

Business Rules Management Systems

November 26, 2004

A technical brief from Chariot Solutions (<http://www.chariotsolutions.com>)

Business Rules Management Systems, or BRMS, are enterprise product suites designed to help companies react more quickly to changes in business policy. A BRMS accomplishes this by providing tools that isolate *business rules* from traditional application code. Business rules are definitions of a company's business policy, typically stated in the form of if-then-else statements. For example:

<p>IF The customer has more than 2 late payments in the past year THEN Only offer the paid-in-full renewal option</p>
--

Traditionally, these policies have been hard-wired into application code. When a change in policy occurs, a developer must find, modify, test and deploy the change. For some companies, this could trigger a process that involves numerous people and spans several days.

With a BRMS, business rules are maintained separately within a rules repository and queried as needed from within trigger points in the application code. In the previous example, the application code would simply query the BRMS for a collection of eligible Bill Plan Options based upon a customer profile. The BRMS would contain the hundreds or even thousands of rules pertaining to bill plan eligibility:

<pre>BillPlanOptions options = rulesEngine.getBillPlanOptions(customer);</pre>
--

Further, since most BRMS come with a full complement of tools for definition, maintenance, and deployment of rules, the more technically adept policy makers could manage simple changes to their own rules. Even if IT resources maintain rules, the value in changing business policy without modifying compiled code can have a tremendous impact on a company's ability to react to changes in business policy.

However, implementing a BRMS can be a difficult and costly exercise for a company with little experience in rules technology. Therefore, the enterprise should consider the following questions in deciding whether to make the leap to incorporating rules technology into their application architectures.

Business Need or Novelty?

Several environmental factors have pushed BRMS into the spotlight, including:

- A rapidly changing business environment that requires more flexible IT applications
- Recent legislation, such as Sarbanes-Oxley, that place greater demands on managing policies, procedures, and decision making criteria as a corporate asset
- Awareness of the need to orchestrate complicated business processes and workflows amongst loosely coupled applications that span across departmental or corporate boundaries

BRMS vendors have responded to these challenges with glossy marketing brochures and impressive technical white papers. However, be aware that licensing rules technology can be costly. Most commercial rules products involve server licenses that can run in excess of \$20K per CPU and developer tools that approach \$10K per seat. Depending on the breadth and load of the application(s), it is not uncommon to spend more than \$100K on licenses before any investment in training, professional services, etc.

In order to justify investments of this magnitude, the benefits must be significant. Of course, one should always investigate whether there are there less costly alternatives. If an organization simply needs to manage a few dozen rules outside of application code, a relational database to abstract the business conditions or perhaps using XSL style sheets to operate on (and

update) XML data passed to and from the application and an XSLT processor may suffice. While not as robust as a full scale BRMS, it can prove to be a viable and very low cost option for externalizing business rules¹.

On the other hand, if the number of rules reaches into the hundreds or thousands **or** the consequence of inconsistent application of the rules (either via manual processes or through home grown technology solutions) becomes high, a BRMS may be the way to go². For example, pure BRMS are extremely popular in industries such as insurance and financial services. These industries are traditionally highly regulated with countless rules related to claim processing, underwriting, policy administration, etc. Furthermore, the rules can change quite often due to the sheer number of regulatory bodies influencing the process. Insurance and mortgage underwriters also leverage rules engines to consistently apply the knowledge acquired over decades of experience when making decisions on pricing, acceptability, and risk classification. Inconsistently applying rules in these cases can result in millions of dollars in claims leakage or an overly risky loan portfolio. Other applications that often utilize BRMS include:

- Sophisticated product configuration applications
- Device or instrument monitoring applications
- Decision support and executive dashboard applications

Business Rules Engine or Business Process Management Software?

As mentioned previously, BRMS are increasing in popularity in part due to the need to integrate workflows and loosely coupled applications. For example, in an insurance company, a policyholder claim may first need to be routed to a coverage verification unit then to an office specializing in triage for personal injury claims before reaching a supervisor who ultimately assigns the file to an available adjuster. Behind each of these steps, a number of business rules may determine routing destinations, escalation thresholds, etc. For this reason, many Business Process Management software providers either bundle a rules engine into their product, or have partnered with a BRMS vendor to tightly integrate the rules component into the offering. Examples include Microsoft Biztalk 2004, Weblogic Integrator, Seebeyond, Sonic Orchestration Server, WebMethods Process Modeling, Staffware, Filenet, and Versata.

If the enterprise requirements fall into more of a business process workflow category, BPM solutions that either embed a rules engine or tightly integrate with third party BRMS may be a better fit. For example, Staffware (from TIBCO) tightly integrates with the Corticon BRMS to provide a complete rule driven work flow solution.

Is there an Enterprise Commitment to Rules Technology?

One of the more difficult challenges an enterprise faces when implementing rules technology is gaining commitment for the necessary investment in training and knowledge extraction. Most corporate developers are not well versed in the declarative programming style nor or can they be expected to immediately trust a relative “black box” such as a rules engine to handle important application decision logic. The enterprise must be prepared to invest in developer training if a deep dependency on external consultants is to be avoided. Regardless of what the marketing brochures imply, a BRMS implementation will require the involvement of technical personnel to integrate the technology. It is only after applications are re-architected to leverage the BRMS that the enterprise will begin to realize the power of turning business rule maintenance over to business people or analysts.

Also, business rule extraction can be a time consuming and costly experience. These rules ultimately must be expressed in the terms the BRMS can understand. Existing rules are often in the form of bulky procedure manuals, existing application code, spreadsheets, memos, or even tribal knowledge. There are experts who specialize in knowledge extraction as part of BRMS projects. These costs must be budgeted for if internal expertise does not exist.

Which BRMS is Right?

If the solution requires a combined BPM/BRMS work low solution, Tom Baeyens has written an excellent article outlining the “State of Workflow” which is published on The Server Side (www.theserverside.com/articles/content/Workflow/article.html). Tom also offers a comprehensive list of open source and commercial offerings in this area.

¹ Unfortunately, open source rules engine projects such as Drools or Mandarax are still relatively immature and provide little in the way of enterprise caliber tools for managing large rule repositories

² Existing application source code and procedure manuals are good resources for determining the number of business rules involved in a business process.

In the pure BRMS space, the two most visible commercial offerings are from ILOG (JRules) and Fair Isaac (Blaze Advisor). These are comprehensive product suites that have matured significantly over time (some may remember Blaze Advisor's heritage as Neuron Data). At the core of their rules engine offerings, these products feature forward chaining inference engines based on the Rete³ (or Rete II) algorithm. The Rete algorithm was initially developed by Dr. Charles Forgy more than 20 years ago at Carnegie-Mellon as an advanced pattern matching algorithm to speed the process of testing conditions in rules. Further, these market leaders boast comprehensive rule repository, optimization, and debugging tools particularly suited for the enterprise.

Other competitors include Corticon BRMS (Corticon), PegaRules (PegaSystems), Aion Business Rules Expert (Computer Associates), Authorete and CafeRete (Haley), Corvid (Exsys), and JESS/Java Expert System Shell (Sandia National Laboratories). JESS is available as a free download for non-commercial use. While it doesn't enjoy the comprehensive tool support provided by other commercial offerings, it does possess a sizable user support group.

One area that distinguishes certain BRMS is in their approach to how rules should be authored and managed. For example, the Corticon product emphasizes a more simplistic approach to authoring rules in a user interface metaphor similar to a spreadsheet. However, the interface is extremely powerful and allows an author to quickly identify inconsistent rules and indeterminate conditions. This is something the more mature Rete-based products have difficulty achieving in an intuitive manner.

Another area to consider when choosing BRMS is the type of decision processing required. For example, does the business scenario typically involve a relatively small set of contextual information (i.e. "facts") but require that potentially thousands of rules be evaluated on that information? If so, a Rete based product will scale well for the needs of most applications. However, if there is a sophisticated and potentially deep object model (with a large number of facts) but a relatively smaller number of rules, a product such as that offered by Corticon may be more scalable over time. This phenomenon exists because the Rete algorithm, while fast in pattern matching (rule elimination), is memory intensive. A sizable working context will expose this trade off more quickly than will a non-Rete based system. Conversely, a rule process with thousands of rules operating on a relatively small number of facts may seem slow when using non-Rete based systems.

Finally, BRMS products should clearly fit within the enterprise's architectural model and platform. While many products claim that a web-services interface eliminates this dependency, it is still likely that a fair amount of integration with current systems will be necessary. Having a clear idea of how a BRMS server will fit within a J2EE Application Server or .NET environment is a significant advantage. Further, it is helpful to have staff technology personnel proficient in the underlying technology of the BRMS in the event troubleshooting is required.

Summary

BRMS can facilitate the management of business rules and policies separate from traditional application code, thus creating the opportunity for tighter control and isolation of change management procedures. However, be prepared to invest significantly in terms of training, knowledge extraction, and integration over and above licensing costs. Consequently, there must be a clear and significant advantage to introducing rules technology to the enterprise. As always, it is wise to research whether lower cost alternatives exist so as to reduce the risk and the payback period. However, a successful BRMS implementation can yield significant tangible and intangible benefits for the enterprise willing to make the investment.

About the Author

Dan Hayes is a technical consultant and rules engine specialist for Chariot Solutions, LLC located in Ft. Washington, PA. Dan has over 15 years experience in the insurance and financial services industry as an executive, entrepreneur, and technical consultant having worked with some of the largest insurance companies on enterprise technology projects including BRMS implementations. Dan has experience with commercial products from Fair Isaac (Blaze Advisor) and ILOG (JRules) as well as JESS and the open source Drools project. He is a Sun Certified J2EE Enterprise Architect and holds Master's Degrees from Columbia University (MBA) and Penn State University (M.S. Software Engineering).

³ Dr. Charles Forgy chose the term "Rete" to describe his pattern-matching algorithm. Rete is Latin for "network".