wrong. Or, by tracing back the rules applied, we can detect that some attributes are missing or should be added for more accurate business rules.

Therefore the Enterprise Model and the Decision Space is continuously updated for keeping the pace with the constant changes that occur both inside the enterprise and in the outside world.

Depending on the nature of change observed, business rules are changed by either choosing other existing rules, modifying existing rules or creating new rules if none of the existing ones meet the new context coordinates.

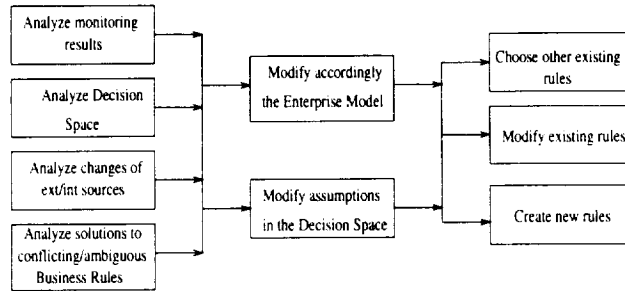


Figure 8: Business Rules Evolution

1.3 Automatic Extraction of Business Rules from Decision Structures and Examples

This section discusses how we support the automatic extraction of business rules. Our objective is to generate a set of business rules that preserve the information in the decision matrix and to reflect knowledge from statistical data about domain assumptions.

1.3.1 Decision Structure Knowledge

Next we will formalize the knowledge contained in the decision structures represented by a decision matrix. This knowledge is used in the rules extraction algorithm, and corresponds to the primitives described in the decision support submodel in Section 1.1. These primitives are naturally expressed in terms of variables and constraints on their values. Thus we distinguish the following variables:

- The *issue* (Iss) that needs to be solved: *Assess Resource Needs* in our example.
- A number of *alternatives* (Alt_i) that are proposed as solutions to the issue Iss . For example, $Alt_1 = Send\ Ambulance$.

- A set of *criteria* ($Crit_j$) against which all the alternative solutions are evaluated in order to decide upon the best alternative. For example, $Crit_1 = Quick Response Time$.
- A number of pro and counter *arguments* that correspond to each pair (alternative, criterion) ($Arg_k(Alt_i, Crit_j)$). For example, $Arg_1(Alt_1, Crit_1) = Slow Response$, related to the more intuitive description “This alternative is too slow in life-threatening cases”.
- A set of *assumptions* that support each argument ($Attr_l(Arg_k)$). Assumptions represent groundable attributes of one or more objects in the Enterprise Model that are relevant to the issue under consideration. For example, $Attr_1(Arg_3(Alt_1, Crit_1)) = Systolic Blood Pressure$. Some attributes may be common for various arguments.

1.3.2 Format of the Business Rules Extracted from Decision Structures and Statistical Data

We are proposing three types of business rules that correspond to different levels of detail. They are the rules obtained at the criteria level, at the arguments level and at the assumptions level of a decision matrix. These different business rule types correspond to different levels of decision making in the hierarchy of an enterprise.

1. Criteria level:

The rules obtained at this level are the most general type of rules. They correspond to high level decision making and express enterprise objectives in very general terms. Therefore these objectives will need to be refined to the point where they can be translated into operational business rules. These objectives will serve as criteria for evaluating the alternatives of various solutions proposed for solving various problems.

For example, in the rule

```

WHEN Assess Resource Needs
IF Quick Response Time [0.6]  $\vee$  Effective Resource Usage [0.4]
THEN Send Ambulance

```

that expresses what solution (alternative) to choose (*Send Ambulance*) for solving the problem *Assess Resource Needs* for an incident, the criteria established by the enterprise for solving this problem are *Quick Response Time* of the resources sent to an incident site and *Effective Resource Usage*. As we

can see, there is no precise definition yet of what the non-functional requirements *Quick Response Time* and *Effective Resource Usage* really mean. The only thing we know at this stage is the importance of each criterion (0.6 and 0.4) in evaluating the alternatives.

The general format of this type of rule is:

```

WHEN Iss
IF  $Crit_1[w_1] \vee Crit_2[w_2] \vee \dots \vee Crit_n[w_n]$ 
THEN  $Alt_i$ 

```

where w_i represent the weights or *Importance* of criterion $Crit_j$ in the process of alternative Alt_i evaluation. The w_i values are given by the decision makers.

Whenever this type of rule is applied we compute the merit of the alternative given in the action part of the rule (Alt_i), based on the weight of each criterion (w_{Crit_j}) and the merit $Merit_{Crit_j}$ of the alternative Alt_i in satisfying each criterion $Crit_j$:

$$Merit_{Alt_i} = \frac{\sum_{j=1}^{n_{crit}} w_{Crit_j} * Merit_{Crit_j}}{n_{crit}} \quad (1)$$

2. The arguments level:

The rules at this level express the heuristics used in deciding how well an alternative satisfies a criterion when several arguments are presented for, or against a solution (alternative). They combine the evidence about the merit of each argument Arg_k correlated with a pair ($Alt_i, Crit_j$), in order to compute the $Merit_{Crit_j}$ of the alternative Alt_i against criterion $Crit_j$. These rules express the fact that the meaning of the enterprise objectives is not always obvious, and therefore requires negotiation among stakeholders.

For example, in the rule

```

WHEN Send Ambulance
IF  $Slow Response = True [-1.0] \vee Acceptable Response = True [0.3] \vee$ 
 $Quick Response = True [1.0]$ 
THEN Quick Response Time

```

we express the fact that in the process of refining the meaning of the *Quick Response Time* objective there have been brought up three arguments with different weights in the context of sending an ambulance to an incident. These arguments correspond to different situations perceived by various stakeholders as being plausible: 1) sending an ambulance may be too slow in life-threatening cases where the location of the incident is far away

from a hospital (and therefore, this is a counterargument for sending an ambulance in these cases); 2) sending an ambulance may be acceptable in life-threatening cases if they are close to a hospital; 3) sending an ambulance is a quick solution in non-life-threatening situations, regardless of the distance from the hospital (and therefore gives a stronger support to the alternative than argument 2)). All these arguments (and others corresponding to other criteria in the level 1 rule) will be taken into consideration when making the decision about whether sending an ambulance.

The general format of this type of rule is:

WHEN Alt_i
 IF $Arg_1[w_1] \vee Arg_2[w_2] \vee \dots \vee Arg_n[w_n]$
 THEN $Crit_j$

w_i represent the weight of each argument in the evaluation process. It can take values on a scale $[-1.0, 1.0]$ meaning: when $w_i = -1.0$, Arg_i is a counterargument, while when $w_i = 1.0$, Arg_i is a strong supporting argument.

The computation of $Merit_{Crit_j}$ is based on the weight of each argument (w_{Arg_i}) and the predicted accuracy of the truth value of that argument ($Merit_{Arg_i}$):

$$Merit_{Crit_j} = \frac{\sum_{i=1}^{n_{arg}} w_{Arg_i} * Merit_{Arg_i}}{n_{arg}} \quad (2)$$

3. The assumptions level:

This is the most detailed level of rules where the business objectives find their operational meanings. Even though they correspond to operational (low level) decision making, there might still be situations that require grounding of some terms inside the rules.

These rules express the operational conditions that need to be met in order for the alternative Alt_i to meet the criteria $Crit_j$ (an enterprise goal), e.g. they assess the truth value of the arguments Arg_i associated with the pair $(Alt_i, Crit_j)$ based on various domain assumptions. The subconditions in the antecedent part of the rules are obtained either automatically by induction from statistical data, or, when this data is not available, by asking the decision maker. Even though these rules might look like the rules of an expert system, they are in fact business rules that achieve a goal.

For example, the rule

IF (*Diastolic Blood Pressure* = High \wedge *Systolic Blood Pressure* = High)
 \vee (*Distance from Hospital* < 28)
 THEN *Quick Response* = True [99.9%]

expresses the operational conditions for the argument *Send ambulance is a quick solution in non-life-threatening situations* to be true. More than that, it shows the perceived accuracy (99.9%) of this assessment.

The condition part of this type of rules is a disjunctive normal form (DNF) formula that contains various assumptions of the argument under consideration. An assumption has the format $As \equiv (Attr < op > value)$, where $< op > = \{=, <=, >\}$. For instance, a rule whose condition has two disjunctive terms is

$$\begin{aligned} & \text{IF } (As_1 \wedge As_2 \wedge As_3) \vee (As_4 \wedge As_5) \\ & \text{THEN } Arg_i [Merit_{Arg_i}] \end{aligned}$$

where $Attr_i$ represent the attributes whose values v_i need to be checked in order to assess the truth value of the argument Arg_i . The $Merit_{Arg_i}$ describes the certainty factor about the truth value of argument Arg_i .

1.3.3 Automatic Generation of Business Rules

The method for the automatic generation of business rules uses the knowledge structure provided by the decision support system through decision matrices. Also it uses statistical data recording how domain assumptions support various arguments.

We distinguish two types of automatically created business rules. The first type are the business rules that capture the heuristic knowledge from the decision matrix. These rules correspond to levels one and two from above. The second type are business rules that are extracted from decision trees induced from statistical data by applying inductive learning techniques. They correspond to level three rules from above.

Decision tree learning is a supervised machine learning technique that uses a collection of training examples and outputs a compact decision structure called a decision tree. The internal nodes in a decision tree correspond to tests on the values of particular attributes, while the leaves correspond to class values. Decision trees logically correspond to a disjunction of conjunctions. They can be further generalized into business rules of level three using a technique of extraction of rules from decision trees (see [6, 8]).

The rules at levels two and one can be automatically generated from the decision structures represented in a decision matrix. The merit of each alternative in the decision matrix can be computed according to the formulae 1 and 2.

For obtaining the rules at level three we apply an algorithm for the induction of decision trees and rules from statistical data [6, 8]. The attributes used for classification are the domain assumptions that underlie an argument. The algorithm tolerates missing values for some of the attributes, therefore allowing for imprecise information about a case. The classes resulted after the application of the algorithm over a data base of assumptions values are the truth values for an argument. Each class has associated the certainty factor of that classification, which represents the likelihood that an argument value is true or false. From an induced decision tree we extract and optimize rules (both at the rule level and rule sets level) that will represent the level three type of rules.

The result of the antecedent phase is a set of rules (DSS rules) associated with every issue ($DSSR(Iss)$). Each such set of rules can be transformed into an operational business rule that follows the ECA format. An operational rule associated with an issue Iss is obtained from $DSSR(Iss)$ by applying the operation of consequent expansion at levels one and two. Consequent expansion means replacing a condition on a variable v in the antecedent of a rule, with the antecedent of the rule that has v as a consequent. If several such rules exist they are combined in an OR logical operation. The weights of the new conditions generated through consequent expansion depend on their initial weights, certainty factors of the rules expanded, and the number of rules expanded.

This way the enterprise objectives (stated in level one DSS rules) are refined to the point where they can be translated into operational business rules that achieve the enterprise goals.

1.4 Decision Support for Business Rules Deployment

After the business rules are defined and integrated into the enterprise process model they become operational. Therefore, whenever a new case is run through a process model defined for a specific enterprise, we see a couple of situations happening in the application of business rules (see Figure 7):

1. Apply deterministic BRs. There is no need for decisions since the rules are applied automatically by the system or people.
2. Apply BRs that are nondeterministic due to conflicts (overspecified situations). For solving the conflicts we introduce the operational de-

cisions that will take into consideration the data available at operation time that was not available at analysis time.

3. Apply BRs that are nondeterministic due to ungrounded terms (ambiguous situations due to subjective judgment). In this case we propose the use of operational decisions for grounding the ambiguous terms.
4. For the situations where the desired enterprise objective, its assumptions, or the desired state of its informational system cannot be clearly articulated, the generated business rules have a conditional flavor. The decisions generated in these cases are conditional decisions and they can be applied to all the three situations above. For these cases we propose monitoring the BRs deployment, looking specifically at the validity of their assumptions related to attributes of objects defined at analysis time. The results of the monitoring activity will represent one of the sources of business rules evolution.
5. In exceptional cases where there is no applicable rule (incomplete set of business rules), we propose to help the user to make an operational decision by presenting rules applicable to similar situations. It is not our purpose here to follow this thread into more details.

2 Current work

Currently we are working on proposing ways for handling business rules evolution. With this respect we need to take into account the various sources of business rules evolution, such as: analysis of the monitoring data, change in the context of a decision (both inside a decision and outside sources of change) and learning from operational decisions.

After this last goal of the thesis will be completed we estimate that the thesis defense will take place by the end of the Spring 1997 semester.

3 List of Publications

This is Ms. Rosca's list of refereed publications connected with her thesis work.

Conference Publications

1. Daniela Rosca, Sol Greenspan, Mark Feblowitz, Chris Wild **A Decision Support Methodology for the Business Rules Lifecycle**, The International Symposium on Requirements Engineering (RE'97), January 1997, Annapolis, MD.
2. Daniela Rosca, Chris Wild, **Business Rules in the Real World: A Decision Support Approach**, The Software Engineering and Knowledge Engineering Conference (SEKE96), June 1996, Lake Tahoe, NE.
3. Daniela Rosca, Sol Greenspan, Chris Wild, Howard Reubenstein, Kurt Maly, Mark Feblowitz, **Application of a Decision Support Mechanism to the Business Rules Lifecycle**, The 10th Knowledge-Based Software Engineering Conference (KBSE95), November 1995, Boston, MA.
4. Chris Wild and Daniela Rosca, **Evolution and Reuse of Formal Specifications Using Decision Structures**, The 9th Knowledge-Based Software Engineering Conference (KBSE94), September 1994, Monterey, CA.
5. Chris Wild Kurt Maly, Chenglin Zhang, Cathy C. Roberts, Daniela Rosca, and Tamara Taylor, **Software Engineering Life Cycle Support - Decision Based Systems Development**. IEEE TENCOM 10, Singapore, August 1994, pp. 781-784.
6. C. Wild, C. Zhang, D. Ray, I. Levinstein, S. Olariu, C. Overstreet, N. Rao, D. Rosca, T. Taylor, T. Ireland, D. Sibol, **A Decision-based Hyper-multimedia CASE Environment for the Computer Productivity Initiative, a Large Undergraduate Project**, World Conference on Educational Multimedia on Hypermedia, June 1994, Vancouver, Canada.
7. Christian Wild, Kurt J. Maly, Chenglin Zhang, Dave E. Eckhardt, Cathy C. Roberts, Daniela Rosca, and Tamara Taylor, **Project Management Using Hypermedia CASE Tools** 1994 Conference on Data and Knowledge Systems for Manufacturing and Engineering, 1994, pp. 722-727, Hong Kong.

Workshop Publications

1. Daniela Rosca, Chris Wild, Sol Greenspan, Howard Reubenstein, Mark Feblowitz. **Decision Support for Business Rules Development and Evolution**, The Third Artificial Intelligence and Software Engineering Workshop held in conjunction with IJCAI'95, August 1995, Montreal, Canada.
2. Daniela Rosca, Chris Wild, **Analysis of Decisions Modification Impact in Specifications Evolution**, The Artificial Intelligence and Software Engineering Workshop held in conjunction with ICSE'94, May 1994, Sorrento, Italy.
3. Mala Mehrotra, Chris Wild, Daniela Rosca, **Role of Clustering Analysis in the Verification and Validation of Expert Systems**, The Verification and Validation Workshop held in conjunction with AAAI'92, San Jose, CA.

References

- [1] J. Bubenko Jr. and B. Wangler. Objectives driven capture of business rules and of information systems requirements. In *Proceedings of the International Conference on Systems, Man and Cybernetics*, pages 670–677, 1993.
- [2] S. Greenspan and M. Feblowitz. Requirements engineering using the SOS paradigm. In *Proceedings of IEEE International Symposium on Requirements Engineering, San Diego*, 1993.
- [3] H. Herbst. A meta-model for specifying business rules in system analysis. In *Proceedings of CAiSE'95*, pages 186–199, 1995.
- [4] Intellicorp. *Livemodel User's Guide, Betaversion*, 1995.
- [5] J. Martin and J. Odell. *Object-Oriented Methods: A Foundation*, chapter 20. Prentice Hall, 1995.
- [6] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

- [7] D. Rosca, S. Greenspan, C. Wild, H. Reubenstein, K. Maly, and M. Feblowitz. Application of a decision support mechanism to the business rules lifecycle. In *Proceedings of the KBSE95 Conference*, pages 114-122, 1995.
- [8] J. Rosca and D. Rosca. Knowledge acquisition facilities within an expert system toolkit. In *Proceedings of the Seventh International Symposium on Computer Science*, Jassy, Romania, 1989.