

## Achieving Decision Consistency Across the SOA-based Enterprise

### Using Business Rules Management Systems in an SOA

James Taylor

October 2005



### Summary:

The adoption of a service-oriented architecture (SOA) provides businesses with the ability to rapidly deploy new applications and easily integrate with other component applications both inside and outside the organization. This decentralized application environment provides a great deal of flexibility for business units and IT departments, but it also creates difficulty in managing the consistency of business decisions delivered through various applications. Business rules management systems (BRMS) provide a mechanism for managing decision logic and act as a conductor in order to align application decision behavior.

The key to BRMS is the use of a centralized rules repository, within which resides decision logic applications. Applications communicate with a rules engine in order to process those business rules specific to the decision required for the particular application and situational context.

This paper will show how business rules management systems fit within a service-oriented architecture, how BRMS can act as intermediary between service-based applications and legacy applications, and how companies can use a BRMS to manage decision processes across the enterprise and achieve decision consistency across an SOA-based enterprise.

"IDC believes that companies should consider business rules management solutions as a way to implement rule changes within their companies and achieve quality control in terms of being able to more accurately interpret the requirements for new rules and implement exactly the changes that need to be made as well as make those changes only once. Although not appropriate for all types of applications, business rules management tools should be assessed for their value to each company considering implementing Web services in the future."<sup>1</sup>

<sup>1</sup> All quotes from "Cost –Effective Solution to Web Services and Application Development Using Fair Isaac Blaze Advisor" by Rikki Kirzner. Filing Information: August 2003, IDC #29929, Volume: 1, Tab: Vendors

## Table of Contents

The parallel benefits of SOA and Business Rules .....	1
A Business Rule Service Architecture .....	3
BRMS Evolution.....	4
Transitioning to an SOA-based Environment Using a BRMS .....	4
Examples of successful rule services .....	5
Conclusion .....	8
About Fair Isaac .....	9

## The parallel benefits of SOA and business rules

The widespread implementation of service-oriented architectures within large-scale enterprise systems has shown that we have moved past the early adoption phase. The era of monolithic all-encompassing enterprise applications is over, and the vendors of such applications (such as the major ERP and CRM systems vendors) have moved towards componentization, while simultaneously redesigning their products for standards-based integration. Both IT customers and vendors have come to realize that the name of the game is to create a set of modular software applications that can work together and allow each component of the system to focus on what it does best. Using an SOA approach, organizations can create and implement new applications more quickly than with other development approaches. These new service-based applications can easily be designed to interact with other “services” both inside and outside of the organization.

Business rules can be found in every business process in every organization. A BRMS allows an organization to centralize discovery, storage and execution of the rules used in their operations. A business rules approach can deliver greater automation, better business agility and reduce maintenance costs. Keeping rules independent of the underlying data and separate from the application structures used to support business processes improves reuse, management and compliance. It allows rules to be changed and executed as needed without changing the application code.

While the decentralized SOA application environment provides a great deal of flexibility for business units and IT departments, it also creates a new set of challenges. How can organizations effectively manage the consistency of business decisions delivered through various applications? And how can organizations reliably implement changes to business decision logic across all their services?

Business rules management systems provide a highly effective and efficient mechanism for managing decision logic and acting as a conductor in order to align decision behavior. The key to BRMS is the use of a centralized rules repository, within which resides the decision logic that applications use in their process interactions. Not only does a BRMS fit within a service-oriented architecture, it can act both as an intermediary between applications and as the decision management component for application behavior implemented as a set of services.

The primary goal of business rules management is the separation of business decision processes from the mechanics of application I/O and control code. A BRMS allows for the storage and management of business rules, as well as the ability to execute decision processes through a rules engine.

There are four main benefits to using a business rules management approach within any enterprise architecture:

- The ability to have disparate operational systems and services act consistently without having to maintain duplicate decision logic in each.
- The ability to change decision logic without having to make any changes within the code of operational systems or services, thereby significantly reducing QA testing when new decision logic is implemented.
- The ability to define decision logic using sophisticated methods not possible through standard coding, such as decision tables, decision trees, scorecards and the use of decision inferencing among chains of interrelated rules.
- The ability to automate many operational decisions that have previously been made manually.

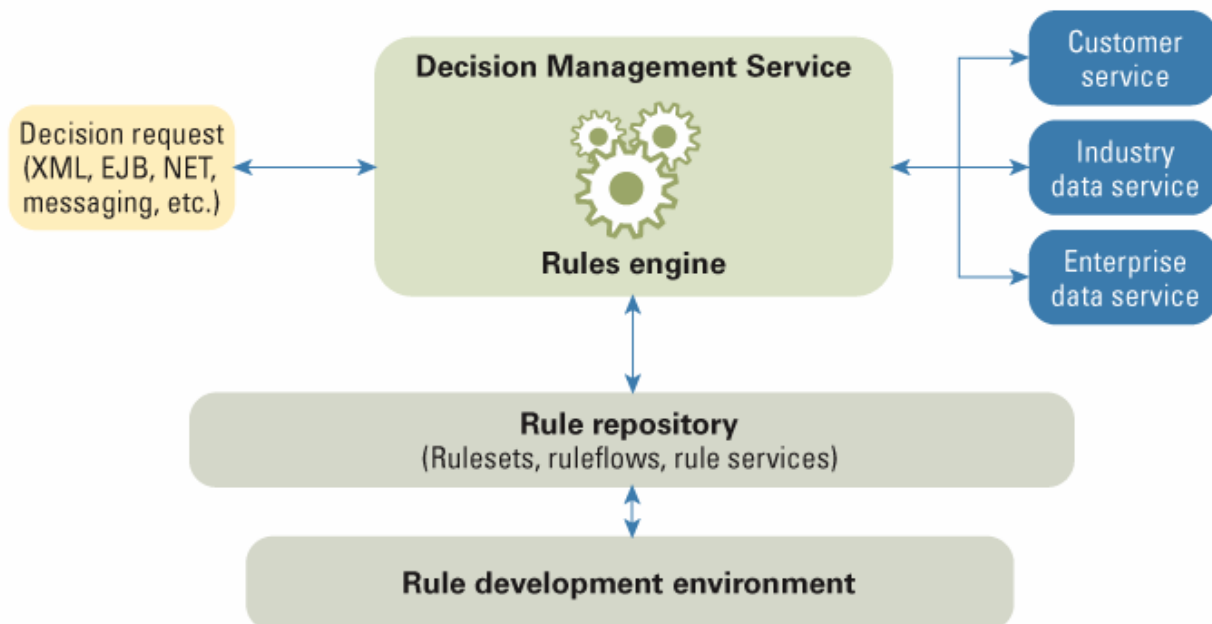
The types of applications using BRMS to handle decision logic cover many different industries and types of functionality. The following list provides a sample of ways that organizations are using BRMS:

- Benefits eligibility determination
- Claims handling
- Credit approval and limit determination
- Credit collection strategies
- Data validation
- Diagnostic advice
- Insurance underwriting and policy pricing
- Problem resolution procedures
- Product assembly configuration validation
- Product recommendations
- Regulatory compliance management

## A business rule service architecture

Business rules management systems define business decision logic as rules, and these rules are stored in a rule repository. Using the BRMS rule development environment, individual rules are then combined into rulesets, functional blocks defining decision logic for a subtask. Rulesets can be combined in a procedural decision flow (ruleflow) to conditionally use the assigned rulesets in a sequence that achieves a desired business decision. This combination of ruleflows, rulesets and individual rules are utilized within rule-based services (rule service) that are used to guide the decision logic for a specific business process. Rule services can be combined and are made executable within a decision management service. Decision management services use the BRMS rule engine to process the inputs from operational system applications through rule services, as well as processing data (from separate data sources) and analytic models (embedded within rule services) as required in order to return the optimal decision output back to the client application.

### A BUSINESS RULES MANAGEMENT ARCHITECTURE



The differentiation between individual rules, rulesets, ruleflows and rule services provides the highest level of flexibility in the reuse and sharing of decision logic across multiple services. As the functionality across each service is different, so too is each service's use of rules. Several services may require a common rule, but each may have its own unique set of related rules or decision processes that need to be associated with that specific rule. Allowing this reuse of rules across services can speed their development and reduce the cost of maintenance above and beyond the improvements that come from a services-based orientation.

Services offer a way to become more responsive and more adaptable compared to traditional application architectures. An SOA approach can reduce the effort to build a new application by assembling services and by providing both standard components and standard ways to assemble them. In theory, this means

that it will be easier to personalize and customize applications, and easier to evolve applications. The challenge lies in developing software constructs (services) that are self-contained and coherent but that can also be assembled and customized.

“IDC believes that business rules are well suited to being easily transformed into the type of components that can help achieve this vision.”

Why is this? The services you need to deliver on the vision of SOA need to be dynamic, configurable, platform-independent and easy to evolve as business needs change—the “guts” of a service can be easily evolved without changing its interfaces and, thus, the interactions of other services with it. Business rules are ideally suited as a way to build these highly configurable services. Additionally business rules offer a way to manage the complexity of assembling these services into applications or processes. This allows changes in how the business operates to be quickly implemented using business rules, resulting in new or changed combinations of services without the need to edit low-level code. In particular, those organizations with large numbers of rules, rules that change often or very complex rules will benefit, as will those that must demonstrate compliance with complex or extensive regulations.

## BRMS evolution

Business rules management systems started to become available in the late 1990s as the next generation of expert systems. While expert systems were designed to help solve specific types of problems, BRMS were designed as “middleware” applications that could be easily integrated with enterprise applications, information systems and databases. The initial group of BRMS vendors developed their products using Java technology; now, Fair Isaac has .NET and COBOL-based versions to allow decision logic to be managed more effectively across a heterogeneous application environment. As “middleware,” robust connectivity is an important aspect of BRMS functionality. In an SOA environment, the primary means of communication between services and BRMS is through the use of XML/SOAP, but other methods of communication include HTTP, EJB, COM+/.NET and messaging protocols (MQ Series and JMS). In many cases, decision management services are implemented within application servers, such as those from IBM, BEA, Oracle, Sun and Microsoft. Application servers provide built-in security protocols, which resolves one of the primary concerns that enterprise architects have when moving their systems to a service-oriented architecture.

## Transitioning to an SOA-based environment using a BRMS

Implementation of a BRMS is a good first step in moving to an SOA-based environment. In fact, the transition to SOA is not an all-or-none proposition; many organizations start by implementing new applications as services. It is also possible to tie new service-based applications and existing applications together using a BRMS as the common decisioning mechanism. This is especially useful in legacy modernization efforts. In many cases, organizations rely on older mainframe systems to handle many crucial business functions, and it just not feasible to completely replace these systems. Instead, a more practical approach is to slowly transition portions of functionality off of legacy systems while allowing them to continue to perform their primary processing responsibilities. Removing decision logic from operational application code is a good initial step in legacy modernization efforts. By doing this, legacy systems are actually stabilized, since fewer changes need to be made to the core code over time. Decision logic code is replaced with a simple invocation call to a decision management service, along with a mechanism for receiving the output from the BRMS. When decision logic (rules) is changed, the legacy system code is unaffected since the changes are made within the BRMS. Core processing functions within legacy systems can then be transitioned to new service-based applications incrementally, and legacy systems can be decommissioned as their functionality is fully replaced by newer systems.

The use of BRMS also allows for faster creation and deployment of new services within a service-oriented architecture. The decoupling of application functionality and decision logic allows developers to focus on core application functionality while having a separate group of business analysts create the rule services that will manage the outputs of the new application. Once both the application and the corresponding rule services are ready for testing, the developers need to create a decision management service to handle the interaction between the application and the rules engine (The major BRMS vendors provide wizard-driven facilities for setting up a decision management service using the standard SOA methods as described previously in the paper). The ability to work in parallel on both application development and decision management definition, along with having developers focus their attention on core application functionality, greatly reduces the time required to bring new services into production. Business rules management tools also allow developers to create interfaces allowing business rules to be maintained by business users. This allows for the customization and evolution of services in even the most dynamic of environments.

“Blaze Advisor offers a way for organizations to implement such business changes more efficiently and more quickly, with reduced chance of errors, less cost, and less operational impact. Because business rules are separated from and independent of the underlying system code that keeps a business application operating, they can be changed without impacting basic system functionality.”

## Examples of successful rule services

### Case study #1

The California Department of Motor Vehicles (DMV) is responsible for collecting approximately \$4.1 billion annually in vehicle registration fees. Centralized computer systems in Sacramento communicate with local systems across 167 field offices throughout the state to handle the complex task of calculating registration fees for the nation’s largest population of new and used autos, trucks, motorcycles, vessels and other vehicle types. In 2000, the DMV realized that it needed to update and unite the two separate vehicle fee systems, and move to a modern system that would meet the state’s strategic realignment towards e-Government with future public access via the internet.

Because of the two different computing platforms, changes and updates required two separate development efforts, two different analyst teams, and two different databases. This made it difficult to coordinate changes and ensure consistency between the two systems. Due to the complexity of the programs and duplication of effort required to make changes to two separate systems, the DMV was challenged to meet legislatively mandated deadlines for fee changes.

The team recognized that a business rules management system would enable the separation of business logic, policies and processes from the actual application programming. The ability to give the power of business policies and business changes to analysts instead of programmers was a monumental change to the current application development process for the DMV.

Any BRMS would have to be understandable and usable by non-technical analysts responsible for overseeing legislative compliance. It would also require an intuitive user interface that would allow comprehensive control and testing of rules without obscure programming syntax. From a technical standpoint, it must run quickly and scale to handle massive numbers of transactions on a variety of computer systems, from the largest mainframes to office servers. And it would have to do all this without requiring replacement or rewriting of the vast majority of the legacy applications and systems in place throughout the DMV infrastructure

The DMV's Vehicle Registration team reengineered a vehicle registration fee service that could work in conjunction with both systems in a manner that would be transparent to the end user. The revised solution used Fair Isaac Blaze Advisor, IBM WebSphere application server, and a Java 2 Enterprise Edition (J2EE) solution operating on a mainframe computer.

Leveraging the power of a centralized decision management service built by the DMV technical development team, the DMV successfully gave the non-technical analysts who are responsible for overseeing legislative compliance the ability to ensure proper implementation of policy rules across the DMV vehicle registration fee systems without having to become programmers.

The service required over 2,000 rules and numerous rulesets, ruleflows, tables and processes for several hundred thousand business transactions per day. While many projects boast tens of thousands of rules, the DMV team's thorough methodology resulted in an effective ruleset of 2,100. Blaze Advisor positioned the DMV to make future system enhancements more easily and to add centralized and consistent rules to new external-facing services such as a self-service web site and telephone response systems. A recent example of this can be seen with the DMV's web-based fee calculator:

<https://vrir.dmv.ca.gov/FeeCalculatorWeb/index.jsp>.

## Case study #2

First American Field Services (FAFS) is one of the largest default mortgage field services company in the United States, providing inspection, maintenance and service repairs on more than 150,000 properties per month. When a property goes into delinquency, foreclosure or bankruptcy, financial institutions retain FAFS to help protect their investments. Many of the properties are covered by government-backed loans and are regulated by US government agencies, such as the Department of Housing and Urban Development (HUD), the Federal Housing Administration (FHA) and the Department of Veterans Affairs (VA). Any work performed on homes backed by FHA and VA loans must be carried out in accordance with voluminous federal regulations, including tight timetables, for a contractor to receive payment.

Using its network of more than 1,000 field representatives, FAFS has built an excellent reputation within the US mortgage industry. But too many processes were paper-based, resulting in costly re-keying and a constant expense to ensure accuracy. In addition to the inefficiencies and delays inherent to a paper-based workflow, FAFS needed a better way of ensuring that HUD regulations were followed. HUD regulations fill hundreds of pages, and must be updated frequently and sent out to the field. Banking customers wanted more flexibility, and FAFS needed to provide this while reducing expenses.

As FAFS looked toward updating its IT infrastructure, it identified key needs, including:

- Eliminating paper-based workflow to enhance project time efficiency and accuracy.
- Embedding HUD regulations and other requirements into the system to ensure compliance.
- Making it easier to customize services for customers.
- Ensuring interoperability with legacy systems that were essential for other roles.

The solution was created using Microsoft® Visual Studio® .NET development system and the Microsoft .NET Framework. FAFS chose Fair Isaac Blaze Advisor as the BRMS to provide an application for automating implementation of rules, a function that serves as the heart of FAFS' service-based Property Inspection and Maintenance System (PIMS) solution. FAFS uses Blaze Advisor for .NET as a rules engine to automatically embed relevant HUD regulations, other requirements and bank-specific customizations into work orders. Blaze Advisor for .NET handles a spectrum of functions that were previously done manually, including:

- Automation of instructions to contractors according to HUD regulations
- Management of order distribution to the field
- Client pricing and vendor payment
- Queue management and escalations for exception processing
- Automatic generation of new orders based on results of other orders
- Maximums and caps for each line item on each order

The .NET-connected PIMS application had a three-tier architecture that included a presentation tier, a business tier and a data tier. The business tier included all the rule services and a .NET data bus to control workflow in and out of the rules engine, and coordinate communication with the database tier.

First American Field Services has enjoyed a number of benefits since deploying its .NET-connected PIMS application, including consistent application of complex regulations, faster and more accurate project completion, rapid return on investment, enhanced developer productivity, and faster time to market for new products. Using Blaze Advisor, First American creates a set of Standard Operating Procedures (SOP) for each banking client and contractor, improving timeliness and reducing costs. Enhanced accuracy provides another benefit: the right rules are applied up front. FAFS is able to give clients what they want, when they want it, without adding resources.

## Conclusion

The powerful advantages of a componentized, service-based application architecture have been shown to help organizations become more productive and nimble in their ability to build new applications. The use of a business rules management system within a service-oriented architecture provides clear benefits through the separation of decision logic from application functionality. Using a BRMS ensures that disparate applications behave consistently, are able to automate complex decisions and quickly adapt to changing business requirements. Business rules management systems also offer organizations the ability to incrementally transition from legacy systems to a service-oriented architecture. The last word goes to IDC:

“An independent business rules system offers the flexibility and capability to work identically within multiple implementations. Business rules should not be intricately bound to specific interaction methods or system functions. If they are as in traditional COBOL or other 3GL programs, any attempt to change the decision logic so the application can run on other types of systems or handle similar business processes requires a rewrite of the rules to meet the new requirements.”

“Companies requiring a cost-effective and proficient way to implement and manage organizational policies and procedures should consider using business rules management tools to develop applications and, particularly, Web services more efficiently, accurately, and quickly. More importantly, using a business rules approach for creating applications and/or Web services can produce significant return on investment (ROI) in terms of reduction in development costs, increased productivity, and decreased time to market.”



## About Fair Isaac

Fair Isaac (NYSE:FIC) makes decisions smarter. As the world leader in decision management solutions driven by advanced analytics, Fair Isaac unlocks value for people, businesses and industries. Companies worldwide use Fair Isaac technology to make billions of faster, more profitable decisions a year in credit management, marketing, fraud, collections, bill payment and other areas. The world's leading banks and credit card issuers rely on Fair Isaac technology, as do insurers, retailers, telecommunications providers, healthcare organizations and government agencies. Founded in 1956, Fair Isaac serves thousands of clients through offices in nine countries, and helps millions of individuals improve their credit health through the [www.myfico.com](http://www.myfico.com) website. Visit Fair Isaac online at [www.fairisaac.com](http://www.fairisaac.com).

**Fair Isaac Corporation** (NYSE:FIC) is the preeminent provider of creative analytics that unlock value for people, businesses and industries.

**Corporate Headquarters:**

901 Marquette Avenue, Suite 3200  
Minneapolis, MN 55402  
1 800 999 2955 *from the US*  
1 612 758 5200 *from anywhere*  
[info@fairisaac.com](mailto:info@fairisaac.com) *email*

**Offices Worldwide:**

Brazil, Canada, India, Japan,  
Malaysia, Singapore, Spain,  
United Kingdom, United States

[www.fairisaac.com](http://www.fairisaac.com)



*It's just a smarter way to do business.®*