From Business Oriented Rules to SBVR

Deepak Kumar Sharma  
Ph.D (CSE) Scholar, IKG PTU, Kapurthala, Punjab, India.  

Naveen Prakash  
ICLC, 21/4, S. Bhagat Singh Marg, New Delhi 110001, India.  

Dheerendra Singh  
Department of Computer Science and Engineering, CCET, Chandigarh, India,  

Manish Mahajan  
Dept. of CSE, CGC College of Engineering, Mohali, India

ABSTRACT
It is argued that business rules to be incorporated in a system have their source in rules as defined in the business and are in accordance with the Business Motivation Model. We use a business rules meta model to express BMM business rules and their properties. This expression uses courses of actions and conditions in rules antecedents and courses of actions in rule consequents. It also introduces notions of preference and priority as business rule properties. Business rules are structured into atomic, complex, and abstract rules. We propose to convert these rules into an abstract representation Language, ARL, in which hierarchical rules structures are converted to flat structures and business properties are expressed using AND, OR, NOT logic. In the final step, ARL expressions are converted into SBVR expressions by replacing all antecedents that use courses of actions by antecedents with conditions. The foregoing two step process is illustrated by an example.

Keywords
Business rules, BMM, SBVR, MDA

1. INTRODUCTION
There is a large body of work around business rules now. The Business Rules Group and OMG have paid attention this area in the last decade or so. Business rules have been classified as[1, 6, 7, 8,16]as operational business rules, policy business rules, structural, intentional, architectural, rules for agreements and definitions, and as an expression of functional and non-functional requirements of systems.

The OMG has developed Semantics of Business Vocabulary and Rules, SBVR [10], for structural and operational business rules. It comes with a structured English language for expressing business rules. SBVR lays down the semantics of the business vocabulary as well as of business rules. The vocabulary consists of names, terms, fact types and keywords. Terms are used to designate noun concepts, for example, insurance company and policy holder whereas names refer to nouns that are instances of these concepts, for example, Life Insurance Corporation of India. Fact types capture n-ary relationships and use verbs to relate terms together. Keywords can be quantifiers, logical and modal operators etc. The semantics of a business rule are captured in SBVR as a logical formulation. A logical formulation is constructed over terms, names, and fact types. A given logical formulation can be extended by introducing quantifications, modal formulations and logical operators. These extensions elaborate, in a step by step manner, the structure of a business rule. Further, SBVR associates three properties with business rules, Practicability, Automate-ability, and direct enforceability.

SBVR is located at the CIM level of MDA. There are other proposals for populating the CIM level and another such is based on ACE [9].

The Business Rules Group, BRG, has proposed a Business Rule Manifesto [3] that promotes the notion of a business rule as a first class concept. In developing this notion Manifesto for business rules independence organized in ten articles. The basic idea is that business rules are primary requirements, independent of processes, represent deliberate knowledge, and are declarative in nature. Business rules should arise from knowledgeable business people and there is need for tools to help in formulation, validation, and management of business rules.

From the governance point of view, BRG has developed the Business Motivation Model, BMM [11]. The idea is to propose elements that go into business governance, their inter relationships, and the purpose/role of these in business governance. Although BMM recognizes the business motivation for business rules, it does not attempt to define business rules but relies on SBVR to provide this definition. Since, according to BMM, business rules govern courses of actions, we must determine the nature of this ‘governance’.

Goverance is interesting in the following cases

1. A course of action standing alone needs to be governed: an individual course of action is governed by its own business rule

2. For inclusion [11] of courses of action in courses of action, governance for both the included as well as the including course of action is needed: if a course of action A that includes courses of action B and C then B and C have their own business rules whereas A has its own business rule.

3. Governance of enablement [11] of a course of action from another is needed: let a course of action A enable course of action B. In addition to the individual business rule of A and B respectively, there is a business rule that governs this enablement.

Evidently SBVR is capable of dealing with (1) above. Regarding (2), SBVR rules are flat structures and it is not possible for a business rule to be contained in another
business rule. Regarding (3) it is to be noted that the SBVR specification has no notion of activity fact type [10][2]. Therefore, “a direct representation of course of action enabling course of action” is not possible.

The first rule says that the COA, pay processing fees, is performed after the occurrence of COA, submit request. The second rule says that only a valid request triggers pay processing fees. This leads us to classify business rules as

- COA - COA business rules having antecedents and consequents both of which are COA
- C - COA rules having conditions as antecedents and COA as consequents.

The ‘affects’ relationship of Fig. 2 says that a COA affects a condition if it either directly changes a variable that participates in the condition or if it forces an indirect change. In other words, if a consequent changes a variable of an antecedent then there is an affects relationship between the two. We express this as Affects (consequent1, variable1). Any antecedent that uses variable1 gets affected by consequent1.

Deadlines in Fig. 2 specify a time limit within which the application of the business rule must be completed. Since antecedent satisfaction has already occurred, this means that, deadlines and specify consequent completion properties.

There are two kinds of deadlines, hard and soft. In the former case, the expiry of the deadline stops the application of the rule upon deadline expiry whereas in the latter, deadlines may not stop the application of the rule but may exact a price or penalty for going beyond the deadline. The figure shows three main kinds of business rules, atomic, abstract, and complex. We consider each of these in turn.
1. **Bunch**: A bunch is a named collection of business rules having a common antecedent. For example, consider the collection of the three C-COA type business rules as follows:

   IF CONDITION borrower type = 'student' THEN register student-borrower
   IF CONDITION borrower type = 'teacher' THEN register teacher-borrower
   IF CONDITION borrower type = 'administrative' THEN register admin-borrower

   These rules form a bunch of business rules named Register Borrower. There are two properties associated with a bunch:

   - **Preference**: There may be a preferential order between the rules of the bunch. Preference provides guidance on what should preferably be done. Consider the bunch, Make Payment, as follows:
     
     **PREFERENCE 1**: IF CONDITION approved payment THEN pay bank transfer
     **PREFERENCE 2**: IF CONDITION approved payment THEN pay draft
     **PREFERENCE 3**: IF CONDITION approved payment THEN pay cheque

     It is preferred to pay by bank transfer and the next preferred by draft and lastly by cheque.

   - **Priority**: Priority is yet another property of a bunch that provides guidance. Priority takes on values from the set of natural numbers {1, 2, ----}. Unlike preference, it implies an order of invocation between the constituents of a bunch. This order is inviolate. For example, consider the bunch as follows

     **PRIORITY 1**: IF COA seat allotted THEN allot hostel room
     **PRIORITY 2**: IF COA seat allotted THEN process educational loan

     Within the bunch, the first rule above is given priority 1 and the second is given priority 2.

2. **Nested**: A nested business rule is one whose consequent has its own business rules. Thus, there is a nested business rule for a business rule. An example of a complex business rule is as follows:

   **BUSINESS RULE NAME**: manage borrowers
   **RULE TYPE**: NESTED
   **BODY**:
   IF registration request THEN register borrower
   **Composed of**
   **BUSINESS RULE NAME**: registers borrower
   **RULE TYPE**: BUNCH
   **BODY**:
   IF CONDITION borrower type = 'student' THEN register student-borrower
   IF CONDITION borrower type = 'teacher' THEN register teacher-borrower
   IF CONDITION borrower type = 'administrative' THEN register admin-borrower

3. **Transitivity**: There are two ways in which transitivity arises, through COA - COA transitivity and through the affects relationship. An example of the former is as follows:

   **BUSINESS RULE NAME**: service registers Borrower
   **BODY**:
   IF COA registration request THEN provide services
   **COMPOSED OF**
   IF COA registration request THEN register borrower
   IF COA register borrower THEN provide services

   Second, the notion of transitivity can be extended to include the ‘affects’ relationship as follows:

   **BUSINESS RULE NAME**: service registers borrower
   **BODY**:
   IF COA registration request THEN provide services
   **COMPOSED OF**
   IF COA registration request THEN register borrower
   Affects (register borrower, registered borrower)
   IF CONDITION registered borrower THEN provide services

4. **Aggregate**: An aggregate is a named collection of business rules meeting a business governance objective. An example is as follows:

   **BUSINESS RULE NAME**: manage user
   **BODY**:
   IF COA registration request THEN register user
   IF COA deregistration request THEN deregister user

   Manage user is a governance objective and the two rules comprise it.

2.3 Abstract

An **abstract** business rule is a generalization of other business rules. This generalization can occur when the business object of the antecedent and/or consequent enters into generalization/specialization relationship with other business objects. An example of an abstract business rule is as follows:

   **BUSINESS RULE NAME**: issue book
   **BODY**:
   IF CONDITION (valid borrower AND number issued less than maximum) THEN give book

   **Generalization of**
   **BUSINESS RULE NAME**: issue book student
   **BODY**:
   IF CONDITION (valid student borrower AND number issued less than student maximum) THEN give book

   **BUSINESS RULE NAME**: issue book teacher
   **BODY**:
   IF CONDITION (valid teacher borrower AND number issued less than teacher maximum) THEN give book

   Here, the business object, borrower, of the antecedent can be specialized into student borrower and teacher borrower respectively. This gives rise to the two specialized rules.
3. CONVERTING TO ABSTRACT REPRESENTATION LAYER

It can be seen that the BMBRM level considers business rules from the purely business perspective. On the other hand, SBVR looks upon business rules for supporting Information Systems. The abstract representation layer, ARL, acts as a bridge between the two. This bridge must do three things:

- Since SBVR deals with atomic rules only, it is necessary to convert BMBRM rules into flat structures and remove all hierarchical structures.
- Express logical connections by introducing conjunctions, disjunctions and negations.
- Express preferences, priorities and deadlines.

Thus, in the Abstract Representation Layer, ARL, a business rule is a conversion of a BMBRM rule into a flat structure that includes AND, OR, and NOT as well as preference, priority and deadline will remain unchanged at this level and it will be handled at the time of SBVR conversion.

A. Obtaining Flat Structures

It can be seen from the previous section that hierarchical business rules arise in for complex and abstract business rules respectively. We consider these in turn.

**COMPLEX**

We need to flatten the following kinds of BMBRM rules:

- Aggregate
- Transitive
- Nested
- Bunch

**Aggregate**: An aggregate business rule is a loose collection of individual rules meeting a business governance objective. Thus, they can be separated and treated as individual business rules in the ARL.

As an example consider “manage user” above. This rule has two atomic rules “user registration” and “user deregistration” as follows:

IF COA registration request THEN register user

IF COA deregistration request THEN deregister user

In ARL, both these rules will be represented individually. **Transitive**: Transitivity can occur in two ways:

- A COA consequent of one rule is an antecedent of another rule. In this case the complex transitive rule can be replaced by the two individual rules displaying transitivity. That is, in the case

  BR1: IF COA1 THEN COA3
  COMPOSED OF
  BR2: IF COA1 Then COA2
  BR3: IF COA2 THEN COA3

  Then BR1 can be replaced by BR2 and BR3.

- A COA consequent change a variable defined in the AFFECTS relationship and such a variable participates in the condition antecedent of another rule. In this case the complex transitive rule is modified such that its antecedent is ANDed with the condition of the second rule. Its consequent remains unchanged. The two constituent rules are removed as is the AFFECTS relationship.

For example, consider “service registered borrowers” business rule above defined as:

IF COA registration request THEN provide services

It is composed of the two rules and the AFFECTS relationship as follows:

IF COA registration request THEN register borrower Affects (register borrower, registered borrower)

IF CONDITION registered borrower THEN provide services

In ARL, The complex rule now gets replaced by:

IF COA registration request AND CONDITION registered borrower THEN provide services

**Nested**: In the case of a complex business rule that contains nested business rules it is required to AND the antecedent of the complex rule with the antecedent of each individual component rule to form the antecedent of the new rules. Further, the main rule is dropped.

For example, consider “manage borrowers” as follows:

IF CAO registration request THEN register borrower Composed of

IF CONDITION borrower type = ‘student’ THEN register student-borrower

IF CONDITION borrower type = ‘teacher’ THEN register teacher-borrower

IF CONDITION borrower type = ‘administrative’ THEN register admin-borrower

The ARL form will be:

IF COA registration request AND CONDITION borrower type = ‘student’ THEN register student-borrower

IF COA registration request AND CONDITION borrower type = ‘teacher’ THEN register teacher-borrower

IF COA registration request AND CONDITION borrower type = ‘administrative’ THEN register admin-borrower

**Bunch**: The interesting question here is whether exactly one or several consequents of the bunch are to be enacted. This leads to XOR and OR specification. This additional information is asked for in the ARL.

For example, consider the “register borrower” bunch

IF CONDITION borrower type = ‘student’ THEN register student-borrower

IF CONDITION borrower type = ‘teacher’ THEN register teacher-borrower

IF CONDITION borrower type = ‘administrative’ THEN register admin-borrower

If the rules designer specifies XOR then the ARL form will be:

Switch borrower type THEN
student: register student-borrower \textbf{XOR}

Teacher: register teacher-borrower \textbf{XOR}

Now consider the situation where OR is specified as in the bunch, ‘make payment’ below

\begin{align*}
\text{IF CONDITION approved payment THEN pay bank transfer} \\
\text{IF CONDITION approved payment THEN pay draft} \\
\text{IF CONDITION approved payment THEN pay cheque}
\end{align*}

This is represented in ARL as

\begin{align*}
\text{IF approved payment THEN pay bank transfer OR pay draft OR pay cheque}
\end{align*}

\textbf{Preference:}

For expressing the preference property of a bunch, we introduce the PRF\_ORDER construct in ARL. This construct specifies the preferential order of the consequents of rules comprising the bunch. For example, consider “make payment” with preferences as follows:

\begin{align*}
\text{PREFERENCE 1: IF CONDITION approved payment THEN pay bank transfer} \\
\text{PREFERENCE 2: IF CONDITION approved payment THEN pay draft} \\
\text{PREFERENCE 3: IF CONDITION approved payment THEN pay cheque}
\end{align*}

The ARL expression of this is as follows:

\begin{align*}
\text{IF CONDITION approved payment THEN} \\
\text{PRF\_ORDER (pay bank transfer OR pay draft OR pay cheque)}
\end{align*}

\textbf{Priority:}

To handle priorities, we introduce the PRI\_SEQUENCE construct to specify order of invocation of consequents. All the members in the sequence are connected by AND connection since this is a case of parallelism among consequents.

Let us consider “seat allotment” bunch.

\begin{align*}
\text{PRIORITY 1: IF COA seat allotted THEN allot hostel room} \\
\text{PRIORITY 2: IF COA seat allotted THEN process educational loan}
\end{align*}

The ARL form is as follow:

\begin{align*}
\text{IF CONDITION approved payment THEN} \\
\text{PRI\_SEQUENCE (allot hostel room AND process educational loan)}
\end{align*}

\textbf{ABSTRACT}

Having finished with complex business rules let us now consider Abstract BMBRM structures. Due to the generalized/ specialized form the abstract rules can be separated and specified as individuals. For example, “issue book” of section II will be replaced by its specialized rules “issue book student” and “issue book teachers”. So, the ARL form is

\begin{align*}
\text{Administrative: register admin-borrower} \\
\text{IF valid student borrower AND number issued less than student maximum THEN give book}
\end{align*}

\textbf{IF valid teacher borrower AND number issued less than teacher maximum THEN give book}

They are then transmitted to the SBVR layer as individuals.

\textbf{8. Handling Deadlines}

All rules of BMBRM that have a deadline associated with them carry the deadline information into the ARL as such. For example,

\begin{align*}
\text{BUSINESS RULE NAME \textbf{register borrowers}} \\
\text{DEADLINE HARD} \\
\text{RULE TYPE TRANSITIVE} \\
\text{BODY} \\
\text{IF COA registration request THEN verify document IF COA verify document THEN register borrower}
\end{align*}

The ARL rule is as follows:

\begin{align*}
\text{DEADLINE HARD} \\
\text{IF COA registration request THEN verify document IF COA verify document THEN register borrower}
\end{align*}

\textbf{4. CONVERSION TO SBVR}

We now consider the last step of our conversion process. Here, the representation of rules in ARL is converted to SBVR. Notice that ARL has already expressed business rules as flat structures.

SBVR does not allow courses of actions in its antecedent part. Therefore, ARL rules that have such antecedents have to be modified to contain conditions only. In other words, all rules that are of the COA-COA form must be recast in C-COA form. This can be done by using the AFFECTS relationship of section II. The procedure for conversion of COA-COA structures is as follows.

Consider a business rule, \textit{BR}_1, which has COA as antecedent and COA as consequent. Further let there be a business rule, \textit{BR}_2 with COA as antecedent and COA then

1. If AFFECTS relationship is defined for COA then in \textit{BR}_2 substitute COA with these variables with an AND connection between them. The AFFECTS relationship is not carried forward to SBVR.

2. If AFFECTS is not defined, then develop this relationship for COA, and then carry out step one above.

We illustrate this with the ARL expression of a business rule as follows:

\begin{align*}
\text{IF COA registration request AND CONDITION registered borrower THEN provide services}
\end{align*}

Let the course of action, registration request change the value of a Boolean valued variable, Request received, as expressed by

\text{AFFECTS (registration request, request received)}

Then, the SBVR expression is

\begin{align*}
\text{IF CONDITION request received AND} \\
\text{CONDITION registered borrower THEN provide services}
\end{align*}

This can be expressed in the Structured English form of SBVR as follows:

\begin{align*}
\text{It is obligatory to provide services to all requests received from registered borrowers}
\end{align*}

Now let us look at deadlines. These can be mapped to the enforcement levels of SBVR. SBVR has 6 levels of...
enforcement. A HARD deadline is converted to ‘Strict’ enforcement level and the SOFT deadline is converted to ‘Deferred’ enforcement level.

Finally, it is to be noted that preferences and priorities cannot be expressed in SBVR. Yet, this information is useful when developing information systems. We propose that SBVR expressions should be annotated with this information so as not to lose it and transmit it to lower layers of MDA.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>ARL</th>
<th>SBVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>IF Antecedent THEN Consequent</td>
<td>It is obligatory that if antecedent then consequent</td>
</tr>
<tr>
<td>2.</td>
<td>IF Antecedent1 AND Antecedent2 THEN Consequent</td>
<td>It is obligatory that if antecedent1 and antecedent2 then consequent</td>
</tr>
<tr>
<td></td>
<td>Switch antecedent THEN value1: consequent1 XOR value2 : consequent2</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>IF antecedent THEN consequent1 OR consequent2 OR consequent n</td>
<td>It is obligatory that if antecedent then consequent1 or consequent2 or consequent n</td>
</tr>
<tr>
<td>4.</td>
<td>IF antecedent THEN PRF_ORDER (consequent1 OR consequent2)</td>
<td>*It is obligatory that if antecedent then consequent 1 or consequent 2 or consequent n</td>
</tr>
<tr>
<td>5.</td>
<td>IF antecedent THEN PRI_SEQUENCE (consequent1 AND consequent2 AND)</td>
<td>*It is obligatory if antecedent then consequent1 and consequent2 and consequent3</td>
</tr>
</tbody>
</table>

*These have to be annotated to indicate that these are cases of preference and priority orders respectively.

5. **EXAMPLE**

The Library Management System of our institution was taken as an example. Since no formal governance model for our library exists, it was our aim to develop one such and put it up for the Library Committee to formally adopt. During development of the set of business rules, we spoke to the senior most authorities, namely, Chairman of the Library Committee, its members and the Librarian. This comprised 9 people in all. A number of business rules expressed in our BMBRM were formulated.

Before we consider the conversion of the BMBRM rules of this system we give a brief summary of the types of business rules and the number of these encountered in this example. This is shown in the table below.

<table>
<thead>
<tr>
<th>Type of Rule</th>
<th>Number of Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic</td>
<td>10</td>
</tr>
<tr>
<td>Aggregate</td>
<td>6</td>
</tr>
<tr>
<td>Transitive</td>
<td>6</td>
</tr>
<tr>
<td>Nested</td>
<td>1</td>
</tr>
<tr>
<td>Bunch with no preference and priority</td>
<td>1</td>
</tr>
<tr>
<td>Bunch with preference</td>
<td>4</td>
</tr>
<tr>
<td>Bunch with priority</td>
<td>0</td>
</tr>
<tr>
<td>Abstract</td>
<td>2</td>
</tr>
<tr>
<td>Rule with Dead line</td>
<td>18</td>
</tr>
</tbody>
</table>

The SBVR conversion of some them is as follows:

The library deals with three broad Courses of Action namely, Manage borrower, Provide Services, and Stock Library material. Manage borrower involves the registration and deregistration of Library uses. Thus, its business rules are a) when request for registration as a borrower is received then the borrower is registered, b) when request for deregistration as a borrower is received then deregistration is done. These business rules are presented below.

**BUSINESS RULE NAME** manage borrower  
**RULE TYPE** AGGREGATE  
**BODY** IF COA registration request THEN register borrower IF COA deregistration request THEN deregister borrower Affects (register borrower, registered borrower) Affects (deregister borrower, registered borrower)

By applying the aggregate conversion procedure we obtain ARL rules as follows:

IF COA registration request THEN register borrower IF COA deregistration request THEN deregister borrower

By applying the SBVR conversion procedure given in Table I, we obtain SBVR Structured English form as follows:

It is obligatory that either student or faculty or staff must request to register as a borrower.

It is obligatory that either student or faculty or staff must request to deregister as a borrower.

**Composed of**

**BUSINESS RULE NAME** register borrower  
**DEADLINE** HARD  
**RULE TYPE** TRANSITIVE  
**BODY** IF COA registration request THEN verify document IF COA verify document THEN register borrower
By applying the transitive conversion procedure we obtain

**ARL rules as follows:**

**DEADLINE**

<table>
<thead>
<tr>
<th>RULE</th>
<th>TYPE</th>
<th>BODY</th>
</tr>
</thead>
<tbody>
<tr>
<td>HARD</td>
<td>IF COA registration request THEN verify document IF COA verify document THEN register borrower</td>
<td></td>
</tr>
</tbody>
</table>

**SBVR Structured English form:**

It is obligatory that document is verified for a requester.

It is obligatory that if document gets verified by library staff then borrower is registered.

ENFORCEMENT LEVEL: strict

The hard deadline specifies that the registration task should be carried out within a specified time otherwise it is aborted and a fresh request must be made.

**BUSINESS RULE NAME** deregister

**DEADLINE**

<table>
<thead>
<tr>
<th>RULE TYPE</th>
<th>BODY</th>
</tr>
</thead>
<tbody>
<tr>
<td>HARD</td>
<td>IF COA deregistration request AND CONDITION registered borrower THEN remove borrower</td>
</tr>
</tbody>
</table>

**SBVR Structured English form:**

It is obligatory that if deregistration request is given by any registered borrower then borrower is removed.

ENFORCEMENT LEVEL: strict

It is to be noted that the aggregate Manage Borrower uses Affects relationship. Registration/Deregistration causes a change in the variable registered borrower.

Registered borrowers produced by ‘manage borrowers’ are provided services for issuing, returning, and reserving library material. Material issued is to be returned within a specified period but returns are accepted even after this period with a monetary fine. Therefore, this constitutes a soft deadline. When returning material it is possible to reissue it by making an explicit renewal request. In the absence of this, the material is returned back. Reservation of material can be done at any time.

**BUSINESS RULE NAME** provide services

**DEADLINE**

<table>
<thead>
<tr>
<th>RULE TYPE</th>
<th>BODY</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOFT</td>
<td>IF CONDITION registered borrower AND COA issue request THEN issue service</td>
</tr>
</tbody>
</table>

**SBVR Structured English form:**

It is obligatory that Book must be issued if it is not reserved and it is within limit.

**Generalization of**

<table>
<thead>
<tr>
<th>RULE</th>
<th>TYPE</th>
<th>BODY</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGRGATE</td>
<td>IF CONDITION ((within material limit) AND (NOT reserved)) THEN issue Book</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IF CONDITION ((within material limit) AND (NOT reserved)) THEN issue Journal</td>
</tr>
</tbody>
</table>

Affects (issue material, within material limit) **By applying the abstract conversion procedure we obtain ARL rules as follows:**

<table>
<thead>
<tr>
<th>RULE</th>
<th>TYPE</th>
<th>BODY</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOFT</td>
<td>IF CONDITION registered borrower AND COA issue request THEN issue service</td>
<td></td>
</tr>
</tbody>
</table>

**SBVR Structured English form:**

It is obligatory that Book must be issued if it is not reserved and it is within limit.
It is obligatory that Journal must be issued if it is not reserved and is within limit.

When reserved material is issued then the material limit should be satisfied and issuing must be done within the alert deadline. Additionally it affects ‘within material limit’ and ‘reserved’.

**BUSINESS RULE NAME**: issue reserved  
**RULE TYPE**: ABSTRACT  
**BODY**:  
IF CONDITION (within material limit AND within alert deadline) THEN issue material

**Generalization of**  
IF CONDITION (within material limit AND within alert deadline) THEN issue Book  
IF CONDITION (within material limit) AND within alert deadline) THEN issue Journal  
Affects (issue material, within material limit)  
Affects (issue material, reserved)

**ARL Rule:**  
IF CONDITION (within material limit AND within alert deadline) THEN issue Book  
IF CONDITION (within material limit) AND within alert deadline) THEN issue Journal

**SBVR Structured English form:**  
It is obligatory that book must be issued if it is within material limit and within alert deadline.  
It is obligatory that journal must be issued if it is within material limit and within alert deadline.

Return Service consists of normal return and renewal of material.

**BUSINESS RULE NAME**: return service  
**DEADLINE**: HARD  
**RULE TYPE**: AGGREGATE  
**BODY**:  
IF COA return request THEN return  
IF COA return AND COA renew request THEN renew

**ARL Rule:**  
DEADLINE HARD IF COA return request THEN return  
IF COA return AND COA renew request THEN renew

**SBVR Structured English form:**  
It is obligatory that if due date is over then material should be returned with fine.  
ENFORCEMENT LEVEL deferred

Renewal is possible if the material is not reserved.

**BUSINESS RULE NAME**: renewal  
**DEADLINE**: HARD  
**RULE TYPE**: ATOMIC  
**BODY**:  
IF CONDITION NOT reserved THEN issue normal

**ARL Rule:**  
DEADLINE HARD IF CONDITION NOT reserved THEN issue normal

**SBVR Structured English form:**  
It is obligatory that material must be issued in a normal manner if it is not reserved.

ENFORCEMENT LEVEL strict

Return alert issues an alert (a) if due date for return is exceeded (b) if returned material is reserved then it gives a hard deadline within which reserver must issue the material.

**BUSINESS RULE NAME**: return alert  
**DEADLINE**: HARD  
**RULE TYPE**: ATOMIC  
**BODY**:  
IF CONDITION due date exceeded THEN alert return

**ARL Rule:**  
DEADLINE HARD IF CONDITION due date exceeded THEN alert return

**SBVR Structured English form:**  
It is obligatory that if due date for return has exceed then raise alert for return.

ENFORCEMENT LEVEL strict

If return is done beyond the due date then a fine is imposed.

**BUSINESS RULE NAME**: return  
**DEADLINE**: SOFT  
**RULE TYPE**: BUNCH  
**BODY**:  
IF CONDITION due date = ‘over’ THEN return with fine  
IF CONDITION due date = ‘NOT over’ THEN normal return

**ARL Rule:**  
DEADLINE SOFT  
**Switch** due date THEN ‘over’: return with fine XOR ‘not over’: normal return

**SBVR Structured English form:**  
It is obligatory that if reserved material has been returned then reservation alert is raised.

ENFORCEMENT LEVEL strict

If CONDITION (reserved AND within alert deadline) THEN reservation alert

**ARL Rule:**  
DEADLINE HARD IF CONDITION (reserved AND within alert deadline) THEN reservation alert

**SBVR Structured English form:**  
It is obligatory that if reserved material has been returned then reservation alert is raised.

ENFORCEMENT LEVEL strict

If CONDITION (reserved AND within alert deadline) THEN issue material

**ARL Rule:**  
DEADLINE HARD IF CONDITION (reserved AND within alert deadline) THEN issue material

52
SBVR Structured English form:

It is obligatory that reserved material must be issued if it was reserved and requested with in alert deadline.

**ENFORCEMENT LEVEL** strict

The third Course of action, Stock material includes all courses of actions from handling a new indent requesting for library material to be procured right up to the final availability of the procured material to registered borrowers. It also includes any maintenance work like binding that may be necessary.

The first four business rules below are straightforward. The fifth requires some explanation. When the library issues a purchase order for periodicals then payment must accompany the order. The material is supplied at its fixed periodicity. Therefore, for periodicals no further payment is to be made at delivery time. On the other hand, for non-periodical material, payment is to be made upon material receipt.

Stock material consists of four major rules, purchase material, receive delivery, process material and provide material.

**BUSINESS RULE NAME** stock material

**RULE TYPE** TRANSITIVE

**BODY**

IF COA purchase material THEN receive delivery

IF CONDITION material received THEN process material IF COA process material THEN provide material

**ARL Rule:**

IF COA purchase material THEN receive delivery

**SBVR Structured English form:**

It is obligatory that receive delivery of material must be received if it is purchased.

It is obligatory that if received material is processed.

**BUSINESS RULE NAME** purchase material

**RULE TYPE** TRANSITIVE

**BODY**

IF COA new indent THEN indent verify

IF CONDITION valid indent THEN fund approval

IF CONDITION fund approved THEN order material

Affects (indent verify, valid indent)

Affects (order material, not periodicals)

Affects (fund approval, fund approved)

**ARL Rule:**

IF COA new indent AND CONDITION (valid indent AND fund approved) THEN order material

**SBVR Structured English form:**

It is obligatory that approved indented material is ordered.

**BUSINESS RULE NAME** receive material

**RULE TYPE** AGGREGATE

**BODY**

IF COA order material THEN take delivery

IF CONDITION (material received AND NOT periodicals) THEN pay

Affects (take delivery, material received)

**ARL Rule:**

IF COA order material THEN take delivery

**SBVR Structured English form:**

It is obligatory that delivery is received for ordered material.

It is obligatory that payment is made for received periodicals.

**BUSINESS RULE NAME** stocking material

**RULE TYPE** TRANSITIVE

**BODY**

IF CONDITION material received THEN catalogue material

IF COA catalogue material THEN shelve material

**ARL Rule:**

IF CONDITION material received THEN catalogue material

**SBVR Structured English form:**

It is obligatory that material is catalogued when received.

It is obligatory that material is placed on shelf after cataloguing.

6. CONCLUSION

The BRG has now established that business rules are first formulated in the business environment and it is only thereafter that they are taken into information systems. The latter is found in the CIM level of MDA that is populated by SBVR. For the former, we have used a business rules metamodel that expresses business rules along with their properties.

In considering the conversion of business rules to their SBVR expressions, we have introduced an intermediate layer resulting in a two-step conversion process. At the intermediate layer, hierarchical business rules are converted into flat structures and logical connectives of AND, OR, NOT are introduced. In converting this to SBVR the main problem is that of converting course of actions in rule antecedents to conditions. This is done by the AFFECTS relationship.

It is to be noted that preference and priority properties cannot be directly expressed in SBVR business rules. However, these are to be passed on further down the MDA layers. Therefore, SBVR expressions must be suitable annotated to not lose this information. On the other hand, deadlines can be expressed through strict or deferred enforcement.

7. REFERENCES


