Implementing Smarter Process decisions

When to use IBM Business Process Manager rules and IBM Operational Decision Manager

Scott Simmons
Hector H Diaz
Irina Singh

March 25, 2015

In our work assisting clients with smarter process applications, we are asked to detail leading practices for integrating decision services with business process management solutions. One question we encounter is when to use IBM Business Process Manager (BPM) rules and when to use IBM Operational Decision Manager (ODM). The answer to this question is use-case specific, but there are criteria to consider for the target solution. This tutorial discusses the decisions to consider when developing applications that integrate business process management with decision services and provides a detailed example to illustrate the concepts.

Introduction

Business process management systems and decision management solutions are naturally complementary. When you look at a business process management solution, key service types include human services, integration services, system services and decision services. The decision services support multiple functions, including validation, data enrichment, aggregation, computation, extended logical functions, and other key flow control aspects. In fact, all processes are dependent on decisions.

If you are interested in the aspects of rules and decision management, there are two very good books on decision services:


Try the Business Rules service

Spend less time recoding and testing when the business policy changes. The Business Rules service from Bluemix minimizes your code changes by keeping business logic separate from application logic. Try it for free!
This tutorial focuses on criteria to consider when implementing rules with an evolving IBM BPM solution. If you are looking for detailed technical information on the underlying integration aspects between IBM BPM and IBM ODM, make sure to review Best practices for designing and implementing decision services by Jerome Boyer. This tutorial offers guidance in selecting the appropriate solution approach for IBM BPM and decision management but does not supersede the extensive detailed technical content in the best practices series.

In nearly all cases, using a separate rules system for developing and deploying decision services is recommended. The rationale for this approach is simple – business agility depends on scalable and extensible decision automation, which normally suggests the use of a separate business rule management systems (BRMS). By adopting the correct practices when implementing IBM BPM-IBM ODM solutions, developers can optimize the flexibility and responsiveness to respond to the changes required by the business.

### The evolution of business process management and rules

Traditional business process management solutions normally required that decision services be implemented in code. Therefore, decision services were tightly coupled to the application in first generation business process management solutions. Any change in the underlying decision required the development, testing and deployment of a new version of the process application. In fact, many current business process management solutions still relegate users to this development model with rules that are embedded in the process application, which is a bit scary.

In some cases, there are decisions that should be implemented directly in the process application. Examples include field-level validations, simple routing decisions, and other static and simple decision services. However, in most cases, developing rules directly in IBM BPM presents significant constraints for supporting a flexible approach to both implementing and governing decisions. IBM BPM provides a basic embedded rule capability to develop if-then-else rules that supports the ability to manage a vocabulary based on the process application data model. The embedded rule function enables the creation of action rules (based on the IBM ODM software development kit). The embedded decision table function in IBM BPM does not use the software development kit. As a general principle, the use cases for implementing rules with the embedded function are limited. In most cases, developers should use JavaScript for developing code-based rules if they decide not to use a complete BRMS, like IBM ODM. This tutorial shows how offering a well documented set of patterns for integrating rules as a separate architectural component is an essential capability for decision management.

Over the past few years, a second-generation process architecture has emerged that focuses on implementing decision services as a decoupled, process-rules solution. This approach is exemplified by the IBM Smarter Process platform and is a natural extension of emerging requirements for enterprise-caliber BRMS. The Making Better Decisions Using IBM WebSphere Operational Decision Management IBM Redbooks publication is a useful guide to understand the rationale for BRMS.

Because rule-based patterns separate decision services into a separate rule application, they offer greater flexibility and adaptability if decision logic changes. As a result, developing rulesets
Implementing Smarter Process decisions

In IBM ODM and supporting a loosely coupled integration pattern with SOAP-XML or REST-XML and JSON web services in IBM BPM is a cleaner and more architecturally sound approach to managing dynamic decisions in a business process application.

Consider the architectural decisions for when to use IBM ODM or embedded IBM BPM rules. The following sections and the solution example focus on these decisions.

Options for IBM BPM-based decision services

Consider the following options for implementing decision services in an IBM BPM application:

- JavaScript or Java (code-based) – This pattern involves developing the rules or decisions using standard code, either with JSAPI or potentially as Java objects are invoked from the process application. Although Java can be used, it requires additional configuration to enable the integration of the Java artifacts with the process application. JavaScript provides a simpler approach because the component can be executed directly as part of the process application. However, JavaScript is normally not a good implementation approach for decision services, as you will see later in the tutorial. If developers do decide to implement rules directly in the IBM BPM solution and not use IBM ODM, then a code-based solution (for example, JavaScript) is the recommended approach.

- IBM BPM rules – IBM BPM contains an embedded rules engine as previously discussed. The rules in IBM BPM are enabled with the embedded IBM ODM editor that provides an easy to use interface for creating rules using Business Action Language (BAL). This development functionality is built with the IBM ODM technology. The rules solution allows the creation of basic if-the-else rules (called Action Rules) and decision tables. These capabilities are further described in the Improve Design and Deployment of Processes Using IBM Business Process Manager V8.0 IBM Redbooks publication as well as the IBM BPM documentation in the IBM Knowledge Center. In most cases, this option is not a recommended pattern for rule implementation in IBM BPM.

- IBM ODM – You can use IBM ODM to externalize rules and manage them as separate artifacts. In most cases, this is the preferable solution for implementing most decision services. This option is a key focus for the remainder of this tutorial.

Key criteria for externalizing rules into IBM ODM

There are five criteria to consider when deciding to externalize rules and manage them in IBM ODM.

Complexity

Consider the underlying complexity and sophistication of the required decisions. Although action rules and decision tables in IBM BPM enable the creation of rules, there is no concept of rule orchestration in IBM BPM. In IBM ODM, rule flows provide an elegant solution for chaining multiple rules together. With IBM BPM, changing multiple rules is accomplished through IBM BPM rule artifacts (for example, action rules and decision tables) and each successive task requires mapping business variables as part of the implementation. As a result, it is difficult (but not impossible) to develop and maintain complex rules in IBM BPM. IBM BPM embedded rules can be used to support simple validations and transforms, but complex decisions are
more easily designed and managed using IBM ODM. Consider using IBM ODM if you have an environment with any of the following complexities: decision services that comprise a large number of variables or a large number of rules, rules that need to be managed as separate versions based on compliance, requirements for rule inference (support for use cases where rules modify rule data objects and require rule re-evaluation), regional variations, or other factors that influence the selection of potential rule artifacts at the point of execution.

As an example, a current IBM client evaluated options for developing an account opening application in IBM BPM. Our IBM team initially considered using the embedded rule functionality with IBM BPM, but the functional requirements indicated a need for orchestrating multiple rules to perform eligibility and for the decisions surrounding potential account options to offer to the actual customers requesting the new accounts. We found that using IBM ODM to enable chaining of rules provided a cleaner and more manageable solution for implementing the decisions.

**Lifecycle management**

In short, externalized rules are much easier to change when compared to an embedded rules approach. By moving the rules into a separate logical container, rules can be managed and deployed more quickly and at a lower cost compared to rules embedded directly in a process application. Placing rules in the IBM BPM embedded rules solution mandates that changes in the rules require a redeployment of the underlying process application.

Decision requirements often change frequently. By managing them in a separate development lifecycle, organizations can provide higher flexibility in responding to business requirements. As shown in Figure 1, managing the rules as part of a decision service lifecycle enables quickly deploying rules without redeploying the application.

**Figure 1. Rule management lifecycle**

The graphic shows that lifecycles of decision services and the business process can be managed independently. Decisions can be modified as business requirements change and evolve. Decisions and processes are effectively managed asynchronously. Teams responsible for managing decisions and policies, and policy managers can design and deploy new rules based on market
trends, regulatory requirements, or other changes that affect decision service articulation, performing these functions independently of the process application lifecycle. Maintaining rules in an externalized rules system lets you control decision changes outside of the base application.

Additionally, externalizing the rules enables the testing and management of the decision services as a separate task, which increases the quality and the reliability of the rule artifacts. For example, a separate governance policy that manages regulatory compliance rules is often governed the different developers and business users. Using a BRMS enables decisions to be changed and deployed as a separate set of tasks and enables higher agility in the face of changes to compliance requirements.

Although organizations can use IBM BPM toolkits to manage IBM BPM embedded rules, the option still requires redeployment of the toolkits and associated testing requirements, which is not agile. By maintaining the rules in IBM ODM, rule changes can be deployed in a matter of minutes instead of months. This deployment advantage is particularly critical for rules and decisions that change frequently.

As an example, an IBM customer uses IBM ODM to manage offers for clients through a myriad of different channels, including ATM, web, and mobile interfaces. Offers can change frequently. By externalizing the rules from the various applications that require offer management integration, the organization can quickly change the offers based on new events and market trends. The IT team often redeploy two to three rule applications each week. This requirement mandates an approach based on IBM ODM to ensure a nimble and agile approach to dynamic offer management.

Reuse of decision services

In many cases, decision services must be used by multiple applications. If these applications are all IBM BPM applications, a case can be made for managing the decision services in an IBM BPM toolkit. However, most of the time, other applications, for example, web applications and data management applications, also need to enable access to these decision services. In many cases, there might be other future applications that require the decision services, which might not be known when developing the initial application. The challenges are clear: either manage the decision services centrally using a tool such as IBM ODM or design and deploy the rule artifacts within each application that requires them. Aside from the lifecycle development and testing issues previously discussed, a key risk is ensuring the consistency of the decisions that need to be made. If you have "replicated" functionality in multiple applications, the approach mandates that each application must ensure that the rules are consistent. The mandate becomes even more difficult as the decisions become more complex or volatile. The upfront development costs are not the key challenge. The real issues are the associated maintenance and governance required to change each instance of the rules in the various applications and the potential risk to data and application inconsistency if the rules are not managed and changed consistently across the enterprise.

An example of reuse involves an IBM banking client that provides loan eligibility rules in IBM ODM. This functionality is used in four different applications that serve retail, small business, corporate, and wealth management clients. The small business loan management application is developed in IBM BPM. By providing decisions in IBM ODM, the client can maintain the rules as a shared
component, which ensures that all applications implementing this set of decisions use the same governed set of rules.

**Business and IT collaboration**

Business users are not normally interested in the details of application implementation. However, they often want to maintain an active role in the specification, testing, and managing of decisions. The ability for business users to have a role in the design, development, testing, and deployment of rules and policies is an optimal function in many business applications.

At the most basic level, business rules and decisions in IBM BPM and IBM ODM are written in an intuitive language, which is easily understood by business users. This simple approach provides a higher level of potential collaboration between business and IT departments.

Take the collaboration one step further: In many cases, business users require direct access to the rules and decision services to review decisions, and in some cases, the ability to modify and extend the rules. Normally the modification of rules is managed as part of a formalized and controlled governance and release policy, but there are many cases where knowledge workers are granted limited management of decision services. If you embed the rules within an application, granting limited management is possible, but it requires creating both user interface components to support this functionality. More importantly, modification of embedded rules lacks the required governance and release management processes needed to ensure the quality and consistency of the rules. IBM ODM provides integrated support for governance and release and deployment management to ensure the consistency and accuracy of the business user initiated modifications.

In the Decision Center in IBM ODM, you can give business users access and change management capabilities in a controlled and governed way. In IBM ODM V8.7, this functionality has been expanded to include business user testing and rules simulation, as well as a governed approach to rule lifecycle management. No custom development is required. This functionality is becoming a growing requirement in industries where subject matter experts on the business team require the ability to change decisions based on changes in the industry and market trends.

As an example, our IBM team recently worked with a client designing a payment processing solution. The client uses Decision Center to enable business users to review the payment processing decisions and to allow specific business users to modify the decisions. The IT team focuses on application implementation aspects and acts in an advisory role to business users to ensure that the right decisions are made and to support rule governance for the decisions.

Similar requirements are emerging for rule scoping and design. Business users want to be involved in the initial requirement analysis and design process for decision services. As part of Smarter Process, IBM Blueworks Live provides users with the ability to model decisions, which is discussed in the *Discovering the Decisions within Your Business Processes using IBM Blueworks Live* IBM Redbooks publication. Blueworks Live is mapped to the emerging work by the Object Management Group (OMG) to develop the Decision Modeling Notation (DMN) standard. This capability provides the required functionality to enable business and IT collaboration on decision service design and development.
Decision services management and visibility

By managing rules externally in a rules system such as IBM ODM, the following additional functions are enabled and can provide ancillary benefits for the decision services:

- Design and development of rules with IBM ODM provides the ability to centrally manage the authoring and testing of decision services as a separate function from the application. This functionality enables developers to support additional simulation functions such as champion-challenge and what-if analysis.
- Centralized management of rules provides for direct governance of decisions and helps ensure the consistency and validity of the decisions across multiple applications by function, domain and user and developer roles.
- The Decision Warehouse tool in IBM ODM enables organizations to support auditing and visibility of decision operations, such as determining what decisions are made and when. This is particularly critical for decisions that provide regulatory and compliance operations where internal and external auditing is needed to provide oversight and analysis of decision outcomes.
- At the same time, it is important to recognize that externalizing the rules is not a panacea. By externalizing process decisions into IBM ODM, architects and developers must understand the following concepts:
  - Executing rules in IBM ODM incurs additional call overhead for processing decisions and requires that the IT team evaluate different IBM ODM deployment approaches based on requirements. See the Best practices for designing and implementing decisions services series by Jerome Boyer to learn about the options that should be considered.
  - Implementing rules in IBM ODM involves additional solution complexity in the implementation of the operational solution architecture and ongoing management, for example availability and performance management due to additional components like Decision Server that are required in the end-to-end solution.

Note that both of these challenges can be easily mitigated with good architecture practices and up-front evaluation of key architectural decisions when designing and implementing solutions.

As a final consideration, comparing the IBM BPM embedded rules function and implementing rules in IBM ODM, there is a capability to migrate embedded rules to IBM ODM. As described in the IBM BPM documentation, embedded rules in an IBM BPM application can be exported to a rule project and then imported into the IBM ODM Rule Designer. The ability to migrate extends the rule functionality through support for ruleflows and a common vocabulary. However, each action rule is exported as a separate project. Consider the impact: If you have 10 action rules, this results in 10 rule projects with separate business object models, which is untenable. This approach normally requires the implementation that developers must refactor and re-implement the decisions. It is optimal to just start with an IBM ODM rules approach if the previously described decision criteria are present in the decisions that are being scoped and designed in the proposed IBM BPM application.
Install the stock review IBM ODM application

Before you can run the Stock Rating process example in this tutorial, you must install an IBM ODM application and then import and configure a process application in IBM BPM. This tutorial requires that you have IBM BPM V8.5.x and IBM ODM V8.7 already installed.

First, install the IBM ODM rules application. Go to the Download section of this tutorial, download the code_sample.zip and extract the files. You see following files:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stocks Demo – rules.zip</td>
<td>Corresponds to the workspace with the Decision Service Main rule project used</td>
</tr>
<tr>
<td>ruleApp_StocksDemoRuleApp_1.0_ddMonthYYYY.jar</td>
<td>Corresponds to the RuleApp file that can be directly deployed to Rule Execution Server.</td>
</tr>
<tr>
<td>res-request-1.xml</td>
<td>Sample requests to test the decision service using the REST Test option from the Rule Execution Server</td>
</tr>
<tr>
<td>res-request-2.xml</td>
<td></td>
</tr>
<tr>
<td>res-request-3.xml</td>
<td></td>
</tr>
</tbody>
</table>

Then, test the rules application with one of the following two options:

1. Ruleapp deployment only: This allows the execution of the StocksDemoRuleApp from any web services client or the provided IBM BPM application.
2. Import workspace into Rule Designer and deploy: This provides all the rule asset details described in the previous section. From Rule Designer, the rule project can be synchronized with Decision Center and deployed to Rule Execution Server.

Option 1: Deploy the StocksDemoRuleApp

To deploy the rule application only, complete the following steps:

2. Select Explorer.
3. From the RuleApps view section, click Deploy Ruleapp Archive.
4. Browse for the ruleApp_StocksDemoRuleApp_1.0_dd_MonthYYYY.jar file that you downloaded and saved.
5. Keep default values and click Deploy.
6. Confirm the StocksDemoRuleapp file is listed in the ruleapps area.

Complete the following steps to test the rule application:

2. Select the Explorer tab.
3. From the RuleApps list, click the StocksDemoRuleApp rule application name.
4. From the RuleApps View section, click the StocksDemoRuleset ruleset name.
5. From the Ruleset view, select Retrieve HTDS Description File. From the Retrieve HTDS Description file section, make sure the following options are selected, as shown in Figure 2:
   - REST as the Service protocol type
• the latest ruleset version
• the latest rule application version

Figure 2. Retrieve HTDS Description File

6. Click **Test**. A new window opens with two sections: Execution Request and Server Response. Make sure that **XML** is selected for the **Execution Request**.
7. Replace the content for the execution request with the content of one of the XML files from the downloaded code sample, for example: `res-request-1.xml`.
8. Click **Execute Request**.
   The Server Response should now have the results of the execution.
   For more information, see the IBM ODM documentation on IBM Knowledge Center:
   • Deploying Ruleapp archives
   • Testing a ruleset for REST execution

Option 2: Import the workspace into the Rule Designer

To import the workspace into Rule Designer and deploy the rule project, complete the following steps:

1. Extract the **Stocks Demo – rules.zip** file, which contains the workspace, in a temporal location, for example: `c:\temp\devworks`.
2. Open the Rule Designer.
3. Select the workspace location, for example: `c:\codes\bpmodm`.
4. Select **File > Export**.
5. From the export window, expand the **General** section and select the **Existing Projects into Workspace** option.
6. Click **Next**.
7. Browse for the location where you extracted the content of the workspace file in the **Select root directory** section. Make sure the **Copy projects into workspace** check box is selected.
8. Click **Finish**. The **Stock – demo – rules** rule project should be displayed in the Rule explorer section on the left.
9. To synchronize the project with Decision Center, complete the following steps:
   a. Select the **Stock – demo – rules** rule project.
   b. Right-click and select **Decision Center > Connect**.
   c. Provide connection information, for example:
      • **URL**: `http://localhost:9080/teamserv`
• User: rtsAdmin
  • Password: rtsAdmin
d. Click Connect. The window is refreshed and the section at the bottom is enabled now.
e. Make sure Create new project on Decision Center under Project configuration is selected.
f. Click Finish.

Complete the following steps to deploy the rule application into the Rule Execution Server:

1. From the Rule Explorer view on the left, expand the deployment folder.
2. Double click the Stocks Demo – deploymentConfiguration deployment configuration.
3. From the Deployment section on the right, click Proceed to Ruleapp deployment.
4. If a message related to Java version opens, click OK.
5. Click Next on the Deployment Summary page.
6. Click Next in the credentials section.
7. Click Finish.

To test the rule application, follow the steps from the Deploy the StocksDemoRuleApp section related to testing the deployed rule application.

For more information, see the following IBM ODM documentation on IBM Knowledge Center:

• Importing projects into Rule Designer
• Publishing decision services to Decision Center
• Deployment configurations

Install and configure the example IBMStockRating.twx application

After you have installed the rule project, you can import the Stock Request process application (IBMStockRating.twx) into the IBM BPM Process Center:

1. Import the process application into IBM BPM.
   a. From the Process Designer, open the Process Center.
   b. Click Import Process App as shown in Figure 3.
      
      **Figure 3. Import Process App**

      ![Create New Process App](image)
      ![Import Process App](image)

   c. Click Browse to search and select the TWX file that you have downloaded. Click OK. You see that other toolkits will likely be imported, such as the BP3 Brazos UI Toolkit.
   d. Click Open in Designer and open the application in Process Designer. You now have the Process open in Process Designer.

2. Configure the process application and the business process to run in your environment.
a. In the Process App Settings, ensure that the server location is correct. The URL shown in Figure 4 is http://localhost:9082. This is the host and port to access the Rule Execution Server console.

**Figure 4. Process App Settings Servers tab**

![Process App Settings Servers tab](image)

b. Expand Processes and double-click the Stock Rating Approval Process to open it. Go to the Diagram tab to review the business process. As you can see in Figure 5, this process consists of 3 swim lanes, which you map to appropriate users.

**Figure 5. Stock Rating Approval Process diagram**

![Stock Rating Approval Process diagram](image)

c. Select the **Junior Analyst** lane and review the behavior pane, as shown in Figure 6.
d. Click **Junior Analyst** to map a user to that role. In the example in Figure 7, Amy is added to the IBM BPM environment. Click **Add User** and select a user to map it to the role. Optionally, you can map the role to a group.

**Figure 7. Mapping a user to the Junior Analyst role**

```
Figure 6. Junior Analyst lane behavior
```

```
Behavior
Default Lane Team: Junior Analyst
Is System Lane: [ ]
```

e. Follow the same steps to map users or groups to the Senior Analyst role. First, select the **Senior Analyst** swim lane in the business process diagram and then click **Senior Analyst** in the Behavior section. Then map a user or group to the role. In this tutorial, Linda is the senior analyst.

3. Configure the decision service in the business process.
   a. Review the **System** lane. This is where the Run Rating Rules decision service runs.
   b. Double-click the Run Rules service, which is an IBM ODM decision service. Click **Properties** and select the **Implementation** tab, as shown in Figure 8.
c. Ensure that the SOAP Port for the RES Server is correct. In the figure the SOAP port is 8882. Enter the user name and password for the RES Administrator and click **Connect**.
If you get errors, ensure that the SOAP port for the RES Server is correct. (Navigate to the Ports section in the WebSphere Application Server administrative console). Ensure that the user name and password are correct.

d. After successfully connecting, click the Rule App field in the Process Designer and select the **StocksDemoRuleApp** that you have downloaded and installed.

e. Select the **StocksDemoRuleSet** and select the latest version. Alternatively you can select a specific version if needed. This is especially important if the ruleset version changes and the business process needs to invoke a specific version. However, for simplicity, this tutorial example selects the latest as the version.

f. Review the Data Mapping tab, as shown in Figure 9. Don't change anything as part of this tutorial. However, if needed, you can change the decision ID to any other string.
Run the Stock Rating process example

A simple process for stock pricing analysis illustrates the recommended approach for using IBM ODM to support decision services. The scenario is intentionally simple: a stock analyst wants to determine the buy/sell/hold action on a given stock based on key information on company performance. The process involves the entry of the given company and related information, the execution of the decision service, and a response returned to a senior analyst for subsequent action. Although the process could have been further elaborated, for example, including the integration of case management functions, monitoring and visualization, requests for additional documentation, and event notifications, this tutorial keeps the process simple to provide a tangible example to support the premise of using IBM ODM compared to either coding the rules in JavaScript or using the IBM BPM embedded rules functionality.

In this section, you run the IBM BPM application, which invokes the decision service, and then you review the result. The tutorial uses the Brazos UI Toolkit from BP3 to implement the coaches in the human services.

1. Make sure you have completed the steps in the Install the stock review IBM ODM application and Install the example IBMStockRating.twx sections.
2. Open IBM BPM and start the Process Designer.
3. Run the IBMStockRating application (the installation of the application and the IBM ODM rules application are covered in the following sections). Select the Stock Rate Approval process and click the button to run this process. Review the application with the Inspector, as shown in Figure 10.
Select the inbox and click the green button to run the process step. You can also run this process through the IBM Process Portal as Amy or any other user that you have mapped to the role of a junior analyst. If you click the Work Tab and then click **Stock Rating Approval Process**, you can launch the process, as shown in Figure 11.

**Figure 11. Stock Rating Approval Process at run time in the My Work tab in Process Portal**

In either case (running the process through Process Designer or Process Portal), you see the initial Stock Rating Request coach.
4. Enter IBM in the Ticker field, as shown in Figure 12. The coach is populated with default values. You can modify these values. For the initial test, just accept the defaults and click Submit.

**Figure 12. Enter Stock Rating Request at run time**

5. Click Submit. The human task completes and the business process definition invokes the decision service that is running on IBM ODM with the stock parameter and gets a response back, which is the rating of the stock. The rule conditions are based on growth, earnings per share, price-to-earnings ratio, and the industry.

6. Run the second human task that receives the results from the decision service in one of the following ways:
   - In Process Designer, refresh the task and run the next step.
   - If you prefer to use the Process Portal like the end users will do, log in as Senior Analyst in Process Portal. In this example, Linda acts in the role of the senior analyst. If you log in with the linda user name with linda as a password, you see a task waiting for you to review. Also note that in the launch section, Linda does not have the ability to launch the stock rating approvals process, because only the junior analyst role has this permission.

7. As the senior analyst, click **Review Ratings Result** and scroll down to see the messages and the ratings that are sent by decision service. Click **Done** to complete this process, and the My Work tab looks like Figure 13.

**Figure 13. My Work tab**

8. Now you can review the underlying decision service implementation from the perspective of the process.
• In Process Designer, review the Stock Rating Approval Process: in the Analyst lane, double-click the view rating result.
• Click **Set Rating message**. The decision service returns a list of messages. The implementation of the Set Rating message contains the code to extract the messages (sent from decision service) from this list, as shown in Figure 14.

**Figure 14. Implementation for Set Rating Message**

Review the implementation of the Stock Rating decision service in IBM ODM

Now you review the actual implementation of the rules in IBM ODM.

In retrospect, the IBM BPM application could have provided the rating decision. However, you see in the following sections why the introduction of a BRMS provides an optimal approach for implementing the decision services.

At this point, look at the implementation of the decision service and consider why it is preferable to implement this decision service in IBM ODM.

As mentioned in the previous section, the use case is related to a Stocks Rating process. The business rules application receives the stock information, based on some simple business rules that determine performance of stock and rating recommendations (Sell, Hold, or Buy).

The business rules application in this tutorial was implemented in IBM ODM V8.7. The application consists of one Decision Service Main rule project: **Stock-rules**, with all required artifacts. For the integration, the hosted transparent decision services (HTDS) feature of Decision Server is used to provide a web service, which IBM BPM uses to invoke the rules, as shown in Figure 15.
Figure 15. Execution of operations

The IBM BPM process invokes the execution of the decision service. You pass a stock as the request and receiving a response of the Stock Response type. When invoking the web service, the process specifies the rule application needed, in this case: StockDemoRuleApp/StocksDemoRuleSet.

From the authoring perspective, the business rules artifacts used for this scenario can be summarized in Figure 16.

Figure 16. Business rules artifacts for Stocks Demo

The rules development lifecycle in Figure 17 illustrates the details of the application.

Figure 17. Rules development lifecycle
Discovery/analysis/model phase

The analysis and modeling phase is very simple, because you only need to perform a simple stock validation. Rule harvesting includes both rule discovery and analysis. During this phase, business policies or requirements are identified in preparation for the authoring of the business rules.

Modeling of the rule application involves defining the artifacts that you are going to use for the application. This tutorial example, based on the simple scenario, uses a dynamic execution object model (XOM), which is based on a schema definition file. The XOM is imported to create the Business Object Model (BOM). Note that the use of an XSD to create a XOM is often an anti-pattern. Modeling of rules based on a specific application is not a good practice – rule design should normally be done as an independent task, which is detailed in Jerome Boyer and Hafedh Mili's book *Agile Business Rule Development: Process, Architecture, and JRules Examples*. Figures 18 and 19 show the Stock class and the Stock Response class in the BOM, which was imported from the XSD file.

**Figure 18. Members of the Stock class in the BOM**

```
<model>
  <com>
    <ibm>
      <stocks>
        <Stock>
          <name>
          <company>
          <ticker>
          <price>
          <sharePrice>
          <currentPrice>
          <toString()>
          <equals()>
          <compareTo()>
          <addToList()>
          <removeFromList()>
        </Stock>
      </stocks>
    </ibm>
  </com>
</model>
```

**Figure 19. Members of the StockResponse class in the BOM**

```
<model>
  <com>
    <ibm>
      <stocks>
        <StockResponse>
          <messagesList>
          <stockName>
          <stockRating>
          <stockTicker>
          <StockResponse()>
          <addToList()>
          <removeFromList()>
        </StockResponse>
      </stocks>
    </ibm>
  </com>
</model>
```

Discovery and authoring phase

Authoring rule projects involves the following decisions:

- Organization of rules: Definition of how the business rules will be put together in a folder organization structure. Organizing rules allows easy access to business artifacts, especially when you have a large volume of rules. Organizing rules also provides a way to classify rules that are common to different scenarios which means that you can reuse them on multiple
cases. Some customers organize their rules by business units, regions or product lines. This organization is something that business analysts, in conjunction with subject matter experts decide during the analysis phase. The rule project sample in this tutorial implements two rule packages: rating and validation.

- Orchestration of rules: During the execution, the rule engine needs to know what set of rules must be evaluated in what order. IBM ODM provides a rule artifact called ruleflow that provides this functionality. Orchestration is very important because it allows you to use a full set of business rules and returns with the result of a complex decision. A ruleflow is one key difference between running rules embedded in IBM BPM and using the business rules approach in IBM ODM.

Figures 20 and 21 present the rule organization and orchestration used in the sample application for this tutorial.

**Figure 20. Rule project organization**

![Rule Explorer](image)

**Figure 21. Main flow orchestration**

![Orchestration Diagram](image)
IBM ODM provides the following components for authoring:

1. Rule Designer: An Eclipse-based development environment targeted for IT users to create the rule project framework, including the verbalization and orchestration of the rules and other authoring tasks involved during the rule project development cycle. Diagrams presented in Figures 20 and 21 were extracted from Rule Designer.
2. Business console and Enterprise console: A set of web interfaces where end users can do all the rule management of the rules, for example: creating, updating and deleting artifacts, taking snapshots, performing deployments, and implementing a governance framework.

The following screen capture examples are from the Business console, which is what end users experience during the rules authoring phase.

The Decision service main rule project was synchronized with Decision Center to continue with the next activities. Figure 22 shows the demo application available in the decision services library.

**Figure 22. Decision services library**

![Decision Center](image)

As previously shown, the rules organization of rules was performed in 2 rule packages, but referred to folders in Decision Center. Figures 23 and 24 present the way rules are visible in the Business console.

**Figure 23. Rules organized by folder**

![Rules organized by folder](image)
Figure 24. Rules organized by name

You can go to the Business console on Decision Center to view implemented rules. See Testing and simulating rulesets in the IBM ODM documentation.

Validation phase

For a basic testing of the rules, IBM ODM testing capabilities in the business console allow you to ensure that the rules are working properly. This functionality is not meant to replace formal testing of rules, but it provides rule validation for business users. You can create a testing file with a list of scenarios like the testsuite-v1.xlsx file in sample files for this tutorial. When working with complex decisions, it is very important to test all business rules authored. IBM ODM provides the Decision Validation Services capability for performing these types of tests. To learn more about testing and validating rules, see Testing and simulating rulesets in the IBM ODM documentation.

Deployment phase

The deployment process of the rules in this scenario is performed from Rule Designer and Business console. In preparation for the deployment, a deployment configuration is created. See Deploying from the Business console in the IBM ODM documentation.

Summary of considerations

As previously discussed, there are other ways to implement the decision service. On one hand, you can develop the decision service in code using either JavaScript or Java. A code-based approach is valid for simple if-then-else conditions that are static. However, the stock recommendation example is neither simple nor static. Building this decision service in code would be possible but would lack the adaptability and flexibility needed for the final solution implementation.

The other option is to develop the decision service using the embedded rules functionality in IBM BPM. As you have seen, there are a number of steps in the actual decision service. This means that you need to have multiple decision modes in the actual IBM BPM decision service. This makes the code difficult to manage given the complexity of the rule execution.

So evaluate your approach using the set of criteria previously discussed:
Complexity | The rules are of a medium complexity. IBM ODM provides a more optimal solution implementation than other alternatives.
----|---
Lifecycle management | Stock rating is a potentially dynamic decision. By externalizing this in IBM ODM, the underlying rules can be more easily managed as business requirements change without incurring the delays associated with a typical application lifecycle.
Reuse of decision services | The rating decision has applicability to potentially other business applications. By externalizing this in IBM ODM, the decisions can be used in a consistent and governed manner across multiple applications.
Business and IT collaboration | IT developers are not (in most cases) stock rating subject matter experts. Development of the decisions in IBM ODM provides a more collaborative approach in the design, development, and testing of the solution between business and IT.
Decision services management and visibility | By maintaining the rules in an English-like syntax and with intuitive modeling, business users can more easily understand the underlying decision services without the complexity of reviewing code or needing to understand the process application details.

If you look at the other options discussed for implementing the decision services, you can quickly review the appropriateness of the approach for this example.

<table>
<thead>
<tr>
<th>Alternative</th>
<th>Solution approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implement decision logic in JavaScript in IBM BPM</td>
<td>The complexity of the rules result in the creation of multiple lengthy JSAPI routines, which would be hard to manage and limit the reuse of the decision across other applications. Rule changes would require a redeployment of the process application. In addition, the reuse of the component across other applications would be minimal (though implementing in a toolkit would provide some reuse across other IBM BPM applications).</td>
</tr>
<tr>
<td>Implement decision logic in the rules editor in IBM BPM</td>
<td>This option provides a more intuitive approach for the business users than using JavaScript. However, it requires multiple rule steps, which is not optimal. Rule changes would require a redeployment of the process application. In addition, the reuse of the component across other applications would be minimal (though implementing in a toolkit would provide some reuse across other IBM BPM applications).</td>
</tr>
<tr>
<td>Implement decision logic in a Java application and integrate from IBM BPM</td>
<td>This option would be moderately difficult to implement. Business user collaboration on the design and modification of the decision would be nearly impossible. At the same time, rule changes would not necessitate a redeployment of the process application. The component could be reused across other applications, but governance and management of the asset would need to use traditional SDLC rigor.</td>
</tr>
</tbody>
</table>
| Implement as a decision service in IBM ODM | This is definitely the optimal approach, as shown in the Decision criteria table.

**Conclusion**

This tutorial looked at the key criteria for determining whether to use IBM ODM as part of an IBM BPM solution. You learned the following criteria to consider:

- Complexity – In most case, decision services of medium or high complexity benefit from an externalized rules approach for example, IBM ODM. IBM ODM provides a number of additional functions for example modeling, simulation which provides organizations a powerful set of capabilities to design and deploy complete decision management solutions – the IBM BPM-based rules approach lacks rigor for anything other than simple rules.
• Lifecycle management – By enabling independence between the rules and application lifecycles, organizations benefit from higher agility and flexibility offered by managing decision services separately from the application development process.

• Reuse of decision services – Though some reuse is possible for decision services given use of toolkits or code-based decision services (for example Java or web), the built-in governance and management of decision services within IBM ODM provides a more optimal approach given decision services need to be enabled for multiple applications.

• Business and IT collaboration – Embedding rules in process applications or using externalized code for example Java and Web restricts the access and the ability for business users to be closely involved in the overall rule design, development and management processes.

• Decision service management and visibility – Lastly, the extended functions of the IBM ODM solution promote a higher level of usability in terms of managing and controlling the decision services for an enterprise which is important for key decision services in use by an enterprise.

Borrowing a line from Forrest Gump, decision management and process management go "together like peas and carrots." Decision Management and Process Management are complimentary to one another, however there are multiple considerations on how to integrate decision services within an application.

Acknowledgements

The authors want to acknowledge and thank Jerome Boyer, Stuart Jones and Bertrand Portier for their thoughtful and immensely helpful review of this article – it is great to have their insight and skills in ensuring a quality discussion of these issues.
### Downloadable resources

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code sample</td>
<td>code_sample.zip</td>
<td>20136KB</td>
</tr>
</tbody>
</table>
Related topics

- Decision Management Systems, an IBM Press book by James Taylor
- developerWorks series: Best practices for designing and implementing decision services
- Agile Business Rule Development, a book by Jerome Boyer and Hafedh Mili
- Making Better Decisions Using IBM WebSphere Operational Decision Management, an IBM Redpaper publication
- Discovering the Decisions within Your Business Processes using IBM Blueworks Live, an IBM Redpaper publication
- Improve Design and Deployment of Processes Using IBM Business Process Manager V8.0, an IBM Redbooks Solution Guide
- Business Process Management with IBM Business Process Manager, an IBM Redbooks Solution Guide
- Object Management Group
- IBM Operational Decision Manager product page
- IBM Operational Decision Manager documentation on IBM Knowledge Center
- Top 10 editor's picks for IBM Operational Decision Manager
- IBM Operational Decision Manager Developer Center
- developerWorks Middleware