



**Enable dynamic behavior changes in
business performance management
solutions by incorporating business rules.**

Contents	
2	<i>Introduction</i>
3	<i>IBM business performance management framework integrates business and IT</i>
4	<i>IBM provides tools to define business processes and incorporate business rules</i>
4	<i>Runtime components execute business processes</i>
4	<i>Common event infrastructure facilitates monitoring of both business and IT processes</i>
5	<i>WebSphere Business Integration supports three types of rules</i>
5	<i>Business rules invoked like any other step in a business process</i>
6	<i>Use business rules for flow decisions</i>
6	<i>Choosing business rules for process steps enables flexible responses to change</i>
6	<i>Enable applications to invoke business rules without requiring re-engineering of those applications</i>
7	<i>Selector executes the proper business rules</i>
10	<i>Leverage a standardized way to exchange rules across the enterprise</i>
11	<i>IBM Business Partners have opportunity to develop a wide variety of rules-related tools for both business and IT purposes</i>
12	<i>Summary</i>
15	<i>For more information</i>

Introduction

Today's business environment demands agility. Businesses must respond on demand to changing customer needs with flexible systems and processes. Static processes that cannot adapt to changing needs are a liability. This is why *business performance management* has emerged as a critical discipline for agile businesses.

Business performance management optimizes business processes through modeling, simulation and monitoring. Business performance management allows a business to:

- *Closely monitor and manage both business and IT activities as they occur—across and beyond the enterprise.*
- *Recognize areas for improvement.*
- *Quickly implement actions that optimize—and bridge—business and IT performance.*

The benefits of implementing business performance management are significant: helping reduce costs, increase ROI, align business priorities with IT resources, and improve operational efficiency and resiliency.

To achieve true business insight, a comprehensive business performance management system should include a set of flexible *business rules* that allow users to optimize business processes without the need for additional modeling and deployment cycles. Business rules externalize business policies so they can be managed independently of application software. For example, as marketing plans change, businesses can easily modify the business rules used to determine customer discounts. As policies regarding customer credit are updated, the business rules that govern approval and rejection based on credit status and history can be modified accordingly. In short, business rules support on demand business because they enable agility and responsiveness in business processes.

Implementing business rules within the business performance management framework allows not only IBM middleware, but also customer-developed and IBM Business Partner-led applications, to participate in a comprehensive management system. This paper explains the role of business rules within the business performance management framework. It also outlines opportunities for IBM Business Partners to extend IBM's business performance management

Highlights

Provide role-specific views for both business and IT executives

capabilities with their own technology and to exploit IBM business performance management software to build domain-specific solutions.

This white paper describes technical concepts for future IBM products. These concepts represent current thinking that could change before product release due to IBM Business Partner feedback, technical discoveries or other reasons.

IBM business performance management framework integrates business and IT

IBM business performance management middleware emphasizes monitoring and managing both business processes and IT events. By managing both event types within an integrated framework, enterprises are better able to align IT resource utilization to meet business priorities. At the same time that enterprise business executives can see the results of core business process execution on their desktops and can utilize those results to enable timely decision-making and business transformation, enterprise IT executives can leverage business views of the IT infrastructure to recommend IT specific actions that will generate the greatest benefits to the business.

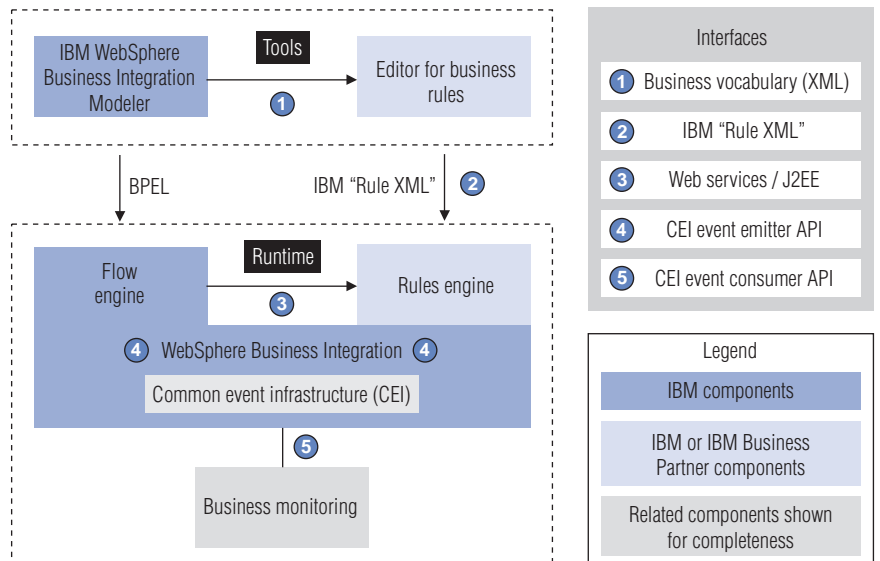


Figure 1: IBM business performance management framework

Understand how business rules fit into IBM WebSphere Business Integration

In the future, IBM's business performance management framework will contain a set of Eclipse-based tools for modeling and simulating business processes, as well as runtime functions for process choreography, event monitoring and rules execution. An explanation of the major components of this framework follows.

Highlights

***Define business processes and
deploy them to flow engines***

IBM provides tools to define business processes and incorporate business rules

The IBM business performance management toolset includes the IBM WebSphere® Business Integration Modeler, an intuitive business process modeler that enables business consultants or developers to define business processes. The tool enables simulation of modeled processes to help ensure that they meet business requirements. WebSphere Business Integration Modeler supports various export formats for deploying business processes to flow engines. The latest release of the modeler, version 4.2, generates FDL (flow definition language) for execution by IBM WebSphere MQ Workflow. Future releases of WebSphere Business Integration Modeler will also generate models in the industry-standard BPEL (business process execution language) format, for execution by the WebSphere flow engine.

Future releases will be further enhanced by the introduction of a “business vocabulary” that codifies the set of business entities that can be accessed by business rules, and an editor for business rules expressed in terms of that vocabulary. Business artifacts will supply the vocabulary for business rules applicable in a process flow; business events will define the vocabulary for rules governing event correlation, KPI calculation and situation discovery. Pending the availability of industry standards, both vocabulary and rules will be produced in an IBM “Rule XML” format. This format is described in the “Leverage a standardized way to exchange rules across the enterprise” section of this document.

***Implement process activities as
business rules***

Runtime components execute business processes

The runtime components of the IBM business performance management system include a flow engine to execute the business processes, defined as XML schema, that were generated by the WebSphere Business Integration Modeler. This flow engine executes the business activities defined in the business process by invoking J2EE or Web services. In the future, some of the business activities may be implemented as rules handled by a rules engine. The rules engine would execute the rules defined in the “Rule XML.” Both the flow and rules engines will execute on top of the operating environment provided by WebSphere.

Common event infrastructure facilitates monitoring of both business and IT processes

WebSphere Business Integration includes a common event infrastructure (CEI) that is notified of all events that occur throughout the systems that

Highlights

Implement simple, event correlation or inferencing rules

monitor business and IT conditions in the enterprise. A presentation about the CEI is available on the IBM Web site.¹ The ability to monitor both business and IT events through the CEI lays the foundation for creating a business performance management framework.

WebSphere Business Integration supports three types of rules

Various rule types may be used within the WebSphere Business Integration business performance management framework:

1. **Simple business rules** provide declarative rules defined by business users. They execute sequentially and maintain no state across invocations.
2. **Event correlation rules** recognize relationships across multiple CEI events. For example, the IBM Tivoli Enterprise Console® “State Correlation” rules detect IT or business situations from patterns in event sequences. IT or business analysts define these rules.
3. **Inferencing rules** implement forward inferencing, backward chaining, Prolog-style unification or other AI-style rules. They develop “facts” that represent states that the rules can recognize. For example, the IBM Agent Building and Learning Environment package supports a variety of inferencing techniques.² These sophisticated rules require programming skills.

Business rules invoked like any other step in a business process

Business rules are often used in the context of business processes. For example, a process step might invoke a “calculate discount” activity using business rules. The process would pass some sort of input to the rules, and expect the rules to return the discount as a numeric value.

Treat activities in the same fashion, regardless of the technology used to implement those activities

From the perspective of the process definition, the important aspects of the activity are the inputs, output and intended purpose of the activity. Whether the activity is implemented using Java™, another process, business rules or some other technology is a separate design choice. This separation of concerns is achieved by using the same calling interface mechanism regardless of the activity’s implementation technology. In other words, the process definition and runtime treat all activities the same, regardless of implementation type.

Highlights

***Facilitate updates without redesign,
redevelopment and reinstallation
of applications***

***Maintain consistency with J2EE and
Web services standards***

Use business rules for flow decisions

Business rules often support decisions in business processes. For example, an order processing flow might include different steps for customers with good versus bad credit. This could be implemented by having an “evaluate credit” flow activity that assesses the credit-worthiness of a customer, and returns as a result the choice among alternate processing paths. Such a flow activity could be implemented using business rules.

Choosing business rules for process steps enables flexible responses to change

The “evaluate credit” activity could be implemented in many different ways—as activities in the overall order process, EJBs (Enterprise Java Beans), Web services or business rules. The advantage of implementing it—or any other activity step in a business process—as a business rule is the flexibility the rule enables. For example, since credit evaluation policies change over time, it’s important that the “evaluate credit” step allow for updates without requiring the complete cycle of process redesign, redevelopment and application reinstallation that other implementation technologies require. Business rules ease updates when business policies change.

Enable applications to invoke business rules without requiring re-engineering of those applications

IBM offers a menu of implementation technologies: Java, Web services, business processes and more. Business rules join this repertoire as another way to implement application function.

As discussed above, calling applications—whether written in Java or as business processes—should not need to be aware of (that is, should be agnostic to) the implementation technology of the processes they invoke. IBM achieves this implementation type independence by maintaining consistency with J2EE and Web services standards across the entire repertoire of technologies. This means that business rules may be invoked as session beans or Web services.

Business rules are invoked with application-specified interfaces, using the standard concept of a “contract” between caller and rules. This provides type-safe calling methods, with the tools validating argument types used in actual calls against the expected parameter types defined in the interfaces.

Highlights

***Dynamically update rules
in accordance with changing
business conditions***

One key difference between business rules and other technologies is the ability to dynamically change the rules, without restarting running applications. For example, in a customer relationship management (CRM) application, a marketing manager might change discount rules frequently in response to changing business conditions.

On demand applications often execute in server environments, in which the same application code executes on behalf of multiple users—each of which needs its own rules. For example, in a multitenant, hosted application, each tenant company should have its own independently managed rules executing in the context of the shared application. The next section discusses a mechanism that supports multiple sets of rules used by a single program.

Selector executes the proper business rules

Within the business performance management framework, a selector component, which sits between a calling application and a rules engine, houses the logic for choosing among alternate rule sets. Selector components are of particular importance to IBM Business Partners because these companies may insert code into selector components to link to their own rules engines.

Select the appropriate rule set

A selector component contains three parts:

1. A shell that implements a type-safe interface on behalf of client applications.
2. A selector routine that chooses which rules and which rules engine to use. On each call, the selector can use arguments passed in the call, context data (such as the WebSphere work area) and exogenous information, such as the current date and time, to select among alternate sets of rules.
3. A locator routine that links to the rules engine chosen by the selector routine.

Selectors can be used in various ways. Consider the following examples:

- *For date- and time-based selections, run one set of rules during January and another set in February.*
- *In multitenant or hosted applications, select the appropriate rules for the current tenant company.*

Highlights

- For insurance companies, which need to deal with varying regulations among the 50 U.S. states, choose the appropriate rule set according to a “state” field passed as an application argument. Organizing rules by state avoids combining the rules for all the states in one cumbersome rule set with “if” conditions such as “if state is New York.”

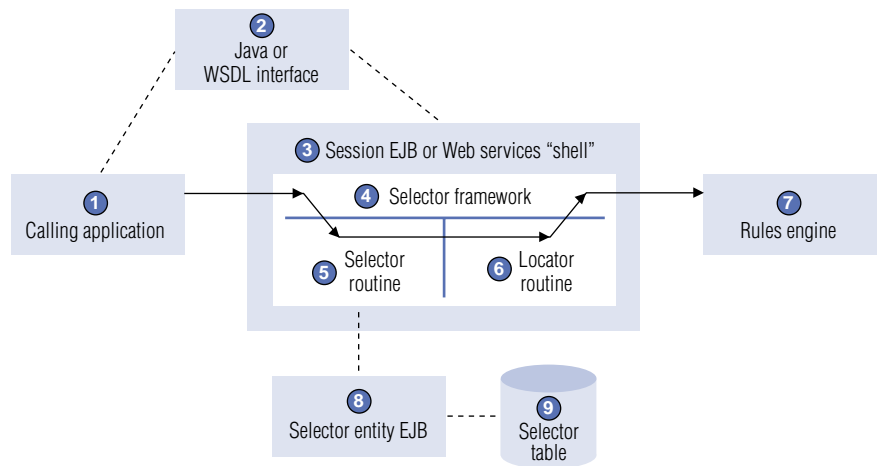


Figure 2: Invoking rules from an application

The connection from a calling program to a rules engine involves the following components, which are also represented in Figure 2:

Invoke business rules from applications

1. A calling application invokes a session EJB or Web services operation.
2. The interface between the calling application and the session EJB or Web services is provided via a Java interface definition or a Web services description language (WSDL) file. These ways of providing an interface enable type-safe calling, ensuring that the arguments passed by the calling application match the parameters expected by the rules.
3. A session EJB or Web services “shell” program—ideally generated by tools directly from the Java interface or WSDL definition—receives the call and converts the application-specific arguments into a normalized format.
4. The shell invokes IBM-supplied selector and locator interfaces, which provide standard APIs for the selector and locator routines.

Highlights

Plug IBM Business Partner rules engine into WebSphere Business Integration

5. A custom selector routine, plugged into the selector interfaces, chooses a set of rules and a locator that represents a rules engine.
6. A custom locator routine, plugged into the locator interface, links to the selected rules engine, passing the selected rule set and the original calling program's arguments.
7. The rules engine executes against the arguments and returns some sort of result, which is passed back through the shell to the calling application.
8. In a typical implementation, the custom selector algorithm might look up the target rules via an entity EJB. Other lookup sources could also be used.
9. The actual rules might be stored in a database accessed through an entity EJB.

Configure remote or local invocation of business rules

Using standard WebSphere features, the selector "shell" can be configured to be remote or local to the calling application. If remote, standard remote method invocation/Internet interoperable object protocol (RMI/IIOP) or Web services distributed protocols are used. If local, WebSphere can be configured to pass arguments by reference, thus making an efficient calling mechanism. In either case, standard J2EE enterprise-class middleware services are available to the shell and rules engine. These include features like the J2EE environment and security access control to the shell.

Business rules can "call back" to methods defined in the calling application's arguments. These methods can execute locally if configured that way, or can execute remotely when EJB references are passed.

Isolate applications from rule execution and centralize rule selection

In summary, the selector provides a way to isolate applications from the specifics of rules and rules engines, while enabling a standard J2EE or Web services invocation pattern with type-safe arguments. It provides a centralized and configurable place for rule selection, and for delegation of execution to rules engines. A preview of the selector mechanism will soon be available in the IBM Selector Software Developer's Kit.³ This preview will provide code and detailed instructions for exploiting the selector concept on a current version of WebSphere.

Highlights

Work with the OMG to standardize rules models

Leverage a standardized way to exchange rules across the enterprise

By storing business vocabulary and rules in files, a database or elsewhere according to the “Rule XML” format, companies can exchange the vocabulary and rules between tools, and also between the tools and the runtime.

IBM is working with the Object Management Group (OMG) to standardize “Rule XML.” The OMG expects standards at several layers, as shown in the following figure:

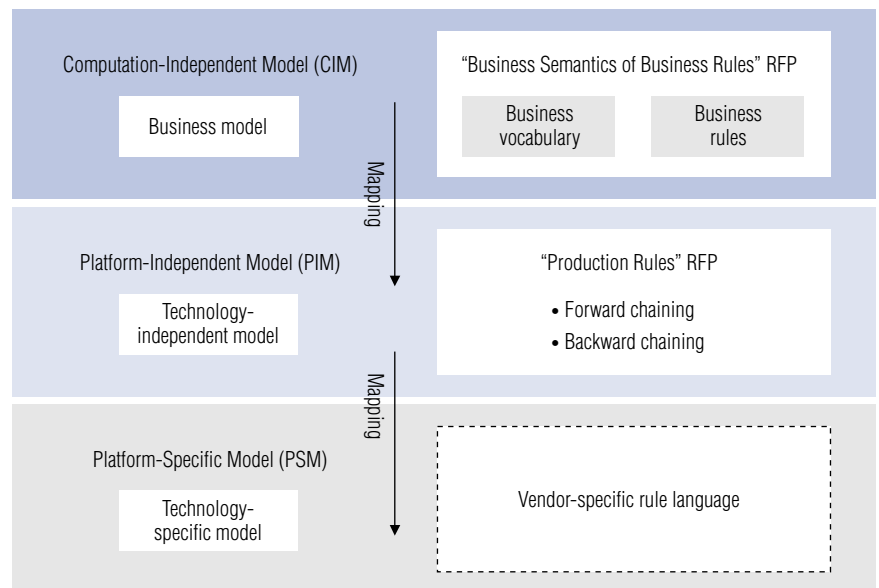


Figure 3: Business rules in the OMG's model-driven architecture

- The top “Computation-Independent Model” (CIM) layer addresses business models; that is, models of concepts intended for business users. At this layer, OMG has requested a proposal for a business vocabulary model and a declarative business rules model.
- The middle “Platform-Independent Model” (PIM) layer addresses rules technology independent of specific vendors. This layer includes concepts such as forward and backward chaining rules.
- The bottom “Platform-Specific Model” (PSM) layer contains individual vendor rule languages. The IBM Agent Building and Learning Environment Rules Language is an example of a vendor-specific rule language.²
- Mappings among these layers permit rules defined at a higher layer to be mapped down to lower layers for execution.

Highlights

***Develop rules for a variety of
rules functions and for a variety
of user types***

***Provide tools for integrated
development environments and tool
widgets intended to integrate with
application user interfaces***

There are several fundamental differences between the top two layers of this diagram. The CIM targets business users. The business vocabulary encompassed in this layer enables business users to think in their own terms, rather than in terms of the underlying IT artifacts. The business rules component of this layer hides execution details that are not meaningful at the business level of abstraction. The PIM is targeted toward skilled developers. It exposes various programming styles that must be understood by users working at this layer.

IBM Business Partners have opportunity to develop a wide variety of rules-related tools for both business and IT purposes

A variety of tools are needed to support rules in a business performance management environment. Tools must exist to define and manage business rules and vocabulary. They must also identify dependencies between rules, and between rules and vocabulary entries. Tools must validate rules for their completeness and identify when rules overlap with one another. Creating and managing selectors requires another set of tools. And finally, testing, debugging and simulation tools for rules execution must exist.

In addition to varying by rules function, tools also vary for the different types of users who will build and interact with business rules. Traditional developers want tools that are integrated into the IBM WebSphere Studio integrated developer's environment. These users work both at the production rules and business model layers discussed above. Business users, on the other hand, work only at the business model layer. Some of these business users will be "professional" tools users who expect the sophistication of an integrated development environment. Others may want application-specific user interfaces (UIs) that can be accessed through IBM WebSphere Portal.

Tools oriented to the business model layer present an abstracted view of rules: they are purely declarative and written in terms of the business vocabulary. Features may include fully spelled-out names for vocabulary terms, such as "Customer's Name" rather than "custName," with translation for multiple human languages. Rules may ignore details of data types (such as integer versus float), and may be defined independent of any concept of execution processing or of how they fit into the overall application. In fact, some rules defined at this layer may be "implemented" via human rather than automated

processes. The business user thinks of the rules as a way to specify business policy, rather than a way to control application execution.

Tools oriented to the production rules layer provide a programmer's view of rules. Instead of a business vocabulary, the rules are written in terms of programming objects with typical short names and explicit data types. The programmer understands the execution sequencing of rules, such as forward and backward chaining. The solution designer carefully considers the interaction of rules with the overall application, including issues such as what parameters are passed to the rules, the expected functions achieved by the rules and the results produced by the rules. In effect, the production rules layer defines a programming language.

Often the tools will be used in the context of a "rules-exploiting" application. For example, when a process step is defined using rules, a rules editor should be launched directly from the process editor. When rules are used to govern or configure business applications such as CRM or ERP (enterprise resource planning), business users will probably use rule-editing tools directly within these business applications. For example, rules for calculating shipping costs might be defined as part of customizing a warehouse application. The WebSphere Studio environment facilitates this sort of composition of tools because it is built on the Eclipse open-source project and includes a well-developed architecture for plugging in tools.⁴

Opportunities exist for IBM Business Partners to develop both rule-specific tools and application-oriented tools that exploit rules.

Summary

There are many opportunities for IBM Business Partners to add value to the business performance management framework by contributing business rules. Consider the following diagram that identifies ten distinct aspects of the IBM Business Partner framework, several of which represent interfaces for IBM Business Partner functions. The top part of this diagram sketches the tools environment for both business-layer and production rules. The bottom portion illustrates the runtime system. The "packaging model" represents the sole connection between these two environments.

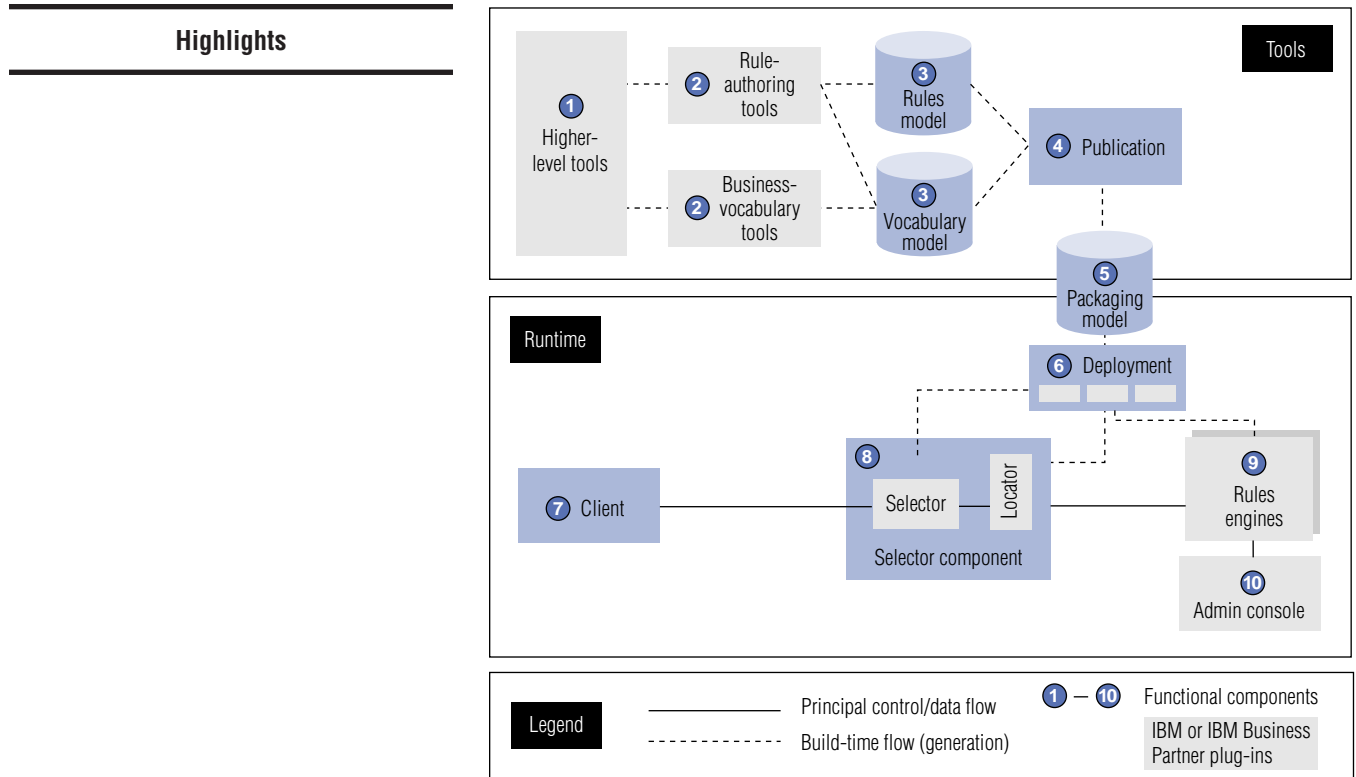


Figure 4: Tools environment and runtime system of IBM BPM framework

Exploit functions provided by IBM Business Partners within the WebSphere Business integration framework

1. **Higher-level tools**—IBM Business Partners can provide application-specific tools that may embed the rule or business vocabulary tools.
2. **Rule-authoring and business-vocabulary tools**—IBM Business Partners can provide editors for authoring and managing the rules and business vocabulary, and utilities for importing business vocabulary from other metadata sources such as Java beans or WSDL.
3. **Rules model and vocabulary model**—Runtime-independent representations of the business rules and vocabulary. In the short term, IBM Business Partners can utilize their own rules models, but we encourage them to work towards standardization in the context of the OMG.
4. **Publication**—IBM intends to provide a standard function for the packaging of rules and vocabulary for transmission to the runtime. In the short term, IBM Business Partners can substitute their own vendor-specific mechanisms.

5. **Packaging model**—IBM will define a packaging model that combines the rules, vocabulary and selector entries needed for a particular application with other application components. To simplify the architecture, this packaging model is the sole interface from the tools to the runtime system. This interface operates one way, passing “Rule XML” from the tools into the runtime.
6. **Deployment**—IBM will define a rules deployment mechanism that will install the rule, vocabulary and selector entries into the runtime rules engine, selector and locator, using vendor-specific plug-ins. IBM Business Partners can support their own rules engine and selector implementations via these deployment-time plug-ins.
7. **Client**—IBM Business Partners can define client components to use business rules. The interface required by the client is specified by the client. Note that the client is indifferent to the use of business rules per se—it is looking for a specific interface that happens to be realized by rules.
8. **Selector component**—As described earlier, this consists of a shell, a selector routine and a locator routine. IBM Business Partners can supply customized selector implementations, and can connect to their own rules engines by providing locator implementations.
9. **Rules engines**—IBM Business Partners can supply rules engines that connect into this infrastructure via locators.
10. **Admin console**—IBM Business Partners can develop functional components for the administration of their rules engines. For example, a component to manage the authorization of users to execute various admin functions, or to manage the starting and stopping of rules engine. The admin console may also include the administration of selectors and locators.

This paper has presented a framework for rules used in business performance management. This enables IBM Business Partners to understand the role of rules in IBM technology, both as potential exploiters and as providers of added-value rules technology. Components of this framework will become available over time. Details may evolve, but the ten aspects of the IBM Business Partner framework outlined here will exist. Most parts of the

invocation model will be made available in the near future via the IBM Selector Software Developer's Kit.³

For more information

To learn more about the role you can play in developing rules in conjunction with the IBM business performance management framework—and leverage IBM resources—consult the sources in the footnotes (on the next page) and also the following Web sites:

IBM, *Establishing a business performance management ecosystem*, IBM white paper, March 2004

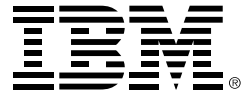
ibm.com/software/bpm

Object Management Group, *Business Semantics of Business Rules Request for Proposal*, OMG Document br/2003-05-01, June 6, 2003

www.omg.org/techprocess/meetings/schedule/Bus_Semantics_of_Bus_Rules_RFP.html

Object Management Group, *Production Rule Representation Request for Proposal*, OMG Document br/2003-09-03, September 12, 2003

www.omg.org/techprocess/meetings/schedule/Prod_Rule_Representation_RFP.html



© Copyright IBM Corporation 2004

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

03-04
All Rights Reserved

The e-business logo, e-business on demand, the e(logo)business on demand lockup, IBM, the IBM logo, Tivoli, Tivoli Enterprise Console and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

¹ The IBM common event infrastructure presentation is available at ibm.com/software/bpm.

² More information about the IBM Agent Building and Learning Environment can be found at alphaworks.ibm.com/tech/able.

³ The IBM Selector Software Developer's Kit will be made available at ibm.com/software/bpm.

⁴ For more information on Eclipse, visit eclipse.org.