The 2006 BPTrends Report on Business Rules Products

Version 1.0
April, 2006

Curtis Hall
Paul Harmon

With an Introduction by
Ronald G. Ross
Contents

Foreword by Celia Wolf ................................................................. i
Introduction by Ronald G. Ross...................................................... iii
The Market for Business Rules Products........................................ 1
A Detailed Analysis of Business Rule Products............................... 14
Corticon Technologies’ Corticon BPM........................................... 29
Fair Isaac’s Blaze Advisor............................................................. 47
ILOG’s ILOG JRules................................................................. 68
Resolution EBS’s Rules (iR) Platform............................................ 90
Rule Product Matrix................................................................. 106
Authors................................................................................... 114
Foreword by Celia Wolf

This is the third in a series of reports that BPTrends plans to produce and publish on business process software products. We are producing these reports to provide our members with an overview of the various types of business process software products available today. In each report we will define the specific market, describe the features important in tools designed for that market, provide detailed reviews of the leading players and their products, and provide a comparison of the key features of the products.

This report focuses on products that companies use to analyze, document or automate the use of business rules. Some of these products are independent and simply catalog business rules. Others automate rule-based decisions. Still others are used in conjunction with workflow or Business Process Management Suites to document rules used in processes and to automate rule-based decisions that occur when instances of processes are executed.

We do not consider either BPM Suites or Software Modeling Tools in this report. There are rule-based BPM Suites and we consider those in a separate report, The BPTrends BPM Suites Report. Most of the BPM Suites do not provide their own business rules capabilities, but rely, instead on one of the independent business rule products that we consider in this report. We indicate which of the rule tools covered in this report are used in which BPMS products. Similarly, some Business Process Modeling or Simulation Tools support business rules but we consider those in the BPTrends Enterprise, Business Modeling and Simulation Report.

Our objective in this report is to describe the various options available to business managers who want to capture and use business rules in their organizations. Different companies are using these tools for very different purposes. It does not make sense to argue that there is, or should be, a “best” tool with a specific configuration that would be best for all. Thus, we have avoided any comparison that suggests that some tools are the “winners” while others are “less desirable.” Instead, we analyze each product, explain what specific markets the vendor is pursuing, consider how the tool is adapted for that market, and identify what is special or unique about each product.

To make comparisons as easy as possible, we have explored the capabilities of modeling tools in several ways. These are discussed in the section entitled, “A Detailed Analysis of Business Rule Products” Some vendors only implement a subset of the features we describe. That does not necessarily detract from their offering. It could mean that the vendor does not think the feature is required for its target market. Or it could mean that the vendor thinks it is more efficient to partner with another vendor who provides that specific feature. As we suggested, it is impossible to say exactly what the ideal business rules tool should have or not have. We have sought to provide a simple, clean description of the current features available in each product so readers can decide if a given product will support their objectives.

Participating Vendors

BPTrends contacted all the Enterprise Business Rules vendors we could identify and solicited their participation in this report. We charged each vendor a fee to defray the cost of developing the report. All products from participating vendors were evaluated in the same manner: Curt Hall and Paul Harmon prepared a detailed questionnaire that each vendor was asked to complete. They reviewed the questionnaires, studied the product documentation and any other relevant materials provided by the vendors, and then requested a product demonstration. Finally, they interviewed each vendor to eliminate any confusion and to make certain they had not overlooked anything. They did not conduct any actual product testing.

We intend to maintain and expand this report in the future. Any vendors who would like to be included in future versions of this report should contact me directly.
Thanks to Our Coauthors and Members

I want to thank our friend and colleague, Curtis Hall, for his contribution as the principal analyst and author of this report. Curt’s knowledge and experience as an editor and analyst proved invaluable in researching and writing this Report. I also wish to thank my longtime friend and business partner, Paul Harmon, for bringing his vision, knowledge, and perspectives on the business process performance market to bear on this report. Finally, I want to thank all our BPTrends members and readers who continue to support us. We hope this report is informative and useful to all of you, and we look forward to hearing your comments and suggestions.

Celia Wolf
CEO and Publisher
Business Process Trends
cwolf@bptrends.com
Introduction by Ronald G. Ross

Looking for new ways to make dramatic improvements in your processes? Here’s a two-part approach: (1) Separate rules from processes. (2) Implement a business rules engine (BRE).

These (and related) ideas are clearly elaborated in the Business Rules Manifesto, a groundbreaking 2003 work product of the Business Rules Group (BRG). To give you some idea of how hot business rules are around the globe, in just over 2 years since its initial publication, the Manifesto (www.BusinessRulesGroup.org) has been translated into nearly a dozen languages, with more on the way. If you are reading this, you are clearly moving – like informed IT professionals in many parts of the world – toward an important new technological mainstream.

The first part of the approach – called Rule Independence – is simply to recognize that rules can and should be separated from processes. The Manifesto puts it this way: “Rules are not process and not procedure. They should not be contained in either of these. Rules apply across processes and procedures. There should be one cohesive body of rules, enforced consistently across all relevant areas of business activity.”

That’s really just common sense, both from a technical and a business point of view. Why? Three crucial benefits come immediately to mind: agility, consistency, and re-usability. The fact of the matter is that if your rules remain embedded in processes, you are unlikely to achieve these goals in full for either your rules or your processes.

The second part of the approach is to implement your rules using a BRE. If BREs are altogether new to you, here is an analogy that might help: A BRE is to rules as a DBMS is to data.

Well not quite. There’s one very big difference – “data” can’t be executed; rules can. Now if you can execute (or at least evaluate) rules directly under a BRE, why would you want to turn them into procedural code?! Even with the best process engine around, without a BRE, that’s the prospect you face. As the Manifesto says, “Executing rules directly – for example in a rules engine – is a better implementation strategy than transcribing the rules into some procedural form”.

This Report explains far better. So, rather than spending time on that, let me address several common fallacies.

One thing I’ve heard some people say is that BREs and a rules approach require you to know all the rules up-front – obviously, a task in real life that could be prohibitively large if not impossible in many or most cases. Nothing could be farther from the truth. BREs are great for incremental development. That’s something they’re designed for; compared with implementation using procedural code, there is far less up-front ‘lock in’. As the Manifesto says, “An effective system can be based on a small number of rules. Additional, more discriminating rules can be subsequently added, so that over time the system becomes smarter.” And that’s the whole point, right? Of course you want your processes to start off as ‘smart’ as possible, but you also want them to get smarter and smarter over time as resources, understanding and opportunity permit.

Another thing I’ve heard people say is that BREs eliminate or diminish the need for a solid process approach and a process engine. In a word, no! How could that be? Rules indicate (speaking very broadly) what behavior is or is not appropriate for given circumstances, where such behavior includes people’s actions, decisions and/or calculations. At the risk of oversimplifying, rules do not, however, tell you in prescriptive manner how those circumstances actually arise. That’s what work and workflow coordination are about – that is, the stuff of good process management.

I think of processes and rules as ying and yang. Processes orchestrate and exploit knowledge and know-how; rules capture and retain it. It’s a great fit; each simply makes the other all that much the better.
So I do hope you will take a long, hard look at this very informative Report. The vendors it covers are substantial players in the space. I am confident that you will come to agree (if not already) that BREs are a perfect complement to your process strategy.

On behalf of www.BRCommunity.com and its tens of thousands of readers around the globe, I applaud Curtis Hall and Paul Harmon for this important and timely Report, and for aggressively pursuing this high-impact technology on behalf of readers of BPTrends.

Ronald G. Ross
Executive Editor, www.BRCommunity.com
Co-Founder and Principal,

Business Rule Solutions, LLC
The Market for Business Rules Products

1. Business Rules

Companies have always had policies and rules to define what should or should not be done. Similarly, business rules have been written down in employee manuals for generations and are currently embedded in many legacy software systems. Today, however, Business Rules have achieved a new status as assets of a company that ought to be explicitly defined and managed.

A business rule is a statement that defines some policy or practice of the business. Business rules, whether implemented by employees or by automated systems, determine that appropriate actions are taken at appropriate times. Changes in company policies or practices invariably are reflected in business rules, and the ability to maintain consistency between policies and the business rules used in business processes, IT applications, and employee practices, especially when changes take place, has become a key characteristic of agile companies.

Today’s efforts to formalize the capture and management of business rules originated in four different movements that have waxed and waned over the course of the last two and a half decades. A review of those movements helps explain the current situation in the business rules market.

2. Business Rules for Software Development

In the late Eighties there were a series of meetings at IBM user group, GUIDE, at which technologists sought to define the business rules that software applications were written to implement. Programmers realized that different elements of their software applications changed at different rates. The data that a company collected, for example, changed relatively slowly. Business rules, which often incorporated specific business assumptions—information about specific interest rates or types of clients, for example – tended to change much more rapidly. Thus, many software architects began to believe that business rules should be formalized and stored independently of the software applications in which they are used. Properly organized, software applications would simply look up rules, as they were needed. This would mean that business managers could change the business rules as needed, without having to reprogram software applications.

Many of those who advocated the formalization of business rules believed that rule formalization should be a top-down effort. Executives ought to define strategies and goals and those should be translated into formal policies. Those policies, in turn, should be translated into high-level business rules, which should then be translated into more specific business rules.

Anyone who has undertaken a rules documentation effort knows that if one isn’t very careful, one soon runs into problems with the specific terms and names one uses in the rules. To create a formal system of rules, one must simultaneously create a formal vocabulary. In other words, everyone in the company must use words like “customer,” “account,” and “primary account number,” in the same way. Ideally, one needs a formal vocabulary to assure that a rule that states that:

All customers are assigned one and only one primary account number,

will be unambiguous and interpreted in the same way by everyone throughout the company. At a minimum, we need to define “communities” that will use the same words in the same way. Thus, business rule methodologists are usually concerned with the formalization of both business vocabularies – sometimes called an ontology – and business rules for companies or for communities within a company.
Most business software products use a repository to store information about rules. In effect, as one writes rules one is also creating and maintaining an object-attribute network that specifies the terms used and the relationships between terms.

Unfortunately, large companies are usually broken into many divisions and departments that are spread throughout the world. Getting management to spend the time required to formalize a corporate business ontology, and to then proceed and define formal business rules has proved very difficult. It’s a huge undertaking and most companies have been unable to justify the effort. Those that have – several insurance companies, for example – have been companies from industries that were already inclined to think in terms of very precise rules. Others have created rules and an associated ontology for only one division or one group within the company.

Figure 1 suggests how someone advocating a comprehensive rule formalization effort might conceive of the effort. In essence, they would start at the enterprise level and work with executives to formalize the company’s policies and create a formal ontology and appropriate business rules. Then they would work down through divisions and departments, formalizing their ontologies and business rules, constantly being sure that lower level ontologies and rules were clearly aligned with high-level rules. Finally, they would reach the implementation level and check to see where business rules appeared, in procedures manuals, training courses, and in software applications and assure that those implementations used rules clearly derived from high-level rules. In the end, if a company persevered, they would have a complete description of all the rules used in the organization. Subsequently, a change in policy would drive changes in high-level rules and those changes, in turn, would work their way down through the entire organization, assuring that all rules were changed to reflect the changes in policy.

Figure 1. business rules approach that advocates a systematic, top-down rules analysis effort.

The theory behind such a comprehensive rule-oriented approach is sound, but the problems involved in actually capturing and maintaining it are significant, and the effort has not been one that most companies have chosen to undertake. One problem that faced anyone considering such an effort in the Eighties was that most of the advocates of this approach were database technologists, and the databases being used at the time were not very well designed to support this approach. Thus,
although many people appreciated the power of the “rules approach,” it didn’t gain much traction until recently, when new tools became available.

3. Rule-Based Systems for the Capture of Expertise

Another approach to rules was undertaken by the Expert Systems movement of the mid-Eighties. Expert system development derived from research in Artificial Intelligence (AI) and focused on capturing the rules used by experts to analyze and solve very hard problems. For example, systems were developed to analyze readings from geological equipment and to determine constantly changing seat prices for airlines. Expert Systems development was facilitated by software tools – expert system-building tools – that stored the rules in a knowledge base and used an inference engine to examine facts and rules when a decision was required and to generate a decision.

While some of the expert system applications that resulted from these efforts proved very valuable, most proved too hard to maintain. It turns out that human experts are constantly learning new things and then synthesizing the new knowledge with their previously acquired knowledge. Thus, although expert knowledge, in the form of rules, could be acquired and used in decision making, the resulting systems proved difficult to keep up to date. In most cases it was easier to pay an expert to constantly work at maintaining his or her expertise than it was to try to capture the expert’s continually evolving knowledge.

In the mid-Nineties, as interest in the capture of expert knowledge waned, many of the vendors who had provided expert system-building products repositioned themselves to provide support for those who were interested in capturing and using business rules. Expert rule sets had proved too unstable and hence too difficult to maintain, but business rules tended to be more stable and to change less frequently. The rule tools originally developed to support expert rule sets turned out to be much better for maintaining business rule sets and supporting the types of rule changes that business managers wanted to make. Thus, in the late Nineties, the IT rules documentation movement and the expert system-building tools vendors had largely joined forces.

4. Risk Management and Compliance Issues

Corporate executives have always been concerned with whether employees are, in fact, following corporate policies. Many industries are regulated and there are laws that require that certain types of companies report on compliance. Recently, Sarbanes-Oxley and related regulations have been promulgated that require that companies demonstrate that they are able to track changes in processes that might lead to a compliance failure. The various concerns have placed a new emphasis on both formal business rule systems that can track compliance from policies to high level rules to specific rules in software programs and employee manuals. At the same time, these same regulations have encouraged companies to develop formal descriptions of key business processes and to show where business rules within those processes assure compliance to government regulations. These legal and management concerns have highlighted the importance of a well managed business process effort that documents not only processes but business rules.

5. Business Rules Used in Business Processes

In the Nineties, considerable attention was focused on reengineering major business processes. To understand a business process, analysts usually began by creating a diagram or model that showed the major steps or activities that occur during the process. At the simplest level, business rules were often pictured as decision points within process workflow diagram. Thus a rule that said that loans should only be granted to applications that meet the company credit standards might get represented in a flowchart as it is in Figure 2.
Figure 2. A workflow diagram with a business rule used to define a branch (decision diamond) in the flow of work.

More complex decisions might also be formalized by means of business or even expert rules. For example, in Figure 2, the process analyst might decide to get very explicit about how one determines the terms and interest for a specific loan application. It could easily turn out that one hundred different rules were involved in determining the terms and interest for a specific type of loan. In this case the rules are not shown, explicitly, as a decision box, but are, in effect, inside the Determine Terms & Interest for Loan activity box. (In many process modeling software tools one can literally click on the Determine Terms & Interest Activity box on a diagram and open a window to the business rules documentation environment.) Clearly the rule represented by the decision diamond was a business rule. Probably the rules used to determine the terms and interest for the load were also business rules, although some decisions in some processes can become so complex that they are, in fact, knowledge rules. In other words the rules are not so much defined by explicit policies as by experts who are hired to make the decisions. As process analysts examined ever more complex processes they found that the capture of business rules was an important part of most business process redesign efforts.

Figure 3 suggests some of the relationships we have been discussing.

Figure 3. Business Rules are derived from common business knowledge, often formalized as policies, while Expert Rules are derived from Human Experts and not formalized. Both are found in business processes analysis efforts.

Just as business rule advocates advocated a top-down approach, most business process analysts have urged companies to begin at the enterprise level and define high-level processes – usually called value chains – and then subdivide those to define a hierarchy of business processes. In a similar way, they have advocated that companies align their strategic goals with their value chains and major processes and develop measurement and management systems to support all their processes. Figure 4 illustrates a pyramid that a corporate process advocate might use to define the development of a comprehensive business process architecture.
In the late Nineties, business process improvement was often aided by two different but complementary software technologies. Workflow software was used to control actual processes. Initially used to manage the movement of documents from one desktop machine to another, workflow systems gradually became more sophisticated and could manage any processes in which people used computers to do work. Most large companies have developed workflow systems to automate all or portions of processes that deal with documents. Workflow systems use business rules to control the flow of work, just as our decision rule controlled the flow of the process shown in Figure 2.

During this same period, packaged software applications became increasingly popular. These packaged applications, usually termed ERP (Enterprise Resource Planning) or CRM (Customer Relationship Management) applications, could be used to automate selected business processes. More sophisticated ERP or CRM systems are managed by workflow software, so that business managers can change business rules and alter how the programs perform. Today, most large companies use suites of ERP applications to automate at least some of their major business processes.


In the past five years, business process change has been given a new impetus. Driven in part by the economic difficulties of the early years of the decade, and the need to become more productive and to reduce costs, and in part by new developments in software technology, companies, today, have a renewed interest in business process analysis and design work. Today’s interest, however, is much more comprehensive in nature and scope than the reengineering efforts of the mid-Nineties.

In the last few years, the separate threads we have been discussing have begun to come together in a new synthesis. A new generation of workflow software, usually termed Business Process Management Systems (BPMS) software, combines traditional workflow with enterprise software application integration (EAI) techniques, the latest Internet protocols and distributed software component techniques, usually termed (SOA). In essence, BPMS developers have created a new generation of tools that can be used to capture and manage the execution of business processes.
Many companies are interested in using BPMS techniques to make ERP and CRM applications more effective.

In addition to combining process modeling and workflow execution techniques, most BPMS products also incorporate business rules engines to automate those aspects of business processes that involve complex decision-making. In most cases the business rules engines are not a part of the core BPMS product, but are provided as an additional capability that can be used when needed. Indeed, several rules products are used by multiple BPMS vendors and some BPMS vendors support more than one business rules product. Figure 5 suggests today that Process and Rule-Based approaches are increasingly complementary.

Figure 5. Rule and Process analysis efforts are increasingly being conceptualized as complementary activities.

Many BPMS products also incorporate monitoring functions that provide managers and senior executives with dashboards that show how processes are performing. The more sophisticated of these BPMS tools rely on BI and Data Warehouse systems to organize and filter process level information for executive consumption.

Figure 6 provides an overview of the various types of software tool vendors that BPTrends tracks. There are a wide variety of rule products being sold and they can be used for various different functions. Thus, the circle representing business rules overlaps many of the other circles.
Some companies continue to develop rule tools designed to help companies catalog and manage business rules while other rule software vendors continue to offer products designed to simply automate specific rule-based decisions. Most business rules vendors sell tools that can be used with business process modeling and simulation, with BPMS, with monitoring products, and with a variety of ERP and CRM systems. Increasingly, rule and process advocates are working together to develop a common approach. Process analysis provides a context for the analysis of business rules, and process redesign efforts that seek to automate decisions often find themselves relying on business rule engines.

No better example of this trend can be found that the current standards work being done at the Object Management Group (OMG). The OMG is formalizing standards that will make it easy for companies to move business and software models from one software environment to another. This is important to companies because they often use more than one software tool, or change software tools when better tools appear, and don't want to get stuck with models that can be migrated to their latest products. The group within the OMG that is working on business models is currently formalizing a business rules standard (SBVR) that was initially developed by the Business Rules Group, (BRG) a descendent of the committee originally formed in the Eighties at GUIDE to formalize business rule systems. The same committee is also working on a standard for Business Process Notation (BPMN) that was originally developed by the Business Process Management Initiative (BPMI), a BPMS group that has recently merged with the OMG. And, finally, the same committee is working on a high-level standard (The Common Motivation Model or CMM) that is designed, among other things, to define the formal relationships between business rules and business processes.

7. **Business Rules Products**

Business rules software products can be broadly divided into three groups.

**Rule Documentation and Management Products**

Some of the vendors offer products designed to document existing business rules and provide traces to show the relationship between specific policies and particular rules. These products tend to be
database or repository products, and the more sophisticated not only keep track of rules but also
document a company’s business vocabulary.

Rule Engines.  
These products provide for the capture of rules and the runtime analysis of rules to either support or
make a business decision. Some of these products are stand-alone products, but most are designed
to work in conjunction with other software, providing a “decision module” that application
developers can embed in a software application. Rules Engines are sometimes called Inference
Engines, and in that case, the products are referred to as Inference-Based Rule Products.

BPMS Products
Business Process Management Systems are software products designed to capture and automate the
runtime execution of business processes. In essence, these products combine traditional workflow
(which manage people processing work) and Enterprise Application Integration (EAI) which
manages the runtime execution of legacy software programs. Not all BPMS products include rules
ingines, but the more sophisticated BPMS products incorporate one or more business rules engines
to automate the flow of process execution and to provide decision automation (a rules engine) when
needed.

These three general types of rule products overlap and some vendors sell their technology in more
than one package. Thus, for example, its common for the rules engine vendors to sell their products
as either a stand alone decision automation tool, or as a component of a workflow or BPMS
product.

Figure 7. The three basic categories of business rules software products

Vendors offering engine-based BRM products primarily define the current market for BRM
products. A new generation of BRM products that are easier to use, and which are designed to
automate repeatable, work-oriented decisions characterize this market. Gartner estimates that the
market for engine-based BRM products is growing at about 10 percent.

The market for BRM products can be considered several ways. One way is by those vendors that
offer stand-alone BRM products and those that market software applications that include embedded
Business Rules capabilities as a subset of some other functionality (i.e., “under the hood”), such as
BPM Suites, CRM, ERP, SCM and other enterprise applications.
Another way to segment the BRM market is according to product functionality. In one camp are vendors whose products are based on a natural language style approach to rules development, whereby developers literally write IF-THEN style rules. These products support less-technical end users by featuring a separate authoring or rule maintenance environment designed just for business users (i.e., that provide a constrained or limited subset of functionality). The other camp consists of vendors offering BRM products that provide a single authoring and modeling environment—typically spreadsheet or grid-based—intended to support both IT developers and business users. A general trend among vendors from both camps is to make it easier for non-technical business users to work with their products.

BRM vendors also take different marketing strategies. Some focus primarily on marketing their products to end-user companies wanting to build and deploy BRM applications within their organizations. While other vendors’ pursue a strategy based on licensing their products to third-party vendors wanting to embed rule-based functionality in their own business software applications.

8. BRM Market Trends

Focus on Decision Automation
One of the most important market trends that has taken place over the past 4 years has been the shift among BRM product vendors in positioning their products for decision automation as opposed to decision support. In the Eighties and Nineties, vendors offering rule-based expert systems tools primarily focused on positioning and marketing their products as development environments for building expert decision support applications that captured some segment of a human expert’s knowledge, and which applied that knowledge to support decision making for a particular task. (For example: the analysis of geological or seismic data to assist in determining where to drill for oil, or expert systems used to assist human operators troubleshoot production line manufacturing issues, etc.). Today, BRM vendors are strongly focused on marketing their products as environments for building rule-based systems that automate decisions in which a high degree of reliability is required. Companies still use BRM products to build expert decision support applications. However, the major use of BRM products today has shifted toward applying the technology to the automation of deterministic work-oriented or repeated decisions—frequently high volume decision making such as on-line fraud detection, credit approval, and other customer management decisions—as opposed to decision support tasks requiring the application of deep expert knowledge.

No Specific Industry Focus
When it comes to marketing, most BRM product vendors do not exclusively focus on one or a few primary industries. Rather, a general trend is that vendors tend to target their BRM products for automating a wide range of decisions that cross industry boundaries.

Templates
The general trend among Business Rules vendors is to offer reusable application templates (as well as the ability to create and catalog templates for reuse) designed to support key aspects of a rules project—thus helping to “jump start” application development.

Packaged Business Rules Logic
Some vendors also offer domain-specific knowledge—packaged in the form of specific processes or rules. This packaged knowledge is typically available as an add-on product offered in conjunction with application consulting and implementation services.

Horizontal and Vertical Industry Templates, Frameworks, and Pre-Built Rule Sets
Some vendors have taken the template approach a step further and are offering horizontal and vertical industry templates and frameworks and pre-built rule sets and vocabularies designed to ease rule-based application development for specific domains and industries (popular areas include
compliance, finance, and insurance). Basically, these templates and frameworks serve as a starting point (instead of beginning from scratch) for organizations developing industry-specific applications.

Packaged Rule-Based Applications
Some vendors offer pre-built or “packaged” rule-based industry-specific applications. Such packaged applications are typically marketed in conjunction with a heavy dose of consulting to assist end-user organizations with implementation, integration and maintenance support.

BPM Suites and Embedded Business Rule-Driven BPM Platforms
A number of the leading BPM Suite vendors have licensed business rules engines (BREs) from the major BRM vendors in order to incorporate advanced workflow and rules-based decision automation capabilities within their BPM platforms, and to support dynamic component configuration and integration.

Conversely, some BRM vendors have repositioned themselves in an effort to capitalize on the growing corporate interest in process management. These vendors tend to bill their products as “rules-driven BPM” platforms, and they strive to differentiate their products from the BPM products that feature “bolted on” rules capabilities.

9. Drivers for Business Rule Product Sales
To summarize what we have discussed, the current drivers of business rules technology adoption include: (1) the ongoing quest for better IT agility; (2) the need for systems that can automate decision making that involves business rules; (3) the need to trace the use of policies and business rules to comply with regulations and reduce risks, and (4) the growing corporate interest in business process management (BPM).

The expressive nature of BRM systems—and their ability to separate business logic from other (i.e., procedural) application and database logic—has made them a popular choice for automating compliance, personalization, marketing, credit/loan approval, fraud prevention and other types of decision-making. In addition, BRM technology is increasingly being used to add advanced workflow and rules processing functionality and dynamic process/application integration capabilities to BPM and other enterprise platforms.

Today many organizations’ business rules are typically defined in manuals or other documents. For example, you might find the statement:

All customer returns require a receipt.

Other policies and business rules are not formally define at all. As a result, many organizations rely on managers and other employees to interpret company policies as business rules. This tends to be inefficient and leads to inconsistencies in the application of the rules. Still other business rules exist or are embedded within the procedural code of different software applications. These embedded rules are basically hidden, making them difficult to identify and interpret as well as reuse or apply in other areas of the organization. In particular, embedded business rules are difficult for analysts and other business people to understand and work with.

Many organizations today are taking an alternative approach in which they seek to identify and capture business rules and then implement them in software systems designed specifically to support business rules management. Their goal is to externalize their business rules so that they can be more easily understood by both business people and IT systems developers. Organizations can then apply their business rules more efficiently and consistently across various decision points as required.

BRM technology offers a number of benefits for end-user organizations, including:
♦ Distribution and reuse of rule-based decision making across decision points
♦ Greater consistency in applying business policies
♦ Flexible application development
♦ Greater involvement of business users in application design and management

Distributed Decision Making/Greater Consistency
Rule-based business logic allows organizations to distribute their business policies across multiple applications, channels, and organizational boundaries. Distributed decision making also supports reuse (of business logic) and imparts consistency in the application of policies and practices throughout the organization. These capabilities are increasingly becoming more important as organizations today strive to offer—yet differentiate—the same services and access to information over multiple channels of communication to their customers, employees, and supply-chain and other partners.

Flexible Application Development
BRM technology can help organizations develop and update business application software more quickly by enabling application developers to externalize business logic from procedural code. This allows application logic to not only be maintained without bringing the production systems down, but it can be viewed and understood more easily.

The argument can also be made that BRM technology can help reduce the total cost of ownership for managing business application software by reducing programming requirements through the use of rule-based modeling techniques and by limiting delays in implementing operational rules.

Greater Control to the Business/Faster Response to Change
BRM technology allows business experts to take a greater role in systems design and maintenance through the utilization of a highly descriptive rules paradigm that makes it easier for them to understand and manage critical business logic. This capability allows non-technical business managers and analysts to assist with the maintenance of application logic as rules in production systems without being as dependent on technical IT resources. This gives them the ability to make routine changes and updates to critical business systems, thus helping to free up IT resources to concentrate on more demanding projects and initiatives, and speeding faster response to changing business conditions.

11. Applications of Business Rules Technology

Business rules technology is now applied to a range of applications, including for decision automation, personalization, and compliance. In addition, BRM technology is increasingly being used to add advanced workflow and rules processing functionality and dynamic process/application integration capabilities to BPM platforms.

Decision Automation
BRM technology is particularly well suited for automating deterministic work-oriented or repeated decisions. As a result, BRM technology is now used for a range of decision automation tasks.

One of the most visible uses of BRM technology is for automating customer management decisions—such as those associated with product selection or recommendation or requests for services. BRM technology is particularly useful for differentiated services offerings, as mentioned previously.

Marketing decisions are another popular area for BRM deployment—for example, to determine the optimum offer to be made to a customer based on their profile and new criteria.
Loan approval and underwriting were two of the earliest applications to utilize rules-based systems. Today, BRM systems are being used to automate a range of credit decisions—for example, should this customer’s credit line be extended based on past history and newly acquired information?

BRM applications are now used to automate decisions associated with supply-chain activities and processes. For example, they can be used to make supplier decisions, such as to determine which suppliers to use based on current price quotes and past performance.

Fraud prevention is another popular application area for BRM technology. A good example is to determine the likelihood that a particular customer transaction is fraudulent based on existing account history and customer account usage profiles.

**Personalized Communications**

BRM technology allows organizations to personalize their business and communication processes with rule-driven data and content displays, policies, interactive dialogs, and Web sites according to general user categories, user history or user preferences.

BRM technology also supports user-centric computing, enabling companies to offer tailored service to individual users. For example, a smart (rules-enabled) Web browser can enable users to configure their own rules governing how they want a diverse set of services on the Internet to function on their behalf.

**Compliance**

Compliance applications have become a major driver for companies to use BRM technology. By externalizing rules from critical processes and applications, organizations can create a centralized repository of compliance rules that they can apply across multiple departments, divisions, and channels, thus helping to standardize and coordinate company-wide strategies for meeting organizational and governmental regulations and laws (e.g., BASEL II, Sarbanes-Oxley, or Do-Not-Call, etc.)

**Business Process Management**

Although practitioners from both camps still tend to view them as separate—yet complimentary—in truth, BRM and BPM technologies are becoming more and more intertwined. This is due to the fact that BRM technology aligns very well with process optimization—in particular, for automating repeatable decisions as part of some process or business activity.

As a result of this synergy, organizations are turning to BRM technology in an effort to streamline business processes that are found to be too rigid or inefficient. For example, a business process analysis team, by analyzing a company’s travel system, might find that managers are taking too long to approve or reject employee travel requests. The company might replace the human approval task component outright with a BRM application. Or, rules may be used to supplement the human steps in the workflow. For example, a BRM system could be used to generate rules-based input forms that provide interactive guidance designed to assist employees with filling out a travel request form completely and correctly (i.e., making sure all data required is provided), thereby ensuring it can be processed by the downstream workflow without incurring unnecessary delay.

Many of the leading BPM Suite vendors have added rule-based capabilities to their platforms in order to incorporate advanced workflow and rules processing capabilities. An example might be to apply rules to consider a large number of facts to determine the next appropriate step in a workflow.

Finally, BRM technology is increasingly being used to add rule-based capabilities to BPM platforms for dynamically configuring/integrating application component functionality. In such a scenario, rules are utilized for a number of purposes, for instance, to automate and specify channel interaction and routing, to streamline interfaces between operational systems, and for automating the mapping and parsing of data between disparate operational systems.
12. The Growing Market for Business Rules Products

Efforts at formalizing business rules began in earnest in the Eighties, and, following some ups and downs, have begun to grow rapidly. The idea that business rules should be formalized and managed to assure that organizations can trace policies to the specific business rules that implement those policies has become popular and widespread. Similarly, the idea that software systems ought to manage rules independent of business process systems or software programs has led to the development of a variety of business rule tools. These ideas will continue to drive new developments in rules and stimulate the growth in the adoption of rule-based software products in organizations in the years ahead.
A Detailed Analysis of Business Rules Products

1. Product Overview

The first section considers the organization and features of the business rules management (BRM) products included in this report in some detail. The tools at the heart of this report consist of environments designed for creating (i.e., authoring) and managing (repository, database and other support, etc.) business rules as well as an engine (for run-time evaluation)—typically referred to as a business rules engine (BRE)—for executing the rules.

Companies use these tools to authorize, organize, and deploy rule-based applications designed to automate specific processes and activities (as discussed further in the Introduction). Consequently, they also include application development facilities—including code generators and APIs designed to support specific languages (e.g., Java, C++, COBOL, etc.)—and for integrating rule-based applications into existing systems and enterprise architectures (e.g., J2EE, CORBA, .Net, etc.).

Many BRM products also offer additional capabilities in the form of various add-on packages and supplemental components designed to assist with developing and deploying rule-based applications. These include tools for automating the creation of business rules as well as domain-and-application-specific products such as horizontal/vertical templates and industry-specific rule sets and libraries. We will describe these add-on offerings as well.

Rather than speak of business rules management products, we will simply refer to the products as BRMs, BREs, or rule-based application development environments where applicable.

We will follow the same outline used in this chapter as we consider each specific product. Similarly, we will follow this same sequence in the Matrix (found at the end of this report) when we summarize the features of each product.

We begin each product review by providing an overview that describes the overall organization and packaging of the tool.

2. Product Architecture

2.1. Architecture Overview

The second section of each product review provides an overview of the general architecture of the tool as well as key modules that support the authoring, execution, deployment, maintenance, and administration of business rules applications. Architecture options for BRM products can vary; however, to some degree they all feature the following standard modules or components. The main differences are found in how the vendors implement them as well as the terminology they use to describe them.

Rule Authoring Environment/Graphical IDE

All BRM products provide some form of graphical IDE for developers to author rules and sets of rules (i.e., rule sets) as well as to create an architecture and structure for business rules applications. Rules sets and the accompanying framework required for their integration with production environments are generated in the form of a software component—typically referred to as either a “decision service” or “decision package.” The authoring environment can be a stand-alone or server-based application. Rules (decision services) can be deployed on an (application) server or embedded as code in various applications (ranging from Web browsers and application servers to third-party enterprise software packages (e.g., CRM, ERP, e-commerce servers, etc.).

BRM products support less-technical users in different ways. Some provide a separate environment designed specifically for business analysts and other non-technical end users to manage and update
rules. Some provide the ability for developers to generate “rule maintenance applications”—highly structured front-ends designed specifically to simplify the management and updating of rules by end users by tightly constraining the features and functionality at their disposal.

Still other BRM products feature a common interface—typically spreadsheet or “grid-based”—designed to support both developers and business users. (Section 3 provides a more detailed discussion of how BRM environments support different types of users.)

**Rules Server/Business Rules Engine (BRE)**
BRM products provide a Business Rules Engine (BRE) for executing rule-based applications built with the tool’s graphical IDE. They can be deployed on application servers using various architecture configurations including as a J2EE Enterprise Java Bean (EJB), Model Driven Bean (MDB) or Web Service. Other configuration options include deployment as a COM/MTS object, a .NET component, or as a functional component that operates within EAI layers and enterprise messaging and middleware systems (e.g., IBM MQSeries, Java 2 Enterprise Edition (J2EE), Java Messaging Service (JMS), RMI, Enterprise Java Beans (EJB) and Web Services APIs (WSDL, UDDI, SOAP, etc.).

The trend is for BRM applications to integrate with enterprise applications via Web Services or service oriented architectures (SOA). In such a scenario, the operational system “calls out” to the rule-based system when necessary. However, some applications—due to process integration requirements, transaction-processing performance considerations, or security reasons—require the “heavy lifting” integration capabilities provided by enterprise messaging and middleware systems.

**Rules Repository/Rules Database**
BRM products utilize a repository or database to store and manage rules and rule sets created using the tool’s authoring environment. The repository is kept separate from the BRE execution engine in order to facilitate easier implementation and maintenance of rules and rule sets. It also supports—to varying degrees—multi-user and team development of rules projects.

Typically the repository either includes its own version control system—or requires the use of a third-party version control system (e.g., CVS, PVCS, SourceSafe, ClearCase, etc.) in order to support multi-user team development (e.g., file access and versioning, etc.).

### 2.2. Business Rules Engine
This section examines the type of engine the BRM product uses to process and execute rules. BRM products can use several different methods/engines for rule processing and management, including a database/transactional engine or an inferencing engine.

Database/transactional-based BRM products rely primarily on a relational database engine, and are designed to utilize triggers and other database transaction facilities to conduct rules processing and maintenance.

Inferencing-based BRM products rely primarily on an inference engine for rules processing (although they also utilize a database for storing the rules created with their authoring environments). The current crop of inference engine-based BRM products can utilize several different types of inferencing techniques, including forward-chaining and backward-chaining to carry out rules processing. A forward-chaining algorithm works from the initial facts to infer the conclusion (i.e., concluding fact). A backward-chaining algorithm works backward from a conclusion to be proved to determine if there are known facts that can be used to prove the truth of a conclusion. A common forward-chaining algorithm used by BRM products is the RETE algorithm, which hails from the early expert system days. Other BRM products use various proprietary algorithms to support forward and backward chaining. In addition, BRM products can support one or both types of inferencing.
2.2.1. Rules Management Approach

This section examines the rules management approach used by the BRM product. Some BRM products use a combined repository and object-oriented approach, which allows various options for specifying and managing rules execution, including ruleflow sequencing for ordering groups of rules, and allowing the developer to choose engine-determined processing or sequential processing of rules in a predefined order. Other methods for controlling order of execution include setting relative priorities, event-based conditions, and explicit calls to rules from other rules.

Other BRM products use a data-based approach combined with the application of advanced mathematical techniques (at design time) to create optimized and compiled rules sets (in the form of decision services) with guaranteed logical integrity.

2.2.2. Enterprise Data Interoperability

BRM products feature the ability to build and execute rules that reference data stored in databases and other information systems. The degree to which they do so and how they do it varies. Most provide database access. Other capabilities range from interaction with XML documents, EAI and BPM systems (e.g., IBM MQSeries), Java/J2EE objects, CORBA objects, COM objects, and customer-defined objects typically associated with third-party data models. The general trend is for products to rely heavily on the use of XML to provide enterprise data interoperability.

2.2.3. Maintaining Data Quality

This section examines a product’s ability to maintain data quality at execution time (i.e., as it carries out rule processing) as well as the techniques used to evaluate the consistency of data or database transactions produced by applications.

BRM applications are able to request data from external data sources as well as end-users when applying rules to reach a decision. Thus, it is possible that a BRM application could encounter “dirty” or missing data, which could impact the decision-making process. Consequently, BREs utilize various techniques to handle this problem. Some engines use exception handling. When a missing data value cannot be resolved, an appropriate exception is generated based on the missing information and the decision process halted. This helps to ensure that the application handles the missing data, communicates the problem effectively to the client application and/or end user, and exits with causing further execution problems.

Some BRM products seek to eliminate the need for exception handling by utilizing advanced algorithms and other techniques in order to assure data quality at execution time.

3. Interfaces and Ease of Use

This section describes a BRM product’s user interface (UI) paradigm in more detail and examines the types or classes of users the product is designed to support and how it does so.

BRM products are designed to support three primary classes of users:

♦ IT developers (i.e., programmers)
♦ Business analysts, and
♦ Non-technical business end users

BRM application development typically consists of business analysts (serving as liaison between the business and IT departments) working in conjunction with IT developers to identify, author and model the necessary business rules and rule sets to fit the scope of a rules project or application.
IT is then charged with implementing the BRM application in software, conducting testing, and then integrating the application into the organization’s existing operational systems architecture.

Once deployed, it is then possible for (non-technical) business users to carry out limited rules changes in order to update and manage the BRM application—for example, to change income, age or other values and criteria associated with a credit approval or marketing application.

It should be noted, however, that it is rare for organizations to allow (non-technical) business end users to effect changes to a production BRM application’s rule set without first conducting rigorous testing in order to avoid deploying conflicting rules which might possibly lead to unsuitable decision processing results.

The general trend among vendors is to enable their products to address both the requirements of the IT department as well as business users, which can range from analysts to non-technical business end users. The business analyst/end-user requires the ease of use and flexibility of a personal productivity tool. IT departments, on the other hand, are looking for a development environment that is both intuitive and which can be used to create applications they can integrate seamlessly within the organization’s existing infrastructure with a minimum of hassle.

Different BRM products support different classes of users in different ways. Some provide what are essentially separate environments for rule authoring and maintenance. One is a fully functional interactive development environment (IDE) intended for IT developers to create the necessary architecture and structure for a business rules application as well as a tool for authoring business rules. The other is a so-called “rule maintenance application”, which provides a constrained user interface designed specifically to allow non-technical people to safely and easily create and change business rules without technical assistance.

Some BRM products take a different approach. They feature a common spreadsheet-style or “Grid” interface designed to provide both IT developers and business users with a familiar, business-like interface for building and updating rule-based applications. This friendly interface helps alleviate the need to create a separate maintenance or interface layer to provide rules editing and maintenance functionality for non-technical end-users.

Both of these designs have their benefits and drawbacks. BRM products that provide a single rules authoring and modeling environment designed to support both IT developers and non-technical business users can avoid or limit a key issue sometimes encountered with BRM products that provide separate authoring and rule maintenance environments for IT developers and business users. That is, the need to somehow synchronize rule sets, languages, and interfaces between the different developer and business-user authoring and modeling environments.

However, it should be noted that some IT developers do not take well to the spreadsheet or Grid-style development paradigm. Rather, as programmers, they prefer a more familiar programming style environment in which they literally write rules. Thus, at some point, choosing a particular BRM environment comes down to personal preferences. Finally, it is important to note that some of the vendors whose BRM products feature separate environments for rule authoring and maintenance have added facilities to their offerings that provide automated synchronization of rule sets, languages and interfaces across the different tool sets, thus helping to avoid or limit synchronization and management issues.

### 3.1. Development Environment

This section examines the product’s development/rule-authoring environment intended for application developers. It includes a sample screenshot of this environment. It also considers any special features that make the tool intuitive or user friendly or which help to differentiate the product from others on the market.
Some BRM products support IT developers with additional application development and deployment tools. These include facilities for creating tailored front ends and templates which are designed to constrain how a business user can write a rule as well as the ability to generate a Web interface that can then be edited to match end-user needs and expectations. Such tools allow developers to create very structured front-ends for simplifying management and updating of business rules (by both IT and by less-technical end users).

Such facilities are important because they give business users and analysts the ability to make routine changes and updates to policies and other business criteria while freeing IT resources to concentrate on more demanding projects.

3.2. End-User Environment
This section examines the product’s rules management application or end-user environment intended for non-technical business users in more detail. It also includes a sample screenshot of this end-user environment.

Not all products, however, provide a separate environment for non-technical end users (as noted in Sections 2.1 and 3).

4. Business Rules Expression
4.1. Defining Rules and Rule Sets
Here we describe the primary type of rule representation scheme the product uses. Different products use different techniques. Some BRM tools are based on an underlying code—employing a natural language paradigm—in which developers write rules using English-language like syntax (i.e., IF – THEN). Some also use the familiar “dot notation” associated with Java and other programming languages (e.g., customer.name, cust.lifetime_value, etc.).

Other BRM products forego exposing the user to any sort of underlying code language for rules development. Instead, they provide an intuitive spreadsheet or “grid” interface designed to mimic the familiar look and feel of today’s popular office productivity suites.

4.2. Creating and Maintaining Rules
Some tools provide additional methods for users to create and manage business rules. These include decision tables, decision trees, scorecards (i.e., scoring models) and other graphical and tabular methods for creating rules.

Decision Tables
Decision tables are a form of structured lookup capability that provides the ability to construct and update conditions and actions for rules as a cross-indexed chart or table. Typical examples of applications and uses for decision tables include pricing charts, actuarial tables, and shipping fee schedules. Decision tables are an important feature because they allow users to match design and maintenance to the way that they deal with structured lookups in their business.

Decision Trees
Decision trees provide the ability to construct and update conditions and actions for rules as a branching tree. Typical examples of applications and uses for decision trees include customer segmentation and outcome selection from a finite set.

Scorecards/Scoring Models
Scorecards and scoring models provide an intuitive way to combine many factors into an overall measurement that can be used to drive decision-making rules. Basically, they allow users to assign proprietary categories, ranges, and associated scores to individual criteria, thereby building up additive scores. Scorecards are useful for a number of applications—for example, to predict rankings, such as a customer’s relative worth, a targeted customer prospect’s likelihood to accept a
promotional offer, or a mechanical component’s probability of failing within a given configuration or time frame.

### 4.3. Rule Templates

This section covers rule templates and other design patterns offered with the product that are intended to assist developers with creating rules and rule sets. It also examines capabilities for creating reusable templates and patterns, which can then be re-used to support other projects.

The ability to create templates is an important feature, because they can help enforce the consistent deployment of rules across different business scenarios, applications, projects and business units. They also provide the basis for creating rule maintenance applications, which only allow end users to modify or create rules within a strict set of constraints appropriate to satisfying different user requirements, application functionality, and security concerns.

It’s important to note that the ability to create templates is not a necessity with some BRM products due to their use of a rules representation scheme that utilizes a common language or interface for both developers and business users. Thus, there is no need to create a separate maintenance language or interface layer to provide editing functions for non-technical end-users.

### 4.4. Rule Syntax Checking

Most BRM products provide some form of automated rule syntax checking through a pre-compilation step or combined analysis and report generation designed to identify problematic code. Here we describe the rule syntax checking supported by a product.

### 4.5. Rule Testing and Tracing

This section discusses rule testing and tracing capabilities provided with a product. Rule testing and tracing is important, because it allows developers to check for potential conflicts and unwanted interdependencies between rules. Different products offer various tools and capabilities for rules testing and tracing. These range from standard rule editing windows to execution browsers and model explorers that let developers visually model rule bases as they execute.

Additionally, some tools feature reporting facilities for generating reports and documentation that describe the structure of a rule set and its individual elements (i.e., rules, ruleflows, decision tables, etc.).

### 4.6. Multi-User/Team Development

Here we examine a BRM product’s features for supporting multi-user team development. The ability to efficiently track and manage versions of projects and rule files becomes crucial when working with teams of developers and rule writers. This is especially true for large projects as well as for re-using and sharing rules or rules sets for other applications or across different organizational units.

Different BRM products support team development to varying degrees. Some offer advanced repository capabilities that include the ability to define and manage shared repositories—complete with check-in/check-out capabilities—thus allowing multiple users to check out and check in rule service components while preventing them from making changes at the same time or overwriting revisions. Some even allow users to work off-line and then synchronize changes with the repository once reconnected.

Some tools offer a built-in versioning system to manages rule changes, including logging historical content changes of rules, rule services, and rule sets. And some BRM product’s repositories are tightly integrated with their BRE, in effect, allowing the rule server engine to monitor changes in the repository in order to (automatically) synchronize or update affected rule services (i.e., decisioning services) without interrupting production operations.
Other BRM products provide more limited support, with multi-user development requiring the use of additional third-party source control systems such as Microsoft Visual Source Safe, CVS, PVCS and ClearCase, etc.

4.7. Rules Management and Maintenance

This section examines features offered by a BRM product’s rules maintenance and management environment, including capabilities provided for developers and non-technical end users.

The general trend is for BRM products to eliminate or reduce the requirement that both developers and end users know programming techniques in order to manage and maintain rules. In addition to providing intuitive GUI-based authoring and management environments, some products also provide additional facilities designed to assist with creating rules, rule sets, templates, as well as other rule development and maintenance aids.

4.8. Rule Logic Validation and Testing

This section examines tools and other facilities provided with the product to assist users in assuring that rules and rule sets correctly achieve the intent of the business. The aim is to avoid the possibility that a rule change could cause unexpected or incorrect results. BRM products provide various tools and methods for testing business logic, including:

♦ Browsers and model explorers—for visually inspecting and verifying business logic.

♦ Performance/Batch Testing—to identify potential problems in the execution of the rule-base with respect to performance; batch mode and interactivity testing allows you to simulate test cases in order to ensure that the application is behaving as expected based on any rule definitions or changes.

♦ Rule Execution Browsers—allow you to follow the execution of a rule graphically by displaying a tree of possible execution paths as well as by highlighting the actual execution path taken by the rule.

♦ Simulation—for simulating the runtime behavior of rules, allowing the tester to view and diagnose interactions between rules to verify expected results without impacting a production system.

♦ Reporting—some products generate reports—in document or linked HTML—to help developers identify potential conflicts with a project

♦ Regression testing—to ensure that previously working functions (e.g., rule interaction, calculations, etc.) will not conflict with, or fail, as a result of newly added or modified rules (or other features).

4.9. Rule Logic Verification

Here we examine tools offered with a product intended to automate the testing of the logic of rule sets for anomalies and conflicts.

Typical facilities offered include analysis tools (for examining rules and rule sets) that can generate conflict reports identifying potentially conflicting or otherwise invalid rules.

4.10. Conflicts and Priorities

This section explains how the BRM product uses priorities to handle conflicts among rules at execution time and the approach used.

4.11. Decision Explanation/Auditing

This section examines a product’s features for providing an explanation for its reasoning or decisions. In general, such facilities record logging and auditing information such as transaction date,
execution date, explanation of decision, data received/collected, and variables calculated during decision processing.

Some tools also provide capabilities for generating audit trail reports (typically in XML and HTML format) detailing rules engine activity, how rules corresponded to one another to arrive at a decision, and the progress of each decision, etc.

5. **Simulation**

Section 5 examines a BRM product’s capabilities for simulating business or application scenarios.

Simulation capabilities vary among products; however, the trend is for vendors to offer facilities, in particular, Wizard-based tools, that allow users to simulate business rules in different scenarios. Such facilities enable the user to select the appropriate inputs—or read in test data files—and outputs for testing. The tool then generates a user interface that can be used to simulate various business scenarios and conditions. This provides a way for developers and analysts to review business rules directly without the need for additional technical resources. Simulation is especially useful for testing new rules or rule changes before deploying them in the production environment.


Many users will naturally want to generate and publish reports detailing performance and other decision-processing information with their rules applications. For this reason, organizations should carefully assess a product’s particular reporting and document management capabilities.

6.1. **Document Generation**

Most BRM products provide some kind of capabilities to analyze rule projects, performance, and generate conflict reports, including:

- Browsers and other GUIs for visual inspection and
- Generated documents—including XML and online (linked) HTML format

For more on documents and reports see Section 4.9.

6.2. **Document Management**

Document management capabilities vary among BRM products. Some only allow the user to create a file and indicate the location for where reports or documents should be stored. Others provide the ability to reference content and sub-content to be included in the presentation or disclosure of documents.

6.3. **Interfaces to Third-Party Document Management Systems**

BRM products tend to be weak in the area of providing specific interfaces to third-party document management systems. Although all the vendors we talked with said they exploring partnerships with major content management systems vendors.

7. **Development Environment**

Section 7 focuses on various technical specifications for the BRM product’s development environment.

7.1. **Programming Languages and Support for Open Standards, Components and Frameworks.**

Here we describe the programming language a BRM product is written in, its support for different programming languages (C/C++, Java, etc.) and open standards (e.g., J2EE, etc.) for application development, as well as any object/component frameworks used or supported.
7.2. **Software Models and Code Generation**

This section examines software modeling and code generation capabilities provided by a BRM development tool—including the ability to work with/generate C, C++, Java, COBOL, C#, etc. and whether code-generation capabilities come standard with the product or are available as an add-on.

We also examine whether a product is available in different versions to support specific programming languages.

7.3. **Maintenance and Update Support**

This section describes how a BRM product supports maintenance, incremental updates, and new versions of rules and/or rule sets. Some tools support these functions with more advanced repositories. Others offer more limited functionality in the form of file and source control systems and other facilities.

7.4. **Integration with Third-Party Development Environments**

Here we focus on a tool’s ability to interface with popular software application development environments (e.g., IBM Rational Rose, MS Visual Studio, Eclipse, etc.) in order to facilitate BRM application development and deployment.

8. **Methodology Support**

Section 8 describes any business rules development methodology support provided by a product. There are a few business rules methodologies, such as those advocated by industry leaders and described in books by Ronald G. Ross (*Principles of the Business Rule Approach*, Addison-Wesley, 2003) and Barbara von Halle (*Business Rules Applied: Building Better Systems Using the Business Rules Approach*, Wiley Press, 2001). Other business rules methodologies are available from companies and consulting firms and include Proteus—from Business Rules Solutions—and the STEP Methodology—from Knowledge Partners, Inc.

Many of the BRM product vendors either do not support any particular business rules methodology, or they have their own proprietary business rules methodologies that they have created based on practices they have compiled from their consulting experiences. Additionally, some vendors’ proprietary methods combine—or are based on (i.e., “are compatible with”) various aspects of the more public approaches just mentioned. Likewise, some products offer documentation describing best practices that the vendor has acquired through the development of rules projects with clients in different industries.

Various standards groups and organizations are also working to create more standard business rules methodologies as well as standards for other aspects of business rules definition and development.

**Business Rules Management Methodologies**

Assuming one takes the more pragmatic approach, then the capture and documentation of business rules occurs in the context of a project of limited scope.

- The first step (after scoping the requirements for a project) is the rule capture phase in which the business rules are identified.
- The next step is to classify them. A number of business rules classification schemes are possible. Basically, it depends on the needs of the organization and the scope of the application. In general, rules are classified into various categories. Typical methods for categorizing rules include by term (e.g., all rules that use the term “customer”); by tasks or activities (e.g., all rules pertaining to order processing, etc.); or by decision points (e.g., all rules used to approve a loan, etc.). These different categories of rules are referred to as “rule sets” or “rule bases” by business rules practitioners.
Once the business rules are identified and classified, the next step involves expressing them in a format that can be easily understood (i.e., in a manner that expresses their business intent or purpose)—in particular, by non-technical business users. This typically involves the use of some form of natural language syntax, for example: “IF_CUSTOMER INCOME IS > $78,000 AND CREDIT_SCORE IS 3 THEN APPROVE”.

Finally, business rules and rule sets are stored in some form of repository or database, allowing users to manage and manipulate them according to their needs.

If Business Rules are developed in the context of a BPMS development effort, then the project usually begins with the development of a model of the business process. In essence, the process model defines the context in which any business rules will be used. Each activity in the business process is examined to determine what business rules are used in the activity. Assuming one is using a business modeling or BPMS software tool, one accumulates information as one defines processes. This, a new process may reuse an activity used in some other process and, hence, reuse the business rules defined for that activity. Similarly, a new activity may use business rules that have already been defined for other activities in other processes. Thus, it is important that the rules used in a process are maintained independent of the specific activities of any given process. The definition of process activities include pointers to the rules used in the activity, and rules, likewise, maintain pointers to the processes and activities in which they are used. In a similar way, the definition of processes and activities often generates a formal vocabulary of standardized terms. The definition of business rules also generates a formal vocabulary, and these two vocabularies should be maintained in a single location so they remain consistent in both process work and business rules work.

As we have already suggested, one can define rules for processes in tools that simply allow for the documentation of the rules, but its more common to specific business rules in inference-based tools that could, subsequently, support the automation of a given decision.

9. Integration

This section examines a BRM product’s ability to integrate with a variety of third-party applications and information management systems as well as its support for open-standards integration specifications.

9.1. BPM Suite Integration

Here we describe how a BRM product is embedded in, and/or supports bridges to, BPM Suites or BPM modeling environments.

9.2. Production Systems and Third-Party Applications

This section examines how a BRM product integrates with other production systems and third-party applications. Emphasis is placed on how the product supports open standards for integration (e.g. XML, UDDI & SOAP, CORBA, JMS, etc.).

9.3. Web Services/Service Oriented Architecture Deployment

This section describes a BRM product’s Web Services features and functionality for deploying applications in a Service Oriented Architecture (SOA)—including any specific APIs, frameworks and conventions supported.

9.4. Messaging and Middleware Integration

Here we describe any specific APIs and integration modules offered for integrating the BRM product with mainstream messaging and middleware systems such as WebSphere, MQSeries, BEA, etc.

9.5. Database Support

This section describes databases and APIs for data access supported by the BRM product.
9.6. Predictive/Analytic Modeling Support
This section examines a BRM product’s ability to work with rules that are automatically generated from data.

It is possible to automate some of the steps involved in developing rule sets by using data mining and induction tools that can convert decision trees and other models generated from their analyses into rules for incorporating into a BRM application’s rule base. (It should be noted, however, that someone still needs to determine which rules are appropriate to apply. This someone must have a detailed knowledge of the domain in which the application will be used as well be skilled in rule-based systems development).

Some BRM products offer such capabilities as a standard feature available with the tool. Other products offer such features in the form of add-on modules. Still others provide APIs for importing decision trees and other models generated by third-party data mining tools and other modeling environments.

10. Templates and Frameworks
The general trend is for vendors to offer horizontal and vertical products that package knowledge they’ve gained from working with companies to develop BRM applications for various industries and applications. Such “packaged knowledge” is typically offered as a add-on product.

10.1. Horizontal and Vertical Industry Templates, Frameworks, and Pre-Built Rule Bases
This section discusses any horizontal or vertical industry templates, frameworks, or pre-built rule bases the vendor is offering (for use with its BRM product) designed to assist with rule-based application development for specific domains and applications.

10.2. Domain-Specific Knowledge Products
Some vendors offer domain-specific knowledge, packaged in the form of specific processes or rules. This packaged knowledge is typically available as an add-on product offered in conjunction with application consulting and implementation services.

10.3. Domain and Industry-Specific Packaged Rule-Based Applications
Some BRM vendors are offering pre-built or “packaged” domain or industry-specific application products. This section describes them.

11. Administration and Security
Section 11 describes systems administration and security features provided by a BRM product.

Security and administration capabilities depend to a large degree on the repository or versioning functionality provided with the BRM product. Some tools provide fairly complex administration facilities, which take advantage of the check-in/check-out, versioning, and logging capabilities of their repository to control user access to rules, rule sets, attributes, calculations, messages and other functionality. In general, repository-based tools allow managers to set role and file authorization levels according to different users roles.

With other BRM tools, access is primarily managed by a source control system (either provided with the product or using a third-party offering), which allows administrators to define access to individual elements (e.g., attributes, calculations, messages, rules, etc.) in at the group or user level.

Control over which users have access to which rule development and maintenance screens should be enabled through standard enterprise Web security controls such as Lightweight Directory Access Protocol (LDAP), lookups, and ID and password logons.
A Detailed Analysis of Business Rule Products

Because BRM applications are typically designed to supplement existing applications with rule-based decision support capabilities, they tend to “live inside” the security functionality provided by the host application. For example, in J2EE Web application scenario, the BRE would leverage the security of the application server. Integration with existing security infrastructure systems is facilitated through various APIs (i.e. single sign-on, LDAP, etc.).

12. Platforms

This section describes the platforms supported by the BRM product, including operating systems (client and server) and application servers.

13. Scalability

Section 13 (13.1, 13.2) examines product features designed to support scalability—both vertically (i.e., small through large numbers of users) and horizontally (i.e., the ability to distribute applications and work across the organizational network)—capabilities which are important for supporting large rules projects.


14.1. Industry Standards Initiatives

This section identifies vendors who are participating in industry standards initiatives.

One of the major roadblocks standing in the way of greater adoption of business rules practices is a lack of industry standards for rules languages and modeling languages. Today, most product vendors are using their own proprietary rules language. Ideally, it would be nice if an organization could apply BRM technology using a standardized rule representation (i.e., production rule representation) and portable rules syntax that could be used with different vendors’ products for different rules projects. In addition, it would also be useful if business modeling tools could reference a common format for expressing their rules that could ultimately be converted into a common production rule representation.

A number of standards organizations and industry consortia have initiatives underway to create standards for business rules production (i.e., portable rules syntax) and modeling languages. For the most part, these are still emerging standards whose completion is some ways off, and they remain in various stages of development. Two of the most important standards organizations include the Object Management Group (OMG) and the World Wide Web Consortium (W3C).

Object Management Group (OMG)

Semantics for Business Vocabularies and Business Rules (SBVR). The OMG—in participation with vendors and end-user organizations—is developing the SBVR. Basically, the SBVR is a high-level set of concepts and categorizations that can be used to capture the meaning of business concepts and rules in business terms, in a formal manner that can be accurately communicated, verified, and, if appropriate, automated. SBVR is designed to be maintained in English by an organization’s business analysts, and can be stored in any MOF-compliant repository (SBVR is formally represented in OMG standard MOF/XMI format). SBVR will be usable with OMG’s Model Driven Architecture (MDA). On September 2005 the OMG approved SBVR and it will soon become a formal OMG standard.

Production Rules Representation (PRR). The OMG is also working with vendors and other members to create the PRR, which is intended to address the representation of production rules in UML models. “Production rules” in this case are not to be confused with XMI production rules as defined in XMI specifications or with other model or grammar transformation rules specified by other OMG standards. Rather, the PRL effort involves work on a MOF2-compliant meta-model with precise dynamic semantics to represent production rules that are executed by a BRE. This meta-model is intended
to support a language that can be used with UML models for explicitly representing production rules as visible, separate and primary model elements in UML models. This effort also involves defining an XMI W3C XML Schema description (xsd) for production rules—based on the proposed meta-model—in order to support the exchange of production rules between modeling tools and BREs.

**Business Motivation Model (BMM)**. The OMG is also considering a high-level description of the relationships between strategies, processes and rules. In late 2005, the OMG approved the BMM and it will soon be an OMG standard.

**World Wide Web Consortium (W3C)**

**Rule Interchange Format (RIF)**. The W3C is developing the RIF, which will specify a standard format for rules, so they can be used across diverse systems. Basically, this language will function as an interlingua into which established and new rule languages can be mapped, allowing rules written for one application to be published, shared, and re-used in other applications and other BREs.

Due to the various existing rule languages and BRE technologies, this common format will take the form of a core language to be used along with a set of standard and non-standard extensions. This project is part of the W3C’s larger goal of enabling the sharing of information in forms suited to machine processing—as witnessed in several application areas presented at the 2005 W3C Workshop on Rule Languages for Interoperability. The RIF specification is expected to receive broad industry support among vendors; however, a final specification is still a considerable ways off and no one is using it today.

**Organization for the Advancement of Structured Information Standards (OASIS)**

**The Business Rules Markup Language (BRML)**. The BRML is an XML Rule interlingua—an XML-based format for expressing rules in engine-neutral fashion. It is used in connection with “CommonRules” from IBM, and was developed in connection with IBM’s Business Rules for E-Commerce Project.

CommonRules is a Java library that provides functionality for business rules inter-operability. Its role is to complement and enhance the functionality of the various rule-based systems currently available. First, it provides a common “interlingua” representation for rules, suitable to grow into an industry standard, for exchange of rules between heterogeneous rule representations employed in various rule-based applications. Import and export formats for rules may be XML, text, Java objects (the applications need not be in Java). Translation between the different rule representations, and thereby rule exchange between the applications, is accomplished by translating in and out of the interlingua. The tentative name for the interlingua is Business Rules Interchange Format (BRIF). The tentative name for the XML version of the interlingua is Business Rules Markup Language (BRML). Second, CommonRules includes features for prioritized conflict handling that make the specification of rules more natural and makes updating/merging of rules simpler and more modular.

**Defense Advanced Research Projects Agency (DARPA)**

Another XML standard has been proposed by DARPA. The DARPA Agent Markup Language (DAML) is designed to enable developers to tag pages of information so that they can be read by BRM products, which can then parse the information and use it in a rules-based context.

**RuleML.org**

The RuleML Initiative (RuleML). RuleML.org is basically an open network of individuals and groups from both industry and academia. Its objective is to provide a basis for an integrated rule-markup approach that will be beneficial to all involved and to the rule community at large. This shall be achieved by having all participants collaborate in establishing translations between existing tag sets and in converging on a shared rule-markup vocabulary. This RuleML kernel language can serve as a specification for immediate rule interchange and can be gradually extended—possibly together with related initiatives—towards a proposal that could be submitted to the W3C. RuleML is an attempt to
A Detailed Analysis of Business Rule Products

support forward-chaining (top-down) and backward-chaining (bottom-up) approaches to building rules. RuleML was proposed as an XML standard but has been extended to Java-based BREs.

The Java Community Process (JCP)

Java Rule Engine API (JSR-94). JSR-94 is a completed standard that defines a number of interfaces that a Java developer can use to interact with a Java BRE. However, it does not attempt to define a standard rules language and thus, has a limited practical impact. Rather, it provides a standard API for: creating a stateful interaction with a BRE; creating a stateless interaction with a BRE; and deploying an executable set of rules from a variety of sources and registering them for execution. A number of BRM vendors support JSR-94 in their products.

14.2. Business Rules Consortia, Groups and Conferences

There are a number of industry consortia and groups dedicated to business rules. One can visit websites, read publications, or visit conferences to learn more about business rules. These include:

♦ The Business Rules Community [www.BRCommunity.com](http://www.BRCommunity.com)
♦ The Business Rules Forum [www.brforum.com](http://www.brforum.com)
♦ The Business Rules Group [www.businessrulesgroup.org](http://www.businessrulesgroup.org)
♦ The European Business Rules Conference [www.eurobizrules.org](http://www.eurobizrules.org)

15. Pricing

Section 15 describes pricing structures for BRM products. Pricing for BRM products involves licensing for development and deployment. The former is typically based on a per seat license (where a seat is expected to be deployed to a machine or user). Deployment is usually available on either a per-processor basis or a named-application basis. Other factors—such as additional components and support options—come into play as well. Consequently, actual pricing can vary considerably, depending on an organization’s particular development needs.

Also, please note that pricing listed in the Matrix is for an introductory version of the product.


16.1. Company Background

This section provides vendor background information and considers any purported company strengths and weaknesses.

16.2. Positioning

Here we describe company and product positioning, target markets and industries, and customer base.

16.3. Educational Training/Product Training

This section describes product-training programs and services offered by vendors.

16.4. Product Support

Here we describe available options for product-support services offered by vendors.

16.5. Consulting Support

This section describes BRM application implementation consulting services and workshops offered by vendors to assist end-user organizations with developing and deploying BRM applications.
17. Case Study
Section 17 offers an actual case study that illustrates a typical example of how each vendors’ BRM product has been used. It examines the context, challenge and the solution and benefits the application of BRM technology has brought to the organization.

18. Company Offices
This section lists vendor’s main office and key subsidiary offices.
1. Product Overview

Table 1 provides an overview of the Corticon Business Rules Management (BRM) platform and its main modules and functionality. Corticon’s approach to business rule modeling and execution differs in a number of ways from the more traditional (RETE-based) BRM products in which developers literally write rules using some form of rule-based language.

Corticon allows users to create rule sets in a completely declarative fashion using spreadsheet-like models. Using Corticon, organizations can model and manage their business policies, guidelines and operational decisions through an interface very similar to Excel.

Corticon features a “design-time” rules analysis functionality that mathematically verifies the reliability of the rules as a part of the modeling process. By applying patent-pending technology known as Predicate Logic Matrix, this design-time conflict-checking feature helps to ensure that modeled rules will deliver their intended result—providing a logically complete and unambiguous decision for every possible outcome. In effect, Corticon business models can be verified for conflicts and other logical errors with a click of the mouse, and then tested against business cases, without writing code.

Verified and tested, rule sets form an executable decision service. Each decision service conforms to a “service contract”, which fully describes the inputs and outputs of the service. Corticon generates compiled, optimized executable decision services (in the form of optimized Java byte code), which are deployed as Web Services or Java Services, enabling standards-based integration to automate decision points across corporate processes and applications.

Corticon decision services are optimized to execute reliably and exceptionally fast on the server. Runtime performance is fast because what is being executed has been optimized and compiled at design time. Decision services also take advantage of the application server’s scalability features.

Finally, Corticon is heavily focused on meeting decision automation needs (as opposed to traditional decision support applications), where fast design and reliable and fast execution are essential. Example Corticon applications include Sarbanes-Oxley and Basel II compliance initiatives, loan pricing and selection in banking, claims adjudication in insurance, complex service pricing in telecommunications, and supplier selection in manufacturing.

| Corticon Business Rules Modeling Studio | Complete and comprehensive modeling, analysis and test environment ready “out-of-the-box”. Features intuitive spreadsheet-style business rules modeling and maintenance environment—based on an extended decision table format; support both developers and (non-technical) business users. Users can capture and model their organization’s best practices, policies and procedures without programming (i.e., coding). Models are fully executable, and can be incorporated into existing applications as Web Services or Java Services. Applies sophisticated analysis and mathematical techniques to identify (and assist users) in resolving logical integrity |
Table 1. Overview of Corticon BRM, Key Modules and Functionality

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Corticon Business Rules Server</strong></td>
<td>Forward-chaining inference engine written in Java (supports backward chaining at design time). Can function standalone, using the features of any J2EE Web Server product embedded in a Java application or deployed into a container (i.e., servlet, session or message bean) within any J2EE application server, including IBM WebSphere and BEA WebLogic. Utilizes a patented “design-time inferencing” technique to determine the optimal sequence of a given set of rules, adhering to all declarative constraints. Targeted at decision automation needs (as opposed to Decision Support applications), where fast design, and reliable and fast execution are essential.</td>
</tr>
<tr>
<td><strong>Corticon Business Rules Collaborator</strong></td>
<td>Repository-based team development environment for business rules development and management via Web interface; provides facilities for managing rule assets through their lifecycle, including rules versioning and access management, workflow and notification for rule approval processes and rule change impact analysis.</td>
</tr>
<tr>
<td><strong>Corticon Enterprise Data Connector</strong></td>
<td>Provides enterprise-class connectivity to back-end data sources during rules execution; uses a model-driven approach to data connectivity that requires no SQL coding. Supports business agility and reduces maintenance by isolating rules from data source changes and isolating applications from data requirement changes.</td>
</tr>
<tr>
<td><strong>Pre-built Templates and Patterns</strong></td>
<td>Corticon provides support for templates that make integrity-safe changes to business rules, but does not offer any specific pre-built templates or frameworks for use with its BRM platform. Rather, Corticon relies on its partners to offer specific industry/application content and templates.</td>
</tr>
</tbody>
</table>

2. Product Architecture

2.1. Architecture Overview

Figure 1 provides an overview of the Corticon platform architecture, which consists of two main modules—the Corticon BRM Studio and the Corticon Business Rules Server. Additional modules include the Corticon Business Rules Collaborator and the Corticon Enterprise Data Connector.
Rule Authoring Environment/Graphical IDE: Corticon BRM Studio

Corticon Studio is a standalone business rules modeling and authoring platform that provides a single spreadsheet-like modeling environment designed to support both developers and business users. Corticon Studio’s spreadsheet-like GUI allows users to capture and model their organization’s best practices, policies and procedures without programming dependency. These models are fully executable, and can be incorporated into existing business applications as standard (XML-based) Web Services.


Corticon Server is the BRE for the Corticon platform. It can function standalone—using the features of any J2EE Web Server product—embedded in a Java application—or deployed into a container (i.e., as a servlet, session or message bean) within any J2EE application server, including IBM WebSphere and BEA WebLogic.

Rules Repository/Rules Database: Corticon Business Rules Collaborator

Corticon Collaborator provides a repository-based environment for managing the business rules (and their associated rule assets) built with Corticon Studio. It supports multi-user team-management features—including check-in and check-out, versioning, security and access control, workflow, and collaborative rules development.

Corticon Enterprise Data Connector

Corticon Enterprise Data Connector provides a direct connection from the Corticon rules engine to relational databases and other data sources. This allows the BRE to retrieve missing data or update data sources directly during rules processing, thereby improving performance and simplifying integration.
Data Connector’s functionality leverages the Corticon platform’s code-free modeling approach and requires no SQL programming skills; all SQL code is generated from the spreadsheet-like rules models created in Corticon Studio.

Enterprise Data Connector offers considerable business agility. This is because rules are isolated from changes to the data sources, and applications consuming decisions (i.e., rule sets) are isolated from changes to input data requirements. For instance, adding rule logic to a loan eligibility decision that requires new data (e.g., customer income, late payment history, etc.) could be accomplished without any modification to the calling application. All of the new data can be retrieved from the external data source by Corticon.

**Deployment**
The Corticon platform affords various options for deployment and integration, including:

- Client deployment—Corticon’s run-time can be deployed directly on a client (with a footprint of as little as 800KB) when deployed as an applet. This option allows rules to run seamlessly within a browser environment, providing a highly-interactive user experience based on business rules.
- Web services—rule sets are deployed as Web Services and integrated with architectures that can invoke and consume Web Services via SOAP documents.
- Microsoft .NET framework—rule sets are deployed as Web Services within the Microsoft environment and invoked utilizing .NET Framework and SOAP messaging.
- Java service with XML payload—rule sets are deployed as Java services (using a J2EE Servlet or EJB interface) and integrated with an architecture that can make Java method calls and transfer XML payloads.
- Java service with Java object payload—rule sets are deployed as Java services and integrated with an architecture that can make Java method calls and transfer Java objects.

**2.2. Business Rules Engine**
Corticon is an inference-based BRM product that supports forward chaining. It uses backward chaining at design time to support optimization and analysis, but does not offer backward chaining within decision services at run-time, as this would break rule set integrity. Corticon representatives make the case that this is where the distinction between decision automation and decision support is critical. They believe that to support the ultimate goal of decision automation (i.e., guaranteed logical integrity, and reliable and fast execution of decisions), one must give up run-time backward chaining—in effect, believing that the two are mutually exclusive.

**2.2.1. Rules Management Approach**
The Corticon platform uses a patented data-oriented approach to business rules combined with the application of advanced mathematical techniques (Predicate Logic Matrix) at design time to create optimized and compiled rules sets in the form of decision services with guaranteed logical integrity.

Corticon’s sophisticated analysis tools are designed to assist users with identifying and resolving logical integrity issues, thus guaranteeing conflict-free rules. Design-time inferencing is also used to determine the optimal sequence of a given set of rules, while at the same time, adhering to all declarative constraints.

**2.2.2. Enterprise Data Interoperability**
Corticon Enterprise Data Connector ensures data consistency of data from external sources through analysis at design time. Corticon’s approach to connecting with external data sources leverages the
analysis features to guarantee logical integrity while isolating rule logic from changes to data sources and hiding the sophistication of external data connection from the user.

2.2.3. Maintaining Data Quality

Corticon deploys and executes decision services (i.e., optimized and compiled rule sets with guaranteed logical integrity) that feature well-defined service contracts with respect to data. And Corticon allows for user-defined data quality rules. Thus, data quality issues are proactively managed at run-time, and the Corticon decision service contract does not allow inconsistent data types to be processed, with data validation rules effectively trapping data values outside a range of interest.

3. Interfaces and Ease of Use

3.1. Development Environment: Corticon Rule Modeling Studio

Corticon’s primary user environment for rules modeling is Corticon Studio, which features one common language that is accessible to both developers and business analysts. This single representation of rules is based on an extended decision-table format.

Corticon has extended the traditional decision-table format to make it capable of expressing the complexity of other technical rule languages, while at the same time, preserving the ease-of-use that has made spreadsheet products ubiquitous. The result is that Corticon Studio looks and behaves much like Excel, and users of all levels of sophistication can use it to model, analyze, test decisions (rule sets), using one common language and modeling environment. In other words, there is no separate language (i.e., developer language) to synchronize with the business user language.

Model-driven Environment

Corticon Studio is completely declarative and model-driven—including rules definition as well as comprehensive rule integrity analysis and business context testing (to verify that rule results satisfy business intent). Even advanced rule language concepts are handled declaratively in the model without requiring any procedural programming.

Corticon Studio features a rich library of operators and business objects that users drag-and-drop to construct rules. In addition, natural language expressions of rules are captured and used as reference documentation and as source for messages posted during the course of rule set execution.

Figure 2 shows a screen shot of the Corticon Studio modeling environment. Note that in Corticon’s extended table format, rules correspond to columns, with the condition part of the rule in one section and the action part in another section. Using Corticon’s library of operators and ability to support expressions, users can model business rule logic in a declarative fashion and then analyze, test, and deploy it.

Figure 2 gives several examples of how rules are defined in Corticon Studio. Column one shows a fairly simple rule: IF AN APPLICANT IS ‘MALE’, THEN ADD $100 TO THE POLICY PREMIUM. Column three (highlighted) shows another slightly more sophisticated rule: IF THE APPLICANT’S AGE IS MORE THAN 30, THEN ADD $10 FOR EVERY YEAR OF AGE MORE THAN 30.
Figure 2. Corticon provides a single rule modeling environment—based on an extended decision-table format—designed to support both IT developers and business users.

Figure 3 provides an example of how you can use Corticon to express sophisticated rule logic in a very compact fashion. These rules were created by dragging and dropping terms (e.g., “holidays”, “holiday.xdate”, “policyB”, etc.) and operators (e.g., “-> exists”) to compare a policy effective date with a collection of holiday dates. Note the natural language description of the defined rules that appears in the bottom pane of the screen.
Figure 3. Corticon rule modeling screen showing sophisticated rule logic

No External Data Access Languages
Corticon’s rule language syntax is free from any external data access language, such as SQL, for accessing database data. This modeling abstraction provides isolation from underlying database data and technology changes. First, separation of rule logic from data access details provides agility to organizations in that rule definitions don’t change whenever data sources change. Second, when connecting to external data sources, users simply identify the external source, and are free from any data access coding.

Comprehensive Analysis and Testing Tools
Corticon’s rule modeling environment also provides comprehensive analysis tools to guarantee the logical integrity of a decision (rule set). This “one-click” rule analysis feature helps ensure that rules are logically consistent (i.e., conflict-free, complete, and free of logical loops). Business case testing is similar: users drag-and-drop desired inputs and enter values, then execute rule sets and view the results and messages displayed in a business-friendly format.

Deployable executables are automatically generated from the model. There is no additional manipulation of rule logic to produce a final executable.

3.2. End-User Environment: Corticon Studio
Unlike some BRM tools that provide a technical language designed for developers and templates that enable editing of a subset of rule logic by non-technical end users, Corticon’s model-driven approach features one language and one UI. Thus, non-technical users can fully exploit the capabilities of the Studio’s spreadsheet-based modeling environment.

Figure 4 provides a screen shot of Corticon Studio showing its rules vocabulary (left side of screen) of business objects and operators. Basically, end users create rules and rule sets by dragging and dropping various operators and objects from Corticon’s rules vocabulary. Figure 4 also shows the results of an ambiguity check, where the conditions of the two rules are not mutually exclusive and the actions are different. The ambiguity is highlighted and multiple options made available to remedy the situation.
Figure 4. Corticon Rule Studio showing Rules Vocabulary of Business Objects and Operators

It's important to note, however, that should an organization desire to extend a subset of rule manipulation functionality (i.e., parameterized rules) to some group of non-technical users, Corticon support this scenario as well. Corticon supports rule manipulation from Web form templates so that tightly constrained, integrity-safe rule editing can be exposed to authorized users. Figure 5 shows a sample of a Web-based template.

The use of Web forms is important, because it can provide additional flexibility and control to organizations implementing Corticon applications. However, for most technical and non-technical users, the Studio modeling environment is the most appropriate and productive rules editing environment, where users define rule sets in the extended decision table layout, and executable code is generated from the model.

4. Business Rules Expression

4.1. Defining Rules and Rule Sets

Corticon’s rule representation scheme is based on an extended-decision table format that makes it capable of expressing the complexity of other technical rule languages, yet still provides the look and feel of a spreadsheet program.

Corticon also supports natural language expressions of rules within a section of Studio, and uses this representation as reference documentation and as source for messages posted during the course of rule set execution.
4.2. Creating and Maintaining Rules
Corticon’s extended decision-table format directly supports users in graphically defining and modeling rules and rule sets. This includes creating decision tables as well as scorecard type rules. It also provides an equivalent, concise representation of decision trees.

Additional rule language constructs enable the construction of scoring rules (increment and decrement rules), and other complex derivation-type rules.

4.3. Rule Templates
As noted previously, Corticon’s primary user environment is Corticon Studio, which uses one common language that is accessible to both developers and business analysts. Thus, there is no underlying technical rules language, per se, and therefore, no need to create an additional language or layer of representation in the form of templates to provide editing functions for non-programmers. In other words, there is no distribution or delegation of responsibility at the individual rule level that might possibly compromise the logical and business integrity of the rule set.

Corticon representatives see this as an important differentiating factor of their product. Because Corticon supports complex rule modeling capabilities (i.e., editing, analysis, testing) using one language, they believe it avoids or reduces some of the issues encountered when trying to synchronize rule sets, languages and interfaces—issues associated with some BRM products that provide separate authoring and rule maintenance environments for IT developers and business users. Basically, with Corticon, there is no “bridging” required to synchronize non-technical user input with the low level rule definitions.

However, as noted (in Section 3.2), Corticon does support manipulation of rules via a Web interface (i.e., Web form templates). Because Corticon can use external data within rules that maintain absolute logical integrity, Corticon can offer Web-based rule manipulation that is integrity-safe.
4.4. Rule Syntax Checking
Corticon Studio checks rule syntax for correct use of operators and correct references to business objects, including data types, and correct syntax in the overall construction of the rules.

4.5. Rule Testing and Tracing
In addition to Corticon’s analysis tools for guaranteeing the logical integrity of rule sets, Corticon offers business case testing tools designed to ensure business intent is satisfied with the rule sets.

Corticon’s primary testing features are the rule set testing tools built into the Corticon Studio. This is a business-friendly environment that allows users to construct test cases in a drag-and-drop manner or by importing from a variety of file formats. Results are highlighted for easier comprehension, as shown in Figure 6.

![Figure 6. Corticon Studio’s Test Environment](image)

Corticon further enables rule testing and debugging with model-level tools, which allow the user, for instance, to disable and enable rules and groups of rules. In addition, Corticon captures rule set execution data and messages for analysis and auditing purposes. With Corticon’s model-driven architecture, testing and debugging at the model level is the most productive means of ensuring rule sets are correctly defined. However, should the need arise to go beyond the model level, Corticon also provides a testing API to its engine, and code-level diagnostic and execution data for code-level debugging.

Corticon’s desktop modeling environment can test rules deployed remotely across intranets and the Internet, as well as rules that access back-end data sources.

Corticon’s rule testing and tracing capabilities allow you to generate reports showing findings. Corticon’s standard reports document the contents of rule modeling assets. Rule execution data is presented in Corticon’s test environment and is captured in a database for reporting and analysis.
4.6. Multi-User/Team Development
The Corticon Business Rules Collaborator provides a shared repository— accessible via a Web interface—and a comprehensive set of facilities to support deployment, management, integration and ongoing rules modeling and development for team-based projects, including:

Rules versioning and source-code type functionality (with granular access controls) for managing rules assets and projects throughout their lifecycle.

Distributed administration—content administrators can set access policies for rule assets in each domain and according to user privileges, ensuring that collaborators, users, and recipients access the appropriate rules and rules set versions.

Advanced workflow—including support for due date escalation actions, and alerts to help manage project schedules, etc.

Rule approval processes and rule change impact analysis to identify vocabularies and rules that have been impacted by changes.

Subscription services—for sending alerts to users who need to be informed of select activities within the repository, such as an update to a particular rule set or the commentary contributed by a reviewer, etc.

Search tools for finding rule assets that incorporate specific objects or logic patterns, rules vocabulary, English language rules statements, or meta-data associated with rules.

Collaborator is intended primarily to serve rule authors, the business community responsible for business policy, and IT personnel responsible for integrating and deploying rule assets throughout the organization. However, it is possible that a company could use Collaborator beyond direct BRM support to manage documents and enable team collaboration.

Corticon Collaborator also makes it possible to gauge the impact of organizational changes on rules. For example, developers can use it to locate all of the business rules that are affected by the change.

4.7. Rules Management and Maintenance
Corticon’s Decision Services Architecture tends to align well with how organizations structure their business authorization: the discrete point of control is the decision, not a part of the decision (rule). With the decision (rule set) as the fundamental point of control, authorized technical and non-technical users can use all modeling capabilities and change and manage every aspect of the rule set defined within Corticon. At any point in time the authorized users can know that the decisions under their control satisfy requirements for logical integrity and business intent. As changes are made to the rule models in Corticon, an impact analysis is generated, showing the business users how the change interacts with the other rules in the application in a way that they can understand and address, without the need to call on programmers.

4.8. Rule Logic Validation and Testing
In addition to tools to ensure rule logical integrity, Corticon offers a testing environment within Studio that enables business case testing to verify rule results satisfy business intent. As with other functions within Studio, business-case testing combines drag-and-drop functionality with the power of the full rules engine.

Test inputs and results are stored as XML, which can be reformatted/transformed to customer’s preferences. (See Section 4.5 for more on business-case testing.)

4.9. Rule Logic Verification
Corticon offers a number of facilities and tools to automate the testing of the logic of rule sets for anomalies and conflicts.

Corticon Studio provides:
Tests for ambiguities and completeness
Analysis tools to check for logical loops
Tests for redundant rules within decision tables

The tool can also collapse tables by removing redundant rules. Corticon has patented unique capabilities with respect to rule logic verification. This is supported by Corticon’s Decision Services Architecture and the focus on declarative rule modeling.

4.10. Conflicts and Priorities
Corticon’s rule management and architecture approach enables the platform to automatically detect and completely resolve conflicts at design time. Thus, it does not require the use of rules prioritization schemes and other techniques to handle conflicts and priorities.

Corticon detects and highlights conflicts and provides an override function to resolve rule conflicts. Corticon Studio also offers a rule expansion feature that assists users in creating a complete, conflict-free set of rules. These features allow Corticon to produce optimized, compiled, conflict-free executable decision services. In short, the engine does not have to try to resolve order of execution issues with priorities, because the order has been worked out at design time, and the resulting execution is both reliable and fast.

4.11. Decision Explanation/Auditing
Corticon logs the rules fired with all associated rule statements each time the engine executes. The rule statements provide full freedom to comment on the reasons for rule execution. This enables both technical and non-technical users to understand how decisions were reached.

Corticon also provides for the documentation of rule execution in the test mode of Studio as well as from the server. In Studio’s test mode, users can get instant feedback on precisely how the logic performed and why, enabling immediate editing to the rule set itself and a very quick iterative design cycle. On the server side, such rule documentation serves as an audit trail as well as for providing calling applications with useful data on rule set execution.

5. Simulation
Users can simulate rules and rule set behavior using Corticon’s testing tools. Business/application data can be entered or imported and run through rule set execution in Studio’s testing mode. Results are highlighted and rule messages are displayed. Multiple decisions can be assembled into a sequence within the tester, enabling the output of preceding decisions to feed input to downstream decisions, and allow visibility into how data changes from decision to decision.

In the testing tool, the user can bring in multiple rule-sets (decision services) and order them. Users can then take data and move the data through the flow of decision services. The testing environment pauses after each decision service has processed and shows the user the result (changed data) for the decision step that just processed (along with the rule statements that caused the data change). The user is then able to click on the execute button and it moves the data through the next decision service. In this way, the user can simulate a complex data decision automation in which multiple decisions all come together to create a final outcome.

In addition, the Corticon platform has also been integrated with BPM products (see Section 9.1) to provide even broader scoped simulation of business and application scenarios.

6.1. Document Generation
Corticon Studio generates XML or HTML files documenting vocabularies, decision services (rule sets), and tests. It provides standard HTML reports, as well as an XSLT transformer to help developers turn XML documents into custom reports.

6.2. Document Management
Corticon Rule Collaborator provides integral document management functions, including version and source control and workflow. It manages all documents in a central library, and provides controls for versioning and access. You can also add documents to the library and access them via the Web.

6.3. Interfaces to Third-Party Document Management Systems
Corticon works with third-party source control and document management systems, enabling companies to employ a single source management product to manage all application assets. Corticon's assets are file-based, which document management systems and source control systems can easily manage.

7. Development Environment

7.1. Programming Languages and Support for Open Standards, Components and Frameworks.
Corticon is written in Java and features a model driven architecture. Studio provides a complete rules modeling environment for defining rules, analyzing rules for logical integrity, testing rules for business integrity, debugging rules and generating executable (i.e., optimized Java byte) code that runs on the Corticon Business Rules Server.

The Corticon platform is extensible, supporting any Java service to participate in rule set execution as an "extended operator", thus preserving rule set logical integrity. Corticon rule sets are deployed as Java services or as standard (XML-based) Web Services that integrate into standards-based environments such as J2EE and .NET.

7.2. Software Models and Code Generation
Central to Corticon’s model-driven architecture is the generation of executables from the models created and maintained in Studio. These executables are deployed, without modification, on the Corticon Server to ensure optimum performance and reliability. In producing compiled, optimized executables directly from the model, Corticon can ensure logical and business integrity, and accelerate the rule set design and maintenance cycles.

This approach differs from the traditional software development model involving generated code, which may require further modification before deployment, and which ultimately gets compiled or interpreted at run-time.

7.3. Maintenance and Update Support
The Corticon platform supports version management, “effectivity” controls (i.e., when a rule set is valid, such as during a specific period of time), and automated change processes coordinated by the Corticon Collaborator tool to manage change. With its syntax and analysis tools, Corticon can detect and highlight changes that create inconsistencies, thus accelerating the maintenance cycle.

7.4. Integration with Third-Party Development Environments
Corticon Studio integrates with IBM Rational Rose in that object models exported from Rose can be imported into Studio to create the rules vocabulary.

Corticon decision services are designed to integrate via Web Services or Java, and can be tested from any number of development environments, including but not limited to, Microsoft Visual Studio.
8. Methodology Support

Corticon representatives stress that Corticon Studio provides a flexible rule development environment that is compatible with any rules discovery methodology. However, the company has developed its own proprietary methodology that is aligned with Corticon’s Decision Services Architecture.

A fundamental premise of the Corticon methodology is that business policies incorporate critical business decisions, and that these top level business decisions often can be decomposed into a hierarchy of business decisions. With this viewpoint, a blueprint of decisions can be formulated, relationships between decisions and with data and processes can be described, ownership can be established, and decision logic requirements initiated or consolidated.

The Corticon methodology extends through design, implementation, test and deployment, as well as change cycles. And Corticon's products—particularly Studio and Collaborator—are integrated with this methodology.

9. Integration

9.1. BPM Suite Integration

Corticon has integration agreements with numerous BPM vendors, including Adobe, Chordiant, Fuego, Global360, Handysoft and TIBCO.

Corticon’s use of Web Services as a primary rules interface makes it straightforward to integrate Corticon rules-processing capabilities with a wide variety of BPM/workflow products.

9.2. Production Systems and Third-Party Applications

Corticon supports several open standards for integration, including XML, UDDI and SOAP, and JMS.

9.3. Web Services/Service Oriented Architecture Deployment

Corticon was designed from the ground up to support SOAs. Rule sets can be deployed as independent or embedded Web Services or as Java artifacts within application programs. And WSDL/XSD Service Contracts are automatically generated for each rule set as part of the deployment process.

9.4. Messaging and Middleware Integration

Corticon provides a set of (general) Java APIs, but has not created APIs tailored for specific middleware systems.

Corticon supports the following application servers/middleware/transport:

- Apache Tomcat
- BEA Weblogic
- IBM MQ Series
- IBM Websphere
- Iplanet
- Tibco Rendezvous
- J2EE
- JMS
- RMI
9.5. **Database Support**
Corticon works with Microsoft SQL Server, Oracle, IBM DB2, and HSQLDB.

9.6. **Predictive/Analytic Modeling Support**
For models that are exposed in database tables, XML, or services, Corticon can map rules vocabulary to the data. The Corticon architecture provides additional integration options, including the use of extended operators to implement any Java method to employ predictive or analytic models.

Corticon also has the ability to integrate with analytic models while preserving the advantages of its declarative rule-modeling paradigm designed to guarantee logical integrity of rule sets.

10. **Templates and Frameworks**

10.1. **Horizontal and Vertical Industry Templates, Frameworks, and Pre-Built Rule Bases**
Corticon does not offer any specific horizontal or vertical industry templates, frameworks, or pre-built rule bases for use with its BRM product. Rather, Corticon partners do offer vertical industry rules content.

10.2. **Domain-Specific Knowledge Products**
Corticon’s partners offer vertical industry processes and rules.

10.3. **Domain and Industry-Specific Packaged Rule-Based Applications**
Corticon does not offer any pre-built or “packaged” domain or industry-specific application.

11. **Administration and Security**
Corticon provides design-time security via Corticon Collaborator. Collaborator’s administration features include ID passwords and logons. Developers can also integrate third-party source control systems for design-time management.

Corticon relies on external authorization systems for runtime security. In addition, like most BRM applications, Corticon is typically deployed within a container where security is provided by the application server itself. Security governing the execution of Corticon decision services, therefore, is enforced in the same way it is for any service within the control of the application server. The protocols and products used to manage security can include those supported by the application server, including LDAP and Active Directory based systems.

12. **Platforms**
The Corticon BRE runs on any operating system with JRE 1.3 or higher, including Windows 2000, Windows XP, Solaris 2.6 and higher, AIX 4.3 and higher, Red Hat, and SUSE.

Corticon can access and update data in a number of sources including relational databases like Oracle, IBM DB2, Microsoft SQL Server, and Apache Derby.

13. **Scalability**

13.1. **Vertical Scalability**
Corticon can be deployed as a service hosted by an application server. In this configuration, scalability can be achieved in a number of ways, including scaling hardware, clustering, and replication. In addition, Corticon decision services also take advantage of the application server’s scalability features.
13.2. **Horizontal Scalability**
Corticon can be deployed as a service hosted by an Application Server. In this configuration, scalability can be achieved in a number of ways, including scaling hardware, clustering, and replication.

14. **Business Rules Standards and Industry Groups**

14.1. **Industry Standards Initiatives**
Corticon monitors and participates in various industry standards rules initiatives, including the OMG Production Rules Representation (PRR), the OMG Semantics of Business Vocabulary and Rules (SBVR), and the W3C’s Rules Interchange Format (RIF) effort.

14.2. **Business Rules Consortia, Groups and Conferences**
Corticon is a Platinum sponsor of Business Rules Forum.

15. **Pricing**
Rule Modeling starts at $10,000, with initial projects averaging about $100,000 in software license.
Corticon also offers a wide range of services and support (see Sections 16.3, 16.4, and 16.5), including tailored services available on a project or time-and-materials basis.

16. **Company, Product Positioning and Support**

16.1. **Company Background**
Corticon is a privately-help company that was founded in 2000 by Dr. Mark Allen, a former MD (and the company’s current CEO) who has extensive experience in implementing rule-based applications for clinical and healthcare best practices.

In addition to its headquarters in San Mateo, California, USA, Corticon has two offices located in Los Angeles, California, and Washington D.C.

16.2. **Positioning**
Corticon has approximately 65 customers, many of which are in the financial services industry, where decision automation within process-driven applications is a critical success factor for their business. It’s important to point out, however, that Corticon does not exclusively focus on this area, but rather targets opportunities for automating a wide range of decisions.

Corticon sells, distributes and supports it products in North America, Europe, and Asia-Pacific.

16.3. **Educational Training/Product Training**
Corticon offers support, implementation and training services, including a focused “train-the-trainer” program. Since a large percentage of revenues are from partners reselling the Corticon solution, attention is focused on enabling those partners.

16.4. **Product Support**
Corticon offers telephone and e-mail support, with support levels offered ranging from basic (i.e., within normal business hours) to premium (24x7). Customers may also request on-site support from Corticon professional services personnel, if desired.

16.5. **Consulting Support**
Corticon offers a range of consulting services and training classes covering business rules modeling (including requirements analysis), business rules automation (including architecture design and prototyping) and application development support (including integration/deployment and post implementation support).
17. Case Study: Corticon at the LexisNexis Group

17.1. Context
The LexisNexis Group provides information to legal, corporate, government and academic markets, and publishes legal, tax and regulatory information via online, in hardcopy print and CD-ROM formats.

The global legal and information division of Reed Elsevier plc, LexisNexis Group offers searchable access to more than three billion documents from thousands of sources with leading edge systems and content management tools. The company offers Lexis service, the first commercial, full-text legal information service, and Nexis news, financial and business information service.

17.2. The Challenge
In 2004, LexisNexis was faced with lengthy, costly, and complex processes for entering, routing and fulfilling new customer orders within its Lexis service. When the company began investigating technologies to help automate order processing, reduce operational risks associated with errors in processed orders typically associated with inconsistent business decisions, and alleviate the burden on information technology (IT) staff, it turned to Corticon Technologies.

The LexisNexis vision was to achieve unified order fulfillment through a new Customer Order Management System that would automate the validation, routing and fulfillment of orders coming into the company via phone, Web, and fax.

17.3. The Solution
Corticon provided LexisNexis with a complete Business Rules Management solution to help create new customer accounts—accounts previously processed and managed by a custom application that required costly continuous development and technical resources.

“At LexisNexis, we’re utilizing Business Rules Management software from Corticon so our business users have the flexibility to create, change and manage the majority of their business decision logic without much IT involvement,” said Steve Iddings, Director of Applications Integration at LexisNexis. “Corticon has demonstrated to us that their rules automation software can reduce project costs and increase the quality of our business rules logic.”

The Corticon Studio modeler environment allows LexisNexis business analysts to capture, validate, test and manage business rules without the need for in-depth programming expertise. As a model-driven environment, the spreadsheet-like rules models are automatically converted to executable Web Services. While Corticon’s service-oriented architecture (SOA) enables LexisNexis to take the common set of business rules that come together to make up a business decision, and package and deploy that decision service as a Web Service. This allows business logic that might be needed in multiple applications to be managed as a common decision service.

17.4. Result/Benefits: Significant Reduction in Operating Costs
The Customer Order Management System at LexisNexis was established to achieve unified processing by automating validation, routing and fulfillment of orders coming into the company via telephone, Web, and fax.

Corticon was tapped to accomplish some significant goals, including ensuring orders were processed consistently, allowing orders to be successfully and accurately handled by fewer people than before, lessening the time it took for the company to fulfill orders, and reducing the time-to-market for new and/or improved LexisNexis products. Because the Corticon solution automates business rules, there is no need for custom coding as LexisNexis required before the implementation.

“The first priority for us was order automation, which would allow us to automatically maintain rules without technical expertise,” said Carter. “Corticon allows us to make intelligent routing decisions about customer orders. For instance, in the case of a new order, the system determines things like the
customer’s market, the parameters for their account, general subscription attributes, and the billing setup. Prior to Corticon, we had to custom code everything, which required a development resource to maintain the process. The costs related to that requirement are now much lower—we need technical input only occasionally now. The development lifecycle is significantly shortened.”

17.5. **Solution Agility Eases Rules Creation and Management**

“With the agility we get with Corticon comes the ability to resource a project more easily by assuming the analyst working on it can get close enough that it doesn’t require three staff months’ worth of development effort to write the rules,” said Carter. “We’re now faster and we’ve offset the allocation of technical resources on each project. I’d estimate it takes 50% to 70% less time to complete a project—that’s a huge savings. The Corticon system also gets the fulfillment organization more involved in the business process, and as they begin to understand each type of customer, they become more knowledgeable with every order.”

17.6. **Operational Risks Reduced**

Decision automation must be precise, and Corticon’s approach is to first analyze rules for correctness and then provide a testing capability that requires no programming or classic QA organization. In other words, Corticon helps LexisNexis determine whether their business decisions are correct and appropriate from the start by providing tools tailored specifically for business users as opposed to IT personnel. This approach is far more beneficial to the bottom line than the alternative approach of trying to test quality into the business logic.

Classic testing is hit-or-miss due to the complexity of decision logic. LexisNexis appreciates the quality control the approach provides for its bottom line. “If an analyst working on a project enters something incorrectly, Corticon will catch that and ensure accuracy,” said Chuck Carter, Senior Software Engineer at LexisNexis. “Once again, it removes the need for programmers to step in, as well as eliminating risk involved with erroneous business decisions caused by human errors made during the process of defining and creating the rules.”

18. **Company Offices**

**Corporate Headquarters**

Corticon Technologies, Inc.  
1000 Bridge Parkway, Suite 125  
Redwood City, CA USA 94065  
Phone: (650) 212-2424  
Fax: (650) 212-2727  
Web: [www.corticon.com](http://www.corticon.com)  
E-mail: [info@corticon.com](mailto:info@corticon.com)
1. **Product Overview**

Fair Isaac’s main BRM product is Blaze Advisor. However, the company also offers various products that can be used to supplement the development and functionality of rules-based applications built with Blaze Advisor. These include the Model Builder tool for creating predictive models that can be integrated with Blaze Advisor’s rule-based applications and the Decision Optimizer platform for optimizing Blaze Advisor applications. Additionally, Fair Isaac offers several pre-built BRM applications targeted at specific domains and industries—including credit card fraud prevention, risk management and marketing. This report focuses primarily on Blaze Advisor’s features for building and managing BRM applications.

<table>
<thead>
<tr>
<th>Blaze Advisor Builder</th>
<th>Rule-based development and management environment for building Java, .NET and COBOL applications. Provides integrated development environment and testing and debugging tools. Integration capabilities include XML, Java, COM, CORBA and Web Services. Deployment facilities for creating front ends and templates for non-technical end users; point-and-click Web interface generation that can be edited to match end-users’ needs and expectations. Features the ability to author rules in various formats, without a need for dictionary overlays and abstractions to an underlying symbolic language representation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blaze Advisor Rule Server (ARS)</td>
<td>Server-based execution environment (forward and backward-chaining inference engine) for deploying rule-based applications built with Blaze Advisor Builder. 100% Java object conforming to J2EE (1.2) and Web Services open standards. Deployable as a J2EE Enterprise Java Bean (EJB), Model Driven Bean (MDB), Web Service, COM/MTS object, .NET or within EAI systems (e.g., MQ Series, JMS and RMI). Runs on any application server or Web server, or can run as standalone server.</td>
</tr>
<tr>
<td>Blaze Advisor Repository</td>
<td>XML-based repository that complies with standard persistence mechanisms (e.g., file, LDAP or database, etc.) and versioning services (Source Code Managers such as CVS, or Blaze, etc.); supports rule service deployment for multiple applications—ensuring consistent enforcement of business policies and procedures across an enterprise—and repository-based team development.</td>
</tr>
</tbody>
</table>
Table 1. Overview of Blaze Advisor, Key Modules and Functionality

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rule Maintenance</strong></td>
<td><strong>Applications (RMA)</strong>: Wizard-based facility for automated generation of complete rule management applications that operate over any Web browser based on predefined templates. RMAs allow non-technical business users to review, modify, delete, or create rules within the structured constraints without knowing a single word of programming or specialized syntax and without using a specialized third-party editing environment. Rule management screen text and cosmetics can be modified using any desired HTML authoring tools to take on the look and feel of familiar applications within the enterprise.</td>
</tr>
<tr>
<td><strong>Quick Deployers</strong></td>
<td>Wizard-based facilities that simplify deployment of Blaze Advisors applications via automated code generation to create configuration files for immediate deployment in the products of major vendors. Can deploy rule services for major J2EE application servers (e.g., WebLogic, WebSphere, Oracle iAS, Sun ONE, jBoss, etc.) and a .Net assembly. Also generates deployments for processing in Java, Microsoft Transaction Server, JSR-94 rule engine interface compliance, and Web Services.</td>
</tr>
<tr>
<td><strong>Pre-built Templates and Patterns</strong></td>
<td>In addition to rule templates for assisting with creating rules, rule sets and rule maintenance applications (for both IT and end users), Fair Isaac also offers pre-built templates for Insurance, Mortgage, and Healthcare that serve as “starting points” for customers developing applications in these areas.</td>
</tr>
<tr>
<td><strong>Model Builder</strong></td>
<td>Optional tool for building scoring, predictive and other types of analytic models; designed to ease development and deployment of analytic models by providing the ability to export models into Blaze Advisor to form the rules set for decision-processing. Allows organizations to more easily build and deploy advanced BRM applications that combine both predictive and rules models within the same environment by eliminating the need for developers to re-code analytic models. Also reduces difficulty associated with deploying analytic models because model functionality is literally embedded within Blaze Advisor rules.</td>
</tr>
<tr>
<td><strong>Decision Optimizer</strong></td>
<td>Optional platform for business users and IT to manage and fine tune decision models for use in Blaze Advisor and Model Builder environments. Provides facilities that allow business users to understand complex logic behind decision-processing models, and a simulation capability for determining the effects that changing model constraints and criteria (e.g., customer scoring values, etc.) will have on decision processing while also taking into account a company’s own constraints. Also provides ability to test and compare effects of different models on decision processing; and offers a feedback mechanism for gathering the results of tuning decisions, which administrators can analyze to improve future model design, deployment, and optimization efforts.</td>
</tr>
</tbody>
</table>

Table 1 provides an overview of Blaze Advisor and its main modules. It also summarizes additional tools that can be used to supplement rules-processing functionality. Blaze Advisor is a rule-based development and management environment for building Java or .NET applications. In addition to an inferencing-based rules engine, Blaze Advisor provides an integrated development environment (IDE)—complete with multi-user repository that supports team development—and testing and debugging tools. Integration capabilities include support for XML, Java, COM, CORBA and Web Services. A COBOL generator version is available for organizations wanting to execute rule-based applications in mainframe environments.
Blaze Advisor supports both IT developers and non-technical business end users through the extensive use of wizard-based development facilities and reusable templates. These deployment facilities allow for creating front ends and templates which both constrain how a business user can write a rule and allows for generating a point-and-click Web interface that can be edited to match end-user needs and expectations.

Blaze Advisor can also execute analytics—in the form of scorecards (i.e., scoring models), decision trees, segmentation trees and other advanced algorithms—directly within its rules processing environment.

Blaze Advisor and its add-on facilities provide a complete set of tools for developing, deploying, and maintaining business rule services as software components, making it well suited for building and managing large, complex, and frequently changing business rules applications.

2. Product Architecture

2.1. Architecture Overview

Figure 1 provides an overview of the Blaze Advisor architecture, which consists of three key modules that support the authoring, deployment, maintenance and administration of business rules.

### Standards-Based Architecture

**Rule Architecture and Design**
- Blaze Advisor Development Environment
  - JDK 1.1.7b or higher
  - W2K, XP
- Blaze Advisor Rule Maintenance App
  - JSP Tags, Beans, MS ASP or Swing UI Webserver / App

**Rule Storage**
- Blaze Advisor Repository
  - XML
  - DB
  - LDAP
  - JNDI

**Rule Deployment**
- Blaze Advisor Rule Server
  - EJB/J2EE
  - MDB / JMS
  - SOAP
  - .NET/COM+
  - Servlet
  - MQSeries
  - Messaging
  - CORBA
  - RMI
  - C++/JNI
  - In-Process
  - etc

**Rule Management**
- Blaze Advisor Rule Maintenance Application (RMA)
- Blaze Advisor Deployment Manager
- Blaze Advisor Repository
- Blaze Advisor Rule Maintenance App

**Application Server**
- JDK 1.1 - 1.4
- Unix, NT, OS/390, AS/400, Compaq NSK, Linux etc.

**Web Application**
- JSP Tags, Beans, MS ASP or Swing UI Webserver / App

---

**Figure 1. Blaze Advisor Architecture**

**Rule Authoring Environment/Graphical IDE: Blaze Advisor Builder**

Blaze Advisor Builder is a graphical IDE for developers to create an architecture for business rules applications as well as for authoring business rules. It is a stand-alone Java application. Deployment of rule projects is done through the Blaze Advisor Rule Server. Blaze Advisor Builder also features a Rule Maintenance Application (RMA) module—a deployment facility developers use to create structured front-ends for simplifying management and updating of rules by business users.

Blaze Advisor Rule Server (ARS) is a server-based execution engine for deploying rule-based applications built with Blaze Advisor Builder. ARS is a 100% Java object conforming to both J2EE (1.2) and Web Services standards. It is typically deployed as a J2EE Enterprise Java Bean (EJB), Model Driven Bean (MDB) or Web Service. Other configuration options include deploying ARS as a COM/MTS object, a .NET component with a C/C++ interface, or within EAI layers such as IBM MQ Series, JMS and RMI. ARS is installable on any application server or Web server, or it can run as a standalone server.

Rules Repository/Rules Database: Blaze Advisor Repository

Blaze Advisor Repository is an XML-based repository that is separate from the execution engine (i.e., ARS). It supports rule service deployment for multiple applications—ensuring consistent enforcement of business policies and procedures across an organization—and repository-based team development.

Blaze Repository is a set of XML entities (files, documents, etc.) which can be stored in any mechanism (JDBC/RDBMS, LDAP, SCCM). Configuration options include saving XML rule files in a LDAP system that supports JNDI, a database that supports JDBC, or saved as flat files. The repository can use CVS or Blaze Advisor’s built in versioning system to maintain history information and to provide check-in/check-out functionality.

All rule structures, rule services, allowed value controls, and audit trail data (e.g., “last modified date”, “last modified by”, and “reason changed”, etc.) are stored in the repository. Likewise, rules created or changed in rule maintenance applications are also stored in the repository. The rule server monitors changes in the repository and updates affected rule services without interrupting production operations. The deployment manager provided with the product handles the assembly process for runtime execution.

Blaze Advisor Repository also provides various management properties, which allow organizations to add their own meta data, in effect, extending the repository to hold any information pertinent to a rules project. (For more on management properties see Section 4.6.)

2.2. Business Rules Engine

Blaze Advisor uses a forward and backward chaining inference engine (to process rules and other decision logic) that is based on an optimized version of the original RETE algorithm. It can efficiently manage and execute rule bases with many thousands of conditions. This algorithm uses rapid pattern matching to relate the conditions that are true at any given moment to the appropriate rule or rules. The rule engine can quickly look through large sets of rules to find the proper ones to apply based on case-specific conditions. No code is required to specify which logic subroutines should be fired in which order.

We must also point out that Blaze Advisor version 6.1 provides the option of using the RETE III algorithm, which is orders of magnitude faster than the original RETE algorithm.

2.2.1. Rules Management Approach

Blaze Advisor uses a rules management approach that is based on a combined repository and OO programming language (i.e., Structured Rule Language) that is designed to make writing and reading business rules as English-like as possible.

The Blaze Advisor rules engine allows several options for specifying procedural execution, including ruleflow sequencing for ordering groups of rules, and within each group (or rule set) the designer can choose engine-determined processing or sequential processing of rules in a predefined order. Other methods for controlling order of execution include setting relative priorities, event-based conditions, and explicit calls to rules from other rules. This allows the engine to primarily work in a forward
chaining fashion, with the capability to do backward chaining as well. An option is also available that allows special performance improvements for situations where rules can be fired in sequence. In addition, a compiled-sequential option (which requires an additional license) for native Java speed is also available.

2.2.2. Enterprise Data Interoperability

Blaze Advisor features the ability to build and execute rules that reference data stored in databases, XML documents, business events (including IBM MQSeries), Java/J2EE objects, CORBA objects, COM objects, and customer-defined objects which might be found in third-party data models. This is done using wizards which examine the definition of external data models and automatically construct Blaze Advisor internal references to their elements so that no dictionary definition step is required before starting to write rules. Wizards also construct runtime linkages so that rules which reference the current value of a data element or change its value automatically invoke the proper underlying interface code to retrieve or update the value in the external data source. Rules can make simultaneous use of data elements from a variety of data sources, including multiple databases.

2.2.3. Maintaining Data Quality

As part of the explicit resolution of missing data, Blaze Advisor is able to request data from external data sources and end-users. Blaze Advisor employs several methods to interact with end-users including question sets, external program calls and messaging (XML based and otherwise). This enables the BRE to request data and information from the user as is required, hence driving the question process flow. This also allows the amount of data requested to be limited to that which is required and reduces the number of questions asked.

When a missing data value cannot be resolved, an appropriate exception is generated based on the missing information and the decision process halted. This process will ensure the application handles the missing data, communicates the problem effectively to the client application and exit with causing further execution problems.

Designers can also write rules specifying how the application should act when encountering dirty or missing data. In this case, the action taken by the BRE in response is entirely up to the designer (i.e., they are not required to throw exceptions and halt the process unless they chose to do so).

3. Interfaces and Ease of Use

Blaze Advisor features two environments for rule authoring and maintenance. One is Blaze Advisor Builder, a full function IDE primarily intended for IT developers. The other is the rule maintenance application(s) (generated by Blaze Advisor Builder) designed for business users.

3.1. Development Environment: Blaze Advisor Builder

Blaze Advisor Builder’s graphical IDE features specialized editing windows and interfaces for different types of rules, functions, question sets, ruleflows, class definitions, and other entities. It allows side-by-side viewing of entities with movable windows or tabs. It also contains separate test interface windows, output windows, and editing windows. It provides a sophisticated navigation tree for immediate access to any object in a rule service, and it supports hyperlink access to any rule referenced in a compilation error or warning message.

♦ Using Blaze Advisor Builder, a developer can:
♦ Import an object model
♦ Create rules, rulesets, functions, variables, and other entities associated with the rules
♦ Test, debug, explore, and report on rule projects
Generate rule deployments

- Create the structure for non-technical users to create and maintain rules outside of builder

- English-like Rule Syntax

Rules can be written using English words and grammar such as “If customer’s average balance is more than ...”, or by using the mathematical symbols and object model “dot notation” familiar to programmers. An extensive selection of rule language keywords is provided, and rules can take advantage of regular expressions and powerful pattern constructs to dramatically reduce the number of rules required.

The Blaze Advisor IDE includes a full set of editors that simplify the creation of entities, and generate much of the syntax. Figure 2 shows the three different syntaxes available for referring to the data values, dot notation (applicant.isEmployed), English-like (the creditEstimate of applicant), and possessive syntax (applicant’s loanType). As shown, the different syntax types can be mixed within a rule. All syntaxes are available as soon as the data is mapped; there is no dictionary definition step required.

Figure 2. The Blaze Advisor Builder IDE

Blaze Advisor defines rules without specifying procedural code for ordering or controlling sequences. It executes appropriate rules based on relevance at time of execution controlled by ruleflow sequencing, rule events, time/date effectivity, and priority.
Visual Ruleflow Editor
This editor makes it fairly simple to break sophisticated conditional business processes into steps. Drag-and-drop icons allow graphical construction of business process definitions as collections of events, steps, branches, actions, conditions, and loops. Tasks can be represented by decision tables, decision trees, drill-down ruleflows, functions or rulesets.

Creation of Advanced Rulesets
Rule sets enable logical groupings of rules to be shared and reused, and activated and deactivated as a single entity. Rulesets make it easy to associate blocks of rules with a step in a ruleflow and can be activated based on context (e.g., only to “qualified” customers) as well as being passed parameters and assigning return values.

Execution Browser and Debugging Tools
Blaze Advisor provides a graphical representation of rule execution within a ruleflow, and can highlight specific rules, functions etc. as they fire. Breakpoints and watch points can be set and rule execution can be traced showing a visual representation of rule-firing sequences during testing. Rulebases can be analyzed for potentially conflicting or otherwise invalid rules.

Cross-Reference Browser
A “cross-reference” browser tool provides a complete graphical representation of interrelationships between rules, functions, and object properties that can be viewed allowing rapid review of all uses of a given function, ruleset, rule, etc. Any object (rule, ruleset, class, enumeration, etc.) may be queried interactively to check various dependencies, such as what it incorporates, what it reads, what it modifies, what other objects refer to it, etc. A dependency report for the entire project may be generated at any time in a hyperlinked Web document or in a static table of contents format.

Quick Deployers
Quick Deployers are wizard-based facilities designed to simplify deployment of Blaze Advisor applications. They automate code generation to create configuration files for immediate deployment of Blaze Advisor rules applications in the products of major vendors. Through a question and answer session, wizards collect deployment information and generate appropriate installation files for any supported application server, ant build files, Web Services, and configuration files. (For more on Quick Deployers see Section 6.)

3.2. End-User Environment: Rule Maintenance Applications (RMA)
The rule maintenance applications (RMAs) generated by Blaze Advisor allow business users to safely and easily create and maintain rules in a graphical environment without technical assistance. Users can review, modify, delete, or create rules within the structured constraints without knowing any programming or specialized syntax and without using a specialized editing environment.

System controlled pick lists, radio buttons, and other controls make RMAs easy to use and safe, as the values that can be selected come from enterprise data sources. This gives users the ability to review and manage rules using everyday terminology without being forced to learn rule syntax. In addition, RMAs can feature decision tables, scorecards and decision trees to make it easy for business users to maintain rules using graphical techniques. (For more on the use of these techniques see Section 4.2.)

RMAs are generated by using the Rule Maintenance Application (RMA) wizard. This wizard allows you to:

♦ Define the RMA’s style and functional characteristics
♦ Define the application scope and decide whether to include banners, a table of contents, list filters, a login screen, etc.
♦ Define labels, page text, data displays, fonts, and colors
Select the style of data entry and display controls

Once the application is generated, you can use any Web HTML design tool to make final stylistic customizations (i.e., incorporate the look and feel of other applications within the organization.) Figure 3 provides a screen shot of the Blaze Advisor Rule Maintenance tool for business users.

Figure 3. Blaze Advisor Rule Maintenance Center (for business end users)

A Rule Maintenance Authorization and Tracking facility allows you to control which users have access to which rule management screens through standard Web security controls like LDAP lookups and ID/Password logons. It also automatically records rule modification documentation in the rule repository associated with each change that is made. User name, date, time, version number, HTML links to external reference sources, and arbitrary comment text can all be associated with any rule modification.

4. Business Rules Expression

4.1. Defining Rules and Rule Sets

The Blaze Advisor Structured Rule Language (SRL) is an OO language designed to make writing and reading business rules as English-like as possible. It shares many features of common programming languages, and is intended for use by programmers to create the entities, control the execution flow, and perform the operations required by the (rule) decision making service. However, it also provides syntax for authoring individual rules and rule sets that are easily understandable by people with little or no programming background. You can use common English words such as “is less than” and
“exceeds” as well as common English representations of objects and properties such as “customer’s age” and “the age of the customer.” (For more on Blaze Advisor Builder see Section 3.1.)

Blaze Advisor rule sets separate rules into logical groups for ease of management and control, for reuse between rule services or business applications, for order dependent operations, and for functional enhancements to existing applications. Blaze Advisor Ruleflows are built in a simple graphical editor. The flow is intuitive to business people and does not rely on technical concepts such as UML. Data can be passed from task to task using ruleflow variables.

4.2. Creating and Maintaining Rules
Blaze Advisor supports three different graphical methods for creating and maintaining rules:

♦ Decision tables—allow vast numbers of conditions and corresponding actions to be shown in a grid. Designers can create tables in columnar, row, or combined formats with multi-level condition groups. A table cell can link to another table for nesting or can contain any number of actions, including anything that can be expressed in rule syntax. A Decision Table editor supports Cut, Copy and Paste functionality for cells, rows and columns. Extra rules can be added before or after the decision table is evaluated.

♦ Decision Trees—show navigation paths displaying condition assumptions leading to the branch in focus. Actions can be changed via a click-and-type interface. New branches can be added to the tree dynamically. A Decision Tree editor supports easy modification of split points, focusing on a portion of the tree and allowing non-exclusive branches leading to multiple actions when appropriate. Extra rules can be added before or after the decision table is evaluated. Blaze Advisor’s Decision Trees work well for situations where the conditions are not symmetrical, which would not work well as a table.

♦ Scorecards—allow weight factors to be assigned and maintained by business experts. Users or system architects can create and maintain reason codes used to track and report on most significant factors in a calculated score. Significance can be determined from pre-set priority levels or from a variety of calculated importance algorithms. For companies in sectors such as finance, insurance, government, or other highly regulated industries, Blaze Advisor offers ways to set, capture, and record significant factors used in calculating a score. By defining customized Reason Codes, organizations can make use of scorecards as an auditable component of their decision processes. And contributing factors to a score can be tracked and reported based on relative corporate priorities or on mathematical rankings considering each factor’s impact on the result.

4.3. Rule Templates
Blaze Advisor supports the creation of reusable rules management templates that can incorporate entire rule services with multiple steps, functions, and rulesets. Templates help enforce the consistent deployment of rule changes as well as provide the foundation for rule maintenance applications—allowing rules to be modified or created within a strict set of constraints appropriate to satisfy different users or tasks. Templates are exposed through a Web interface so that end users can update the values, choosing from a list or range of values that is set by the developers. Business users do not need to know whether they are editing rules, or functions, or any other part of the project, as the rule maintenance environment does not show any of the syntax. The whole spectrum—from simple rules parameterization to full condition and action selection—is supported. Templates and associated rule maintenance applications are able to reference any property, object, rule, database table etc. in Blaze Advisor.

(Pre-built templates offered for specific industries are covered in Section 10.)
4.4. Rule Syntax Checking
Complete rule syntax checking is automated through a pre-compilation step with interactive links to problematic code.

4.5. Rule Testing and Tracing
Extensive rule testing and tracing capabilities allow you to trace all rule execution operations in real-time, with interactive test values, user-defined breakpoints, traces, event monitors, single-step operation, and graphical depiction of rule execution.

An analysis tool allows checks for potential conflicts and unwanted interdependencies between rules. And the developer has control over what tests will be run and which rulesets will be checked. Results may be seen on screen or included in project reports (described below).

Rule definitions may be viewed from dedicated rule editing windows, or may be viewed in system reports (both hyperlinked and static). Rules may be accessed from an online navigation tree that allows a jump to a named rule, or from within rulesets accessed by clicking on task boxes in the visual ruleflow.

An Execution Browser lets you follow the execution of a rule agent graphically. By displaying a tree of possible execution paths and, as the rule agent is executing the project, it highlights the current execution path. It also provides the ability to interactively build dependency trees showing relationships between rules, parameters, and data objects.

A reporting facility lets you run reports describing the complete structure of a rule service, with detailed documentation of individual elements including ruleflows, decision tables, functional calls, and rules. You can review reports online (with interactive jumps between objects), or formatted with tables of contents for hardcopy or archival documentation. Rule conflict detection is included to point out potential problems with inconsistent, incomplete, or self-referential rules. Finally, all debugging and testing operations can be performed on the production system in a separate processing thread while end-user operations continued unaffected.

4.6. Multi-User/Team Development
Multi-user/team development is supported through shared repositories and check-in/check-out capabilities. Users can work offline and synchronize changes with the shared repository. The repository prevents multiple users from making changes at the same time or overwriting revisions by providing a way for users to check out and check in rule service components. Information about an older version can be retrieved and viewed simultaneously with the current version. And previous versions may be promoted into production to undo errors and problems.

Blaze Advisor’s built-in versioning system manages rule changes, including logging historical content changes of rules, rule services, and rule sets. Author, date and other comments are also captured. You can view the information from within the IDE, from a Web-based RMA, or as exported to a data file. The rule server monitors changes in the repository and updates affected rule services without interrupting production operations.

Blaze Advisor Repository also provides important management properties. For example, companies can add their own meta data to a project, thus allowing them to store additional information pertinent to the rules project (i.e., beyond basic information like author, date, and comment, etc.)—such as the stage of development, the office in charge of maintaining the specific code, and so on. These management properties can be queried on and used to determine which files should be moved to testing and production servers.

Companies can also use third-party tools should they have more familiarity with a different system. Blaze Advisor supports file-based, LDAP, and database repositories that will handle file access and versioning, including CVS, PVCS, SourceSafe, and ClearCase.
4.7. **Rules Management and Maintenance**

A number of wizard-driven development tools for building rules, rulesets, global rule templates, and importing definition classes are provided. Wizards for developing decision trees, decision tables, and scorecards are also available.

Templates can be used to automatically generate Web-based rule maintenance applications (RMAs) allowing the use of a company’s desired terminology, value choices, and interaction styles for non-technical users. (See Section 3.2.)

4.8. **Rule Logic Validation and Testing**

Blaze Advisor provides a comprehensive set of testing and debugging tools:

- Rule Execution Browser—allows you to follow the execution of a rule agent graphically by displaying a tree of possible execution paths and, as the rule agent is executing the project, highlights the current execution path.

- Project Reports—available to help developers identify potential conflicts with a project. The report is generated as linked HTML.

- Performance Testing—to identify potential problems in the execution of the rule-base with respect to performance. The tool provides both hot-spot report to aid identification and execution path to identify the specific process or processes.

- Regression testing—can be performed by storing test inputs in a persistent data source and having the system read them in for debugging and QA.

- Integration with RuleTrack (a tool for capturing and managing business rules that is available from Business Rule Solutions).

- All tools available through the development environment can be used to test and debug a live Rule Server whether it is in a test environment or production.

4.9. **Rule Logic Verification**

Conflict reports analyze potentially conflicting or otherwise invalid rules. You can analyze rules interactively in the IDE or by generating project reports. Testers can select the conflict tests for inclusion in a report and the rule sets for testing. This narrows the amount of output that needs to be perused.

4.10. **Conflicts and Priorities**

Blaze Advisor uses the RETE rules inferencing algorithm, which is capable of both forward and backwards chaining. Each rule has a priority setting where you can make one rule a higher priority over another. Blaze Advisor supports the concept of rule-priority and the models associated for evaluation of layers of rules from general to more specific (or general to layers of exceptions). A combination of rule-repository architecture, rule-flow and rule-templates hides the complexity of these multi-level rules from the business analyst who can directly access the “general” or “exception” rules using Web-pages to make necessary rule changes.

There is no limitation on the number or complexity of special rules or exceptions that can be applied to any product or scenario. All rules whose conditions are met will be executed by default, but you can control the order of execution with priorities, and you can have rules that cause you to exit from the ruleset without executing the rest of the rules.

4.11. **Decision Explanation/Auditing**

Blaze Advisor provides a facility for recording logging and auditing information. Recorded information is configurable to meet the requirements of the specific deployment. Examples of information which can be recorded include (but is not limited to):
• Transaction date
• Execution date
• Decision logic version
• Initiated system or client
• Rules executed
• Explanation of decision
• Data received and collected
• Variables calculated during the decision process

Recorded information can be used to report on the activity of the rules engine and the progress of each decision. The amount of information recorded can also be determined at runtime to increase levels of audit information, potentially to assist in debugging activities. Such information can be used in conjunction with Blaze Advisor’s debugging tools including the execution flow browser where detailed interrogation of a decision process is required (see Sections 4.4 and 4.5.)

5. Simulation

Rules may be tested within the development environment with interactive inputs or by reading test data files. In the production environment, candidate rules may be tested in parallel with production rules through the use of parallel ruleflow branches.

Regression testing can be performed by storing test inputs in a persistent data source and having the system read them in for debugging and QA.


6.1. Document Generation

Blaze Advisor provides rule project, performance, and conflict reports. Project reports show object model components, ruleflows, rule syntax, conflict analysis, cross reference information, comments, and more.

Performance reports analyze rule times and number of calls on rule processing for both locally running rule projects and deployed rule services. Timer options allow data gathering over multiple sessions to evaluate averages, totals, minimums, and maximums. Remote performance tracking allows users to start monitoring the performance of a rule service on the fly without have to take down the server.

Conflict reports analyze potentially conflicting or otherwise invalid rules. Rules can be analyzed on demand in the IDE or can be reported in complete project reports. Testers can select the conflict tests to be included in a report and the rulesets that should be tested. This narrows the amount of output that needs to be perused.

6.2. Document Management

Blaze Advisor does not provide any inherent document management capabilities—when reports are created, the user is prompted for the location to store the files. There is no management of the reports beyond that.

6.3. Interfaces to Third-Party Document Management Systems

No specific interfaces to third-party document management systems are provided; however, integration of Blaze with major document management systems products (e.g., Interwoven) is possible using API calls.
7. Development Environment

Blaze Advisor Builder provides a graphical IDE for building and maintaining rule-based applications. (See Sections 3.1 and 3.2 for more on Blaze Advisor Builder).

Of particular importance are the Blaze Advisor Quick Deployers—wizard-based facilities designed to simplify deployment of Blaze Advisors applications. Quick Deployers automate code generation to create configuration files for immediate deployment in the products of major vendors. Wizards collect deployment information and generate appropriate installation files for any supported application server, ant build files, Web Services, and configuration files. Quick Deployers ask a series of questions that determine the specific configuration of the production environment and the desired operation of the rule service. They can deploy rule services as session beans in an EJB container, packaged and deployed for use with most of the major J2EE application servers (such as WebLogic, WebSphere, Oracle iAS, Sun ONE, and jBoss). If the selected EJB container supports the features, you can choose to run as a Message-Driven Bean, a stateless service, or a stateful (synchronous or asynchronous) service. They can also deploy as a .Net assembly. In addition, Quick Deployers also can generate deployments for processing in Java, Microsoft Transaction Server, JSR-94 rule engine interface compliance, and Web Services.

Quick Deployers are driven by a plug-in architecture, so customers can also create their own customized deployment specific to their environment and have use of that custom deployment repeatedly through the wizard. In addition, new plug-ins to support new environments can be made available to customers immediately instead of waiting for the next release of Blaze Advisor. (For more on Quick Deployers see Section 2.1)

7.1. Programming Languages and Support for Open Standards, Components and Frameworks.

The Blaze Advisor IDE is written in Java; a version written in J# is also available for users wanting a .NET version of the product. A version is also available that can read from and write to COBOL copybooks.

Blaze Advisor is designed to work with any structured data format, including Java and .Net objects, SQL databases, XML documents/schemas, and message queues. Blaze Advisor also provides extensive XPath operations for the interrogation of XML strings and objects. It can receive and return or issue XML strings and objects based on the structure used across the architecture. Native support for database operations is also built in to the rule language.

The Blaze Advisor Rule Server is a 100% Java object conforming to both J2EE (1.2) and Web Services open standards. Supported deployments include J2EE Enterprise Java Bean (EJB), Model Driven Bean (MDB) or Web Service. Other configuration options available include deploying the server as a COM/MTS object, a .NET component with a C/C++ interface, or within EAI layers such as MQ Series, JMS and RMI.

7.2. Software Models and Code Generation

Blaze Advisor Java can generate deployment code for deployment as a Java object, a J2EE session bean, message driven bean, or Web Service. Blaze Advisor .NET allows you to create a .NET deployment, generating client code in C#, J#, or Visual Basic. Blaze Advisor for COBOL can generate an entire rule project as COBOL code, complete with the ability to read from and write to COBOL copybooks.

7.3. Maintenance and Update Support

The repository supports rule service deployment for multiple applications, ensuring consistent enforcement of business policies and procedures across an enterprise. Rule structures, rule services, allowed value controls, and audit trail data (such as “last modified date”, “last modified by”, and
“reason changed”) are stored in the repository, where the rule server monitors changes in the repository and updates affected rule services without interrupting production operations.

The repository also has release management facilities, so that you can save off a set of files as a specific release of the application so that subsequent changes to the files do not affect the released version of the product. Configuration options include saving XML rule files in a LDAP system that supports JNDI, a database that supports JDBC, or saved as flat files. (For more on Blaze Advisor Repository see Section 2.1)

7.4. Integration with Third-Party Application Development Environments
Blaze Advisor Builder is a standalone Java application. It does not integrate with any other ADEs; however, it is possible to use Blaze Advisor’s RMA API calls to integrate the business user view into those environments.

8. Methodology Support
Blaze Advisor does not provide any specific support for any particular (public) business rules development methodology, (however, it does support integration with Rule Track from Business Rules Solutions—an automated tool for recording and organizing business rules). In addition, a plug-in for IBM Rational Rose is also available. Developed jointly with Knowledge Partners Inc. (KPI), the RUP Plug-In for Business Rules lets organizations take advantage of KPI’s standardized business rules discovery and implementation methodology within Rational Rose. Finally, Blaze Advisor product documentation includes a section covering best practices: development, including roles and responsibilities; repository management; security; repository content organization; authoring practices and patterns; rule writing; release management; quality control; deployment and execution; and performance tuning.

9. Integration

9.1. BPM Suite Integration
Blaze Advisor has demonstrated integration with BPM products from Fujitsu Software, FileNet, Lombardi Software and Savvion. Integration for Blaze Advisor is built into the Fujitsu Software product, and similar joint development work is in progress with FileNet.

9.2. Production Systems and Third-Party Applications
Blaze Advisor’s open architecture communicates with external systems and data sources via CORBA, MQ Series messaging, EJB, JDBC, COM, .NET, Web Services, XML, JMS, and other middleware transports such as Tibco Rendezvous.

9.3. Web Services/SOA Deployment
Quick Deployer Wizards supports Web Service generation for the following EJB Containers:

- iPlanet 6.5
- WebLogic 6.1, 7.0 and 8.1
- WebSphere 4.0 and 5.0
- Sun ONE 7.0
- Oracle 9iAS Release 2 (9.0.3)

Generated files for the Web Service are the same as for an EJB, but include additional files to support the container’s Web Service implementation. (i.e., Blaze Advisor supports the same set of capabilities around deploying in an SOA as the application server that a customer chooses to use.)
9.4. **Messaging and Middleware Integration**
Quick Deployer wizards can generate container-specific files for the following EJB containers:

- iPlanet 6.5
- Sun ONE 7.0
- J2EE 1.2.1
- JBoss 2.4, 3.0, and 3.2.2
- Oracle OC4J
- Oracle 9iAS Release 2 (9.0.3)
- WebLogic 6.1, 7.0, and 8.1
- WebSphere 4.0, and 5.0

Blaze Advisor communicates with external systems and data sources via MQ Series messaging, JMS, and other middleware transports such as Tibco Rendezvous.

9.5. **Database Support**
Blaze Advisor supports any relational database vendors with JDBC database drivers, including Oracle, Microsoft, IBM, Informix, and Sybase.

9.6. **Predictive/Analytic Modeling Support**
The ability to generate rules from data for use with Blaze Advisor is facilitated through the use of Fair Isaac’s Model Builder tool (optional). You can also use Model Builder to create optimized scorecard models and decision trees, and for identifying policy weighting factors and thresholds for building more optimized business rules. Blaze Advisor has built-in capabilities to bring in models from Model Builder.

10. **Templates and Frameworks**

10.1. **Horizontal and Vertical Industry Templates, Frameworks, and Pre-Built Rule Bases**
Fair Isaac offers three industry-specific templates. In Insurance, Quick Start for Insurance Underwriting includes rules built around the ACORD Standards and a framework enhanced by lessons learned from Fair Isaac’s successfully deployed BRM applications. Similar offerings are available for Mortgage and Healthcare.

10.2. **Domain-Specific Knowledge Products**
These offerings are primarily made available as consulting engagements where the framework code is included as part of the offering.

10.3. **Domain and Industry-Specific Packaged Rule-Based Applications**
Fair Isaac plans to eventually develop the above-mentioned frameworks into full packaged applications. (Note: Many of Fair Isaac’s existing applications have added Blaze Advisor as a rules engine component, such as Falcon Fraud Manager.)

11. **Administration and Security**
Blaze Advisor provides a built-in authorization manager that allows role and file authorization levels to be set independently. An API for user authentication and authorization is also provided. This facilitates integration with existing security infrastructure systems (i.e. single sign-on, LDAP, etc.), allowing security administration for Blaze Advisor within the existing infrastructure.
12. **Platforms**

The production system is certified to run on hardware from IBM, HP, Dell, Sun, or Windows-based PCs/workstations/servers. The execution engine can be deployed on mainframes, Unix, or PCs, essentially anywhere where it can run in a Java 2 virtual machine. The development environment is a client application that runs on a workstation or PC.

13. **Scalability**

13.1. **Vertical Scalability**

Blaze Advisor provides automatic scalability for thousands of client sessions. Performance scales linearly as the number of CPU’s is increased.

Blaze Advisor is architected to run multiple simultaneous rule execution processes in separate threads. Each thread is dedicated to handling the complete rule service processing for a service requestor (which may be a physical requestor, such as a Web client in an interactive Internet application, or a logical requestor, such as a queued service request in a batch application). Each thread is an instantiation of the Rule Engine processing code held in memory—Fair Isaac refers to these threads as “Rule Agents.” Only the context information is maintained in these agents, which makes their instantiation fast and cost effective. The rules structure is shared by all Rule Agents participating in a rule service. The number of Rule Agents and their system properties are configurable by the system administrator.

Administrators can even make configuration parameters dynamic, based on load factors as seen by an application server. Each Rule Agent that is running and available uses system memory, but adds more parallel load capacity to the overall application. This architecture ensures that Blaze Advisor is infinitely scalable to handle any desired load volumes, simply by adding more processing threads and spanning multiple processors and servers if needed.

13.2. **Horizontal Scalability**

Blaze Advisor provides an enterprise rules repository for native deployment on multiple platforms (Java, .NET, COBOL). A single repository offers the ultimate flexibility for interactions between applications, data, and rules. This architecture makes rule maintenance and deployment far less complex for accessing and executing decision logic in many types of environments with no translation or conversion steps necessary. A single repository accommodates multiple projects simultaneously and delivers a high degree of business logic re-use and extension across the enterprise. This promotes increased accuracy of your decision management policies and lowers development costs and time.

A built-in versioning feature manages historical content of rules and rule services including author, date and additional comments. Rules created or changed in a RMA or in the IDE are stored in the rules repository.

The repository can be used to prevent multiple users from making changes at the same time or overwriting revisions by providing a way for users to check out and check in rule service components. The system allows offline work with synchronization upon re-linking to the shared repository. Both IDE and RMA changes can be tracked with automatic assignment of incremental version numbers and audit information. The collection and reporting of audit information is a configurable element of the version control and security features of the repository, allowing administrators to review alterations to the rulebase, tracking who made the changes, when, what was changed and additional management information as required. Other third-party tools may be used as well. (For more on Blaze Advisor Repository see Section 4.6.)

14.1. Industry Standards Initiatives
Fair Isaac leads the OMG Production Rule Representation submission team, is a supporter of the OMG SBVR, and advises on BPMN/BPDM at OMG. Fair Isaac is also involved in business rules initiatives with the W3C and supports the JCP JSR-94.

14.2. Business Rules Consortiums, Groups and Conferences
Fair Isaac is a primary sponsor of BRForum and EBRC (refer to their web sites), and is active in certain domain standards bodies (eg ACORD, PMML). Conference presentations and support include BPM conferences, DAMA, and Semantic Technology.

15. Pricing
Pricing for Blaze Advisor starts at around $50,000; however, a typical enterprise application is in the $250,000 - $500,000 range. Fair Isaac also offers departmental and enterprise wide licenses above that range.

Blaze Advisor and its optional tools offers one of the most comprehensive set of products currently available for building and managing BRM applications.

16.1. Company Background
Fair Isaac was founded in 1956 and is a leading provider of analytic development tools and applications. The company markets its various offerings as “Enterprise Decision Management” products. These products include predictive models; customer data management and data analysis services; and rule-based decision management systems. Fair Isaac also provides tools and services that help businesses develop and deploy their own systems for enterprise decision management.

Many of the leading government agencies, banks and credit card issuers rely on Fair Isaac solutions, as do insurers, retailers, telecommunications providers, and healthcare organizations.

Fair Isaac serves global markets through offices in nine countries and has approximately 3,075 employees. Of these approximately 130 are in BRM (100 product development and 36 in product sales. The company also has about 230 employees in consulting and over 100 in support.

16.2. Positioning
In the BPM space, the industries most represented in Fair Isaac's customer base are insurance and finance, followed by government, healthcare, and telecommunications; however, the company has customers across a wide variety of industries.

16.3. Educational Training/Product Training
Both on-site and scheduled training is offered for Blaze Advisor directly at the fundamental and advanced levels. Training is live, with instructors available to assist students. Students have access to training computers and hardcopy training guides. Blaze Advisor software also comes with numerous code examples and tutorials in HTML descriptions and code files.

16.4. Product Support
A dedicated team of engineers provide technical/product assistance for clients from solution deployment through production. Product Support teams provide clients with a single Fair Isaac point of contact. Support teams also have multilingual skills. To better assist in phone conversations with non-English speaking clients, Fair Isaac may engage translation services during the call (currently available through the AT&T Language Line Service).
Support engineers are trained in problem analysis. They also retain ownership of the incident and involve developers or Professional Services specialists as needed in the resolution process. Inquiries are logged, tracked, resolved and closed only with client agreement.

Various levels of support for Blaze Advisor are offered, including general (within specific business hours), 24-hour services (based on maintenance contract) and “Temporary After Hours” support available as an add-on service. In addition, Fair Isaac provides online support.

### 16.5. Consulting Support

Fair Isaac Professional Services is a stand-alone business segment with over 400 professionals delivering solutions to organizations worldwide, and with personnel spanning 16 offices in 10 countries. It is currently engaged on nearly 400 projects and has delivered solutions for 46 of the world’s top 50 financial services institutions.

Projects are guided and delivered within the context of a structured delivery methodology (The Fair Isaac Business Integration Methodology). This methodology consists of a series of detailed steps, tools, and templates encapsulated within a framework that follows an industry standard approach to addressing core project management processes. All projects undertaken are also supported by a centralized Program Management Office, which is responsible for ensuring adherence to methodology principles, facilitating project staffing, and consolidating weekly reporting of project status.

### 17. Case Study: California Department of Motor Vehicles (DMV) Revolutionizes Vehicle Registration Using Blaze

#### 17.1. Context

The California Department of Motor Vehicles (DMV) is responsible for collecting approximately US $4.1 billion annually in vehicle registration fees. Centralized computer systems in Sacramento, California 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 autos, trucks, motorcycles, vessels and other vehicle types.

#### 17.2. The Challenge

In 2000, the DMV realized that they needed to update and converge the two separate vehicle fee systems, and move to a modern system that would meet the state’s strategic realignment towards eGovernment with future public access via the Internet. The DMV began the Vehicle Registration Fee Computation (VR Fee Comp) project to consolidate and streamline the fee generation processes that were implemented across two different computing platforms, two computing languages and two distinct systems:

The DMV Automated fee system (DMVA) is deployed on servers at each of the DMV’s 167 field offices and its headquarters. It processes customer initiated vehicle registration transactions in real time in IBM’s proprietary Event Driven Language (EDL).

The DMV Batch fee (DMVB) systems are deployed on mainframe computers at the Teale Data Center, where renewal notices are generated and the Remittance Processing system handles most of the “DMV by mail” COBOL.

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.

“Most of the computer programs composing the DMV legacy fee systems have been used for more than three decades with constant updates and workarounds grafted into the code by multiple people over the years,” says Jerrianne Seitz, Data Processing Manager at the California DMV. “Making changes to the system was no easy task as one change could introduce numerous side effects.”
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. Even minor changes required extensive analysis and programming efforts by the legacy system’s development staff. There also was a real and practical need to update the systems as they had met their physical limitation; the DMV reached a point where they couldn't add more statements to the system.

17.3. The Solution

After discovering all of the places where Vehicle Fee calculation rules had been programmed, the VR Fee Comp team's analysis uncovered that instead of simply recoding the existing systems, they needed to consider newer technologies and methodologies for managing their business processes.

The team recognized that BRM software 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, as often a limited amount of programming resources resulted in latency to system updates. In addition to faster system changes, a business rule system could enforce the greatest levels of compliance with the legislative mandates.

Any BRM system would have to be usable by non-technical analysts responsible for overseeing legislative compliance. It would also require an intuitive 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 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 in place throughout the DMV infrastructure. According to Seitz, “Many people within the organization felt our rules were so complex that a third-party software package wouldn’t be able to handle the task. We needed to evaluate and demonstrate functionality and performance in any solution we planned to bring in.”

Upon delving into the project, however, the DMV realized the necessity existed to also conduct a best practice review of their business policies and practice methodologies. The resulting clearer understanding of their decision infrastructure proved to be an invaluable process. They examined a business rules methodology, created a blueprint of their business processes, and looked at normalizing and understanding the terms and policies that were already in place. It was necessary to look at existing business policies, the legislation that affects them, and how a consolidation of look-ups and interrelations could be architected to simplify future updates.

The DMV’s Vehicle Registration team reengineered a vehicle registration fee system that would work in a manner that would be transparent to end users. The revised solution resulted in a Blaze Advisor, IBM WebSphere application server, and a J2EE solution operating on a mainframe.

The development team, however, faced the challenge of having little or no experience in Java software development. Most of the team was experience with COBOL and Assembler programming. Fair Isaac Professional Services offered Java software training and on-site assistance during the pre-project implementation phase. The DMV was able to move smoothly from design and structure of conceptual rules to development and execution without fear of overlooking or misinterpreting key decision factors.

Leveraging the power of Blaze Advisor and its RMAs, the DMV successfully gives the non-technical analysts responsible for overseeing legislative compliance the ability to ensure proper implementation of the policy rules across the DMV vehicle registration fee systems without having to become programmers. They also defined templates for certain types of rules that business analysts should be able to create on their own. These maintenance facilities are available through automatically generated Web pages that eliminate formal rule syntax and custom editing environments.
17.4. Result/Benefits

The DMV has successfully developed a revolutionary solution that boasts a Blaze Advisor business rules system as the cornerstone of its vehicle registration fee systems.

The first phase of the project implemented the new technical environment and fee business rules for vessels (i.e., boats and other watercraft). Vessels were selected for the first implementation because it represented a lower volume transaction count and therefore a lower impact to DMV field office operations. This phase was completed and successfully released to production in March 2003.

The second phase of the project implemented the new fee business rules supporting expedited legislation for the Vehicle Registration Penalty (Reg penalty) fees which could not be implemented in the legacy fee system due to its limitations. This phase went live across the DMVA in April 2003. The Reg penalty fees system processes 60,000 to 75,000 business transactions per day.

The third phase of the VR Fee Comp project went into production in February 2005. Phase three implemented the rules and fees for autos, commercial vehicles, trailers, motorcycles and off-highway vehicles. It required over 2,000 rules and numerous rule-sets, rule flows, and 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. With an efficient use of intermediate rules, the DMV proudly reduced the overall ruleset that results in less coding. It also provides better visibility into the rules that are in effect, an understanding of the levels of business change and how changes to the rules will interact and affect subsequent business processes. The DMV’s new system leverages business rules to support mandated and changing legislative laws.

Thus far the fee comp team has focused on updating the DMVA aspect of the project. With the deployment of the fourth phase of the project to production the team moved towards the consolidation of DMVA and DMVB. The DMV’s new system leverages business rules to support the mandated and changing legislative laws.

Coupled with the IBM’s WebSphere Applications Server, Blaze Advisor positions the DMV to make future system enhancements more easily and to add centralized and consistent rules to delivery channels such as self-service websites and telephone response systems. “The implementation of the VR Fee Calculator on the Internet further supports our strategic business and IT goals by improving the integrity and quality of its products and services to customers,” added Seitz. “The new rules powered system and resulting business practices should continue to lower our operating costs by decreasing the number of telephone calls and transactions in the CA DMV Field Offices and Telephone Service Centers.” The system went live on June 30, 2005 as the main feature on their webpage and is a huge success. Now citizens can go see how much it would cost to register a vehicle or boat in California before purchasing it and will be able to see how much their registration fees will be in future years. To see the application run live, visit [https://vrir.dmv.ca.gov/feecalculatorweb/selectcalculations.do](https://vrir.dmv.ca.gov/feecalculatorweb/selectcalculations.do).

18. Company Offices
USA
Corporate Headquarters
901 Marquette Avenue
Suite 3200
Minneapolis, MN 55402 USA
Tel: +1 (612) 758 5200
E-mail: info@fairisaac.com

Brazil
Rua Luigi Galvani, 200 - 8º Andar
Brooklin Novo
04575-020 São Paulo – SP
Brasil
Tel: +55 (11) 5501 9222
Fax: +55 (11) 5506 7703
Email: Lac_Info@Fairisaac.Com

Canada
The Exchange Tower
130 King Street West, Suite 1800
Toronto, Ontario, M5X 1E3, Canada
Tel: +1 (416) 865 3390
Fax: +1 (416) 865 3379
Email: Info@Fairisaac.Com

Germany
Fair Isaac Corporation
Maximilianstr. 35
D-80539 München
Deutschland
E-mail: emeainfo@fairisaac.com

India
Fair Isaac India Software Pvt. Ltd.
Titanium Building
135 Airport Road
Kodihalli
Bangalore 560017
India
Phone: +91 80 51371700

Japan
Meiji Yasuda Seimei Bldg. 5F
2-1-1, Marunouchi, Chiyoda-ku
Tokyo 100-0005 Japan
Tel: +81 3 6212 2770
Fax: +81 3 6212 2769
E-mail: Info@FairIsaac.Co.Jp

Malaysia
Unit 902C, Level 9, Uptown 5
5 Jalan SS21/39
Damansara Uptown
47400 Petaling Jaya
Selangor Darul Ehsan
Kuala Lumpur,
Malaysia
Tel: +60 3 7727 5413
Fax: +60 3 7727 5244

Singapore
No. 8 Shenton Way
#40-01 Temasek Tower
Singapore 068811
Tel: +65 6323 9700
Fax: +65 6323 3449

Spain
Fair Isaac
Calle Nuria 36, Planta 2
28034 Madrid, Spain
Tel: +34 917355020
Fax: +34 913721534
E-mail: Madrid_Info@Fairisaac.Com

United Kingdom
5th Floor
Cottons Centre
Hays Lane
London SE1 2QP
United Kingdom
Tel: +44 0207 403 1333
Fax: +44 0207 403 8981
E-mail: Emeainfo@FairIsaac.Com
New London Bridge House
25 London Bridge Street, 16th Floor
London SE1 9SG
United Kingdom
Tel: +44 0207 403 1333
Fax: +44 0207 403 8981
E-mail: Emeainfo@FairIsaac.Com
Fair Isaac House
International Square
Starley Way
Birmingham, B37 7GN
United Kingdom
Tel: +44 (0870) 420 3600
Fax: +44 (0870) 420 3601
E-mail: EMAinfo@fairisaac.com
1. **Product Overview**

ILOG automates business policies in custom and composite business applications through their business rule management system (BRMS) for the Java and .NET platforms. Today, ILOG strategy for the .Net and Java platform is to provide tools that fully leverage their respective platform while integrating toward full rule inter-operability and functional equivalence between the two platforms. This report focuses primarily on JRules features for building and managing rules on the Java platform.

ILOG JRules provides a comprehensive set of tools designed for both the business and technical teams. The ability to focus on both business and IT positions ILOG's delivery of BRMS without compromise. JRules 6 is a new generation BRMS that meets the needs of both teams with tools and features designed specifically for their particular needs and way of working. Table 1 provides an overview of JRules 6 and the key capabilities.

JRules empowers the business team with the complete functionality called Rule Care. Rule Care combines a deep, scalable rule repository with a thin client rule management application designed specifically for the needs of policy managers engaged in rule authoring, management and maintenance. The thin client web application gives the policy manager a complete, wizard-based rule editing, management, and deployment facility. Rule Care also provides the business team with the ability to simulate changes in business rules and perform what-if analysis to determine the potential impact of rule changes on business operations.

Rule Tech is ILOG’s tech-centric rule environment. Rule Tech provides the productivity functionality for the technical team including an extensive Java library for defining and extending rule execution and management environments. Rule Tech delivers rule application development without compromise: a full feature set for realizing the promise of rule applications capabilities, fully integrated with developers’ existing tools, methodologies and skills. JRules is designed to integrate with any Eclipse-based integrated development environment (IDE).

FullCircle BRM provides the functionality for the business policy manager to manage rules through the entire business rule lifecycle. This includes the ability to reference, share and reuse rules among rule authors with different professional backgrounds and languages. ILOG has achieved this by decoupling the business models and technical execution models, while still providing the ability to share and reuse information between the two models.

To assist in managing these different authoring and development communities, ILOG JRules provides automated (“on-demand”) synchronization of rules (and rule artifacts) associated with developing and managing rule projects across the different tool sets as well as across the enterprise. These features enable organizations to implement a business rule management cycle that is independent of application development cycles so that they can implement changes to rules as business needs change, as opposed to on an IT schedule.

Finally, ILOG’s tools are backed up by the company’s extensive maintenance services and technical support offerings.
Table 1. Overview of ILOG JRules Key Capabilities

<table>
<thead>
<tr>
<th>Rule Care</th>
<th>Rule Care empowers the business team to:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Learn business rule management quickly</td>
</tr>
<tr>
<td></td>
<td>• Author, maintain and deploy business rules safely, confidently and directly</td>
</tr>
<tr>
<td></td>
<td>• Satisfy regulatory and business imperatives for security, traceability and auditability of policy changes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule Tech</th>
<th>Rule Tech allows tech teams to:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Apply corporate development standards, best practices and processes to rule applications</td>
</tr>
<tr>
<td></td>
<td>• Master BRM quickly, working with familiar techniques</td>
</tr>
<tr>
<td></td>
<td>• Integrate BRMS naturally, natively within current and future technical infrastructure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FullCircle BRM</th>
<th>With FullCircle BRM, our customers can:</th>
</tr>
</thead>
<tbody>
<tr>
<td>FullCircle BRM</td>
<td>• Share and reuse rules among rule authors with different professional backgrounds and languages</td>
</tr>
<tr>
<td>FullCircle BRM</td>
<td>• Work on several release cycles simultaneously, maintaining the release in production while constructing the next release in development</td>
</tr>
<tr>
<td>FullCircle BRM</td>
<td>• Control the rule management life cycle</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Continue Performance Engineering</th>
<th>ILOG’s performance engineering makes it possible to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continue Performance Engineering</td>
<td>• Process large, complex rule sets against large amounts of data efficiently</td>
</tr>
<tr>
<td>Continue Performance Engineering</td>
<td>• Execute large decision tables faster than source code</td>
</tr>
</tbody>
</table>

2. Product Architecture

2.1. Architecture Overview

Figure 1 provides an overview of the ILOG JRules platform, which consists of four main modules: Rule Studio, Rule Team Server, Rule Execution Server, and Rule Scenario Manager mapped to the JRules 6 key capabilities outlined in the above section.

![Figure 1 - ILOG JRules 6 Capabilities and Modules](image-url)


**Rule Studio – Rule Tech Developer Productivity Tool**

ILOG JRules Rule Studio provides a set of tools for IT development teams (i.e., programmers) that integrate as plug-ins into the Open Source Community’s Eclipse, IBM’s Rational Application Developer (RAD), and IBM’s Rational Software Architect (RSA) IDEs. Rule Studio supports deploying and debugging of rule sets to the JRules Rule Execution Server and enables collaboration with business rule authors through synchronization with Rule Team Server. Rule Studio is a desktop tool that communicates with the server-based modules via pre-defined interfaces.

Rule Studio is J2SE- and J2EE-compliant with interfaces to application servers from IBM, BEA, JBoss, Oracle and Apache. Rule Studio components for synchronous, asynchronous and web-based invocation of business rules and includes a web administration console. It also integrates with JMX-compatible systems management tools such as IBM Tivoli and HP OpenView.

**Rule Team Server – Rule Care Empowerment for Business Policy Managers**

ILOG JRules Rule Team Server is a highly scalable rule management server and repository with a collaborative web environment for business policy managers and business analysts to author, manage, validate and deploy business rules. The Rule Team Server repository is designed to be an enterprise repository, storing in a single instance multiple independent or dependent rule projects and their histories. The RTS repository supports automatic, rule-level locking, as well as user-managed persistent locks. The RTS repository scales to dozens of users working on the same or different projects, and hundreds of thousands of individual rule artifacts.

**Rule Execution Server – Managed, Monitored, High Performance Rule Execution for the Enterprise**

JRules 6 Rule Execution Server wraps the high performance ILOG rule engine into a scalable, manageable and monitoring service that provides for business rule execution for all server based applications, service-oriented architectures and embedded rule applications. ILOG Rule Execution Server integrates into J2SE- and J2EE-compliant platforms. It is designed for deploying business rule services to the Web and application servers from IBM, BEA, JBoss, Oracle, and Apache.

Rule Execution Server includes components for synchronous, asynchronous and Web-Service-based invocation of business rules. It communicates with the Rule Studio and Rule Team Server modules to support business rule deployment by both IT developers and business policy managers, and is managed via a Web administration console or any JMX-compliant commercial systems management tool.

**Rule Scenario Manager – Rule Test and Simulation for Developers and Business Policy Managers**

JRules 6 provides Rule Scenario Manager as part of both Rule Tech and Rule Care, to provide a common environment for testing the function of rules for both developers working in the application development context and policy managers who author and validate rules as part of the business rule lifecycle. Rule Scenario Manager is a modular, extensible framework for all aspects of rule testing and simulation. Rule Scenario Manager provides facilities for simulation in which the results from a modified “candidate” rule set executed against a suite of test cases, is compared to a baseline rule set running against the same data using customer-specified key performance indicators (KPIs).

**2.2. Business Rules Engine (BRE)**

JRules uses an inference-based BRE that offers a choice of execution modes used during rule engine operation.
Different kinds of rule based decisions require different approaches to rule execution. For decisions requiring inferencing, such as underwriting, computations or clinical decision recommendations, the JRules engine offers RetePlus, a third-generation implementation of the Rete algorithm that delivers industry leading performance on up to hundreds of objects and thousands of rules. Rete is widely used because of its ability to handle large numbers of rules, and its performance in handling rules that reference dynamically changing data.

For simpler decisions, such as validation, retail promotions and compliance monitoring, the engine offers sequential execution. The sequential execution mode utilizes dynamic rules compilation that can improve the speed of rule processing. (The performance of the engine will improve using the sequential execution mode if it is provided with a large rule set made of basic but test-intensive rules with static priorities.)

For sequential applications with significant condition sharing between rules, the engine’s globally optimized Fastpath technology delivers execution throughput up to 15 times that of standard sequential execution and up to 10 times the speed of hand-coded Java. The Fastpath execution mode provides augmented sequential compilation and execution of a rule project. Fastpath may significantly improve the runtime performance of the execution of decision tables and particularly decision trees. (The Fastpath execution mode is a variant of the sequential mode. It is designed to further optimize the execution of the compliance and validation, which constitute a substantial part of business rules.)

ILOG JRules engines supports forward chaining inferencing. Backward chaining can be simulated using JRules’ forward chaining inference engine and Truth Maintenance System (TMS).

The JRules Java rule engine provides a 100% Java class library that includes an API that can be used in an applet. The BRE can be deployed as a directly invoked service using the engine’s API, or as a remote or local service in a J2EE container. The JRules engine supports local and remote debugging on a running set of rules; it also exposes an interface that allows remote profiling on a running set of rules.

The JRules BRE is tightly integrated with the ILOG JRules Rule Execution Server that provides a pre-packaged J2EE application that exposes the engine via a standard J2EE Connector Architecture (J2C) and EJB interface. Using the ILOG JRules Execution Server, architects can expose any rule-based decision-making process as a Web Service or via EJB session or message bean interfaces.

### 2.2.1. Rules Management Approach

ILOG JRules provides a rules management approach through FullCircle BRM. FullCircle BRM is the processes and tools built into JRules 6 that enable full-lifecycle enterprise business rule management from rule inception through retirement. Business policy managers can work on several release cycles simultaneously while maintaining the release in production and preparing for the next major or minor release. This unique functionality enables business policy managers to continue designing their strategic business policy strategies while continuing to work on their tactical business policy implementations.

JRules 6 RuleSync technology provides the ability to synchronize the business vocabulary and business rules between the business user environment (Rule Team Server Repository) and the development environment (Rule Studio Source Code Control Repository). This unique feature allows the business and technical teams to work independently through separate environments but kept in synch on demand across multiple projects and project phases.
The ILOG Rule Team Server repository can store and manage multiple rule projects, independent of deployment and execution. Each project is associated with a deployer that specifies which rules should be extracted and deployed for each application. Rule Extractors provide query-based extraction of rules for deployment. Further flexibility is available at run time using Ruleset Interceptors and Dynamic Rule Selection capabilities, which select the applicable rules from a rule set based on incoming data.

### 2.2.2. Enterprise Data Interoperability

ILOG JRules can be integrated with most Data Access Object (DAO) products with little or no additional infrastructure code. If performance is not adequate, the data to be processed by the rule engine can be stored in Data Transfer Objects prior to adding them to the rule engine.

Some applications may require more advanced solutions compared to either CMP or JDO to allow persistence strategies to be swapped with close to zero code change and persistence to be evolved wholly independent from business logic. To help get your data out of your data stores and into the rule engine and then persist the results, design patterns can be used. The core patterns of interest described here are data access and distribution patterns.

Patterns are proven, reusable, and expressive. They reflect the experience, knowledge and insights of experts in fields where there are no simple “one size fits all” answers. They provide a ready-made solution that can be adapted to different problems as necessary. Finally, they provide a common vocabulary of solutions that can express large solutions concisely.

The most standard data that you can run your rules on is a Java object model. However, ILOG JRules lets you integrate rules with XML and Web Service data dynamically. Rules run directly on the data without converting it to Java objects, using a reflection mechanism. You can also seamlessly integrate database data or access third party data.

### 2.2.3. Maintaining Data Quality

Designers can define whole rule packages dedicated to input data validation. It is a common practice to begin a Ruleflow with a data validation step, where a specific set of rules is applied to examine the input data for correctness, completeness and self-consistency.

### 3. Interfaces and Ease of Use

The ILOG JRules platform provides a separate environment for developers (Rule Studio), business users (Rule Team Server), and systems operators (Rule Execution Server), with each designed to provide a set of skill-specific, role-appropriate tools to cover the full life cycle of a business rules project or application.

#### 3.1. Development Environment: Rule Studio

IT developers (i.e., programmers) interact with JRules using Rule Studio, with support provided by Rule Scenario Manager and Rule Execution Server for testing and deployment. Basically, the Rule Studio development environment adds ILOG JRules’ rule modeling, debugging and deployment tools inside the Eclipse IDE.

Rule Studio integration with Eclipse provides a familiar environment for Java developers to construct rule-based applications. Developers use Rule Studio to both write rules and organize them into project-based “packages”, which can be shared across projects.

Key features of Rule Studio include:

- Eclipse plug-ins for rule development, debugging and deployment
- Cross-debugging between business rules and Java application code
Developers have two primary tools for expressing rules in Rule Studio. The Business Rule Text Editor enables users to type rules in the BAL language using ILOG’s IntelliRule code-completion system. The Technical Rule Text Editor does the same thing using the ILOG IRL language. (See Section 2.2.1 for more on these rules languages.)

Figure 2 provides a screen shot of Rule Studio’s Business Rule Text Editor within Eclipse. Writing rules in Rule Studio is easy, as it provides guided assistance to users with auto-correction in rule editing, and the text format allows easy copy/paste and other keyboard operations. In addition, code generation wizards enable users to quickly execute rules within Eclipse or to integrate execution within existing applications.

Developers can share rule projects created in Rule Studio with other developers through any third-party source-code control repository supported by the Eclipse IDE. Additionally, developers can share rule artifacts from Rule Studio with business policy managers by publishing a rule project to Rule Team Server, while RuleSync, a new feature in JRules 6, enables simultaneous changes to be managed across Rule Studio and Rule Team Server.

### 3.2. End-User/Rule Management Environment: Rule Team Server

Rule Team Server provides business rule authoring, analysis, simulation and maintenance capabilities for business policy teams and other business users. As a rule management environment, Rule Team
Server is easy to use due to its Web interface and natural language style rule language. In addition, you can tailor Rule Team Server’s features and functionality to accommodate various levels of end-user skills and responsibilities based on permissions.

Figure 3 provides a screen shot of the Business Rule Guided Editor—the primary tool used by business policy managers to interact with Rule Team Server. The Guided Editor provides a point-and-click editor that allows users to build rules by selecting items from drop-down lists and by entering text or numbers into specified fields. As its name implies, the Guided Editor also helps prevent mistakes and assists the user by offering relevant options based on previous choices.

Rule Team Server has a number of features that make the tool intuitive as well as productive, and which help to shield non-technical end users from the complexity of large, overwhelming environments.

**On-line Help and Automated Error Detection**

On-line help capabilities include static rule analysis that provides error detection and helps the user correct his or her mistakes.

**Scenario-based Testing and Simulation**

Scenario-based business policy testing and simulation tools assist business users in assessing the correctness and effectiveness of business rules they author prior to deploying them to the production environment.
Smart Views
Similar to Microsoft Outlook search folders, Smart Views permit users to organize, filter and browse rules created by policy managers according to any combination of rule properties. Smart Views also allows policy managers to define custom views into the rule repository that match their perspective or area of responsibility within the business team.

Rule Flows
Users can inspect rule flows, which show the order of rule set for a particular domain (e.g., loan validation) in order to assess their operation.

Audit Support
Rule Team Server provides provisions for safeguarding the integrity of rules projects and production rules environments. For example, when a rule is changed, both its prior state and new state are saved. Prior versions are accessible for browsing or reporting, and baselines (complete snapshots of earlier project states) can be saved for later browsing or modification. Finally, project state can be rolled back to a prior saved state.

Business Object Model Filtering
“Categories” let users filter the expressions available to the author of a particular rule, so that only elements relevant to that rule are presented. Applications requiring very large models can define multiple rule types or templates, each of which present concise, targeted subsets of the whole model to the rule author.

Query Access of Rules through Database Backend
Policy managers can query all Rule Team Server artifacts across projects enabling enterprise-wide business rule management. The Query feature uses relational database queries for fast, scalable access to 100,000s of rules.

Rule Execution Server
Rule Execution Server integrates with the Rule Studio and Rule Team Server tools to support business rule deployment and management by both IT developers and business policy managers. It is managed using a Web administration console, as shown in Figure 4, which allows system administrators to manage deployed rule sets and monitor run time execution.

4. Business Rules Expression
4.1. Defining Rules and Rule Sets
A fundamental concept of ILOG technology is the provision of a unique toolkit for each major user role that provides a set of tools specific to the BRM activities and which is appropriate to the skills and knowledge associated with the role.

To assist in managing these different authoring and development communities, ILOG JRules provides facilities designed to automate the synchronization of rules and rule assets associated with developing and managing rule projects across the different tool sets as well as across the organization. These capabilities enable organizations to implement a business rule management cycle that is independent of application development cycles so that changes to rules are implemented as business needs change, as opposed to on an IT schedule.

The ILOG business rules languages are designed to support this role-based user concept:

- Business analysts and other non-technical users can use familiar business terms to create rules
- IT programmers can author rules using Java-or XML-like syntax
Organizations can even create new rule authoring languages to support specific types of users or development roles.

The problem domain that business rules manipulate is expressed with ILOG JRules in the form of a business object model (BOM). Similar to a UML object model, the BOM is a structured representation of all the concepts, data elements, and relationships in the problem domain. Within an ILOG JRules project, the object model is represented with Java classes or an XML schema representing an executable version of the model.

When it comes to defining rules and rules sets to implement the BOM, business rule artifacts all use the BAL. Whether creating or editing business rule artifacts, they are, in fact made up of combinations of modifiable building blocks (objects), which represent vocabulary elements, rule set parameters and variables, and BAL constructs and operators.

4.2. Creating and Maintaining Rules

ILOG JRules directly supports users in creating decision tables and decision trees as well as other graphical and tabular methods for creating rules, as discussed below. JRules does not provide direct features for creating continuous scorecards; however, it is common practice to create discrete scorecards using JRules’ decision tables or decision trees.

4.3. Rule Templates

ILOG JRules includes tools for creating and editing templates for business rules, decision tables, and decision trees. JRules’ template capabilities include:

♦ Creating simple, form-like business rules suitable for untrained users
♦ Creating many business rules with the same form
♦ Restricting the type of business rules that end users can write or modify
An example of the use of templates is found in Rule Team Server, which provides guidance to users in the form of templates consisting of preformatted rules, which the user instantiates simply by filling in values (See Section 3.2).

4.4. Rule Syntax Checking
ILOG JRules provides various tools and techniques for rule syntax checking—including automated capabilities and reports.

A ruleset can be automatically built when saved or manually built using the appropriate menu selection in Rule Studio. The build operation merges all the rules contained in the rule project to produce a single executable rule set file. During this operation, Rule Studio automatically detects problems related to the execution of the rule set and reports them in the Problems View and Rule Explorer tools.

Reports are also available, including identified errors and warnings. Warnings are generated for valid parts of the generated rule set file that contains awkward or deprecated syntax, and errors are for syntax or structural defects contained in the generated rule set.

4.5. Rule Testing and Tracing
When executing a rule set in JRules, a Debugger tool is automatically activated, enabling users to test rules in several ways. Rule Studio provides tools for inspecting the execution of the rules and the state of the engine, and to set breakpoints on classes, objects, rules, decision tables and trees, and ruleflows. Rule Studio also lets you debug both IRL code and Java code in a project.

Debug views are used for testing and debugging the execution of rules. Debugger monitors business rule execution by, for example, checking the rule set, connecting to a rule engine, sending the rule set to the engine, resetting the engine, and firing all the rules.

You can also trace rule execution using the Rule Execution Server. The execution trace is amassed during the entire process of rule set execution. Information reported by the execution trace includes the following:

♦ Time taken to complete rule set execution
♦ Rules fired
♦ Tasks executed
♦ Number of rules fired
♦ Number of tasks executed

4.6. Multi-User/Team Development
Rule Team Server is a rule management server and repository whose features include a collaborative Web environment for business analysts and policy managers to author, manage, validate, and deploy business rules.

The ILOG repository provides a common location for storing rules and rule metadata. It is based on standards like MOF, XMI and JMI, providing a foundation for metadata exchange with other repositories and modeling tools. It stores rules and their properties, including name, creation date, last modification date and associated documentation. Various facilities are provided to allow for efficient rules editing and collaboration, including:

♦ Project management—The repository can store and manage multiple rule projects, independent of deployment and execution
♦ Concurrent access—a locking mechanism supports simultaneous access to rules
♦ Versioning—allows storage and management of multiple versions of business rules:
♦ New versions of business rules can be created when policy changes
Previous versions of business rules can be viewed to analyze policy changes

Previous versions of business rules can be selected or reactivated

Source code control integration—API and repository enhancements support use of third-party source-code control tools for package-level configuration

Rule Team Server provides tools for managing rules stored in the repository. Users can query and version rules, track their status and effectiveness, view change history, and control their access.

**Query Support**

Users can query the repository to find and view specific rules. A query editor locates rules implemented in production, or related rules that will be affected by a policy change. Users can also write queries to make global changes to sets of related rules.

**Customizable Logging**

A history feature allows users to track changes to rules and other repository elements. Users can specify which repository operations are logged, such as creating or deleting a rule, changing the value of a rule property, or creating a new version of a rule.

**Permission Manager**

A permission manager lets organizations control access to the repository. Administrators can assign specific roles and permissions to developers and business users to restrict access and modification privileges. (See Section 11 for more on security and administration.)

### 4.7. Rules Management and Maintenance

ILOG JRules provides rules maintenance and management, capabilities through Rule Care. Rule Care provides the features and environment for business policy teams that covers business rule authoring, analysis, testing and simulation and maintenance tools, including:

- A Web-based, guided rule management user interface in Rule Team Server
- Rule-level locking to support concurrent management by large numbers of distributed policy managers
- Audit support in Rule Team Server:
  - When a rule is changed both its prior state and new state are saved
  - Prior versions are accessible for browsing or reporting
- Baselines (complete snapshots of earlier project state) can be saved for later browsing or modification
- Project state can be rolled back to a prior saved state
- Scenario-based business policy testing and simulation tools to assist business users in assessing correctness and effectiveness of business rules they author prior to deployment
- Support for rule inheritance and overrides to simplify automation of complex policies

### 4.8. Rule Logic Validation and Testing

Rule Scenario Manager enhances JRules 6 with comprehensive scenario-based testing and simulation capabilities. It provides an integrated environment for business policy managers to verify the correctness of rules and simulate changes in business policy. (See Section 5 for more on Rule Scenario Manager’s simulation capabilities.)

### 4.9. Rule Logic Verification

ILOG JRules offers various tools to automate the testing of the logic of rule sets for anomalies and conflicts (see Sections 4.5, 4.8, 4.10 and 5).
4.10. Conflicts and Priorities
ILOG JRules offers a number of techniques for handling or averting conflicts among rules at execution time. JRules stores rules whose patterns are all matched in its agenda. Any rule that enters the agenda is a rule instance. In the agenda, rule instances are ordered according to three criteria that determine which rule should be fired first, including:

♦ Refraction—a rule instance that has been fired cannot be reinserted into the agenda if no new fact has occurred, that is, if none of the objects matched by the rule are modified, or if no new object is matched by the rule.

♦ Priority—the second criterion taken into account to decide at which position a rule instance should be placed in the agenda is the rule priority.

♦ Recency—if two rule instances have the same priority, the rule that matches the most recent object (the most recently inserted, modified, or retracted object) will be fired first.

Priority and recency are used to resolve conflicts when several rule instances are candidates for firing at the same time. The priority of a rule can be static or dynamic.

A static priority may be used to alter the sequence of rule execution among different rules. Static priorities are any integer between -10^9 and +10^9; the larger the number, the higher the firing priority of the rule.

A dynamic priority may be used to alter the order of execution between several instances of the same rule, when several rule instances are in the agenda. Dynamic priorities are expressions whose value depends on variables bound in the condition part of a rule.

4.11. Decision Explanation/Auditing
ILOG JRules supports decision explanation and audit trails of rule execution through the execution report. The execution report is generated by the Rule Scenario Manager module working in conjunction with the Rule Execution Server. The report captures and filters information from the execution trace. Information reported by the execution trace includes:

♦ Time taken to complete the rule set execution
♦ Rules fired
♦ Tasks executed
♦ Number of rules fired
♦ Number of tasks executed

5. Simulation
Simulation and testing in JRules is provided by Rule Scenario Manager. Rule Scenario Manager is an add-on module that provides an environment to verify the correctness of rules and simulate changes in business policy. It is also designed so that organizations can customize it to support their particular data stores, deployment processes and reporting requirements.

Rule Scenario Manager lets business policy managers create reusable business scenarios and apply them against their business rules to perform what-if analysis by simulating changes to the rules or the data being processed by the rules. This includes conducting tests that invoke the rules engine to apply reference input and output data used in tests to make sure that authored business rules don’t violate defined business criteria before deployment.

Rule Scenario Manager applies KPIs that allow users to visualize the impact of changing rules or input data sets. Execution reports and validation templates are used to verify results. This feature allows policy managers to write validation tests using easy-to-use Web forms.
Execution reports can be generated in text or Excel formats. Users can view a summary of the validation history for a rule set, as well as the detailed results for a given execution test, including failures and errors.

Rule Scenario Manager makes rule testing accessible even to non-technical business users; users can interact with the scenario manager through JUnit, ANT, Eclipse or a simple policy manager Web interface. Analysts and statisticians can run batch simulation runs. In addition, simulation scenarios can run multiple rule sets against the same input data sets.

Rule Scenario Manager integrates into existing client applications via a service-based API. XML configuration files and schemas allow developers to configure the framework.

Finally, execution reports can be stored in a data warehouse, allowing organizations to analyze and manipulate rule set execution results using standard reporting and business intelligence tools.


6.1. **Document Generation**

JRules' main modules (i.e., Rule Team Server, Rule Execution Server and Rule Scenario Manager) provide various reporting capabilities.

Rule Team Server lets you generate business rule reports and consistency checking reports. The former provide detailed information on the artifacts in rule packages. The latter are used to detect and troubleshoot situations when a conflict in business rules occurs.

**Business Rule Report**

A business rule report contains detailed information about the content of a rule package, including:

- Ruleset parameters
- Rule artifacts (e.g., business rules, technical rules, functions, decision tables, trees, etc.)
- Ruleflows
- Packages
- Functions
- Variables

You can generate reports for the following repository elements:

- A BOM
- A package or sub-package containing business rules, decision tables, decision trees or a set of query results, etc.

**Consistency Checking Report**

A consistency checking report is used to identify and troubleshoot a conflict in a business rules application—such as conflicts that might occur when new rules are created, added, removed, or modified.

A consistency checking report contains:

- The name of the package in which the conflict occurred
- A description of the error
- The definition that needs to be changed

Rule Execution Server and Rule Scenario Manager generate Execution Reports, as described in Section 5.
6.2. **Document Management**  
ILOG JRules generates reports in XML or HTML format.

6.3. **Interfaces to Third-Party Document Management Systems**  
JRules does not include any explicit integration with third-party content management or document management systems; however, the database schema for the Rule Team Server repository is published to help users understand its operation. Use of the schema to aid in viewing or reporting from the repository is supported.

7. **Development Environment**

7.1. **Programming Languages and Support for Open Standards, Components and Frameworks.**  
ILOG JRules is written in Java. In addition to general Java programming standards (JDK, JDBC, etc.), JRules incorporates and supports an extensive range of open standards, object components, and frameworks for application development, including:

- J2EE
- EJB
- JAXP
- XML (schema)
- WSDL
- UDDI/SOAP
- CORBA
- Servlets
- JSP
- CSS
- MOF
- XMI
- JMI
- JMS
- JSR-94

7.2. **Software Models and Code Generation**  
JRules does not provide code generation itself or through a third party. However, JRules 6 can map one or more BOMs to automatically generate code from other tools.

ILOG also offers different versions of ILOG Rules to support specific programming languages and object frameworks, including C/C++ and .NET—as indicated in Table 1.

7.3. **Maintenance and Update Support**  
ILOG JRules provides various features and functionality to support maintenance, incremental updates, and new versions of rules and/or rule sets.

The repository supports rule service deployment for multiple applications, ensuring consistent enforcement of business policies and procedures across the organization. Rule structures, rule services, allowed value controls, and audit trail data (such as “last modified date”, “last modified by”, and “reason changed”) are all stored in the repository.
The repository also has release management facilities, so that you can save a set of files as a specific release of the application so that subsequent changes to the files do not affect the released version of the product. Configuration options include saving XML rule files in a LDAP system that supports JNDI, a database that supports JDBC, or as flat files. (For more on ILOG Repository see Section 4.6)

7.4. **Integration with Third-Party Development Environments**

JRules Rule Studio implements a plug-in architecture to integrate directly into the open source Eclipse, IBM Rational Application Developer (RAD), and IBM Rational Software Architect (RSA) IDEs.

8. **Methodology Support**

ILOG JRules implements a proprietary methodology—the ILOG Solution Implementation Standard (ISIS). ISIS is a set of practices, tools and guidelines based on IBM’s Rational Unified Process, and which have been extended to explicitly support the business rule dimension of custom or composite application development. ISIS is currently available through ILOG’s consulting team. JRules serves as the basis of support for all business rule-related activities prescribed by ISIS.

9. **Integration**

9.1. **BPM Suite Integration**

ILOG’s BRMS technology is used in various vendors’ BPM Suites or BPM modeling environments. ILOG offers specific connectors to integrate ILOG JRules into popular workflow engines. ILOG BPM partners include BEA, FileNet Corporation, Fuego, Fujitsu, IBM, Vitria, and W4.

9.2. **Production Systems and Third-Party Applications**

As discussed in Section 7.1, JRules supports numerous open standards, object components, and frameworks for application development and integrating rule-based functionality into both production systems and commercial software packages.

9.3. **Web Services/Service Oriented Architecture Deployment**

ILOG JRules supports all Web Services standards, APIs and frameworks for deploying applications in a Service Oriented Architecture (SOA)—including: UDDI/SOAP, WSDL, J2EE, EJB, JAXP, and XML.

CHECK SOA POSITIONING FOR MORE HERE???

9.4. **Messaging and Middleware Integration**

The JRules Rule Execution Server is a run time execution environment for integrating the JRules BRE into J2SE and J2EE-compliant platforms. It is designed for deploying business rule services to the Web and application servers from IBM, BEA, JBoss, Oracle, and Apache.

ILOG also offers specific JRules messaging and middleware connectors, including:

- ILOG JRules Integration for BEA WebLogic Workshop
- ILOG JRules Integration for IBM WebSphere MQ
- ILOG JRules Integration for IBM WebSphere Process Server

9.5. **Database Support**

Rule Team Server and Rule Execution Server interact with third-party databases, including IBM DB2, Oracle, Pointbase (bundled with WebLogic), MS SQL Server, Sybase, Cloudscape (WebSphere), Derby (JBoss), MySQL (JBoss) and HSQLDB (JBoss).
9.6. Predictive/Analytic Modeling Support

JRules provides integration with data mining and analytic tools through ILOG’s PMML connector for SPSS Inc.’s data mining products.

10. Templates and Frameworks

10.1. Horizontal and Vertical Industry Templates, Frameworks, and Pre-Built Rule Bases

ILOG Lending Practice (ILP) is designed for financial services institutions implementing consumer-lending solutions. Available through ILOG Consulting Services, ILP combines ILOG consultants’ expertise from building lending applications with tools and templates to address specific lending process functionality. It targets policy automation and streamlining initiatives related to loan origination, underwriting and servicing.

ILP includes implementation roadmaps, methodologies and templates for eligibility determination, pricing and risk assessment (scorecard, rating, etc.), as well as modeling tools, support for industry standards such as MISMO, best practices and access to ILOG expertise for lending applications. It also complements the expertise of ILOG partners serving the customer lending and credit industry.

10.2. Domain-Specific Knowledge Products

ILOG JRules is the first BRM to earn MXCompliance certification from the Mortgage Industry Standards Maintenance Organization Inc. (MISMO), a not-for-profit subsidiary of the Mortgage Bankers Association (MBA). MXCompliance certification confirms that ILOG JRules meets specific mortgage industry standards related to XML data exchange.

ILOG JRules also recently earned certification from the Association for Cooperative Operations Research and Development (ACORD) for meeting ACORD’s XML for Life standards. This certification confirms that ILOG JRules fully supports ACORD XML-based messages. This the second ACORD certification for ILOG, which in 2004 earned a certification for meeting ACORD’s Property and Casualty (P&C) XML standards.

10.3. Domain and Industry-Specific Packaged Rule-Based Applications

ILOG offers a range of industry solutions through its Professional Services division, including for banking, lending and credit, capital markets and securities, government and defense, insurance, logistics, manufacturing, retail, telecommunications, and travel and transportation. Solutions combine consulting expertise with best practices, methodology, implementation roadmaps, modeling tools and templates designed to address lending-specific functions like eligibility determination, rating, score cards, credit scoring, pricing, risk assessment and compliance.

11. Administration and Security

Security and administration features of ILOG JRules are tied to the functions of each main module.

Rule Studio

Rule Studio is expected to be used in a development environment and provides for all JRules artifacts to be managed by a source code control (SCC) or similar content management system, either directly via the SCC user interface, or via an Eclipse plug-in for the SCC system. Rule studio thus defers access control and administration of the JRules artifacts to the SCC or content management system.

Rule Team Server

Rule Team Server runs in a J2EE application server, and gets all authentication services from the application server. Authorization for access to specific features of Rule Team Server is controlled by the roles assigned to authenticated user at login. For example, administrative features are available only to users who are assigned the role “rtsAdmin” by the application server authentication process.
Authorization for access to specific rule project artifacts is controlled by a fine-grained access control mechanism within Rule Team Server. The administrator determines whether projects use fine-grained security, assigns roles that allow general access to the project, and assigns access privileges to individual artifact classes, and individual properties of artifacts for each project, using Rule Team Server’s administrator user interface. Access privileges can be assigned permissively or restrictively.

A Permission Manager allows administrators assign specific roles and permissions to developers and business users to restrict access and modification privileges. For instance, business analysts might be permitted to write and edit rules, but only developers might be authorized to deploy them. Marketing staff might be restricted to writing and editing marketing rules, with accounting staff restricted to writing and editing accounting rules.

Permission Manager interfaces with any external authentication and authorization system, including LDAP, JAAS, Unix, Windows, Kerberos or Keystore. For large organizations, a locking feature allows multiple users to safely access and modify rules.

In addition, administrators and other specified user roles have access to general Rule Team Server configuration, and to baseline creation, modification and rollback features, and to deployment of rule sets to the Rule Execution Server.

**Rule Execution Server**

Rule execution server includes an administration console from which authorized users can promote staged rule sets to production status, and test, manage, and monitor multiple rule applications on multiple execution servers.

### 12. Platforms

JRules supports a range of JDKs and JVMs including Sun JVM, IBM JVM, and BEA JRockit.

Rule Team Server is supported on J2EE. Supported application servers including BEA WebLogic, IBM WebSphere and JBoss.

Rule Execution Server supports BEA WebLogic, IBM WebSphere, JBoss, Oracle OC4J and Tomcat.

### 13. Scalability

#### 13.1. Vertical Scalability

JRules 6 addresses vertical scalability for development teams by adopting the application development best practice of source code control. Each developer works through a local copy of Rule Studio to develop, test and deploy RuleApps. Coordination and concurrent development across a team of developers is managed through integration with third-party SCC tools. Because rule artifacts are persisted in the same repository with the source code and other non-rule artifacts of the custom or composite applications, they can be accessed concurrently and remotely by as many developers as the SCC tool can support.

JRules addresses vertical scalability for policy management teams by basing rule artifact persistence in Rule Team Server or third-party databases, as well as by providing our own concurrency control mechanisms that lock rule artifacts during editing. Through locking and DBMS efficiency, Rule Team Server can support large, distributed policy management teams working concurrently.

JRules also addresses vertical scalability during rule execution through clusters that provide mission-critical services to ensure minimal downtime and maximum scalability. Rule Execution Server supports BEA clusters, IBM clusters, JBoss clusters, and clusters on Oracle 10.1.3 preview 4.

#### 13.2. Horizontal Scalability

Rule Team Server is designed as a multi-project, multi-application repository for business rules. A single instance of Rule Team Server can support:
ILOG, Inc.


14.1. Industry Standards Initiatives
ILOG helped start the OMG’s business rules standards efforts and continues to work with OMG to further business rules standardization efforts. ILOG is currently leading a new standard rule-exchange language effort within the W3C to be coordinated with the ongoing OMG work.

14.2. Business Rules Consortia, Groups and Conferences
ILOG has been a fundamental sponsor of the Business Rules Forum and the European Business Rules Conference for the last several years, contributing time, resources, money, and customers to help make both events more valuable.

15. Pricing
Pricing for JRules 6 starts at $50,000 and it is common for total license costs to run between $150,000 and $300,000 for medium-sized, mission-critical custom application projects.


16.1. Company Background
ILOG was founded in 1987. The company is public, with headquarters located in France and the USA. In addition to BRM products, ILOG develops and markets optimization, data visualization, and planning and scheduling technology.

ILOG has approximately 650 employees worldwide, with approximately 230 involved in business rules technology. It has approximately 2,500 customers worldwide. For 2005, ILOG reported overall revenues of approximately US $125.3 million.

16.2. Positioning
ILOG serves a number of industries including, but not limited to:

♦ Banking and financial institutions
♦ Manufacturing
♦ Communications (equipment makers and network operators)
♦ Government
♦ Retail

16.3. Educational Training/Product Training
ILOG training covers core concepts, methodologies and tools for building business rule applications. Courses are offered for business analysts and technical users. Both curricula build practical experience through hands-on exercises. Sessions are conducted at ILOG facilities or on-site.

Training courses cover the following BRM product lines:

♦ ILOG JRules for developers
♦ ILOG JRules for business analysts
16.4. Product Support

ILOG’s standard product support services include:

- Online/Web support
- Release upgrades—support customers receive latest product releases and updates
- Phone support—customers can call ILOG support engineers regarding issues like installation, product use and problem resolution
- Discussion forums—support customers can tap into a worldwide online community of ILOG users and technical staff members
- Product newsletters—support customers can subscribe to free ILOG newsletters covering new product information, customer profiles, expert tips, performance-enhancing recommendations, application examples, technical training opportunities and related information

Additional programs are available for customers with expanded support requirements:

- After-hours support—ILOG maintains support centers in the major geographic regions. Developers needing support outside regular business hours can call any open ILOG support center on a 24x5 basis when the center in their region is closed. A 24x7 option is available that extends support to weekends.
- Premier support—guarantees that a product expert will arrive at a company location within two days whenever needed.
- VIP support—assures subscribers that their “Severity 1” support situations are given top priority by the ILOG support manager in their region. Subscribers receive the commitment that (at a minimum) a workaround will be found by the end of the next business day.

16.5. Consulting Support

ILOG consulting provides various resources, including industry best practices and product experts all utilizing ILOG’s proprietary methodology (ISIS).

ILOG consultants can provide guidance and expertise, formulate business goals, design the right technical architecture, develop the solution, and support its deployment. ILOG consultants can take the lead in several key areas:

- Technical oversight—architects and project managers ensure that projects get off to the right start, with a targeted application assessment
- Project involvement—product consultants work with developers, ensuring that a project’s ILOG components perform properly
- Project responsibility—ILOG consultants help tackle the optimization or business rules part of a project, delivering guaranteed results

17. Case Study: ILOG JRules at eBay

On-line retailer eBay uses ILOG JRules to enhance marketing promotions, among other applications, and to improve business response time across eBay’s product categories, creating an even better user experience, while improving time-to-market of eBay’s products and services.

eBay has standardized on ILOG JRules across several applications to create a unified platform for managing the business rules related to the buying and selling of items on the eBay site. Because of
the dynamic nature of eBay’s online business, with major updates to the software occurring regularly, the company needed a flexible solution that reduced time-to-market for these changes.

ILOG JRules allows eBay employees across departments to create and maintain business rules, allowing rule changes to be deployed instantly. Using ILOG JRules, new trends can be identified and policies can be updated in a matter of minutes. Meanwhile, ILOG JRules’ central rule repository makes managing rules across multiple sites and geographies possible in a quick, scalable and cost-effective way.

eBay’s worldwide ILOG JRules implementation is believed to be the most visible and demanding to-date for business rules technology—with more than 19 million listings on eBay across 35,000 categories. ILOG JRules Business Rule Management System, representing the latest generation of business rule software, met the demanding technical requirements associated with managing complex business policies across multiple sites and applications at eBay.
18. Company Offices

USA

Corporate Headquarters

ILOG, Inc.
1080 Linda Vista Ave.
Mountain View, CA 94043
Tel: 650-567-8000
Toll free: 800-FOR-ILOG
(800-367-4564)
E-mail: info@ilog.com

ILOG, Inc.
Nevada Office
889 Alder Avenue
Suite 200
Incline Village, NV 89451
Tel: 775-881-2800
Toll free: 800-FOR-ILOG
(800-367-4564)
E-mail: info@ilog.com

ILOG, Inc.
Boston Office
144 Turnpike Road
Suite 130
Southborough, MA 01772
Tel: 775-881-2800
Toll free: 800-FOR-ILOG
(800-367-4564)
E-mail: info@ilog.com

Directions

ILOG, Inc.
Washington DC Office
4350 North Fairfax Drive
Suite 800
Arlington, VA 22203-1619
Tel: 775-881-2800
Toll free: 800-FOR-ILOG
(800-367-4564)
E-mail: info@ilog.com

France

ILOG S.A.
9, rue de Verdun
BP 85
94253 Gentilly Cedex
Tel: +33 1 49 08 35 00
Fax: +33 1 49 08 35 10
E-mail: info@ilog.fr

R&D Office

ILOG S.A.
1681 Route des Dolines
Les Taissouinieres HB2
06560 VALBONNE
E-mail: info@ilog.fr

Germany

ILOG Deutschland GmbH
Ober-Eschbacher Strasse 109
61352 BAD HOMBURG
Tel: + 49 6172 40 60 0
Fax: + 49 6172 40 60 10
E-mail: info@ilog.de

Spain

ILOG SA
Gobelas, 21
28023 Madrid
Tel: +34 91 710 24 80
Fax: +34 91 372 99 76
E-mail: info@ilog.es

United Kingdom

ILOG Ltd.
Gentilly House
Bracknell Beeches
Bracknell Berkshire RG12 7BW
Tel: +44 1344 661600
Fax: +44 1344 661601
E-mail: info@ilog.co.uk

Australia

ILOG Australia
3, Spring Street
Sydney 2000
Tel: +61 (0) 2 8249 4355
Fax: +61 (0) 2 8249 4898
E-mail: info@ilog.com

China

ILOG
Room 601, Level 6,
Tower W3, The Towers ,
Oriental Plaza No. 1,
East Chang An Ave.,
Dong Cheng District,
Beijing 100738,
Tel: (86-10) 8518 1080
Fax: (86-10) 8518 1081
E-mail: info@ilog.com.sg
Japan
ILOG Co., Ltd.
3F YS Sanbancho Building
24-14 Sanbancho, Chiyoda-ku, Tokyo 102-0075
Tel: (03) 5211 5770
Fax: (03) 5211 5771
E-mail: info@ilog.co.jp

Singapore
ILOG (S) Pte Ltd
77 Science Park Drive
#03-24/27 Cintech III
Singapore 118256
Tel: (65) 6773 0626
Fax: (65) 6773 0439
E-mail: info@ilog.com.sg
19. Product Overview

Table 1 provides an overview of the Resolution Interactive Rules (iR) Platform and its main modules. The Resolution iR product takes a different approach to business rules management than the traditional “IF/Then” English language-like syntax model so closely associated with rule-based systems development. It employs a technique that provides the ability to express business logic as data, not code, and without using “IF/THEN” statements, thereby allowing logic to be analyzed, organized, combined and optimized mathematically using patented algorithms. The result is complex logic that looks very simple and executes very quickly. Resolution EBS’s approach has enabled the company to create a BRM product that employs a “Rule Grid” authoring environment that helps simplify BRM application development. This design also offers a very compact business rules engine (BRE) that can be deployed in various architectures—ranging from standalone applications to embedded components to smart applets that execute in a Web browser. Whether integrating via XML, Web Services, or Java APIs, organizations can deploy rule-based decision processing wherever it is most appropriate. In addition, the Resolution iR engine’s light footprint makes it very attractive to ISVs wanting to embed rule-based functionality into their software applications.

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iR Manager</td>
<td>Creation (authoring) and maintenance environment for building rule-based applications. Uses an innovative “Rule Grid” approach that provides an intuitive way for IT and non-technical users to develop and maintain applications. iR Manager consists of Core and Desktop. Core is a Java-based environment that enables centralized deployment of the iR Manager functionality on enterprise servers. Desktop is a .Net-based environment that provides a set of graphical interface components designed to enable business users to build and update business logic without creating code.</td>
</tr>
<tr>
<td>iR Engine</td>
<td>Extremely compact (70 KB) forward-and-backward chaining inference engine written entirely in Java; supports multiple deployment options ranging from standalone applications to embedded components to smart applets. Integration: provides full set of Java APIs, including JSP, EJB, Web Service (SOAP), XML interface, JDBC, integration maps, custom handlers to external systems, and HTML interface tools. Supported platforms include JBoss, SunOne, Sun Java System Application Server, Tomcat, WebLogic, and WebSphere.</td>
</tr>
<tr>
<td>Decision Packages</td>
<td>Publishing rules is accomplished through Packaging. Highly scalable (run-time) representation of the infrastructure, data and business rules that provides the linkages between the iR Manager and the runtime iR Engine. Packaging manages the interaction between all rules, and automatically combines the rules into a highly optimized Decision Package.</td>
</tr>
</tbody>
</table>
Table 1. Overview of Resolution iR, Key Modules and Functionality

| Interactive HTML Interface | Allows Web developers to design Web pages that the Resolution Platform can “plug” results of decision processing into; handles Web page controls, including for managing drop lists, radio boxes and text entry fields, as well as for managing other selection input mechanisms and for driving the output to data entry fields. |
| Pre-built Templates and Patterns | Resolution does not offer any specific templates or frameworks for use with Resolution iR. However, the company’s Professional Services team does have specific industry experience in healthcare, manufacturing and financial services. |

20. Product Architecture

20.1. Architecture Overview

Figure 1 provides an overview of the Resolution iR Platform architecture, which consists of three main components: the iR Manager, the iR Engine, and the “decision packager.”

Figure 1. Resolution iR Platform Architecture

Rule Authoring Environment/Graphical IDE: Resolution iR Manager

The iR Manager is the application development and maintenance tool for the Resolution iR Platform. It provides a simple, non-programming environment to document and manage business rules. Built to run on a Windows 32-bit platform (Windows 2000, Windows XP), the iR Manager provides a familiar, Microsoft Office-like interface, making it easy for the subject matter expert to capture and maintain their business logic.

The iR Manager allows the rules modeler to represent complex business logic as a simple Rule Grid, where different business attributes are matched in acceptable or unacceptable combinations.

All business rules and application components created with the iR Manager are stored as simple, individual XML files that can be managed, secured, and versioned using any commercial source code control system. The open schema of these XML files also enables rules and associated data elements to be created and modified programmatically by other applications. In addition, a set of API’s is available for the iR Manager, making it easy to embed.

The Resolution iR Engine is a packaged executable BRE that is standard across every Resolution-driven application. The iR Engine has a very small footprint—occupying less than 70 KB of disk space—and is entirely written in Java. It requires no infrastructure other than a Java Virtual Machine (JVM) to run.

The iR Engine executes in conjunction with one or more Decision Packages, responding to input value changes by executing the logic defined and triggering appropriate events such as calculations and guidance. The Resolution Engine is inherently stateless, enabling inputs to be supplied in any order and supporting “backward chaining” through even complex logic. This becomes especially helpful when driving interactive, UI-based applications.

Integration to the iR Engine is open and flexible, including direct access to individual data objects, mapped interfaces to XML data sets, and accessible API’s. Once again, embedability is the focus of this architecture.

Decision Package

Linking the two architectural elements is a “decision package” (i.e., decision support service)—a run-time representation of the business rules, business data, and deployment dependencies that can be processed by the iR engine. Decision packages are solely responsible for controlling the behavior of Resolution iR runtimes.

“Packaging” consists of converting the rules in the model (originally created using the iR Manager and stored as XML) into highly-optimized mathematical representations called “Directed Graphs” for efficient run-time processing. In addition, the other data elements in the model are converted to Java objects for easier integration. The entire decision package is very compact and easily deployed to the system hosting the iR Engine. Together the iR Engine and the Decision Package form the core of any Resolution iR-based application.

Rules Repository/Rules Database

All business rule data elements (e.g., attributes, calculations, messages and rules) created with iR Manager are stored as XML. Storage of these elements requires use of either a file system or industry leading source control systems such as Microsoft Visual Source Safe or CVS. The iR platform does not provide its own repository and can be operated with no repository, if desired.

Deployment

The Resolution iR Platform is specifically designed with the flexibility to run either standalone without any server or infrastructure requirements, or as a component of an application server (e.g., BEA WebLogic, etc.).

Both the iR Manager Core and the iR Engine are lightweight Java applications that are exposed via multiple interfaces such as Java class, EJB or Web Service. This affords various options for integration and deployment, including:

♦ Client-side deployment—the engine executes on a remote machine through a Web browser to function as a highly interactive “smart” applet. In this configuration, the iR engine can function with a constant Internet connection (i.e., “tethered”) or untethered—for example, to support mobile workers.

♦ Server-side deployment—the iR engine executes on the Web browser, as a JSP, communicating with the browser on the remote client to provide an interactive UI with full access to legacy data and systems residing behind the corporate firewall.

♦ Server-side component deployment—the iR engine executes on the server as a server-side component—such as an EJB or Java Class Library—for high-demand decision processing. (For more on iR Resolution Core and desktop architecture see Section 7.1.)
20.2. Business Rules Engine

Figure 2 shows the Resolution iR Engine architecture. The iR Engine is a constraint and inference-based BRE that supports forward and backward-chaining to process rules and other decision logic. It is a packaged executable software program that is standard across every Resolution-driven application regardless of deployment mode.

![Figure 2. Resolution iR Engine architecture](image)

The flexibility and compact nature of the Java runtime iR Engine allows it to be deployed on a wide range of platforms and architectures. And the same common calling interface layer is used to access all functions of the Engine, regardless of the method of integration.

20.2.1. Rules Management Approach

The iR Platform uses a patented code-free, data-driven approach to business rules that is based on the branch of mathematics called “Directed Graph Combinatorics (DGC).” With DGC, logic representations can take many forms, but the underlying concept is the “Rule Grid.”

This visual structure can replace many (sometimes dozens or even hundreds) of individual statement-type rules in a compact form that is easily created, changed, and interpreted. This representation of business rules enables “omni-directional” chaining in which all facts and decisions can be explored at any time. This provides several benefits. First, it powers an interactive "Goal Oriented" experience where user-defined criteria can be explored quickly and easily. Second, it enables very fast execution.

20.2.2. Enterprise Data Interoperability

The iR Engine provides multiple integration layers including standard integration maps for tabular data (e.g., XML, KVP, INI files, etc.). In addition, the iR Engine has a full set of Java API’s that enable flexible, programmatic integration between enterprise software applications and the iR Engine.
20.2.3. Maintaining Data Quality

Resolution’s approach uses mathematical algorithms to create a data-oriented representation of the business rules that is assured of data quality. Completeness and consistency of the logic are inherent in the way Resolution combines rules during the design-time packaging step.

21. Interfaces and Ease of Use

Resolution iR is specifically designed to address both the requirements of the business analyst and the IT department.

21.1. Development Environment

The iR Manager is the creation and maintenance application for the Resolution iR Platform. It provides a familiar Microsoft Office-like interface that makes it easy for subject matter experts to capture and maintain business logic. No coding, scripting, or statements are needed to model business rules.

The iR Manager allows the business modeler to represent the most complex logic using Rule Grids, where different business attributes (e.g., business data inputs, user selections, etc.) are matched in acceptable or unacceptable combinations. Basically, the Rule Grid allows the modeler to define one, two, or many attributes for the rows or columns, making these multi-dimensional Grids a fairly simple yet powerful way to express a large number of relationships in a single place.

Figure 3 shows a screen shot of the iR Manager GUI displaying a four-dimensional Rule Grid used for credit rating. This example demonstrates “Unacceptable” combinations being defined. The modeler simply has to click in the cells (Number of Employees, Industry, Company type and Financial Rating, etc.) that are “Unacceptable” and the rule is created. In addition, the modeler can define different events (messages and calculations) that are to be triggered for each cell in the Grid. There is no need to learn programming languages, English-like syntax, or anything else other than the facts of the rule.

Figure 3. iR Manager GUI Displaying 4-dimensional Rule Grid for Developing a Credit Rating Application.
Resolution Rule Grids are not a mask over a code-generator. Resolution rules are stored as pure data, so they can be mathematically combined and automatically optimized. The Resolution technology ensures Rule Grids are always complete, consistent, and unambiguous, so complex business logic can actually be broken into separate grids for ease of understandability and maintenance. In short, Rule Grids provide a powerful yet compact method of expressing business policies and actions, giving subject matter experts the ability to:

- Define relationships between data elements as “Acceptable” or “Unacceptable” as precisely as needed;
- Trigger specific events (e.g., unique calculation formulas to be used; and/or messages to be displayed) when those conditions are encountered.

While most rule-based systems require that the designer carefully consider the order in which rules are executed, Resolution iR does not. Rather, Resolution technology with its ability to represent rules as pure data actually executes all the rules in a rule set (or “model”) altogether (i.e., all at once and non-sequentially). This ability to define logic in separate Grids that get automatically combined helps ease development of complex applications by piecing together understandable segments.

21.2. End-User Environment
The runtime engine provides a built-in interface to handle Web page controls, including for managing drop lists, radio boxes and text entry fields, as well as for managing other selection input mechanisms and for driving the output to data entry fields. Web developers can design Web pages that the Resolution Platform can “plug” the results of decision processing into, allowing organizations to integrate Resolution iR rules processing into Internet and intranet sites. The Resolution Platform also offers “Conflict Advice” and “Selection Advice” to guide end users through the decision selection process. Selection Advice informs the user which choices are acceptable with their previous choices. Conflict Advice guides the user to the selection categories where they have made unacceptable selections visually represented in the application by a red “X.” Thus, Resolution presents the user with alternatives at every step in the decision making process, rather than forcing them into a predetermined set of conclusions.

22. Business Rules Expression

22.1. Defining Rules and Rule Sets
Resolution iR does not employ traditional natural-language style rules (i.e., IF-Then…) Rather, the iR Manager GUI features a grid-based interface that makes it easy for the subject matter expert to capture and maintain business logic without any technical expertise.

22.2. Creating and Maintaining Rules
The iR Manager allows the modeler to represent the most simple to complex business logic as a “Rule Grid”. There is no need to learn programming languages or special syntax. The Rule Grid enables the modeler to define specific actions for rules such as the presentation of a message or the computation of a calculation. Each combination or “cell” functions as an event to trigger the action. In addition, the iR Manager automatically ensures that no rules conflict and that all rules are complete and non-ambiguous. Therefore, rules generated in Resolution are 100 percent accurate and unbreakable on creation obviating the need to go through a manual conflict resolution cycle. Rule testing focuses on the proper definition of business logic as well as the interactivity these rules embody in the solution.

22.3. Rule Templates
Rule templates are a mechanism to simplify the creation of business rules and enable existing rule sets to be easily duplicated. Resolution’s rule grid approach enables business analysts to model
business rules without the need for custom-designed rule templates. The underlying, open XML schema of Resolution’s rule definitions allows third parties to add their own templates if needed.

22.4. Rule Syntax Checking
The iR Manager uses an iconic versus syntactic approach to defining business rules. Thus, the Rule Grid automatically handles rule syntax. Rule syntax can also be extended to support any special business requirement via User Defined Functions written in Java. The iR Manager automatically ensures that no rules conflict and that all rules are complete and non-ambiguous. Generated rules are 100% accurate and unbreakable on creation obviating the need to go through a manual conflict resolution cycle. Testing focuses on validating business logic instead of rule execution, rule exceptions, correct syntax, etc.

22.5. Rule Testing and Tracing
Resolution’s Rule Grids represent complete sets of rules, not single rule statements. Unlike rule statements, this graphical representation of logic eliminates the need to debug code, rule order, and context, completeness, ambiguity or conflict. Therefore, rule testing focuses on the proper definition of business logic as well as the interactivity these rules embody in the application.

Resolution provides a number of methods and tools to test business logic, including:

♦ Visual Inspection—Rule Grids make direct inspection a highly effective way to verify business logic. The tester can quickly and easily see all the possible logical conditions and verify that the appropriate outcomes have been defined.

♦ Model Explorer—the iR Manager includes a tool for inspecting the combined business logic defined in all the Rule Grids. The Model Explorer (Figure 4) simulates the runtime behavior of rules, allowing the tester to view and diagnose interactions between rules to verify expected results without impacting a production system.

♦ iR Engine State—the iR Engine features a debug mode that presents the state of the engine during runtime. This facilitates review of attribute state, calculation values, message presentation and model validity.

♦ Batch Mode Testing—batch mode and interactivity testing is supported to simulate test cases in order to ensure that the application is behaving as expected based on any rule definitions or changes.

Figure 4. Resolution Model Explorer Tool
22.6. Multi-User/Team Development
The iR Manager supports multi-user development via integration source control systems such as Microsoft Visual Source Safe or CVS. This enables multiple users to check in and check out different elements of the model. In addition, the iR Manager provides guidance to resolve any discrepancies or conflicts caused by any changes to the model.

22.7. Rules Management and Maintenance
The Resolution iR Manager GUI is specifically designed to allow business users to author, publish, and modify business rules without the need of a programming resource. The iR Manager allows the modeler to represent the most simple to complex business logic as a “Rule Grid”. There is no need to learn programming languages or special syntax. (See Sections 3.1 and 4.1 for more on the Resolution iR development environment.)

The Rule Grid also enables the modeler to define specific actions for rules such as the presentation of a message or the computation of a calculation. Each combination or “cell” functions as an event to trigger the action.

All business rules are stored as simple, non-proprietary data, instead of coded statements. The rules can be secured and maintained in a file system or source control system, and can leverage existing IT infrastructure and procedures for backup and restore capabilities, without the introduction of additional processes.

Publishing rules is accomplished through Packaging. Packaging manages the interaction between all rules, and automatically combines the rules into a highly optimized Decision Package. This shelters the user from the more complex task of describing the interactions between the rules. The Decision Package is deployed to the system hosting the iR Engine. Together the iR Engine and the Decision Package form the core of any rules-based application. (For more on “packaging and “decision packaging” see Section 2.1.)

22.8. Rule Logic Validation and Testing
Business rules in Resolution’s Rule Grids represent complete sets of rules, not single rule statements. Unlike rule statements, this logic representation eliminates the need to debug code, rule order, context, completeness, ambiguity or conflict. Therefore, rule testing focuses on the proper definition of business logic as well as the Interactivity these rules embody in the solution. (For more on logic validation and testing see Section 4.5).

22.9. Rule Logic Verification
See Sections 4.5 and 4.9 for more on verifying rule logic.

22.10. Conflicts and Priorities
Business rules in Resolution’s Rule Grids represent complete sets of rules, not single rule statements. Unlike rule statements, this code-free representation of logic eliminates the need to debug code, rule order, context, completeness, ambiguity or conflict. In cases where rules are overlapping or multiple conditions apply the iR Manager provides a mechanism for prioritization. Once these priorities are established testing focuses on the proper definition of business logic as well as the Interactivity these rules embody in the solution. (See Sections 4.5 and 4.9 for available testing and debugging tools.)

22.11. Decision Explanation/Auditing
Resolution provides two mechanisms to examine business rule reasoning and decisions. First, the iR Manager provides tools and reports to examine model elements and how they correspond to one another to arrive at a decision. Second, the iR Engine provides a complete audit trail of engine processing in XML form.
23. Simulation
The iR Manager provides a tool that enables quick simulation of business rules via a built-in HTML interface. It automatically generates a user interface that can be used to simulate various business scenarios and conditions. This provides a way for business analysts to review business rules directly without the need for a technical resource.

   24.1. Document Generation
   Resolution offers three reporting capabilities:
   - The Model Explorer tool provides a visualization of attribute, calculation, message and business rule cross references.
   - When used in conjunction with a source code control system (e.g., Visual SourceSafe, etc.) the iR Manager can provide an audit report of deleted and/or modified elements in a model.
   - The iR Manager also allows users to export the contents of a model in XML for details on individual rule elements.

   24.2. Document Management
   The Resolution iR Platform can be used to drive decisions and references to content and sub-content to be included in the presentation or disclosure of documents. These content references are assembled into a META tag and then output to the document management system.

   24.3. Interfaces to Third-Party Document Management Systems
   The Resolution iR platform does not currently provide any specific interfaces to third-party document management systems; however, the company is exploring partnerships with major content management systems.

25. Development Environment
   25.1. Programming Languages and Support for Open Standards, Components and Frameworks.
   The iR Engine is a 100% Java technology that is exposed via Java class, EJB and Web Service interfaces. Web application integration uses standard HTML, JavaScript and JSP technologies. The integration layer uses XML.
   The iR Manager provides the components that enable creation, validation, and automatic recombination of business rules. Subject matter experts as well as developers can use the intuitive “Desktop” application to construct and maintain rules, or developers can create business logic programmatically by directly accessing the “Core”.

   iR Manager Core
   The compact Java-based iR Manager Core enables deployment of the iR Manager functionality on a variety of platforms, including desktop clients (in conjunction with the Manager Desktop) and centralized build servers. The Core offers APIs that enable programmers to directly create rules and related data elements, validate them, export them as XML for storage, and package them into highly compressed logic files for execution by the iR Engine. The Core also accepts XML data elements created externally (with the Desktop or a third party authoring tool), validating and packaging these elements as noted above.
iR Manager Desktop
The iR Manager Desktop is a .Net-based environment that provides a set of graphical interface components designed to enable business users to build and update business logic without creating code. These desktop components are in essence a set of UI forms and widgets that utilize the iR Manager Core. Resolution’s office-user-focused tools and signature “Rule Grids” make it fast and easy to create data elements and define the relationships between them. All rule model elements (e.g., attributes, calculations, messages and rules) are built using XML.

25.2. Software Models and Code Generation
The Decision Package is created via the Core using a set of base methods supplied as standard Velocity code templates. The packaging step, performed at design-time, takes each data element in the model and creates a corresponding runtime Java object for integration. In addition, the rule grids defined are converted into directed graphs, combined, and reduced to remove irrelevant combination data. The runtime representation of these graphs amounts to compact data sets, rather than coded statements. (For more on Resolution’s logic representation scheme/model see Section 2.2.1)

25.3. Maintenance and Update Support
The iR Manager supports maintenance, incremental updates, and new versions of rules through the use of a source control system. The iR Manager can store all business rule data elements (e.g., attributes, calculations, messages and rules) in a source control system such as Microsoft Visual Source Safe or CVS. This enables easy maintenance and versioning of rules. In addition, users can add new elements to the system or check out existing elements for incremental changes.

25.4. Integration with Third-Party Development Environments
The iR Manager Core provides a set of Java API’s and a corresponding set of .Net API’s for developer access.

26. Methodology Support
Resolution iR does not implement any specific business rules development methodology. However, it is compatible with open standard rules methodologies such as those advocated by industry leaders like Ronald G. Ross (Principles of the Business Rule Approach, Addison-Wesley, 2003) and Barbara von Halle (Business Rules Applied: Building Better Systems Using the Business Rules Approach, Wiley Press, 2001).

Customers are not forced to adopt a proprietary methodology, but are instead encouraged to investigate publicly accessible rules-oriented approaches and tailor them to the specifics of their organization. The Resolution iR Platform supports these approaches through flexible, type-free specification of attributes and rules. In addition, the iR Platform’s code free approach simplifies several aspects of these methodologies by removing the need to exhaustively cross check rules for completeness and continually re-sequence rule sets to achieve performance.

27. Integration

27.1. BPM Suite Integration
The Resolution iR Platform is specifically architected for embedding into third-party products such as BPM suites. Both the iR Manager and the iR Engine are lightweight Java applications that are exposed via multiple interfaces such as Java class, EJB or Web Services. This enables the iR Platform with unmatched integration and deployment flexibility.

As an example, the Resolution iR Platform is embedded in the Insession Workpoint Advantage product. Workpoint Advantage uses the Resolution iR Platform for business rules in three key areas:

♦ Complex business logic and computation. An example would be using Resolution to perform a rating computation for an insurance underwriting business process.
♦ Complex routing and perform logic. An example would be using Resolution to consider a large number of facts in order to determine the next appropriate step in a workflow.

♦ Interactive data input form. Certain steps in a workflow may require interaction with a key player in an organization. The iR Platform can generate rules-based input forms that provide interactive guidance to complete the data and ensure it can be processed by the downstream workflow.

27.2. Production Systems and Third-Party Applications

The iR Platform provides two levels of integration:

Runtime/iR Engine
The runtime environment of the iR Platform is powered by the iR Engine. Business processes requiring rules processing call the iR Engine, which returns the appropriate response back to the workflow.

The Resolution iR Engine has multiple integration layers including standard integration maps for tabular data (e.g., XML, KVP, INI files, etc.). In addition, the iR Engine has a full set of Java API’s that enable flexible, programmatic integration between the BPM software and the iR Engine.

Rule Design/ iR Manager
The rule design environment of the iR Platform is powered by the iR Manager Core and Desktop. The Core contains a set of API’s to create, edit, validate and deploy business rule models. The elements (e.g., attributes, calculations, messages and rules) that comprise the business rule models are XML files that conform to a specific schema. Integration with a third-party application is achieved in two ways. First, the Core API can be leveraged to generate business rules programmatically. Alternatively, the third-party application can create the appropriate model elements conforming to the appropriate XML schema defined by Resolution.

The Desktop is a .NET application that uses a .NET Object library to design the business rule models. Third-party applications written in .NET have the option to leverage the modular window panes and/or the .NET Object library.

(For more on the iR Manger Core and Client Services see Section 7.1.)

27.3. Web Services/Service Oriented Architecture Deployment

The iR Platform typically fits into a SOA in one of two ways:

♦ The iR Engine can run as an Enterprise Decision Service where applications leverage the iR Platform for any rules based transactions;

♦ Resolution can also be used within front-end applications that require flexible rule-based capabilities. This enables more interactive and contextual user interfaces for applications powered by Web Services.

27.4. Messaging and Middleware Integration

The iR Engine can operate on a number of application servers including:

♦ Apache Tomcat
♦ BEA Weblogic
♦ IBM Websphere
♦ JBoss
♦ Macromedia JRun
27.5. Database Support
The Resolution iR Engine has no specific requirements for database. Rules are deployed in compact, stand-alone decision components called “Packages.”

The Resolution iR Manger requires no database. For archival, security and versioning, any commercial source code system can be used.

27.6. Predictive/Analytic Modeling Support
The Resolution iR Platform can import data from relational databases or the results produced by data mining tools. Since iR rules are based on data, the output of these tools does not need to be converted into programming syntax or natural language text. This allows the iR Manager to easily use tabular and XML based induction tool results.

28. Templates and Frameworks

28.1. Horizontal and Vertical Industry Templates, Frameworks, and Pre-Built Rule Bases
Resolution EBS does not offer any specific templates or frameworks for use with the Resolution iR Platform. However, the company’s channel partners have specific industry expertise in healthcare, manufacturing and financial services.

28.2. Domain-Specific Knowledge Products
Resolution EBS does not offer any domain-specific knowledge products for use with Resolution iR. However, the company’s channel partners have specific industry expertise in healthcare, manufacturing and financial services.

28.3. Domain and Industry-Specific Packaged Rule-Based Applications
Resolution EBS does not offer any packed domain or industry-specific BRM application products. However, the company’s channel partners have specific industry knowledge in healthcare, manufacturing and financial services.

29. Administration and Security
The Resolution iR Engine is a component and “lives inside” of the security provided by the application itself. For example, a J2EE Web application powered by the iR Engine leverages the security of the application server.

Security in the Resolution iR Manager is managed by a source control system. Privileges and access to individual elements (e.g., attributes, calculations, messages, rules, etc.) in the business model can be defined at the group or user level within the source control system.

30. Platforms
The Resolution iR Manager and iR Engine are lightweight Java technologies that only require JVM 1.4 or higher. This gives the Resolution iR Platform flexible deployment options. Customers can choose to run either stand-alone/disconnected or fully integrated into an application server environment.

The iR Engine runs on all application server platforms including:

- BEA Weblogic
- IBM Websphere
- Macromedia JRun
- Apache Tomcat
Supported operating environments include Microsoft Windows 2000, XP and Sun Solaris.

31. Scalability

31.1. Vertical Scalability
The Resolution iR Engine is scalable in both browser-based and server-based implementations. Browser-based deployments offer near zero overhead when compared to normal Web pages. Server-based deployments utilize a stateless engine for high throughput. The high throughput and the small footprint (i.e., 70 KB) of the iR Engine delivers industry-leading performance and scalability on both platforms. For example, according to EBS representatives, one customer running a server-side deployment tested 2,000 concurrent user connections and response times to entire rule sets were under 100 milliseconds.

31.2. Horizontal Scalability
The iR Platform has proven horizontal applicability with production solutions leveraging the iR Platform including:

- Health care—individual and small group rating, online quoting and coordination of benefits
- Property and casualty insurance—policy quotes
- Manufacturing—product configuration and pricing, needs analysis and proposal generation.
- Financial services—360 degree customer view enabling fine tuned product recommendations, actionable leads and an interactive online application for products

32. Business Rules Standards and Industry Groups

32.1. Industry Standards Initiatives
Resolution EBS representatives stress that the Resolution iR Platform is well aligned with the model driven architecture initiatives that have recently come into prominence in the software community. Resolution is also supportive of industry groups such as the OMG, the Business Rules Community, the Business Rules Forum, and the Business Rules Team.

32.2. Business Rules Consortiums, Groups and Conferences

33. Pricing
Resolution representatives declined to provide specific pricing information, indicating instead that pricing is at the “lower end of the mid-range for enterprise BRE’s,” with entry-priced end-user licenses typically offered on the initial application. OEM pricing is individualized to suit the requirements of each partner.

34. Company, Product Positioning and Support

34.1. Company Background
Resolution EBS was founded in 1996. It is privately held, and based in Columbus, OH, USA. The company has approximately 50 employees (15 in BRM Product development; 15 in sales; and 20 in consulting and product support).
34.2.  Positioning
Resolution EBS focuses primarily on manufacturing, financial services and healthcare industries, although the company has customers outside these verticals as well. Resolution EBS BRE product transcends market segments. Resolution EBS has 25 customers and partners (representing approximately 150,000 users).

34.3.  Educational Training/Product Training
Resolution EBS offers Role Based Training (RBT) that focuses on the way a specific person will ultimately use and interact with their products. This is conducted as close to the time that they will start using the product as possible in order to improve the retention rates of the trained users as they undertake work on their projects. Training covers three areas:

♦ Modeling
♦ User Interface Development
♦ Engine Integration
Each training class is hands-on and personalized with examples designed to reinforce the way the user will utilize the product. Classes are tailored to a specific organization’s business and designed to encompass actual business problems pertinent to that organization. According to EBS representatives, this tactical focus helps to reduce one of the most typical training complaints—that “the training was not relevant to what I do.” It also serves to maximize attendees’ ability to apply what is learned to their own work after leaving the classroom.

34.4.  Product Support
Resolution provides full service Technical Support via the Internet, email or telephone for any commercially released and supported product. Both standard support between 8 a.m. and 6 p.m. CST and premium 24x7 support is provided.

34.5.  Consulting Support
Resolution works through channel partners who have extensive experience in deploying rules based applications with design, development, implementation, support, and project management expertise to implement and support organizations through the process of deploying these solutions.

The goal is to ensure that every project is designed, developed, and implemented using the most reliable and effective services delivery methodologies, processes, and standards available in the industry today, directly addressing an organization’s specific goals and objectives. All services, and the processes designed to support them, are intended not only to provide clients with the maximum amount of risk protection possible, but also to address and support the cost-efficiencies businesses require.

35.  Case Study: Resolution iR for Policy Underwriting at BlueCross BlueShield of Arizona

35.1.  Context
The policy underwriting function in a health plan—BlueCross BlueShield of Arizona (BCBSAZ)—is complicated and labor-intensive. It is not unusual for this mission-critical process to encounter cumbersome decision-making, dependence on key staff members.

35.2.  The Challenge
Related to this challenge is a sales dilemma. With underwriting’s inherent complexity, there exists the possibility of a significant dollar variance between the initial rate quote generated by the sales broker and the final proposed rate determined by the underwriting group. If this variance is notably large, it can cause a prospective customer to look at alternative plans with competing carriers, thus complicating the sales cycle.
35.3. The Solution
In light of this, BCBSAZ initiated a project to find a better long-term solution for automating its underwriting business decisions. The deployed solution had to address fast changing market and competitive conditions, accommodate fluctuating decision criteria, support multiple product plans, efficiently integrate into its application infrastructure, and demonstrate a real, timely return on its initial investment.

BCBSAZ conducted a proof-of-concept demonstration with Resolution EBS and quickly determined that the Resolution iR Platform offered the right answer to its underwriting challenge. Resolution’s first implementation is automating individual rating decisions and migrating this decision process to its public website (www.AzBlue.com). The next implementation phase will include small and large group rating. Resolution is also applicable in other management areas beyond underwriting, including group and member enrollment, renewals, provider and medical management, and claims processing.

35.4. Result/Benefits
The primary benefit provided by the Resolution iR Platform is that it enables BCBSAZ’s underwriting department—the true subject matter experts—to easily define and manage critical rating, underwriting, and risk management decision rules. Under this scenario, greater efficiency is achieved because there is considerably less inter-departmental dependency between underwriting, sales and IT staff members. Underwriting correctly focuses on a decision rule’s validity for the business issue at hand, while the IT department concentrates on application integration and network infrastructure questions. No more waiting for limited, in-demand IT resources or cross-training on nuances of underwriting procedures.

Benefits BCBSAZ has realized from this project include:
♦ Assurance of complete, accurate census information from the sales broker and individual
♦ Consistent application of rating guidelines and policies across all plan types
♦ Lower dollar variations between initial quotes and final rate proposals
♦ Improved turnaround time for underwriting proposals, reduced sales cycle timeframes
♦ Elimination of time-consuming spreadsheet models
♦ Improved exception handling and reduced error rates
♦ A superior end-user experience with BCBSAZ and its online sales application

From a technology perspective, Resolution’s BRE fully integrates into BCBSAZ’s .Net computing architecture and meets the robustness, flexibility, and scalability needs required of an enterprise-level decision management technology.

For BCBSAZ, implementing a single BRM system that streamlines today’s underwriting process and provides a dynamic automation answer for future workflow opportunities demonstrates sensible strategic management.
36. Company Offices

USA
Corporate Headquarters
2240 City Gate Drive
Columbus, OH 43219
Phone: (614) 471-1471
Toll free: (866) 204-4427
E-mail: info@resolutionebs.com
sales@resolutionebs.com
### Rule Product Matrix

<table>
<thead>
<tr>
<th>1. Company, Product &amp; Version</th>
<th>Corticon Corticon BRM Platform Version 4.0</th>
<th>Fair Isaac Blaze Advisor Ver. 6.0.4</th>
<th>ILOG JRULES Ver. 6</th>
<th>Resolution EBS iR Platform Ver. 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X = Vendor claims to have feature</td>
<td>X = Vendor will add in next version</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>2. Product Architecture</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2.1 Architecture Overview</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Desktop</td>
<td>Authoring/IDE</td>
<td>Authoring/IDE</td>
<td>X</td>
<td>Authoring/IDE</td>
</tr>
<tr>
<td>Client Server</td>
<td>BRE, Repository</td>
<td>BRE, Repository</td>
<td>X</td>
<td>BRE</td>
</tr>
<tr>
<td><strong>2.2 Business Rules Engine</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database/transactional Engine</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inferencing-based Engine</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>&gt;Forward Chaining</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>&gt;Backward Chaining</td>
<td>(Design-time only)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>2.2.1 Rules Management Approach</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repository/OO</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Any SCS</td>
</tr>
<tr>
<td>Data-based</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Any SCS</td>
</tr>
<tr>
<td><strong>2.2.2 Enterprise Data Interoperability</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB Access</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>XML Documents</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>EAI/BPM</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Java/J2EE Objects</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CORBA Objects</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>COM Objects</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Legacy Systems, (objects found in Third-party data models, etc.)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>2.2.3 Maintaining Data Quality</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exception Handling</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>Data Quality Assured at Design-Time</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>User-defined data quality rules</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>3. Interfaces &amp; Ease of Use</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supports IT Developers</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Business Analysts</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Non-technical Business Users</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>3.1 Development Environment</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Separate Development Environment</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1. Company, Product &amp; Version</td>
<td>Corticon</td>
<td>Fair Isaac</td>
<td>ILOG</td>
<td>Resolution</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------</td>
<td>------------</td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>X = Vendor claims to have feature</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>O = Vendor will add in next version</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generates Rule Maintenance Applications</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>3.2 End-User Environment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Separate End-User Environment</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Generates Rule Maintenance Applications (constrained end-user UI)</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Web Forms Templates (limited subset/constrained rules)</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. Business Rules Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Defining Rules &amp; Rule Sets</td>
</tr>
<tr>
<td>Rules Language Approach</td>
</tr>
<tr>
<td>Spreadsheet-based/Grid-based UI</td>
</tr>
<tr>
<td>4.2 Creating &amp; Maintaining Rules</td>
</tr>
<tr>
<td>Decision Tables</td>
</tr>
<tr>
<td>Decision Trees</td>
</tr>
<tr>
<td>Scorecards/Scoring Models</td>
</tr>
<tr>
<td>4.3 Rule Templates</td>
</tr>
<tr>
<td>Rule Templates Supported</td>
</tr>
<tr>
<td>4.4 Rule Syntax Checking</td>
</tr>
<tr>
<td>Automated Rule Syntax Checking</td>
</tr>
<tr>
<td>4.5 Rule Testing &amp; Tracing</td>
</tr>
<tr>
<td>Rule-editing Windows</td>
</tr>
<tr>
<td>Execution Browsers</td>
</tr>
<tr>
<td>Model Explorer</td>
</tr>
<tr>
<td>Reporting Facilities</td>
</tr>
<tr>
<td>Test Case Construction</td>
</tr>
<tr>
<td>4.6 Multi-User Team Development</td>
</tr>
<tr>
<td>Repository Offered</td>
</tr>
<tr>
<td>Built-in Versioning Provided</td>
</tr>
<tr>
<td>Uses Third-Party SCS</td>
</tr>
<tr>
<td>4.7 Rule Management &amp; Maintenance</td>
</tr>
<tr>
<td>Wizards</td>
</tr>
</tbody>
</table>
1. Company, Product & Version

<table>
<thead>
<tr>
<th>Feature</th>
<th>Corticon Corticon BRM Platform Version 4.0</th>
<th>Fair Isaac Blaze Advisor Ver. 6.0.4</th>
<th>ILOG JRULES Ver. 6</th>
<th>Resolution EBS iR Platform Ver. 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Rules Templates</td>
<td>X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to Import Rule Definition Classes</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td>Rule Grid</td>
</tr>
<tr>
<td>Impact Analysis</td>
<td>X X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.8 Rule Logic Validation &amp; Testing</td>
<td>X X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Browsers &amp; Model Explorers</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td>Rule Grid</td>
</tr>
<tr>
<td>Rule Execution Browsers</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance/Batch Testing</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulation</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Reports</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration with Third-Party Rule Maintenance Tools</td>
<td>NA X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.9 Rule Logic Verification</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rules/ Rule Sets Analysis Tools</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td>Rule Grid</td>
</tr>
<tr>
<td>Conflict Reports</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>4.10 Conflicts &amp; Priorities</td>
<td>NA X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule Prioritization</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule Sequencing Decided at Design Time</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.11 Decision Explanation/Auditing</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Record Logging (e.g., transaction date, execution date, etc.)</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audit Trail Report Generation</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Simulation</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule Testing</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interactive Inputs</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to Read Test Data</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wizard-based Tools</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.1 Document Generation</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule Project Reports</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Reports</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conflict Reports</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.2 Document Management</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Specification</td>
<td>X X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Rule Product Matrix

<table>
<thead>
<tr>
<th>1. Company, Product &amp; Version</th>
<th>Corticon BRM Platform Version 4.0</th>
<th>Fair Isaac Blaze Advisor Ver. 6.0.4</th>
<th>ILOG JRULES Ver. 6</th>
<th>Resolution EBS iR Platform Ver. 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X = Vendor claims to have feature</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>O = Vendor will add in next version</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

- **Central Document Library**
  - X
  - X
  - X

- **Interfaces to 3-rd-Party Document Management Systems**
  - X
  - X
  - X

- **Specific Interfaces to 3rd-party DMS Provided**
  - X
  - X
  - X

<table>
<thead>
<tr>
<th>7. Development Environment</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>7.1 Programming Langs. &amp; Support for Open Standards</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>7.2 Software Models &amp; Code Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
</tr>
</tbody>
</table>

- **Separate, Code-specific Versions Available**
  - X
  - X
  - X

<table>
<thead>
<tr>
<th>7.3 Maintenance &amp; Update Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
</tr>
</tbody>
</table>

- **SCS** (optional, not required)
  - X
  - X
  - X

<table>
<thead>
<tr>
<th>7.4 Integration with Third-Party ADEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eclipse</td>
</tr>
<tr>
<td>IBM RAD</td>
</tr>
<tr>
<td>IBM RSA</td>
</tr>
<tr>
<td>IBM Rational Rose</td>
</tr>
<tr>
<td>MS Visual Studio</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. Methodology Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proprietary Methodology</td>
</tr>
<tr>
<td>Standard Business Rules Methods</td>
</tr>
<tr>
<td>No Methodology Supported</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1 BPM Suite/M&amp;A Tools</td>
</tr>
<tr>
<td>Adobe</td>
</tr>
<tr>
<td>BEA</td>
</tr>
<tr>
<td>Chordiant</td>
</tr>
<tr>
<td>FileNet</td>
</tr>
<tr>
<td>Fuego</td>
</tr>
<tr>
<td>Fujitsu</td>
</tr>
</tbody>
</table>
### 1. Company, Product & Version

* X = Vendor claims to have feature
* O = Vendor will add in next version

<table>
<thead>
<tr>
<th>Company &amp; Product</th>
<th>Corticon Corticon BRM Platform Version 4.0</th>
<th>Fair Isaac Blaze Advisor Ver. 6.0.4</th>
<th>ILOG JRULES Ver. 6</th>
<th>Resolution EBS iR Platform Ver. 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global 360</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Handysoft</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDS Scheer</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insession</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lombardi</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Savvion</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIBCO</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vitria</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W4</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.2 Production Systems &amp; Third-Party Applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UDDI/SOAP</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CORBA</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java Class Library</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>JMS</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EJB</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MQseries</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COM</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.NET</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.3 Web Services/SOA Deployment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EJB</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>J2EE</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>UDDI/SOAP</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.4 Messaging &amp; Middleware Integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apache</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>BEA WebLogic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>IBM MQseries</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM WebSphere</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM WebSphere Process Server</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>iPlanet</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J2EE</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>JBoss</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>JMS</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Rule Product Matrix

<table>
<thead>
<tr>
<th>1. Company, Product &amp; Version</th>
<th>Corticon Corticon BRM Platform Version 4.0</th>
<th>Fair Isaac Blaze Advisor Ver. 6.0.4</th>
<th>ILOG JRULES Ver. 6</th>
<th>Resolution EBS iR Platform Ver. 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X = Vendor claims to have feature</td>
<td>O = Vendor will add in next version</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macromedia JRun</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Oracle Application Server</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle OC4J</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMI</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Sun ONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tibco Rendevous</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.5 Database Support</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Infromix</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sybase</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSQLDB</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MySQL</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supports JDBC</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>9.6 Pedictive Analytics/Modeling Support</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capabilities Provided</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offered as Add-on Module/Product</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrates with 3rd-Party Data Mining/Induction Tools</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Templates &amp; Frameworks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.1 Horizontal &amp; Vertical Industry Templates, Frameworks, Rule Bases, etc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Financial Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Healthcare</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insurance</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Mortgage</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>10.2 Domain-Specific Knowledge Products</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific Processes, Rules</td>
<td>Via Consulting</td>
<td>Via Consulting</td>
<td>X</td>
<td>Via Consulting</td>
</tr>
<tr>
<td>10.3 Domain &amp; Industry-Specific Packaged Rule-Based Applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fraud</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. Administration &amp; Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Built-in Authorization Facilities</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
### Rule Product Matrix

<table>
<thead>
<tr>
<th>1. Company, Product &amp; Version</th>
<th>Corticon Corticon BRM Platform Version 4.0</th>
<th>Fair Isaac Blaze Advisor Ver. 6.0.4</th>
<th>ILOG JRULES Ver. 6</th>
<th>Resolution EBS iR Platform Ver. 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X = Vendor claims to have feature</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>O = Vendor will add in next version</td>
<td>(optional, not required)</td>
<td>(optional, not required)</td>
<td>(required)</td>
<td>(required)</td>
</tr>
<tr>
<td>Role-defined User Access</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Third-Party SCS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LDAP/Active Directory</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ID &amp; Password Logons</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>12. Platforms</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>OSs</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Unix</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Linux</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>13. Scalability</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>13.1 Horizontal Scalability</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Supports Large Number of Users</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>13.2 Vertical Scalability</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Can Distribute Workloads/Work Across Organizational Network</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>14.1 Industry Standards Initiatives</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>OMG</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Production Rules Representation (PRR)</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Standard for Business Vocabularies &amp; Business Rules (SBVR)</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>W3C</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Rule Interchange Format (RIF)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>JCP</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>JSR-94</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>OASIS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Business Rules Markup Language (BRML)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>14.2 Business Rules Consortia, Groups &amp; Conferences</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Business Rules Forum</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Business Rules Community</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Business Rules Team</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>European Business Rules Conference (EBRC)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
## Rule Product Matrix

### 1. Company, Product & Version

<table>
<thead>
<tr>
<th>Feature</th>
<th>Corticon Corticon BRM Platform Version 4.0</th>
<th>Fair Isaac Blaze Advisor Ver. 6.0.4</th>
<th>ILOG JRULES Ver. 6</th>
<th>Resolution EBS iR Platform Ver. 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAMA</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic Technology</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predictive Modeling Markup Language (PMML)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 15. Pricing (starting, single seat)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Corticon Corticon BRM Platform Version 4.0</th>
<th>Fair Isaac Blaze Advisor Ver. 6.0.4</th>
<th>ILOG JRULES Ver. 6</th>
<th>Resolution EBS iR Platform Ver. 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing</td>
<td>$10,000 (rule modeling)</td>
<td>$50,000</td>
<td></td>
<td>NA</td>
</tr>
</tbody>
</table>

### 16. Company Background

<table>
<thead>
<tr>
<th>Feature</th>
<th>Corticon Corticon BRM Platform Version 4.0</th>
<th>Fair Isaac Blaze Advisor Ver. 6.0.4</th>
<th>ILOG JRULES Ver. 6</th>
<th>Resolution EBS iR Platform Ver. 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Private</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

### 16.2 Positioning

<table>
<thead>
<tr>
<th>Feature</th>
<th>Corticon Corticon BRM Platform Version 4.0</th>
<th>Fair Isaac Blaze Advisor Ver. 6.0.4</th>
<th>ILOG JRULES Ver. 6</th>
<th>Resolution EBS iR Platform Ver. 3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Markets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Airlines</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Banking &amp; Finance</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Communications/Telecom</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Government</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Healthcare</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Insurance</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Manufacturing</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ports &amp; Shipping</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.3 Educational Training/Product Training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRM Training</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phone</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dedicated</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24-hr Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-lingual Support</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.5 Consulting Support</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRM Implementation Consulting Services</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workshops</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Authors

Curt Hall

Curt Hall is a well-known industry analyst, consultant, and newsletter editor. Curt’s expertise includes business process management, business intelligence, data warehousing, business performance management, data mining, business rules engines, knowledge management and other analytic technologies. He also focuses on the commercial applications of intelligent software including rule-based systems, intelligent agents, and speech recognition.

In addition to working as an analyst for BPTrends, Curt is a senior consultant with Cutter Consortium's Business Intelligence Advisory service, where he serves as editor of the weekly Business Intelligence Advisor e-newsletter. He is also co-author (with Paul Harmon) of Intelligent Software Systems Development: An IS Manager's Guide (John Wiley & Sons) and a contributing author to James Martin and James Odell's Object-Oriented Methods: Pragmatic Considerations (Prentice Hall). Curt's work has appeared in numerous technical journals and IT publications. His study on the corporate use of data warehouses and the issues associated with data warehousing projects has resulted in the in-depth report Corporate Use of Data Warehousing & Enterprise Analytic Technologies.

Curt lives in Berkeley, California and can be reached at curt@curt-hall.com

Paul Harmon

Paul Harmon is the Executive Editor and Founder of the Business Process Trends website.

Paul is a noted consultant, author, and analyst concerned with applying new technologies to real-world business problems. Paul's most recent book is Business Process Change: A Manager's Guide to Improving, Redesigning, and Automating Processes (Morgan Kaufmann, 2003). He has written a wide variety of articles that have been published on BPTrends in the past two years.

Paul has worked on major process redesign projects with Bank of America, Wells Fargo, Security Pacific, Prudential, and Citibank, among others. Paul is a widely respected keynote speaker and has developed and delivered workshops and seminars on a wide variety of topics to conferences and major corporations throughout the world.

Paul lives in San Francisco and can be reached at pharmon@bptrends.com