Article

A Business Rules Management System for Fixed Assets

Sabina-Cristiana Necula [ID]

Department of Accounting, Statistics and Business Information Systems, Faculty of Economics and Business Administration, Alexandru Ioan Cuza University of Iasi, Iasi 700505, Romania; sabina.necula@uaic.ro

check for updates

Abstract: The goal of this paper is to discuss the necessity of separating decision rules from domain model implementation. (1) Background: can rules help to discover hidden connections between data? We propose a separated implementation of decision rules on data about fixed assets for decision support. This will enhance search results. (2) Methods and technical workflow: We used DROOLS (Decision Rules Object Oriented System) to implement decision rules on the subject of accounting decisions on fixed assets; (3) Results: Making the model involves: the existence of a domain ontology and an ontology for the developed application; the possibility of executing specified inferences; the possibility of extracting information from a database; the possibility of simulations, predictions; the possibility of addressing fuzzy questions; and (4) Conclusions: The rules, the plans, and the business models must be implemented to allow specification of control over concepts. The editing of meta models must be directed to the user to ensure adaptation and not implemented at the level of control of the data control.

Keywords: decision rules; decision support system; DROOLS

---

1. Introduction

The first approach in modeling decision-making process was the rational approach, which considered that decisional choices are available and their effects are known and evaluated in terms of utility. Simon et al. [1] stated that decisions are taken by actors that are led by the purpose of achieving a maximum benefit by having a satisfactory behavior. The scientific management of Taylor et al. [2], the administrative theory of Fayol et al. [3], and M. Weber's [4] bureaucracy are good examples of rational-based systems. The biased rational approach considers that the decision-making process concerns many actors that are rational-biased. In this kind of system, obtaining information is important.

Problem-solving led to the emergence of many IT solutions for business management, in more or less advanced forms of integration.

From an informational point of view, decisions are inputs for other decisions. Decisions' alternatives overlap with inputs for other decisions. Models intertwine and form a graph of concepts in causal and constraint relationships that determine decision-making contexts.

The decision support systems heavily rely upon large volumes of data, information, and knowledge arriving from different sources [5].

Although it is not considered good practice, many real-world process models contain detailed decision logic, encoded through control data structures. This often results in spaghetti-like and complex process models and reduces maintainability of the models [6].

Providing information is the goal of any decision support system. Modeling decisions is a concern for formalizing problem solving and involves knowledge formalisms. Most often, computer science proposes breaking the problem into sub-problems. A computer program implements functions. The decision maker uses the functions and formalizes the structured part of the decisional problem. Artificial Intelligence seeks to discover the functions, and, in this way, seeks to give the decision makers the knowledge. The decision

maker uses knowledge discovery to create new knowledge and proposes an approach to model the decision by enunciating logical inferences. DROOLS (Decision Rules Object Oriented System) [7] is a business rules management system that separates the domain level from the decision level, so that decision makers infer rules on data and query results.

Knowledge is expressed by inference rules, and rule execution will infer the result of logical implications in the knowledge base. In literature, the implementation of rules is considered to be feasible with inference engines, considered programs that have control, procedural or operative knowledge, exploiting the knowledge base, and are designed to combine and chain knowledge to infer new knowledge through judgments, plans, demonstrations, decisions, and predictions.

Concepts from business rules are used in the specifications needed to define control. This paper presents a separated decision rules implementation from the domain model with the main purpose of establishing good practices for developing decision support systems.

2. Methods

Our article has a research hypothesis:

Hypothesis 1. *Rules help to discover hidden connections between data on fixed assets. We propose a separated decision rules implementation on data about fixed assets. This implementation will enhance search results. In order to pursue this endeavor, we intend to build a rule-based application interface for editing and executing rules on data in an expressive manner.*

Inferential Engines have special interaction interfaces with programming languages and run rules to solve a problem [8]. The rules must be stored separately and defined on the concepts of a common ontology for the members of the organization involved in the same process. The rules provide answers close to the members of the organization who ask questions (queries) on the concepts defined in the common ontology.

As for the decision-making process, we can say that there are two general principles accepted in decision-making:

(1)   The decision is made in terms of gain-loss;
(2)   The decision depends on the context; even if the decision-making rules are known, the context of the decision may be different, which means that the meaning of the concepts involved in the decision-making rules changes.

We try to introduce the idea that the specification of rules and constraints of concept use should be part of the level of knowledge modeling rather than the functional level of the developed application. Specifying constraints at the data processing level contextualizes the problem and limits the modeling domain. Specifying constraints at the place and time of manifestation of their knowledge characterizes the context and adapts the decision model.

We chose the fixed asset accounting decisions as the practical example to demonstrate that accountants use accounting techniques and valuation methods to plan and monitor the performance of an organization's assets. Information on the cost of assets over their life cycle is particularly important if we consider that the life of an asset often exceeds the time period used in the organization's strategic plans.

Decisions concern the monitoring of the lifecycle of assets and result in acquisition for replacement, upgrading, or repairing. The monitoring follows relationships between values related to the capitalized value (market or utility value), the value of the repairs, the cost of maintenance or disposal, and efficiency indicators in the achievement of the planned production. In principle, this process is the pursuit of the relationship between the demand for assets (expressed in units of value of products and services planned to be achieved) and the supply of assets (production capacity). Accounting decisions monitor the true image of the recognized asset value. Table 1 presents the information sources for accounting decisions on fixed assets.

An asset is recognized when it is probable that future benefits (income, expenditure cuts, economic, and social benefits) will flow into the enterprise and the asset has a value or cost that can be reliably measured.

Table 1. Information sources for accounting decisions on fixed assets.

Fixed Assets Specific Issues	Information Sources
Establishment of financing conditions for fixed assets	Organizational plans, strategies and rules Information on active markets
Determining the life duration	Organizational rules relating to production plans Existing regulations and norms
Determining the amortization method	Information on active markets Existing laws and norms
Determination of depreciation	Information on active markets Efficiency indicators of assets
Fair value evaluation	Information on active markets Evaluation models based on stream updates
Providing information for making economic decisions about fixed assets: repair, scrapping, replacement, upgrading	Information on active markets Active efficiency indicators Production plans

Impairment of value refers to the deterioration in the value of an asset, a deterioration that has little chance of disappearing in the future due to moral wear, unfavorable market conditions, etc.

Internal information sources:

The existence of evidence of moral or physical wear.

Information extracted from internal reports that the economic performance of the asset is, or will be, less than estimated, for example, information about the cash flow generated by that asset.

The issues concerning the formalization of accounting decisions about fixed assets are determined by:

- various sources of information
- the need to make estimates of benefits, depreciation
- the difficulty in monitoring the performance of assets

In the following study, we will present solutions for the implementation of the decision models regarding the fixed assets by exemplifying the knowledge necessary for their management activities.

We propose a system IT architecture for substantiating organizational decisions (Figure 1).
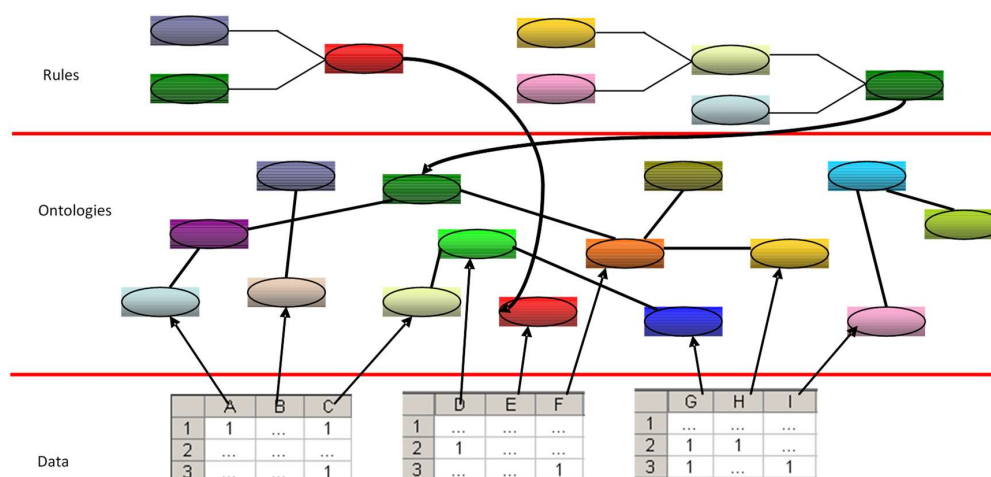

Figure 1. Data, ontologies, and rules.

This kind of architecture takes into account the following premises:

(a) the specific problem-solving knowledge needed in human communication processes is local, context-dependent, and belongs to a human-being. We choose DROOLS, which is a Business Rules Management System (BRMS) solution.

(b) ontologies are necessary for sharing common meaning; statistical and mathematical modeling is required in calculations, forecasts, simulations, and data processing under constant risk and uncertainty conditions; modeling allowed by neural networks is required for simulations. We implement the data model in Java.

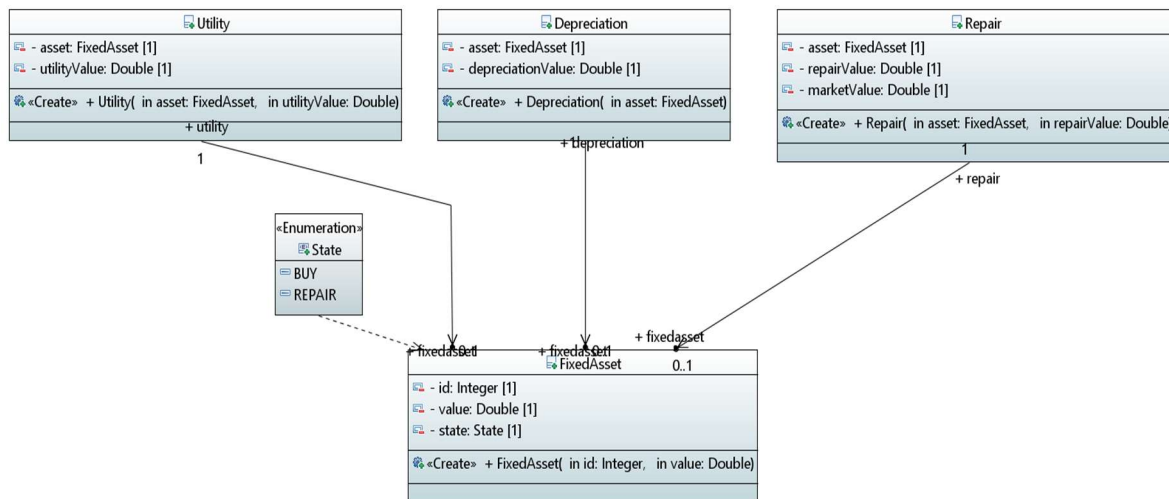The class diagram is the one presented in Figure 2.

Figure 2. Class diagram for the domain model.

The intelligence feature may be provided to a system under the conditions that an internal logic of an application could be used by the internal logic of a second application, and the information manipulated by the system would be organized on the basis of general knowledge in a domain.

We discuss accounting decisions about fixed assets, but for the consistency of the examples presented, we will present several scenarios.

Scenario 1: The accountant must assess fixed asset impairment. He or she knows that in making such assessment, he/she needs to compare the net value with the utility value of the fixed asset but does not know what the utility value of the asset would be. It is highly probable that the net value can be extracted from the company's database, as it is historical information stored in a database. Unlike the net value, the utility value results from estimates of possible updated benefits. Such information is not stored in a database due to its volatility. Thus, we can say that there is a difference between the database schema and the ontology of the application.

Obtaining the net present value of estimated future benefits from using the asset will most likely be achieved with an actuarial function.

At this point, the decision maker could make the assessment of depreciation by specifying and executing a rule. The rule is presented in Figure 3.

```
rule "The value of depreciation is"
   //include attributes such as "salience" here...
     when
         a:FixedAsset()
         u:Utility(asset.equals(a))
         d:Depreciation(asset.equals(a))
     then
         d.setDepreciationvalue(a.getValue()-u.getUtilityValue());
         System.out.println("The value of depreciation is "+d.getDepreciationvalue());
                              End
```

Figure 3. The decision rule concerning assets' impairment.

The rule's execution results in attributing the impairment characteristic to individuals that comply with the rule.

To visualize depreciable assets, the decision maker will be able to ask a question, and in this way, existing individuals in the ontology will be displayed, and the condition will be displayed (see Figure 4).

```
query "Depreciated"
a:FixedAsset()
asset: Depreciation(asset==a)
end
```

Figure 4. Querying the knowledge base.

Scenario 2: At the end of the financial year, it is necessary to reassess the fixed assets in order to determine any impairment and to make provisions. The accountant should estimate whether the recoverable amount may be less than the net book value in the future.

Where there are internal sources of qualitative information on declining performance or moral wear, the estimate of impairment will be based on the professional intuition and judgment of the decision-maker.

In the absence of such sources of information, appreciation can be made from historical data about assets that are part of the same asset class and that have once been depreciated. Because it is not a trend set of values, we cannot talk about an estimation based on statistical methods. The best solution is to use a neural classification network. The training set must consist of the input values for asset returns and the output values (0 for underestimated and 1 for depreciation). In Figure 5 we present the realization of such a neural network.
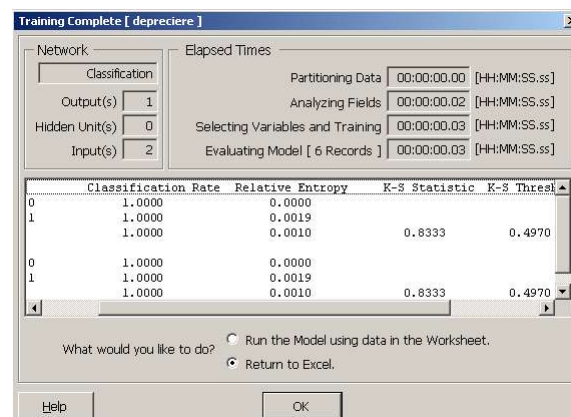

Figure 5. Training the neural network.

Scenario 3: The decision maker knows the value of possible repairs, the purchase price of a similar asset, and he/she wants to know what the decision to take for the impaired assets is.

The accountant states two rules:

(1) if the net book value added to the repair value does not exceed the purchase price, repair is recommended;
(2) if the net book value added to the value of the repair exceeds the purchase price, replacement is recommended.

The rules are presented in Figure 6.

```
rule "Buy or repair"
when
     a: FixedAsset()
     r: Repair(asset==a, ((r.asset.value+r.repairValue)>=r.marketValue))
then
     a.setState(BUY);
     System.out.println("There is an asset with the state "+a.getState());
                                  End
```

Figure 6. The decision rule concerning assets' repair or replacing.

We tested our rules on a publicly available dataset of Bureau of Economic Analysis—Detailed Data for Fixed Assets and Consumer Durable Goods [9]. To test a single instance, the lines of codes are those in Figure 6.

```java
public class Test {

  public static void main(String[] args) {
    // TODO Auto-generated method stub
    KieServices kService=KieServices.Factory.get();
    KieContainer kContainer=kService.getKieClasspathContainer();
    KieSession kSession=kContainer.newKieSession("ksession-rules");
    FixedAsset f1=new FixedAsset(1, 50000.0);
    Depreciation d1=new Depreciation(f1);
    Utility u1=new Utility(f1, 55000.0);
    kSession.insert(f1);
    kSession.insert(d1);
    kSession.insert(u1);
    kSession.fireAllRules();
  }
}
```

The output is "The value of depreciation is −5000.0". We notice that the depreciation value was inferred by the Drools rule "The value of depreciation is". If we decide to define another rule that establishes that the utility value is equal to the market value, we will be able to infer the state of the fixed asset. The main concepts from our domain model are those from the class diagram, namely: FixedAsset, Utility, Repair, and Depreciation. By inferring rules, we were able to set the state of an asset, and after that, we were able to query the knowledge base about the fixed assets list.

3. Results

The conclusions drawn from this experiment are:

- Although specific IT and technology are suitable for any field of activity with real benefits for the society we live in, the limits imposed on integration require common standards recognized for software producers, so we appreciate the fact that the direction of research on the development of ontologies is important in the context of implementing information systems specific to the knowledge society.
- Any IT technology must be tested and accepted by the user. For now, the degree of user acceptance of some technologies is small enough. Unless software developers are more geared to the real needs of the knowledge society, they can one day provide a common work platform that integrates different models from different fields of science.
- Decision models should take into account qualitative factors that can be formalized with variables and organized as concepts in ontologies.

- Neural networks are useful tools in predictions and classifications when several factors are known to determine the value of another indicator and the function is unknown.
- To be interactive, decision-making models must have implementation capabilities for production rules.

The main contribution of this study is the experimentation of the applicability of intelligent technologies in modeling decisions. We analyzed various decision models capable of interacting with the decision-maker and extracting real-time knowledge from it. With respect to other implementations, our approach separates the business rules layer from the data layer. Our approach is better suited to exposing the business rules layer as a service layer and showing that it could belong to the domain service in an application developed by applying domain driven design.

The main conclusion is that rule-based systems are essential in modeling decisions and extracting specialized knowledge from decision-makers. Organizing knowledge in the form of decision rules allows the rules to be executed by an inference engine to answer the questions of decision makers.

We chose DROOLS because, for the moment, there is no other rule engine available as open source. Moreover, it is possible to integrate DROOLS in any Java application, and the rules layer might be added to any applications developed by respecting the Layers pattern.

4. Conclusions

The implementation of a decision model involves knowledge formalization. The knowledge is not generally accepted as true, but relative to the organization and its purposes. Moreover, over time, organization's objectives and how they are measured change due to the organization's adaptation to the economic environment in which it operates. Rules, plans, and business models change and involve changing the constraints on concepts/entities formalized by the information system. The rules, the plans, and the business models must be implemented to allow specification of control over concepts. The editing of metamodels must be directed to the user to ensure adaptation and not be implemented as accessor or mutator methods on data.

In developing the decision support system, we started from the need to extract real-time knowledge from the decision maker to adapt the decision model to the context. Making the model involves:

- the existence of a domain ontology and an ontology for the developed application;
- the possibility of executing specified inferences;
- the possibility of extracting information from a database;
- the possibility of simulations, predictions;
- the possibility of fuzzy questions.

Intelligent systems technology can thus potentially annihilate the negative effects of potentially wrong mathematical optimizations by capturing heuristics using qualitative factors and discovering knowledge that is allowed by neural network technology or genetic algorithms. The use of optimizations and simulations proves to be unreliable in estimation cases regarding information (as a stage of the decision-making process).

Conflicts of Interest: The author declares no conflict of interest.

References

1. Simon, A.H. Organizations and markets. *J. Econom. Persp.* **1991**, *5*, 25–44. [CrossRef]
2. Taylor, F.W. *The Principles of Scientific Management*; Harper & Brothers: New York, NY, USA, 1911.
3. Van Vliet, V. 14 Principles of Management (Fayol), 2009. Available online: https://www.toolshero.com/management/14-principles-of-management/ (accessed on 15 May 2019).
4. Mulder, P. Bureaucratic Theory by Max Weber, 2017. Available online: https://www.toolshero.com/management/bureaucratic-theory-weber/ (accessed on 15 May 2019).

5.   Smirnov, A.; Levashova, T.; Shilov, N. Patterns for context-based knowledge fusion in decision support systems. *Inf. Fusion* **2015**, *21*, 114–129. [CrossRef]

6.   Batoulis, K.; Meyer, A.; Bazhenova, E.; Decker, G.; Weske, M. Extracting Decision Logic from Process Models. In Proceedings of the 27th International Conference on Advanced Information Systems Engineering, Lecture Notes in Computer Science, Stockholm, Sweden, 8–12 June 2015; Zdravkovic, J., Kirikova, M., Johannesson, P., Eds.; Springer: Cham, Switzerland, 2015; Volume 9097, pp. 349–366.

7.   Drools. Available online: https://www.drools.org/ (accessed on 15 May 2019).

8.   Turban, E.; Aronson, J. *Decision Support Systems and Intelligent Systems*; Prentice Hall: Upper Saddle River, NJ, USA, 1988; p. 43.

9.   Bureau of Economic Analysis, Detailed Data for Fixed Assets and Consumer Durable Goods. Available online: https://apps.bea.gov/national/FA2004/Details/Index.htm (accessed on 15 May 2019).