IEMS Vol. 6, No. 1, pp. 72-82, June 2007.

Modeling Dynamic Business Rules using A Dynamic Knowledge Approach

Nasser Karami†
Department of Industrial Engineering and Management
Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550, Japan
Tel: +81-3-5734-2587, Fax: +81-3-5734-2947, karami@ml.me.titech.ac.jp

Junichi Iijima
Department of Industrial Engineering and Management
Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550, Japan
Tel: +81-3-5734-3942, Fax: +81-3-5734-3943, iijima@me.titech.ac.jp

Received Date, May 2006; Accepted Date, February 2007

Abstract. Business Rules are formal statements about the data and processes of an enterprise. They present projections of the organization's constraints and ways of working on their supporting information systems. Therefore, their collection, structuring and organization should be central activities within information systems. In an enterprise, business rules are used to represent certain aspects of a business domain (static rules) or business policy (dynamic rules). Hence, regarding problem domains in the organization, business rules are classified into two groups: static and dynamic business rules. The paper introduces a new concept of business rules, Extended Dynamic Business Rule (EDBR) which contains the results of the occurrence of business rule's action. The focus of this paper is in the organizing, defining and modeling of such business rules using Mineau's approach. Mineau's approach is an extension of Sowa's Conceptual Graph theory.

Keywords: Business Rules Approach, Extended Dynamic Business Rules (EDBR), Conceptual Graphs, Mineau's approach, Locomotive Repairs.

1. INTRODUCTION

Business rules are the basis of any organization. They represent decisions that are made to achieve enterprise goals and reflect the business policies of an enterprise. Business rules are specifications of the business policy, conditions and knowledge and they are set by the organization. Apart from being important as an organization asset, their value has been recognized also in the Information System community. First, they represent significant inputs in the determination of requirements for an information system, and second, they act as a means through which the information system can be aligned with the real business environment. According to Martin and Odell (Martin and Odell, 1998), business rules allow user experts to specify in small, stand-alone units using explicit statements. Business rules are defined as statements on how the business is done, i.e., guidelines and restrictions regarding states and processes in an organization (Bell *et al.,* 1990). The term business rule can be understood both at the level of a business domain and at the operational level of an information system. According to Demmy *et al.* (2002), in an enterprise, business rules are used to represent certain aspects of a business domain or business policy. In the former, they are called static rules whilst in the latter, they are called dynamic rules.

When developing information systems to support organizational change, an important facet of enterprise knowledge is that of business rules. Information systems development methods examine business rules from a number of different perspectives. However, a number of open questions about business rules on the information systems development remain, and we believe they are still unresolved and thus present challenges for future research. For example:

- **Scope of business rules**: What exactly are business rules? How can those be classified with regard to organization and Information System's people?

† : Corresponding Author

- **Modeling of business rules**: How to represent business rules so that they will be understandable to both business and Information System's people? How to model business rules?
- **Acquisition of business rules**: How to acquire business rules from business objectives and business people?
- **Implementation of business rules**: How to implement business rules? What technology to use?
- **Management of business rules**: Where to store business rules? How to manage business rules for an entire organization?

To answer these questions, it has to be clear that the role of business rules should be precisely analyzed at each phase of the information systems development lifecycle. We are going to consider the former two topics in this paper as follow:

1. Definition and classification of business rules concerning a domain business,
2. Business rules modeling using Mineau's Approach.

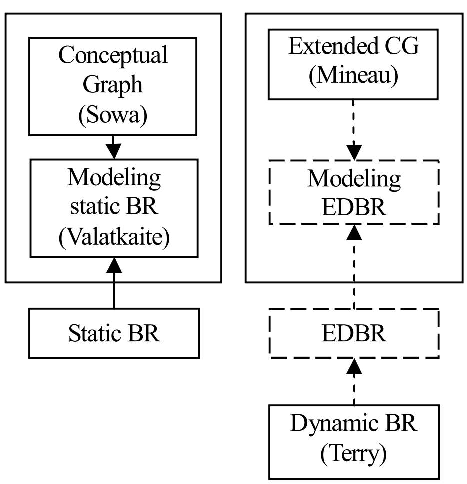

Figure 1. The general view of this research

As shown in Figure. 1, static business rules have been modeled by using conceptual graphs (Sowa, 1984). Conceptual graphs cannot represent dynamic business rules because they can only be used to represent static knowledge. In order to represent dynamic knowledge, Mineau extended Sowa's original conceptual graphs (Mineau, 1998). Hence, one purpose of this paper is to model dynamic business rules using Mineau's approach.

The work presented in this paper concerns the definition of *extended dynamic business rules (EDBR)* dealing with the effects of an action after it happened. According to the prior research and work on the topic, there is a need to specify and define a different concept of dynamic business rules based on the effects of an action. If these effects would not be defined and considered in a business process, we can not precisely model a sequent of business rules in the business process. We can not also solve, for example, defining the exact behavior of business agents in response to events or specifying and expressing complex flows of actions in processes, in traditional approaches. To do so, we propose and define EDBRs that help recover some properties of dynamic business rules that may not be explicitly available from existing business rules. Thus, we first suggest a *When-If-Then-Then Do* pattern for such business rules in comparison with the *When-If-Then* or *If-Then-Else*, and then define them.

Our work makes several contributions related to the definition and modeling of business rules in the design of conceptual databases. The main contribution for proposing EDBRs is the consideration and organization of the relationships among rules in the business process so that people involved in the Information System can easily study their interaction and use this study in decision making as well as in the efficient processing of rules. For example, the effects of an action (*Then Do* part in EDBR's structure) can be considered as a condition triggering the next business rules in a business process (see Figure 2). In this figure, a business rule is seen as a mean to transit from one state to another within a business process step which is described by rules with pre- and post-conditions. Another contribution of our paper is to offer a logical approach which covers the modeling of EDBRs in the business processes area. There is a number of modeling languages and approaches for business rules modeling, and some methods, like Ross's, are quite complicated for inexperienced users, and some other methods, like OCL statements, do not have any graphical notation and thus are not easily understandable by business people. In our research, CGs and Mineau's approach are used as a business rules modeling language because of their simple graphical and linear notations. They are also directly mapped to first order predicate logic. So, in this paper, first we explain and define EDBRs that clarify prior related works to consider the results of an action after it happened, and then we model these business rules using a logical approach and show how dynamic business rules can be modeled using an extension of the Conceptual Graph theory. Such business rules exist in any organization that uses information systems.

To underscore the practical focus of our approach, we have conducted a case study. The case study of the locomotive repairs factory at a large railway company demonstrates how the proposed approach can be applied to model EDBRs within the existing conceptual database. The paper is organized as follows. In Section 2, we review the literature on business rules and we give a brief overview of the business rules system. In this section, we will describe and define the EDBR. A short brief of Mineau's approach and dynamic business rules modeling using Mineau's process are presented in Section 3. The case study of the locomotive repairs' business rules is discussed in Section 4. Finally, the conclusion and future work directions are outlined in Section 5.

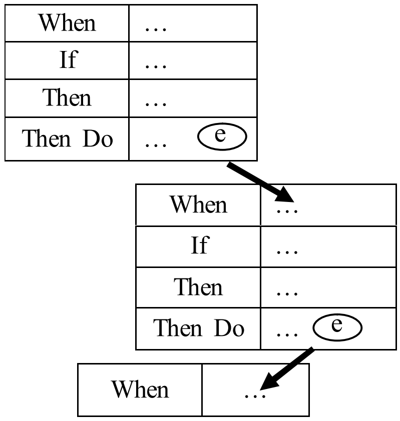

Figure 2. a sequence of EDBRs in a business process

2. BUSINESS RULES

Our approach is related to initiatives in two different areas, namely business rules classification and business rules modeling. The former is mainly concerned with the approaches of business rules classification whilst the latter is concerned with the representation of business rules as structured.

2.1 Business Rules Definition and Classification

In recent years there has been an increasing interest in business rules in the Information System community, which has resulted in dedicated rule-centric modeling frameworks and methodologies (Zaniolo *et al*., 1997). Much knowledge and many rules exist in an organization to prescribe and/or restrict the way in which organizational goals are achieved. Some of these rules exist in a formalized way, e.g., an organizational handbook; others exist only informally. Some rules are precisely defined, others allow for some discretion of human actors. Originally, business rules were defined in connection with integrity constraints.

The term "business rule" has been used by different approaches in different ways. For example, business rules are "statements of goals, policies, or constraints on an enterprise's way of doing business" (Rosca *et al.*, 1997) or they are defined as "statements about how the business is done, i.e. about guidelines and restrictions with respect to states and processes in an organization" (Bell, 1990). Business rules can be stated as "programmatic implementations of the policies and practices of a business organization" (Krammer, 1997). Halle states that "depending on whom you ask, business rules may encompass some or all relationship verbs, mathematical calculations, inference rules, step- by-step instructions, database constraints, business goals and policies, and business definitions" (Halle, 1994).

For the purpose of this work, we consider the definition of business rules related to restrictions and conditions regarding processes in an organization. Thus, the adopted definition in this paper is Bell's definition and we will describe our approach according to this definition.

There are a number of methods and approaches that formalize and classify business rules. One of the most famous approaches is the report offered by the Business Rules Group. The Business Rules Group (formerly the GUIDE Project on Business Rules), a non-commercial peer group of IT professionals, published a report in 1995 with the last edition published in 2000. The goal of this work was to cover all aspects of business rules and set the standards for understanding the term "business rule" in general. Basically, the GUIDE Business Rules Project was organized to formalize an approach for identifying, classifying and articulating the business rules which define the structure and control the operation of an enterprise. This group classified business rules into three main types: structural assertions, action assertions, and derivations. Figure 3 illustrates the classification and categories of business rules in the Business Rules Group's results (Hay and Healy, 2000).

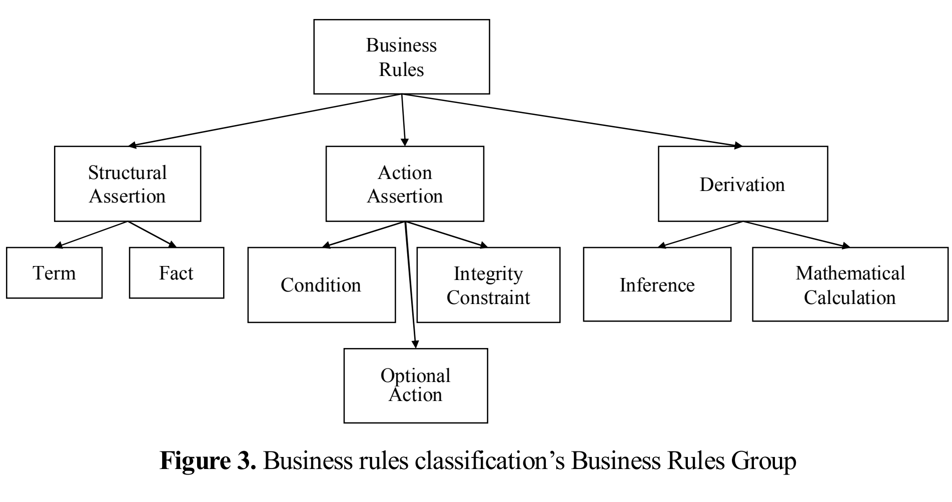

Figure 3. Business rules classification's Business Rules Group

2.2 Extended Dynamic Business Rules (EDBR)

In general and from some business rule's researcher viewpoints, business rules can be *static* or *dynamic*. A *static business rule* is a *constraint (integrity)* or *derivation* rule that applies to each individual state of the business, taken one state at a time. The purpose of a static constraint rule is to restrict the set of valid states of one or more items of data. A derivation rule may define a derived fact type.

A *dynamic business rule* is a transition constraint that restricts how the business may change to new states. A kind of dynamic rules is the dynamic action rule. A dynamic action rule defines the conditions for the invocation of an operation. Dynamic action rules have a three-part structure, consisting of a *trigger*, a *precondition*, and an *action*. The trigger and preconditions describe the conditions under which a rule becomes active, whilst the action part of the rule generates messages to activate operations (Terry, 2005; Oelmann, 1991). A dynamic action rule's form is illustrated in Figure 4. In the form, *WHEN* indicates a Trigger, *IF* a Precondition and *THEN* an Action.

Rule:	
WHEN	**T**rigger
IF	**P**recondition
THEN	**A**ction

Figure 4. The dynamic action rule

There is a lack of business rules classification based on the occurrence of an action, illustrating the need for another concept of dynamic business rules. Let's consider the following example: assume that the minimal amount for a cash withdrawal is $30. If the amount is greater than $30 or equal to it, a customer can withdraw the money. After withdrawing the money (the action), the previous balance should be changed to a new balance. Thus, we propose an extension of the dynamic business rule concept and introduce an *extended dynamic business rule (EDBR)* that considers the results of the occurrence of the action. In order to consider those effects, we add another part to *dynamic action business rules* called a *Postcondition*. Therefore, in contrast with dynamic action rules, EDBRs have four parts that consist of a *trigger*, a *precondition*, an *action*, and a *postcondition*. Postconditions happen after the action has ended. Based on the above discussion, an EDBR will be defined as follows.

Definition: *an Extended Dynamic Business Rule (EDBR) is a constraint or derivation rule that applies to each individual state of the business process with regards to the results of the action after it occurred.*

The general form of EDBRs is illustrated in Figure 5, where the Postcondition is described as *THEN DO*. In this rule, an action is specified with the help of a trigger, precondition and postcondition. It means that if the trigger and precondition hold, the postcondition of the action is satisfied after the occurrence of the action. The postcondition of an operation expresses the effect of the operation under the condition that the precondition holds. EDBRs can also be called *Event-Condition-Action-Effect (ECAE) rules*. Figure 6 shows the previous example by using the EDBR classification. As shown in the figure, the postcondition (*THEN DO* part) has been described by "Balance=pre@balance- amount" that represents the new balance by subtracting the withdrawn amount from the previous balance.

Rule:	
WHEN	**T**rigger (**E**vent)
IF	**P**recondition (**C**ondition)
THEN	**A**ction
THEN DO	**P**ostcondition (**E**ffect)

Figure 5. The extended dynamic business rule

Rule:	Withdrawal_Money
WHEN	Withdrawing
IF	Withdrawable amount >= $30
THEN	Receiving the money
THEN DO	Balance = pre@balance - amount

Figure 6. The Withdraw_Money business rule

2.3 Business Rules Systems

A business rules system is an automated system in which the "rules" are separated (logically, perhaps physically) and shared across data stores, user interfaces and applications. Figure 7 illustrates a simplified representation of this concept.

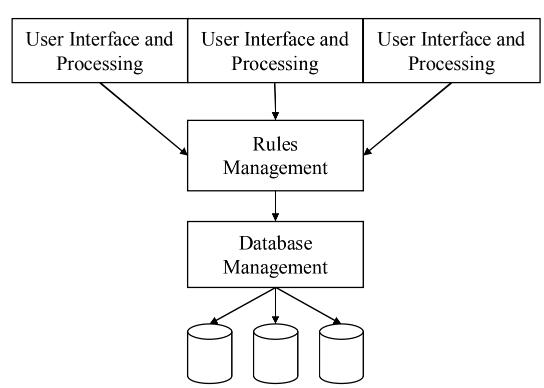

Figure 7. A high-level conceptual architecture for a Business Rules System

To achieve this, the system must be developed according to a business rules approach. A business rules approach is a development methodology where rules are in a form that is used by, but does not have to be embedded in business process management systems (Ross, 2003). So, a business rules approach is a methodology where aging rules are captured, challenged, published and positioned for ongoing change. A business rules approach is a vehicle to drive change across large business scopes. A business rules approach in these cases includes technology for expressing, accessing, publishing and managing rules from a business perspective.

2.4 Business Rules Modeling

A number of languages and approaches were considered for the business rules modeling:

2.4.1 Using UML Associated with OCL Statements

The most popular modeling language is UML which was created through joint the efforts of researchers and commercial organizations. In this approach, business rules modeling are fulfilled by the UML Object Constraint Language (OCL). Business rules are expressed in OCL statements. For example, a simple business rule is that all persons which are married should be at least 18 years of age. The respecttive OCL expression is given below (Booch *et al*., 2000; Demuth *et al*., 2001):

context Person
inv ageOfMarriage: (isMarried = true) implies (age>=18).

Although OCL is an expressive formal language, it does not have any graphical notation and thus is not understandable by business people. Moreover, it does not provide any methodological guidance for the collection of rules.

2.4.2 Ross Method

The Ross Method is one of the most complete methodologies which model business rules. Ross has created the original graphical notation to represent business rules in a data model. It is formal, in accordance with the underlying data models of an organization, offers sufficient methodological guidance and specific constructs for each of the rules families together with a big number of accompanying constructs, such as special symbols, invocation values, special interpreters, and special qualifiers (Ross, 1997). The mentioned simple business rule is presented below by using Ross method. The below rule (Figure 8) is an integrity constraint of Limited type (LIM) which states the instance of a Person is 18 or older as the married age.

However, these properties do not seem to be an advantage, as the complexity of the resulting diagrams and the vast amount of graphical symbols make the language quite complicated, at least for inexperienced users.

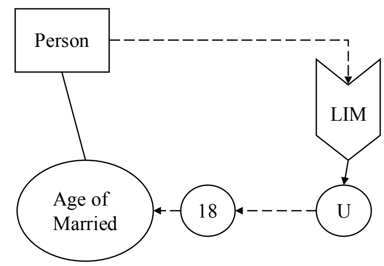

Figure 8. Formal representation of business rule using Ross notation

2.4.3 Conceptual Graphs

Conceptual graphs (CG) may be viewed as one of the suitable modeling languages. CGs created by Sowa (Sowa, 1984) are a knowledge representation language. As defined in the Conceptual Graph Standard, a conceptual graph (CG or graph) is an abstract representation of logic with nodes called concepts and conceptual relations, linked together by arcs. They express meaning in the form that is:

1. logically precise
2. humanly readable
3. computationally tractable.

Hence, a CG is a structure of concepts and conceptual relations where every arc links a concept node and a conceptual relation node (Sowa, 1984; Sowa, 2000). CG is a formal logic-based language which can be used as a business rules modeling language because of its simple graphical and linear notation. It is also directly mapped to a first order predicate logic. In order to model business rules by using the knowledge base framework as defined by Sowa, additional constructs are needed; A Type Rule and a Base Rule. A Type Rule is a CG which has a concept of Event type as initiator and a concept of Conditional Action type in the form of if– then rule as a result. It specifies that each rule must have the initiator of Event type that is specified by INIT and the type result [if: *z[Then: *w]] by RESULT in a linear form (Valatkaite and Vasilecas, 2003). For example, the Married_Person business rule would be represented in the following way:

T*ype Married_Person(*x) is*
Rule(?x)−
*←(INIT)←[Event: [Person: *y]→(Attr) ← [age: *z]]*
→ (RESULT)→ [If : [Age : ?z]→ (>=)→
[Number: 18]
[Then: [Person : ?y]→ (Chrc)→ [Married]]].

Although CGs are expressive formal languages because of their simple graphical and linear notations, they cannot represent dynamic business rules or EDBRs. The conceptual graph formalism provides all necessary representational primitives needed to model static knowledge.

However, the representation of dynamic knowledge falls outside the scope of the actual theory (Mineau, 1998). Sowa's Theory can be used for expressing various kinds of static business rules (integrity and derivation rules).

3. BUSINESS RULES MODELING USING MINEAU'S APPROACH

3.1 Mineau's Representation of Dynamic Processes

Mineau (1998) proposed a representation for dynamic processes. This approach is more oriented toward the automatic translation of algorithms into an executable but declarative format. Lukose and Mineau point out knowledge representation formalisms like Frames, Conceptual Dependency Graphs, and CGs represent declarative information. (Lukose and Mineau, 1998).

In his paper (Mineau, 1998), Mineau uses the idea of processes to represent dynamic knowledge. Basically, Mineau's processes are one kind of executable conceptual graph formalism. Generally, processes can be described using algorithmic languages. In this approach, a process can be described as a sequence of state transitions. A transition transforms a system in such a way that its previous state gives way to a new state. These previous and new states can be described minimally by conditions, called respectively pre and postconditions, which characterize them. Thus, transitions can be represented by pairs of pre and postconditions (Mineau, 1998).

Mineau's approach which can be applied to knowledge modeling is an informal (easily understood by humans) or formal logic-based language.

Therefore, Mineau's approach is suitable for dynamic business rules modeling. The approach allows defining dynamic business rules using processes. Within each process (business processes for business rules), there are a number of steps (business process steps) where each step involves a different set of actors. Business process steps are an important construct in the approach to building business rules engines. Hence, business processes must be decomposed into business process steps, each of which representing a set of business rules (Chisholm, 2004).

3.2 Mineau's Approach

According to Mineau's approach, each of state transitions (here business process steps) has a three-part structure, consisting of a precondition, an actor, and a postcondition. Preconditions describe the conditions under which an actor becomes active, whilst the actor part of the state transition generates messages to activate postconditions. The actor is presented in linear form or graphical form. In linear form, an actor is defined by a list of its parameters (input and output parameters). The input parameters will appear in the preconditions of the actor; while the output parameters will appear in the postconditions of the actor. An actor can be defined as the following statement:

Actor name (in_1 u_1; … in_n u_n; out_1 u_{n+1}; … out_m u_{m+n}) is:
u_1, … u_n, u_{n+1}, u_{m+n}.

The statement specifies the actor that comprises a set of inputs and outputs as pre and post-conditions. Each input or output parameters are represented by using a CG (here u_i). In fact, the statement declares that the input graphs will be the preconditions and the postconditions of the actor.

In graphical form, the actor is represented by a double-lined diamond box labeled with the name of the actor. Each of the parameters appears as a separate statement linked to the actor symbol by an ingoing (for input) or outgoing (for output) arc. Figure 9 shows a simple graph of the actor, where u_1, u_2, u_3, and u_4 are a CG.

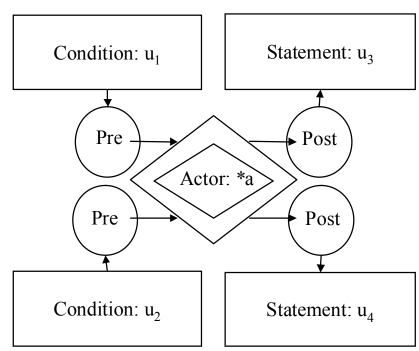

Figure 9. The graphical representation for an actor

The graph of Figure 9 states that actor *a will be triggered (will activate) if, from the current state of the system, graphs u_1 and u_2 are both true (are logical consequences of the actual state of the system). In this case, two assertions will be made: $u_3$will be asserted and u_4 will be negated (Lukose and Mineau, 1998). Let's consider the mathematical operator PLUS for more explanation about the actor. PLUS operator will be implemented on three variables by using basic actors. The PLUS actor would be presented the following way:

Actor PLUS (in_1 a, in_2 b, out c) is:
<PLUS> -
← [Statement: [Integer] ← (Value) ← [Variable: *a]]
← [Statement: [Integer] ← (Value) ← [Variable: *b]]
→ [Statement: [Sum: *c]]

Figures 10. illustrate how a PLUS actor is represented in linear and graphical form.

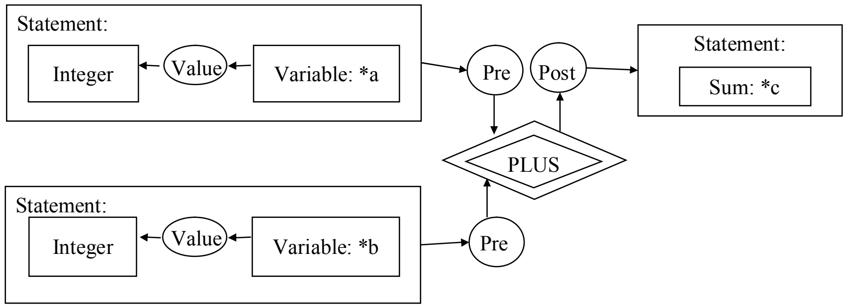

Figure 10. The graphical representation of the PLUS actor

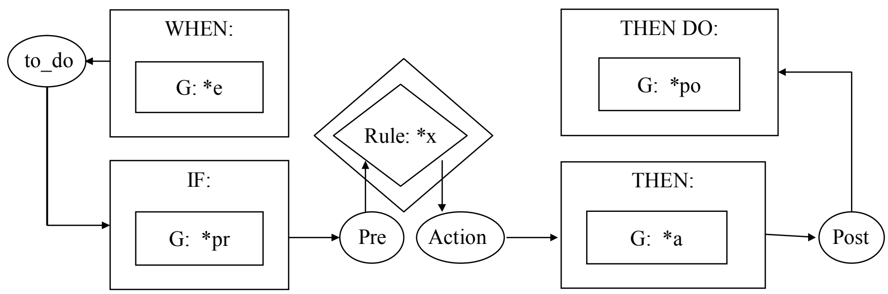

Figure 11. A simple view of EDBR using Mineau's appr1oach

gAs mentioned above, dynamic knowledge could be represented by processes which Mineau defined in terms of transitions between states; previous and new states can be characterized by the pre and post conditions associated with them. The triggering of a transition to a new state depends upon the truth of the preconditions.

3.3 Dynamic Business Rules Modeling Technique

As already stated, a business process will be represented as a set of actors linked by an execution sequence explicitly described by predicates in their pre and postconditions. When the preconditions of an actor become conjunctively true, its postconditions will be asserted, updating the current state of the system. As mentioned in Subsection 2.2, EDBRs consist of four parts, a trigger (event), a precondition, an action, and a postcondition. In our approach, each business rule is considered as an actor and defined as a pair of pre and postcondition. So, in order to model such business rules by using Mineau's approach, two additional elements are needed in order to model dynamic business rules clearly. The action (*THEN* part) and triggering (*WHEN* part) of the business rule should be considered as two important elements when using Mineau's approach. The action is specified with the help of a trigger, precondition and postcondition. The trigger is an event that initiates the precondition. As a result, the business rule will be activated if the precondition is true. The trigger component indicates when a rule has to be executed. Figure 11 shows a simple aspect of EDBR using Mineau's approach. In the figure, **e*, **pr*, **a* and **po* are the referent fields for the trigger, the precondition, the action and the postcondition concepts. Let's consider an example, Withdraw_ Money business rule, assume that the minimum amount for a cash withdrawal is $30. This rule is composed of a single pair of pre/postconditions. The triggering of the business rule will start by the cash withdrawal of a customer. The precondition includes a graph to impose the business rule on the withdrawal event. Therefore, the withdrawal event imposes a precondition on the amount to be withdrawn. After paying the money, in the postcondition of the business rule, the balance of the account is decreased by the withdrawn amount. In the postcondition, the assertion cannot hold if a withdrawal with an amount smaller than $30 occurs. Consequently, the balance of the account is left untouched in case of a withdrawal with an amount smaller than $30. The example would be presented the following way in linear form:

Business rule Withdrawal_Money
(in_1, u_1; out_1, u_2) is:
< Withdrawa_Money >-
← (Pre) ← [[IF:[WithAmt:?z]→ (≥) → [Number:30]]
*← (to_do) ← [WHEN:[Customer:*y]-*
*← (Agnt) ← [withdraw] → (Thme) → [WithAmt:*z]]*
→ (Action) → [THEN: [Customer: ?y] ← (Agnt)
← [Receive] → (Thme) → [WithAmt: ?z]]
→ (Pos) → [THEN DO: [Balance: ?v] ← (MINUS)
← [Balance: ?v] ← [WithAmt:?z]]

4. CASE STUDY: THE LOCOMOTIVE REPAIRS' BUSINESS RULES

To illustrate the discussion above, we introduce a subset of business rules which may be relevant to the Locomotive Maintenance & Repair's Business Rules of the Iran Railway Company. The Iran Railway Company, as the only railway system in Iran, has the responsibility of transporting large number of goods and passengers. The Iran railway network annually transports about 35 millions freight and 20 millions passengers in 2006 (Iran Railways, 2006).

The case study demonstrates how the proposed approach can be applied to model EDBRs within the existing conceptual database.

4.1 About Locomotive and its Repairs

A locomotive is a traction vehicle that pulls a train. Regarding energy consumption, it is classified into electrical and diesel locomotives. A locomotive is repaired after the occurrence of a defect. Locomotive repairs are generally of four kinds in our case: "slight", "minor", "special", or "heavy" repairs. For example, heavy repairs involve all parts of the locomotive being brought up to standards, while special repairs only involve, normally, the repair of one major component on the locomotive or defective part so it can be returned back to service. Such repairs are described according to the basic recommendations of the locomotive manufacturer and compiled in technical manuals.

The repairs of diesel locomotives are carried out in two repair shops called the running shop and the workshop. The running shop is responsible for slight and minor repairs, whereas the workshop is in charge of basic and special repairs. Figure 12 shows how the information is processed between the two repair shops. As shown in the figure, repairs begin after the locomotive arrives at the running shop. If the initial inspection of the locomotive deems it irreparable by the diesel running shop, the locomotive is dispatched to the diesel workshop for basic or special repair ("Cold Locomotive" situation).

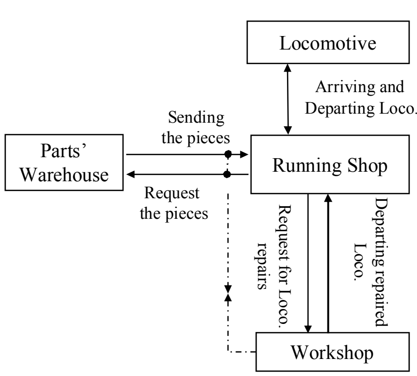

Figure 12. The information processing of locomotives repairs diagram

After all necessary repairs have been completed, the locomotive is ready to work ("Warm Locomotive" situation) for pulling the trains which carry goods and passengers.

4.2 Locomotive repairs' BR System architecture

The architecture of Locomotive repairs' BR System incorporating the proposed approach for modeling dynamic business rules (DBR) is presented in Figure 13. The target business (Domain) contains two repair shops, the locomotive's running shop and the locomotive's workshop. Each repair shop has its own business rules. Using a logic language (in our approach, CG for static business rules and Mineau's approach for dynamic), these business rules are formalized and modeled. The modeled business rules are enforced by a business rules engine, which consists of two major components: Antecedent Evaluation and Consequent Implication. Business rules are enforced (implemented) upon the result of the Antecedent Evaluation part. Users can select some options and submit their data invocation to the system via their interfaces. The system sends output information to the user's interface after analyzing information using business rules in the BR engine.

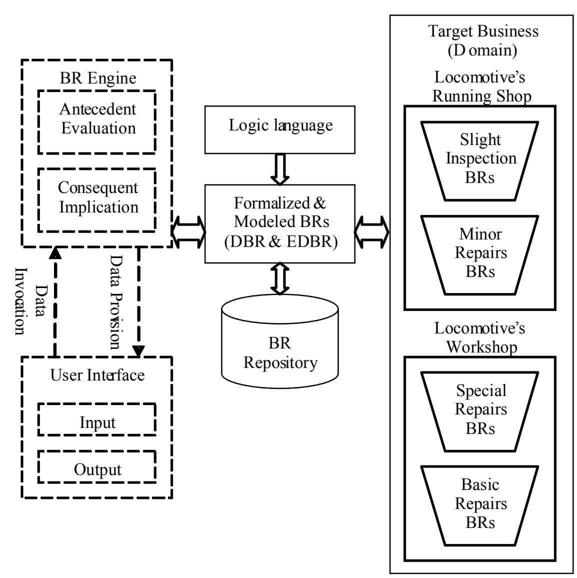

Figure 13. Locomotive repairs' BR system architecture

Figure 14 depicts the semantics of the locomotive repair's business rules that are explained using an event schema.

In our case study, we extract business rules from the existing system. Regarding the event schema, there are six rules that can be transformed into dynamic business rules form. The structure of those rules related to the running shop and workshop is illustrated in Figure 15.

As mentioned, the trigger of the business rules depends on the arrival of the locomotive at the running shop or workshop. Rules 1, 3, 4, 5 and 6 have four parts:

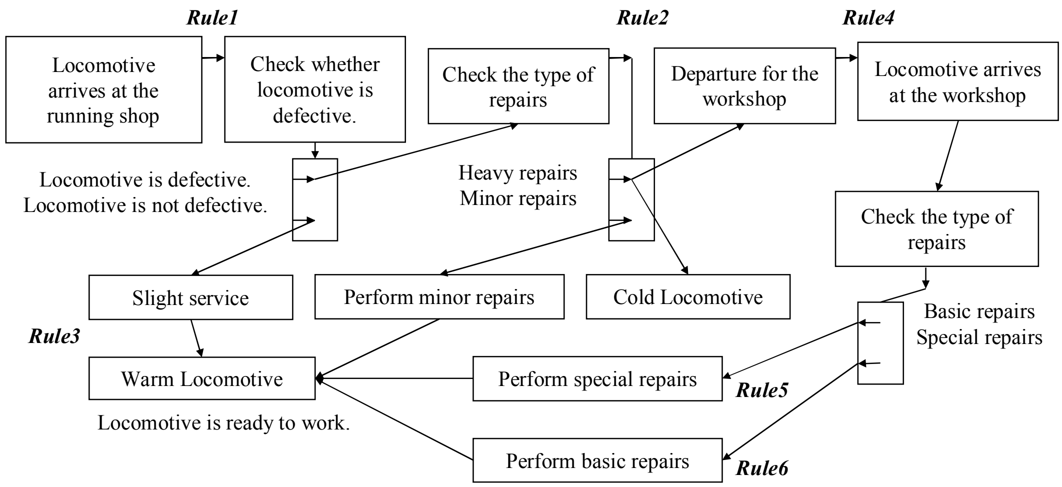

Figure 14. Locomotive Repairs' Business Rules using Event Schema

Rule1: Running Shop_Slight Repairs

WHEN: Locomotive arrives at running shop.
IF: Locomotive is intact.
THEN: Perform slight inspection or service
THEN DO: Locomotive situation is "Warm."

Rule2: Running Shop_ Loco. Inspection

WHEN: Locomotive arrives at running shop.
IF: Locomotive is defective.
THEN: Locomotive is inspected and type of repairs is determined.

Rule3: Running Shop_Minor Repairs

WHEN: Locomotive arrives at running shop.
IF: Locomotive needs minor repairs.
THEN: Perform minor repairs
THEN DO: Locomotive situation is "Warm."

Rule4: Workshop_Heavy Repairs

WHEN: Locomotive arrives at workshop.
IF: Locomotive needs heavy repairs.
THEN: Locomotive is inspected and type of repairs is determined.

Rule5: Workshop_Special Repairs

WHEN: Locomotive arrives at workshop.
IF: Locomotive needs special repairs.
THEN: Perform special repairs
THEN DO: Locomotive situation is "Warm."

Rule6: Workshop_Basic Repairs

WHEN: Locomotive arrives at workshop.
IF: Locomotive needs basic repairs.
THEN: Perform basic repairs
THEN DO: Locomotive situation is "Warm."

Figure 15. The Structure of Locomotive Repairs' Business Rules

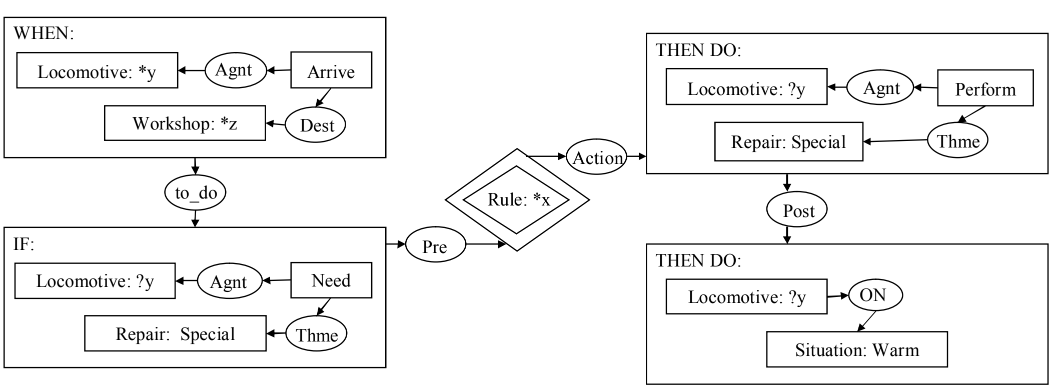

Figure 16. The Workshop_Special Repairs business rule

a trigger, a precondition, an action, and a postcondition (see Figure 15). These business rules follow the EDBR structure and can be represented as EDBR format by using the proposed EDBR form in the Business Rules definition and classification's Subsection. The Postcondition of each business rule is the locomotive's readiness for dispatch. The defined situation for locomotive is warm locomotive. In Figure 16 the rule Workshop_ Special Repairs is modeled in the graphical notationusing Mineau's approach.

5. CONCLUSION AND FUTURE WORK

An information system can be described through two components: static and dynamic. Business rules that support these two components are classified into two main categories: integrity constraints to control the static aspect, and dynamic rules to describe the dynamic aspects. In this paper, we focused on the dynamic business rules and introduced a new concept of dynamic business rules to consider the effects of an action after its occurrence, EDBRs. There are a number of methods and approaches that model business rules based on a *When-If-Then* or *If-Then-Els*e pattern, but our approach follows a *When-If-Then-Then Do pattern*. In this model, the Then Do part indicates a postcondition in an EDBR. Such business rules exist in any organizations that use information systems. In this paper, we modeled these business rules using a logical approach which is quite readable in linear or graphical form. We modeled EDBRs by using Mineau's approach and extended this idea by allowing a business rule to have a CG as input and output parameters. Since EDBRs can directly be mapped to first order predicate logic, they can easily be implemented in business processes area.

However, this paper offers a contribution to modeling business rules at the design level of any part of an information system. In our approach, the EDBR structure allows to specify single business rules (e.g. encompassing dynamic integrity constraints) and define entire processes consisting of business rules.

Future work includes further development of the discussed business rule structuring and implementation mechanism with particular emphasis on the development of Institutionally-Dependent Business Rules. So, our plans for future work are as follow.

1. Evaluate the input and output parameters of a modeled dynamic business rules based on KB and represented as a set of CGs.
2. Implement business rules using Prolog-Java program based on the proposed architecture.

REFERENCES

Bell, J., Brooks, D., Goldbloom, E., Sarro, R., Wood, J. (1990), *Re-Engineering Case Study Analysis of Business Rules and Recommendations for Treatment of Rules in a Relational Database Environment*, US West Information Technologies Group, Bellevue Golden.

Booch, G., Rumbaugh, J., and Jacobson, I. (2000), *The Unified Modeling Language User Guide*, Addison-Wesley Longman Publishing Co., Redwood City, CA.

Chisholm, M. (2004), *How to Build a Business Rules Engine: Extending Application Functionality through Metadata Engineering*, Morgan Kaufmann, San Francisco, CA.

Demey, J., Jarrar, M., and Meersman, R. (2002), A Conceptual Markup Language that supports interoperability between Business Rule modeling systems, *Proceedings of the Tenth International Conference on Cooperative Information Systems (CoopIS 02)*, **2519**, 19-35.

Demuth, B., Hussmann, H., and Loecher, S. (2001), OCL as a Specification Language for Business Rules in Database Applications, *Proceedings of the 4th International Conference on the Unified Modeling Language*, *Modeling Languages*, *Concepts*, *and Tools*, **2185**, Toronto, Canada, 104-117.

Halle, B. V. (1994), *Back to business rule basics*, Database Programming and Design, 15-18.

Hay, D. and Healy, K. A. (2000), *Defining Business Rules What Are They Really?*, Technical Report Rev 1.3, The Business Rules Group.

Herbst, H. (1997), *Business Rule-Oriented Conceptual Modeling*, Physica-Verlag.

Iran Railways (2006), http://www.rai.ir/eng/.

Krammer, M. I. (1997), *Business rules: automating business policies and practices*, Distributed Computing Monitor.

Lukose, D. and Mineau, G. W. (1998), A Comparative Study of Dynamic Conceptual Graphs, *Proceedings of the 11th Knowledge Acquisition for Knowledge Based Systems Workshop (KAW-98),* Banff, Alberta, Canada, Section VKM-7.

Martin, J. and Odell, J. (1998), *Object-Oriented Methods: A Foundation (UML Edition)*, Prentice-Hall.

Mineau, G. (1998), From Actors to Processes: The Representation of Dynamic Knowledge Using Conceptual Graphs, *Proceedings of the 6th International Conference on Conceptual Structures*, Montpellier, France, 65-79.

Oelmann, A. (1991), Representing a system specification with a temporal dimension in an object-oriented language, *Proceedings of the third international conference on advanced information systems engineering*, Trondheim, Norway, 540-560.

Rosca, D., Greenspan, S., Feblowitz, M., and Wild, C. (1997), A decision support methodology in support of the business rules lifecycle, *Proceeding of the International Symposium on Requirements Engineering (ISRE_97),* Annapolis, MD, USA.

Ross, R. G. (1997), *The Business Rule Book, Classifying, Defining and Modeling Rules*, Business Rules Solutions, LLC, Houston, Texas.

Ross, R. G. (2003), *Principles of the Business Rule Approach*, Addison-Wesley Longman Publishing Co., Boston, MA.

Sowa, J. F. (1984), Conceptual Structures: *Information Processing in Mind and Machine*, Addison-Wesley Longman Publishing Co., Boston, MA.

Sowa, J. F. (2000), *Knowledge Representation: Logical, Philosophical, and computational Foundations*, Books Cole Publishing Co., California.

Terry, H. (2005), Fact-orientation meets agent-orientation. *The 6th International Bi-Conference Workshop* (*AOIS 2004)*, New York, USA, 97-109.

Valatkaite, I. and Vasilecas, O. (2003), *A Conceptual Graphs Approach for Business Rules Modeling*, ADBIS, 178-189.

Zaniolo, C. (1997), *Advanced Database Systems*, Morgan Kaufmann.