

GETTING STARTED WITH BUSINESS RULES MANAGEMENT



Applies to:

SAP NetWeaver Composition Environment 7.1.1

Summary

This document gives an introduction to business rules management (BRM). It also describes the methods of BRM, for example rules in databases and rules in application code. It explains how to get value from a BRM system in your company.

Author: Rajagopalan Narayanan

Company: SAP Labs India Pvt. Ltd.

Created on: 19 February 2008

Author Bio



Rajagopalan Narayanan is a Product Manager in the SAP NetWeaver Product Management Business Process Management team. His focus area is Business Rules Management. He has many years of experience with Business Rules Management.

Table of Contents

What are Business Rules?	3
Examples of Common Kinds of Business Rules	3
Business Rules in Applications – Conventional Methods.....	4
Rules in Databases	4
Rules in Application Code	5
Rules in Scripts	5
What are the Challenges with Conventional Methods?.....	5
Lack of visibility into how applications enforce business rules	5
Software maintenance issues	5
Scattered knowledge across the organization	6
Introducing Business Rules Management Systems	6
The Value Proposition of Business Rules Management Systems.....	7
Accurate and precise decision automation	8
Improved visibility for business users	8
Faster turnaround times	8
Centralization of knowledge	8
Copyright.....	9

What are Business Rules?

Business rules describe or represent constraints on the behavior of the business. Business rules represent the core business logic of each organization; they guide and control the basic business processes that form the back bone of any business transaction.

Business rules represent a set of standard business practices or policies that need to be applied consistently across business activities. Therefore it is important to understand the difference between a “Rule” in software applications and a “Business rule”.

A “Rule” can be defined as a set of conditions with some associated actions. These are typically introduced in the form of an ‘If -> Then’ statement. Examples include: ‘Database Constraints’, ‘Object Model Rules’, etc. These “Plain Rules” are owned by IT and not by the Line of Business (LoB).

“Business rules” on the other hand represent organization policies and guidelines and are always owned by the Business. Examples of business rules include ‘Loan Product Eligibility Guidelines’, ‘Product Configuration Rules’, etc.

When introducing business rules organizations have to take in consideration that rules are inherent in:

- Corporate charters
- Management practices and regulatory forces
- Human resources management
- Marketing strategies
- Pricing policies
- Products and services offerings
- Customer relationship practices

Business rules are the most dynamic component of any application. Therefore their constant and correct identification and externalization improve the organization's adaptability to industry changes and competition. Externalizing business logic in the form of simple rules ensures that these rules are easily communicated across the organization and managed separately from the application code: *business adaptability and flexibility via IT stability.*

[Back to top](#)

Examples of Common Kinds of Business Rules

Here are some examples of different kinds of business rules that you may encounter:

Validation Rules	Form Validation Claim Validation
Rate Calculation Rules	Computation of Taxes Interest Rates Commissions Premiums

	Bonuses Duties, etc.
Decision Rules	Loan Eligibility Customs Clearance, etc.
Recommendation Rules	Marketing Promotions Medical Recommendations Online Product Chooser, etc.
Personalization Rules	Website Product Relationship Personalization
Internationalization and Localization Rules	International Products (Insurance) Custom Duties, etc.
Exceptions and Special Rules	Exceptions Special Accounts Special Arrangements, etc.
Optimization and Configuration Rules	Product Configuration Deal Configuration, etc.

Although there are some industry specific wordings, business rules are typically worded in simple language that can be understood easily across different organizational levels and roles. Some examples are given below:

- If a customer's spending habit and credit standing are good, grant a 10% discount on the total bill amount.
- If the employee is a programmer with a good performance, give them 15% of their basic as a bonus.
- Depending on the principal amount, set the interest rate as shown below (include table and /or picture).

[Back to top](#)

Business Rules in Applications – Conventional Methods

Many enterprise applications require enforcement of business rules. Developers of these applications typically use one of the many available programming techniques to model these business rules, such as:

Rules in Databases

This is one of the most common ways to implement business rules. You will find business rules in stored procedures and triggers.

One common example would be the rules that calculate sales commissions of Insurance Field Agents. Many applications that cater to this use case implement the involved calculations inside of stored procedures.

Rules in Application Code

A huge variety of business logic may be found in application code. Examples include Tax Calculations, Billing Rules, etc.

Rules in Scripts

Another popular way of implementing business rules is using a particular scripting language. Popular scripting languages include java script (rules in web pages), ruby and proprietary scripts. Examples of rules implemented using scripts include budget forecast models.

There are also many cases where business rules simply exist as tacit knowledge in the minds of different business users. A good way to start business optimization may be to identify these rules and employ informed decision automation in order to automate them and enrich the standardized practices base.

[Back to top](#)

What are the Challenges with Conventional Methods?

There are some unique business rules characteristics that may bring challenges in the implementation process or in case the use of conventional programming techniques is enforced. These characteristics are:

- Business rules are owned by the Business and therefore IT will not understand their impact
- Business rules can be very volatile, and change often
- Compliance to regulations is a direct function of how well business rules are enforced in the organization

Because of these characteristics, conventional methods of realizing business rules automation can face organizations with some of the following challenges:

Lack of visibility into how applications enforce business rules

- The IT systems become a big 'black box'. The LoB cannot look into the system and understand how business decisions are being taken
- Critical business rules are locked in code-lines or databases

Software maintenance issues

- Many business policies are very volatile. Changing them as fast as the business wants, normally becomes a real challenge. It is very difficult to estimate what may be the impact of a simple change in the code base
- IT does not understand business rules, which means that hardwired/hard-coded business rules are very difficult to change
- This results in long testing cycle and change cycle times of business rules, which in turn increases the cost of correction and rework
- Another big risk for organizations is that hidden business rules increase the risk of compliance violations
- Communication cycle of understanding business rules, when intermediaries are required to communicate the business requirements to the IT, may lead to delays and errors - one of the main reasons why organizations wanted to start automating in the first place

Scattered knowledge across the organization

- When business rules are implemented using conventional methods, they tend to get scattered across code, databases, instruction manuals, handbooks and excel spreadsheets
- This creates a situation where organizations end up with inadequate, ambiguous business rules spread across software systems, and even duplicates in many cases
- An even more serious situation may occur when same rules provide conflicting results, because of the way they were implemented

[Back to top](#)

Introducing Business Rules Management Systems

Business rules define the manner in which organizations conduct their business. It is, therefore, imperative that they keep changing in accordance with market requirements.

Managing changes to core business logic can be a challenge in itself for the following main reasons:

- Business rules may not be available in one single location - they may be spread among multiple policy documents and often across different geographical locations
- Visibility of business rules may suffer when they are buried deep within application logic, when business policies are locked away in code, database stored procedures or in spreadsheets
- Business rules stored in spreadsheets cannot be automated; policies which are in code, in the form of database stored procedures or any other IT artifacts are less likely to be discovered and understood by business users
- Many policies are very complex and very difficult for IT to implement or manage
- IT professionals often lack the domain knowledge to understand the implications of a change in business rules; business users, who have the knowledge, do not have the required expertise to make these changes at the code-level. The opportunity cost of such delays can be very high in a dynamic market
- There may be a need for adequate skilled manpower to handle changes to business policies in a demanding business environment positioned in a dynamic market

Organizations need a solution that:

- Will allow IT to model the complex business policies and execute them at the same time
- Will help IT to externalize the business policies so that changes to policy have a minimal or zero impact on the IT system
- Will allow to make changes to business policy much faster
- Will enable IT to provide an executable plan for eventual involvement and transfer of control of business policies to the business

Business Rules Management Systems (BRMS) aim to address these problems and handle changes to business rules efficiently. One of the biggest advantages of a BRMS comes from externalizing rules from application code. Separating out business logic from application logic ensures that business users can take charge of changes to rules without depending on IT professionals at all times. BRMS enables organizations to represent rules in a natural-language-like format, as the business goes and changes, instead of in the form of complex application code.

A well-designed BRMS can offer the following key features:

- Low memory footprint and optimized performance
- Ability to represent rules in a natural-language-like format for easy understanding among all users
- Ability to externalize rules from application logic and store them in one central location
- Ability to model complex business rules and execute them at the same time
- Ability to build rules and edit them with ease without being dependent on IT professionals all the time
- Allow versioning of rules to maintain history of changes over time
- Access control to restrict access to business policies
- Ability to access rules remotely from any geographical location and change them in real-time
- Easy integration with the existing systems and IT infrastructure of the enterprise
- Easy-to-use, extensible and flexible API

The keywords here are “model complex policies”, and “externalize”. Now, any solution that your organization would employ **must** provide at least the following capabilities:

- ❖ *Provide different ways for representing business policies, because business policies come in all kinds of flavors and shapes:*
 - There are many policies that live in spreadsheets. It is not uncommon to see Insurance or Loan Agents consulting a print-off or a file version of pricing/rate tables. Now, lets call these ‘*Decision Tables*’
 - Policy constraints on business information and data. For example, consider this rule, ‘If AssetValue > 5 million’, then classify as a ‘Super Platinum Customer’.
 - Sometimes decisions are a result of certain policies that have to be applied in a specific sequence, like in a flow-chart. Let’s call these ‘*Sequential Rules*’
 - There are of course information validation rules that need to be applied, like in a form, or on supplied business data like a ‘Stock Trade’
- ❖ *Provide a runtime decisioning component (engine) that will apply business policies on the business information supplied to reach a certain business decision*
- ❖ *Provide ways for the business and IT to validate and refine business rules*
- ❖ *Provide Change Management tools to manage fast changing and volatile business rules*

A well-designed Business Rules Management System (BRMS) is a class of Enterprise software that provides all these capabilities.

The components of a BRMS include the following:

- A Development Environment for modeling, testing and refining business rules
- A Rule Engine for executing business rules (Application Integration point)
- A Rule Repository for managing rules (security, versioning, alerts, etc.)
- A Business User interface for modeling, changing, testing, analyzing and optimizing business rules

[Back to top](#)

The Value Proposition of Business Rules Management Systems

To understand completely the need of implementing a BRMS, let’s revisit its value proposition in solving some major business problems:

1. Most complex rules can be captured using the rule formats that are supported by the BRMS. This greatly resolves the automation problem. There are many well documented case studies of enterprises achieving 99% of Business Decision Automation

2. The rules are externalized and not hidden inside of the applications; therefore changing the rules can now be done independently without touching of the application code itself. This will immediately bring down any change management efforts and constraints
3. The business rules (policies) captured as decision tables, flows, etc., can be shared. These are far easier to understand and work with than code. Therefore visibility on decision making process is achieved instantly once the rules are written

The value proposition offered by employing a BRMS can be summarized in the following way:

Accurate and precise decision automation

- ❖ Precise and informed decisions are taken driven by rules that users know are correct
- ❖ Consistent and unambiguous application of business rules across all enterprise systems
- ❖ Additionally, business rules can be reused in multiple IT systems consistently

Improved visibility for business users

- ❖ Users can translate business rules into a natural-language format to improve visibility - business rules are visible and understandable to the LoB
- ❖ Business users own and influence automated decisions
- ❖ Business users share and collaborate on business rules and this helps reduce errors and eliminate eventual waste of time due to collaboration issues

Faster turnaround times

- ❖ Because rules are externalized from application code, users are able to change them faster than if they were locked up in code
- ❖ Users can build rules, test them and integrate them with applications
- ❖ Business users can quickly and easily implement changes same as IT users and this immediately reduces maintenance costs of enterprise applications
- ❖ Real-time responsiveness to market changes is faster and more reliable

Centralization of knowledge

- ❖ Using a BRMS to manage business rules ensures sharing 'One single version of the Truth'
- ❖ Business rules can be centralized, administered and maintained in isolation from the application
- ❖ A BRMS also enables a standardized deployment of business critical rules consistently across the enterprise

Introducing a Business Rules Management System immediately brings in many benefits to every organization. Here is an example: a pricing service used by a mortgage company is driven by a set of pricing rules managed by a BRMS for all the company's loan products. This organization can then use the same pricing service across all their sales channels ensuring: accurate decision automation, centralized knowledge of best business practices, which in turn minimizes confusion and errors, and allows reduction of processing turnaround times. Last but not least, rules transparency improves training and collaboration, where users share and build on same policies, following a consistent approach towards all business processes.

[Back to top](#)

Copyright

© 2008 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.