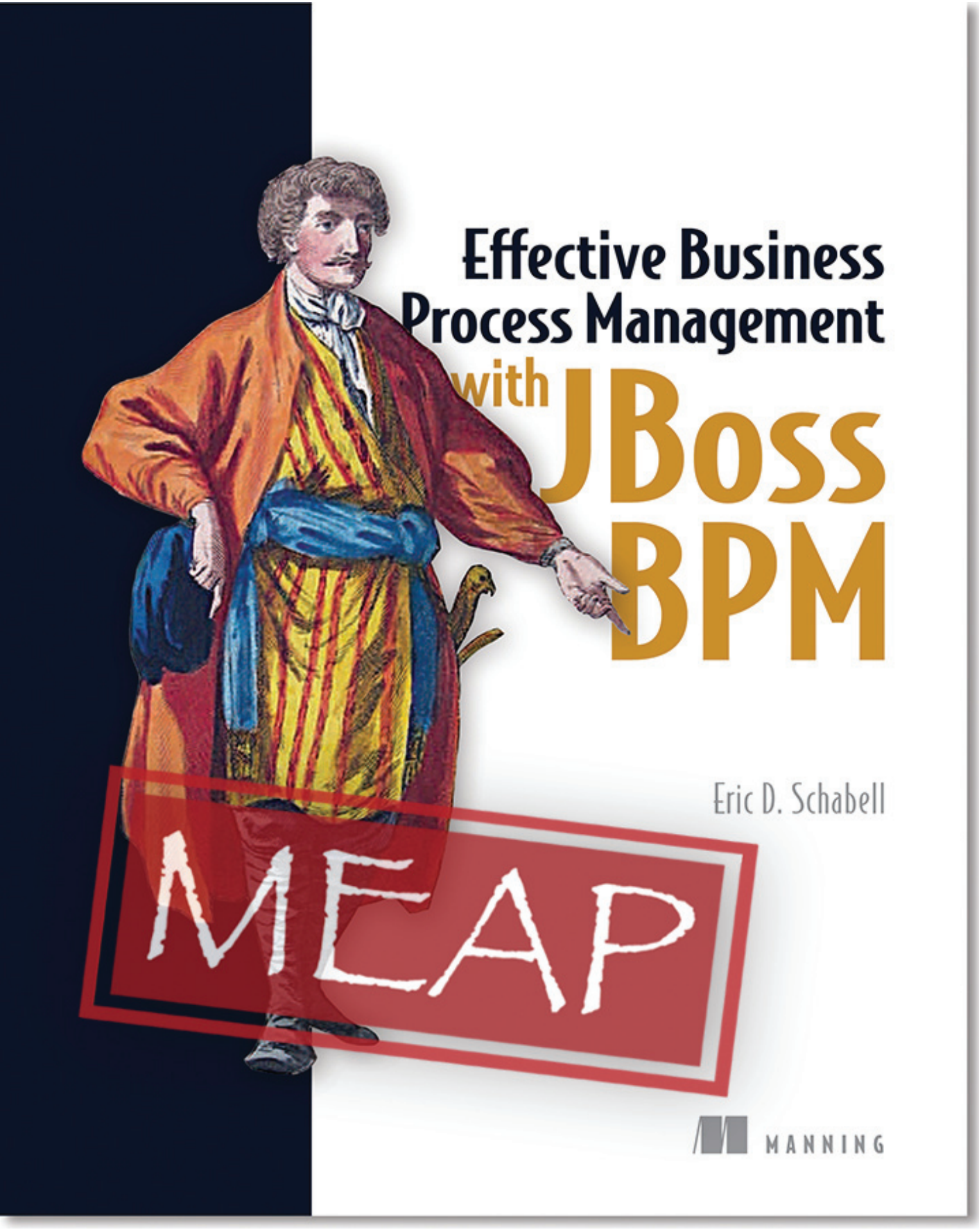
Effective Business
Process Management
with JBoss
BPM
Eric D. Schabell
MEAP
MANNING

MEAP Edition
Manning Early Access Program
Effective Business Process Management with JBoss BPM
Version 6

For more information on this and other Manning titles go to
www.manning.com

welcome

I would like to personally thank you for purchasing the MEAP for *Effective Business Process Management with JBoss BPM.* The foundation of this book comes from my start in JBoss BPM many years ago, when I started to work daily with JBoss Business Process Management (BPM) technologies. All the work and time spent implementing, teaching and writing about JBoss BPM has finally arrived at the stage where I am able to start sharing my experiences with you in this book.

This is an intermediate level book designed for developers, process analysts and architects engaged in designing or writing applications that leverage JBoss BPM. I have received several requests from users who have read, used and engaged with my JBoss BPM content online. The one that kept resurfacing was the request to put my material into a cohesive book covering the practical usage of the JBoss BPM.

I have tried to make this technology understandable and easy to consume for the readers of this book by providing practical examples that take readers on a journey towards mastering JBoss BPM. Whether you are a developer, an architect or a process analyst, there should be something for everyone in this book. I have tried to address all three of these audiences, sometimes together and sometimes individually. They all have a role to play in the use of JBoss BPM for their development projects.

As you're reading, I welcome you to make use of the Author Online forum to provide feedback and comments about the content you are reading. I am looking forward to engaging with you, making use of your feedback and working together towards creating the best JBoss BPM in Action book possible. I would like to provide a book that you will want to use regularly to help you become a valuable member of any BPM project.

— Eric D. Schabell

brief contents

Part 1

Getting started

Business process management (BPM) brings together the capabilities to orchestrate all manner of systems, people and services while allowing you to structure their timeline of execution. Very few disciplines in application development will give you the rewarding feeling that BPM gives when you are able to orchestrate all the services, systems and people an enterprise has to offer its customers into process solutions.

The first part of this book focuses on practical BPM concepts and guide you in setting up your first JBoss BPM project.

1

What's in a process

This chapter covers

- Introducing basic BPM concepts
- Providing an introduction to rules, events and processes
- Discovering the community origins of JBoss BPM
- Walking through JBoss BPM Suite

This chapter introduces you to Business Process Management (BPM), and the important terminology used, as you embark on the path to learn this technology. I begin with a process and what is within the context of BPM. As you explore JBoss BPM there are three important aspects that support integrating business activities into the processes you are developing. These aspects are business rules, business events and business processes. I discuss each and provide an overview showing how each can be used to support your process development projects.

Next up is a tour of the community of Open Source projects that make up the JBoss BPM product eco-system. Projects are highlighted, specifically how project code is integrated into a supported JBoss BPM Suite product. These communities allow you to keep an eye on the research and development being done in the area of rules, events and processes. You have the ability to influence the direction of this technology by providing direct feedback or code contributions.

Finally, you are taken on a tour of the JBoss BPM Suite architecture and explore the components that allow you to develop process projects. This is an introduction to how JBoss BPM Suite supports development, testing and runtime execution of processes. Testing is integrated into the chapters where you create artifacts and these tests demonstrate that your artifacts are working correctly. For more in-depth looks at how a specific component works,

refer to the chapter devoted to it. The path you are taking in this chapter is highlighted in figure 1.1.

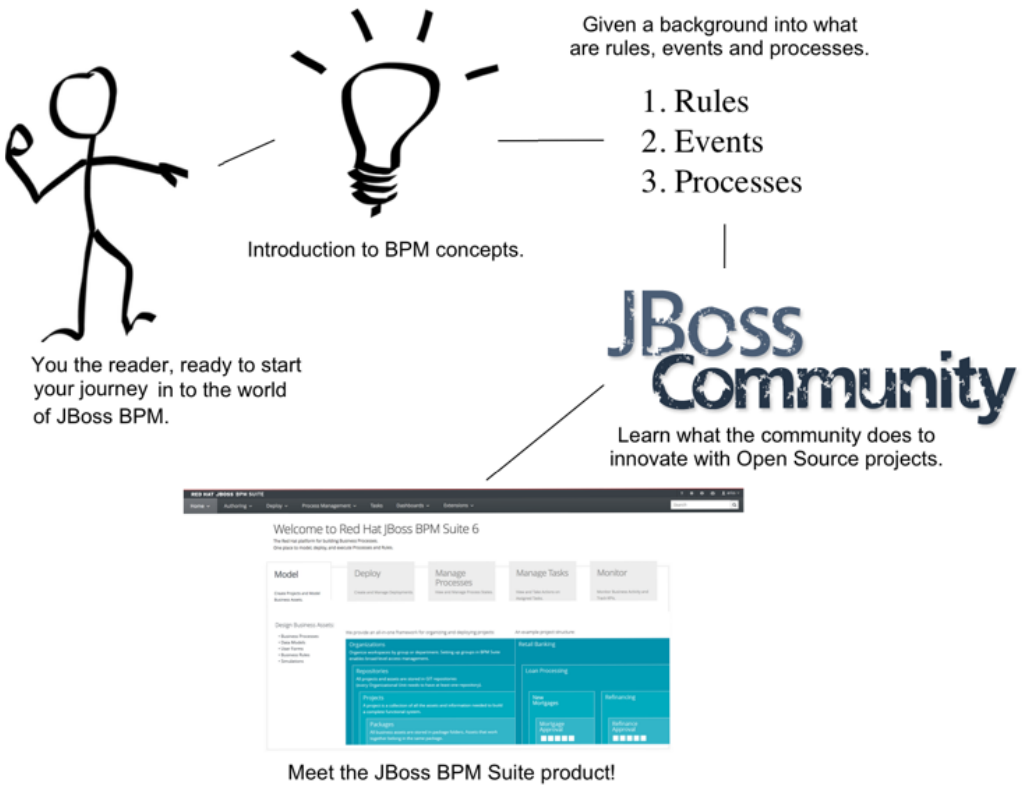

Figure 1.1. The path you the reader is taking through this chapter, where you are introduced to BPM concepts and more to get you started with JBoss BPM.

1.1 Introducing BPM

Organizations are constantly being tested in the markets they operate in by shifting expectations of their customers, and by competitors looking to provide better value at a lower cost. This tension is the catalyst that continually pushes organizations to search for ways to improve their services, improve the speed which they deliver value to their customers, enable employees to get more done with less administrative overhead, and most importantly, to constantly grow by generating more revenue. This is the basis of BPM, to be able to identify and capture processes in an organization to create repeatable, measurable and consistent execution of goals to drive their business forward.

When an organization studies its operations, it discovers there are many processes used in their daily business. These processes are often not well thought out, or were created to complete some aspect of the daily business, with little thought given to improving efficiency. At this point the organization looks for the first steps to improve their processes, through automation, that represent business value.

Business value could be anything that drives organizational goals forward to make customers happy, and thereby generate more revenue. Value to your business can be anything, such as keeping track of interaction with a customer. If that data could be captured, the marketing department could search a customer's behavioral patterns to decide what products and services to market to a particular person. It would take mass marketing out of the equation and allow for direct, specific marketing towards individual customers' needs.

While I choose to call this business value, it is sometimes referred to in this domain as knowledge and the people working within a process are called knowledge workers. Either one works, but in this book I'm sticking with business value to capture the spirit of the organizations we all work for.

It is possible to identify pieces of business value that are not worth automating because they are inconsistent or have too many potential paths to justify the effort to automate. Others require traditional human brain power, which is not easy to capture in automated process form. An example of this would be the hiring of employees, a process that can be largely automated, but the actual decision to hire a specific candidate remains a factor of human intelligence. You can automate the process of handling applications, scheduling interviews, and the post process of on-boarding a new employee once hired. Let's leave the 'hire or not to hire' task to humans.

Another important facet of capturing business value in processes is that you can monitor processes and tasks as they are completed to provide business owners with valuable information. You can provide insights into aspects that interest the business owner, and make intelligent decisions about when and where to improve a process as the business evolves.

Imagine your business is running a retail process to sell products online. This process has a user task to approve large orders and is staffed by a set number of people during certain business hours. Bottlenecks may develop in your process as the business grows. What can you do when the Christmas holidays arrive and you expect a surge in orders? Will the current staffing of the user task allow you to process ten times the number of orders? What about one hundred times the orders?

By using historical data captured in previous process instances it is possible to determine how many orders are large enough to require human approval, and on average how long each approval took. If you simulate your process using tooling provided by BPM process engines, you can set the number of humans working on the approval task in the process and set how long they take. By simulating hundreds or even thousands of process instances you can watch the results of the orders flowing through your process and determine that you need to staff your user task differently. During the normal months of the year you are able to process large orders with just two employees assigned for eight-hour days. During the holidays, due to

expected increases, simulation testing reveals a need for twenty-four hour shifts to approve large orders and you need to increase staffing to four employees. It is always better to know this before hiring new employees for the holidays and discovering it did not help process orders fast enough to justify the costs.

This process looks at a step-by-step plan to accomplish a set of tasks that deliver business value. The basic series of events that leads to process discovery begins with identifying the piece of business value to be automated. This piece of business value is selected for its potential to improve the business as a whole when:

- the process can be automated
- the process can be consistently executed in the same way
- you can clearly define human involvement in the process
- automation of the process removes current ad-hoc or inconsistent behavior
- measuring the process gains insights into current business behaviors
- insights provide a better means to decide when changes can and should be made to improve a process
- you can reduce resource waste by efficiently handling *wait-states*

Once a potential business process is identified, a workshop can be held by stakeholders from the business responsible for executing the process. For example, the human resource department might be in charge of registering new employees, getting them a workstation, starting their benefits, and assigning them an e-mail address. This process is discussed and dissected to identify the exact steps and order that they would need to be accomplished to register a new employee. This results are put in a diagram with each step, from start to finish, drawn up as tasks to be completed.

These are the beginnings of a process, known as a process diagram. A process diagram contains all the elements needed to capture the steps in the processes without any of the execution details. You see task nodes, start nodes, end nodes, transition arrows, gateways that split paths of the process, and gateways that join paths of the process together again. These are all standard elements that are part of the *Business Process Modeling Notation (BPMN)* specification[1].

The executions details needed to complete a process could be that a task in the process needs to send an email. The details missing to send an email from the task would be the sender, receiver, subject line, and body of the email. These need to be defined in the e-mail task. Another task might be to fetch data from an existing service in the organization, therefore requiring definitions for the location of the service, the service name, any data it might need, and whatever else is needed.

[1] The BPMN specification is a standard owned by the Object Management Group (OMG) that was put together to provide a single specification that tool vendors could then implement against to provide process definition portability. For more information see http://www.omg.org/spec/BPMN/2.0.

A process is not complete until defining the data needed for the process, what tasks are to be automated, what tasks humans must complete, what systems are to be integrated, and the execution details for each task.

To illustrate this, imagine you are developing a process definition for a travel agency booking system. A part of the larger project is to define a process for registering a selected flight, hotel, and charge the credit card provided, if not fraudulent, and notify the customer of their travel details. If the credit card be fraudulent, you want to cancel the booked flight and the hotel. Figure 1.2 shows a process diagram of the travel agency booking section of the project. It is a fully implemented business process once the execution details have been added to each task, such as the flight booking service details, the hotel booking service details, the flight cancellation service details, the hotel cancellation service details, the credit card payment details, the e-mail details needed to notify the customer of their travel arrangements, and sorting out the details for cancellation services before they are signaled to undo a booking.

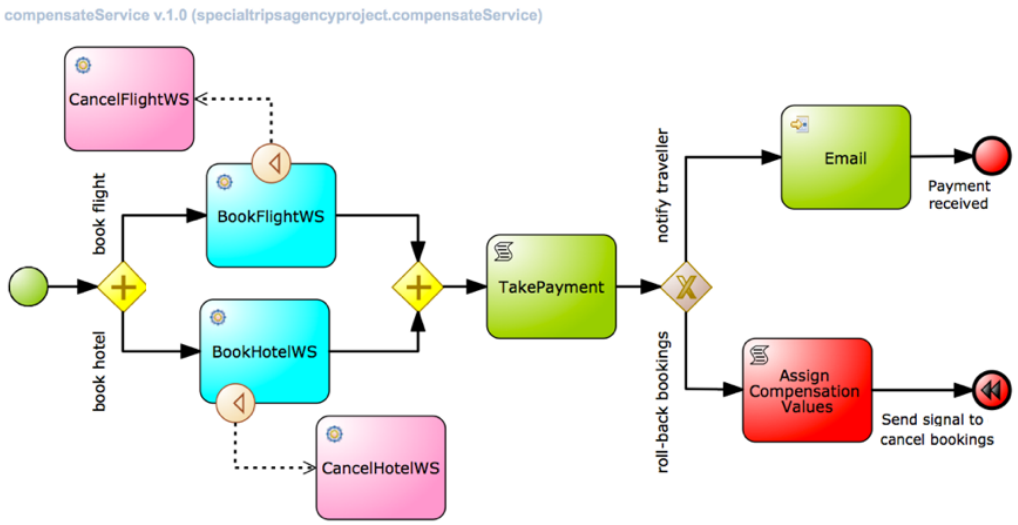

Figure 1.2.A process that captures a piece of business value, that of registering a booking for a hotel and a flight by taking payment from the provided credit card before notifying the buyer or determining that the payment was fraudulent and triggering a roll back of the bookings.

The ideal process is a fully-automated process that removes human involvement. This is a process without any user tasks and is referred to as Straight Through Processing (STP).

By capturing a process in a static diagram and removing all human interaction you ensure that each instance completes in a consistent manner. Any ad-hoc activities that are part of human nature, like taking a coffee break or visiting with a colleague, are no longer occurring when it's captured as an STP process.

Figure 1.3is an example of an STP process, where the tasks and decisions are made without human involvement from start to finish. This process always reaches one of the end nodes in the process diagram for each and every instance of the process that is started.

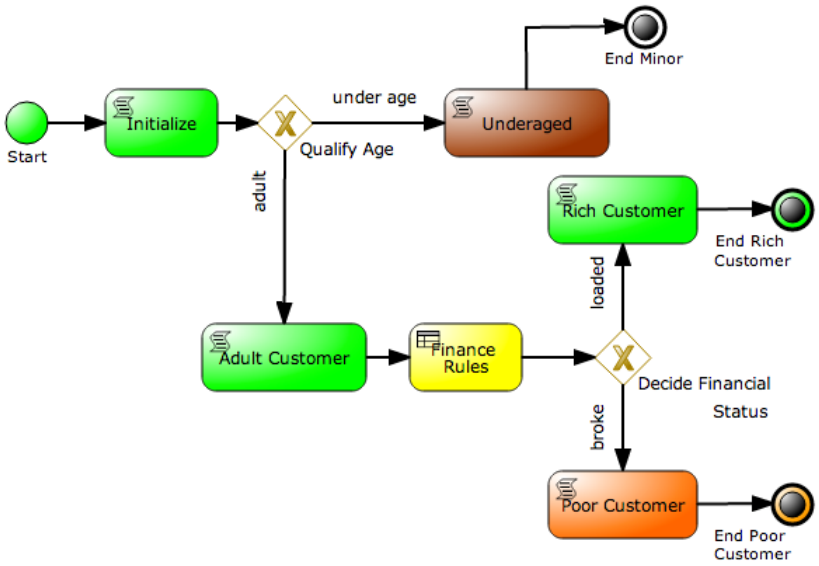

Figure 1.3. An example of a Straight Through Process (STP), this Customer Evaluation process has only tasks and decisions from start to finish.

Sometimes a piece of business value that is to be captured as a process cannot be fully automated. There remains human involvement to complete these types of processes, yet automating tasks improves the business enough to justify turning it into a process. By capturing a process that contains user tasks, you've stumbled upon a very important feature of BPM which allows you to manage *wait states.* A wait state is any task that requires that you pause processing and wait for some external event to notify the process to move onwards.

In classical application delivery it is very hard to keep track of state. Waiting means putting information into storage so the application can be put to sleep until you are ready to continue. With BPM technology you are provided with a state engine that manages wait states by keeping track of where you are in the process, by releasing resources that other process instances can use while the process instance is put to sleep. It also manages the *rehydration* of the process instance when it is ready to wake and continue from the current wait state. When a process instance reaches the wait state, it saves all state information needed to run the process instance and persist, or go to sleep and wait for something to trigger a restart. When a trigger arrives to restart a process instance, it rehydrates by gathering the necessary state data, and populating the process instance exactly as it was before the wait. It then uses

the trigger provided information, such as a user task form with input data, to continue moving the process instance forward from where it stopped previously.

Figure 1.4shows a process that leverages a user task called *Approve Reward*. When a process instance is started, the first task reached is the user task *Approve Reward*, at which time the task is assigned to a manager and waits until a manager has time to work on the task. When this user task is reached, the process engine sets up the task, and then puts it to sleep so that all resources are released for other process instances to use. This is how a single BPM process engine can execute many, many process instances at one time; there is a small set of active process instances. Most are either in a completed state or in a wait state and not utilizing any computing resources.

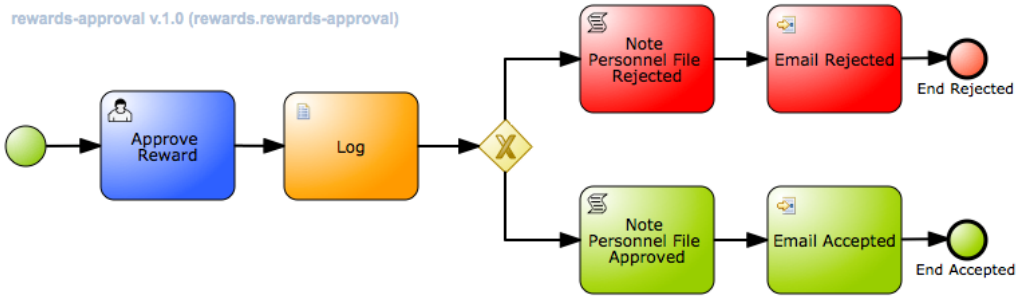

Figure 1.4. The Rewards process has a wait state in the form of a user task that is used by a manager to approve or deny a submitted employee reward. The manager decision determines the completion path to be taken, approved or rejected.

Once the task is claimed, worked on and completed, the process instance signals that it's ready to move onwards. The process engine rehydrates the process instance, putting it back into a state to move onwards with the data provided by the user task and evaluate if it should take the reject path or accept path for this particular employee reward.

Now that you have a feel for the basics of what BPM is, let's take a look at the three main elements that make up a BPM solution and need to be supported by any BPM product you might use.

1.2 An introduction to rules, events and processes[2]

The basic building blocks for any BPM project requires a product to have the ability to integrate business rules, business events and business processes. In this section I introduce the concept of a business rules, discuss what they are, look at how events differ from rules,

[2] This section comes from an introductory presentation found online.

and look at business processes. This approach starts with foundational building blocks that then leads to the high level business process used to tie it all together in your project.

1.2.1 What are business rules?

In traditional application development, you see business logic is often put into the application itself. This logic is implemented in static application code, becoming part of the artifacts that are compiled, tested and delivered into production. Each change to the logic within such an application requires a complete release cycle. Code is changed, code is compiled, it's tested and finally delivered to production. This costs time and is susceptible to errors. Changes to any logic is passed from the business owners during requirement discovery phases, to a project team and developers, whose interpretation might differ from what was intended.

The next time you are looking at applications in your organization, look for constructs like *if-then-statements* and *case-statements*, which are basic indicators of logic that can be extracted as business rules. These constructs are indicators of business rules that should be externalized from applications. Once such business rules are externalized you can deliver applications, and later modify the externalized business rules without needing new application code. It is now possible to put the business rules into the hands of the business owners, who understand how to maintain the life-cycle of the applications using the business rules.

Business rule management systems are designed to provide exactly this kind of support and tooling to business rules owners and application developers. Business rule management systems provide tooling to express rules in terms that the business owners understand, and allows the developer to focus on application delivery while the business owner retains visibility of the business rules serving customers. Finally, with business rules centralized in an external location it becomes easier to maintain consistency across applications leveraging the business rules. If business rules are spread out across multiple applications, there is a risk of duplication, and rule maintenance becomes difficult as the application landscape expands.

1.2.2 What are business events?

Business rules are applied based on a condition that has to be met, and when that condition is met the rule triggers an action. Rules are evaluated one-by-one, and they are either triggered or not. Rules can be grouped together, but they are still evaluated one-by-one to determine if a rule has a condition that is met that causes its action to be executed.

Business rules are also part of a concept called business events. Events can be triggered when a rule, or set of rules, match their conditions over a defined time period. Events that take place within the context of a business rule management system are still business rules, but now you add a *temporal* element.

For example, traditionally there are rules that can be applied to credit card transactions. Imagine a rule that requires a purchase must have a total value that fits the credit limit for that card. Should the purchase being attempted exceed that credit limit, the action taken rejects the purchase. This rule can be applied time and again, without regard for any sort of

time sensitive information. It is only when you add a time element that it becomes a business event, such as looking at a period of transactions to determine if anytook place in locations that are not physically possible. Such a series of purchases, say in Tokyo, San Francisco and Amsterdam in a span of 24-hours, would result in a business event triggering the blockage of the credit card, and a notification process to alert the card holder of fraudulent usage of the credit card.

A more modern example is how enterprises use business events to monitor their corporate image across all manner of social media channels. In the travel industry for example, you see event monitoring coupled with large customer contact centers, which are manned by hundreds of employees who receive notifications whenever online comments reference their company. If they can use event monitoring to detect and respond to messages, negative or positive, directly with the customer who wrote them they can have a positive effect on their image in the market. This is a powerful use of business events and is only possible with a business rule management system that has event processing.

1.2.3 What are business processes?

I have discussed how business processes can be discovered in an organization to reduce inefficient manual processes by automating as much as possible. What are business processes used for besides automation? Business processes are used to improve consistency in completing a series of tasks, increases visibility, and reduces errors.

Before an organization starts using processes to streamline their business activities, they are just a large pool of employees that are trying to bring value to their customers. This can be done by filling orders or providing services. These employees are assigned tasks that might require interaction with back-office systems like shipping, financial, or inventory, or it might require they contact a transport company to handle order delivery.

A modern organization evolves over time, automating some interactions with back-office systems, and adding technology to provide a service layer that communicates with various applications. The problem is these services are used by applications for specific tasks, and not linked together to handle a complete series of tasks that makes up a business process.

Once the back-office and external organizations have been put behind an automated layer of services, business process discovery can identify the processes to be automated. Business processes become the layer of organization or integration that brings a series of identified tasks to complete a part of the business. This can sometimes involve human interaction which are referred to as user tasks. By managing user tasks in your business processes, they are repeatable, can be measured for efficiency and reduce human errors. Finally, an overview of business activities can be monitored and reports generated to keep track of how various parts of the organization are functioning. This can lead to quicker decisions around adapting existing processes, or implementing new ones, to further accelerate the earning potential of the organization as a whole.

To review, it starts with experts from the business helping to identify needed tasks and the sequence in which they are to be completed. A small group of human resource employees

could be used to discuss the hiring process. They would tell their stories about how they put together a job description, place advertisements for the job opening, handle incoming reactions from applicants, schedule interviews using existing calendaring systems for employees selected to interview candidates, gather interview impressions from the employees, obtain a decision from the hiring owner of the job, inform rejected candidates, notify the hired candidate, begin the onboarding, etc. Portions of this process are worth automating, like the onboarding process that uses manual tasks, when automation would save valuable time and resources.

A BPM suite is used to automate the process by integrating services in the organization, directly with systems, or managing human interaction with tasks users need to complete. The BPM suite captures the process instance data and generates reports to provide business owners up to date information and visibility into every aspect of business operations.

An often asked question is when *not to use* a BPM suite. While there are many cases that can be made to fit into a BPM type of solution, sometimes the complexity of a process lies in the judgments of human interactions. For example, a process to onboard new hires is a good candidate, but the decision process of who to hire after interviewing candidates requires that human decision process that you don't want to automate. This is a process that is not a good candidate nor should it be.

When looking at BRMS products, note that they support rules and events. A BPM suite product needs to be a super-set by encompassing BRMS functionality and adding in support for process development and execution. Therefore, when talking about a BPM suite, you are referring to rules, events, and processes in a single suite.

1.3 Understanding the role of community projects

When looking at Open Source software solutions it is important to understand how the products in that market are created, maintained, and from where they originate. This is no different when looking at JBoss BPM solutions, as the entire portfolio of Red Hat JBoss products is based on Open Source software.

Before a product can be created, there are community projects that are *upstream*. Upstream is where research and development takes place, where new features are tested in an open community of coders. These coders can be employed by companies, work on projects that use community code and want to contribute back fixes and findings they encounter in daily use. Some members of these community projects are only interested in rules, events and processes, so they use these projects to explore ideas that interest them.

Whatever their reasons are, there is an ever-shifting group of developers working on projects that make up the community known as Drools (http://www.drools.org) and jBPM (http://www.jbpm.org). The Drools and jBPM projects are the foundations for rules, events, planning, processes, and tooling that can be found co-hosted on GitHub (https://github.com/droolsjbpm) for anyone to use and explore. Where the Drools project is focused on business rules and events technology, jBPM is focused on business process

management technology. They are both released under the Apache License 2.0, a common Open Source license offered by the Apache Software Foundation that gives users freedom to use the software for any purpose, to distribute it, to modify it, and to distribute modified versions of the software, under the terms of the license, without concern for royalties (http://www.apache.org/licenses/LICENSE-2.0). The user is only required to preserve the copyright, notice and disclaimer.

Let's take a look at a few of the projects that are found in the Drools and jBPM communities to get an idea of some of the work available. There are projects listed for maintaining websites, specific tools to support users in their tasks, core functionality or engines, that provide developers with application programming interfaces (API's), and special projects that provide functionality for users.

Here are a few selected projects that give you an idea of what is available:

- *drools-website https://github.com/droolsjbpm/drools-website)* – project used to generate the Drools project website.
- *jbpm-designer (https://github.com/droolsjbpm/jbpm-designer)* – project for the web-based process designer found in jBPM web console.
- *jbpm (https://github.com/droolsjbpm/jbpm)*– main project for core jBPM engine.
- *droolsjbpm-knowledge (https://github.com/droolsjbpm/droolsjbpm-knowledge)* – project for the common API for Drools and jBPM.
- *jbpmmigration (https://github.com/droolsjbpm/jbpmmigration)* – small side project for migration from jBPM 3.x jPDL process format to standard BPMN.

There are many more projects in the community. You are encouraged to explore and keep an eye on the work done there to see what the future might hold for JBoss BPM products.

A final question that is often posed is why not just use the jBPM community projects instead of a product like JBoss BPM Suite. It is easy to understand that free software is easy to obtain, but hard to integrate and maintain in your organization's architecture. I mentioned that there are many individual projects that are taken from the community by Red Hat and then tested, integrated, quality assured and more before you can obtain a new copy of JBoss BPM Suite. This product is then also able to integrate with other products in the JBoss portfolio.

It remains easy to obtain the community projects, but it is not easy to integrate, maintain, keep them updated and secure over time. Many organizations choose to have a supported version of a product where they can rely on the stability offered and service level agreements to ensure their projects run well in production. Community is about innovation and fast moving changes to the projects found there as new features, engines and theories are tested on the fly. In a product, you are less interested in the risks involved in running community versions and more interested in a solid platform for developing and running your applications.

1.4 Meet the JBoss BPM Suite

In this book I focus on the JBoss BPM Suite product. JBoss BPM Suite provides you with all the engines, tools, testing and execution environments needed to develop and deploy projects containing rules, events and processes. It brings together, in one easy to use web console, the designers, modelers, reports and other information that reduces the complexity of having to install each one individually. It covers the needs of process project teams, and allows for good visibility into the activities of team members as they work on project artifacts. It attempts to make common rules, events and process tasks easier to accomplish, and with less code development than if done without using JBoss BPM Suite.

JBoss BPM Suite is a collection of components made up of projects found in the Drools and jBPM communities. Specific projects are hard to list as certain features are sometimes deemed not ready for inclusion in a supported product. These almost-ready features are sometimes released as *technical previews* and in later releases become fully-supported features.

The components that are found in JBoss BPM Suite are shown in figure 1.5, an architectural overview of the suite layers. Starting at the bottom, I describe the components shown in each architectural layer.

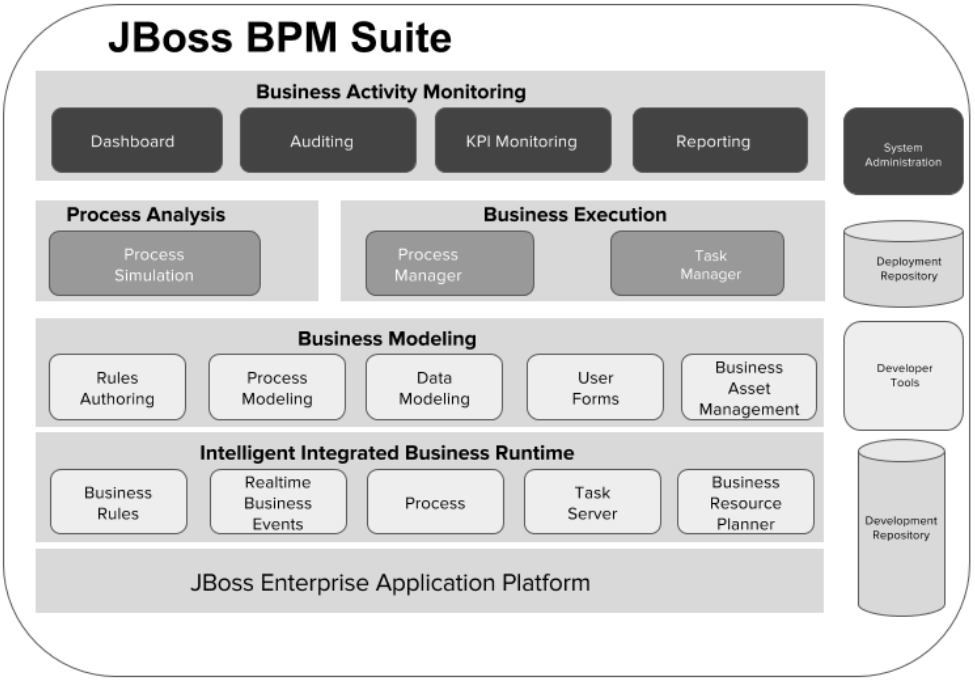

Figure 1.5. JBoss BPM Suite component architecture shows the layers that make up a BPM suite. From engines, modeling tools, execution monitoring, analysis tools and reporting, the JBoss BPM Suite has everything you need to design, develop, run and manage your business processes.

In the first layer, this architecture shows JBoss BPM Suite running on the JBoss Enterprise Application Platform (EAP). This is a certified configuration, meaning a configuration the Red Hat supports and recommends you use with JBoss BPM Suite. There is a true spirit of Open Source and interoperability with JBoss BPM Suite, so all support outside of the certified JBoss EAP platform is based on the use of certain Java Virtual Machine (JVM) configurations. For more on certified and supported configurations see the product documentation (https://access.redhat.com/articles/704703).

1.4.1 Introducing the core runtime engines

The next layer is called the *Intelligent Integrated Business Runtime* and contains the engines that support the core aspects of the suite:

- Business rules engine
- Business events engine
- Process engine
- User task engine
- Business Resource Planner engine

The business rules engine in this layer is responsible for rule parsing, evaluation, and execution. It is called by the application code and is supplied with the gathered data, and the rule set used to test the data. If data matches a rule it triggers predefined actions. For example, a persons' age is passed to a validation rule set that determines if the person is a minor or an adult. If the data puts the age under eighteen years old, the person is labeled a minor, otherwise they are labeled an adult. A second way to use a rule engine is through a business process. A business process can have a rule task, which is supplied with details for calling a rule set with data from the process. Often the outcome of the rule is used in a decision later in the process to take a specific path over another.

A business events engine is used to keep track of business events, the temporal aspects, as they occur and trigger rule executions as needed. It is used by the same applications when they are setup to monitor events. It must watch a flow of events for any matches with a set of rules within a given time period as defined. When an event is triggered, the event engine leverages the rules engine to execute the defined actions.

The process engine in this layer manages the state of a process, keeping track of which task a process instance is in and the data involved in that process instance, managing resources by releasing on wait states and rehydration of a process instance when needed. It also manages interactions with core rule engines, task servers, services and more. The process engine is the runtime integration component at the heart of a BPM suite.

A separate engine is used to manage tasks. These are the user tasks that require forms to display data, and to provide fields for input by users in order to complete the defined task. This engine is used to parse the form templates generated from user tasks defined in processes. These form templates can be embedded into other applications while tied to the task engine for processing.

The final engine is the Business Resource Planner, an engine used to determine the best way to allocate resources, to plan a roster, to determine the most effective route to travel, to optimize bin packing, and much more. It works with a limited set of constrained resources, such as employees, assets, time and money, to do more business with less resources.

1.4.2 Modeling tools for all your BPM needs

To facilitate business users and developers with their rule, event, and process designing, there are several business modeling tools that allow guided help in the creation of artifacts the above runtime engines use for process execution. These modeling tools are integrated into a central web interface known as Business Central. They are also available in the developer tooling that Red Hat offers as an integrated solution called JBoss Developer Studio (https://www.redhat.com/en/technologies/jboss-middleware/developer-studio). It is based on the popular Open Source Integrated Development Environment (IDE) called Eclipse (https://eclipse.org).

There are various types of rules such as:

- guided rules
- technical rules
- decision tables
- score cards
- guided decision tables
- guided score cards

Each of these has a modeling tool provided in the web interface to make the development of specific rule artifacts easier than coding them by hand. Most of the modelers guide the rule developer through the steps needed to design and implement specific rule types. For example, the guided rule modeler, as shown in figure 1.6, has an interface to guide the developer from data imports, to the layout of required conditions that trigger actions. The modeler also has views that allow you to inspect the source code generated by the rule being designed and allows the rules to be stored in the development repository while displaying change history. It gives the developer a chance to select a past version and rewind to that point in time of developing the rule. Various rule modelers are discussed in more detail in later chapters.

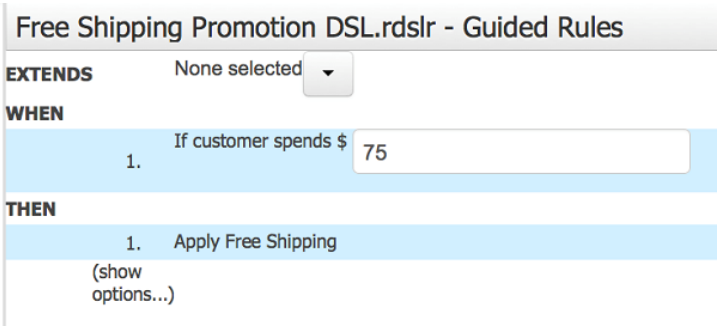

Figure 1.6. An example of the Guided Rule Modeler where a developer is guided through designing a rule to apply free shipping (shipping value set to zero dollars) if the total value of the customer purchases exceeds seventy-five dollars.

The process modeler is a rich web-based modeler that provides the process developer with basic and advanced elements for process design. It uses drag and drop from a list of tasks, including property editors for each task type. This provides further detail for integration with back-end systems, and provides features covered in later chapters.

A data modeler provides developers the ability to define data objects that are part of the process. It provides a simple to use front-end, and allows the user to view the generated Java source code. This modeler is covered in later chapters.

The user form modeler gives the developer a drag-and-drop based tool where task forms are created or generated task forms are modified. It manages data imports to bring in models that tie fields in a form to data passed between user tasks and the form. It has lists of HTML based elements to enhance forms being designed, and includes specific property editors to allow for customization of elements. Form design is covered in a later chapter.

An asset manager provides help with importing external data models, and managing assets that make up a process project. This also includes deployment management at runtime as well as development assets.

1.4.3 Looking at the BPM analysis tooling

Imagine a retail process for selling products online and a few user tasks that require manager approval on larger orders. Normally two employees are tasked eight hours per day to process these approvals, and this works. During the Christmas holidays the retail process is expecting a surge of product orders, and by doubling the number of employees working on the user tasks they can manage the expected increase of orders.

This is an example of how, during development, you can examine or test how a process reacts to severe loads or special situations before being put into production. It might also be desirable to examine the effects of a process change before they are put into production. The tooling provided here is found in the process designer. It allows you to set properties for each

task to denote simulated costs and time aspects throughout the process. For example, the user task above takes, at most, fifteen minutes to complete, and at least five minutes. Furthermore, the task costs, on average, twenty dollars to complete based on the employee salary and task completion times. The user tasks can be adjusted to set the simulated number of work hours spent, and how many users are assigned to completing incoming tasks during the defined work hours. These properties are then used for input to calculate the time spent on each task, the cost of the tasks in the path taken through the process, to count the user tasks completed, and provide an overview of the totals in each category. The results can also be displayed in a variety of output styles, such as shown in figure 1.7.

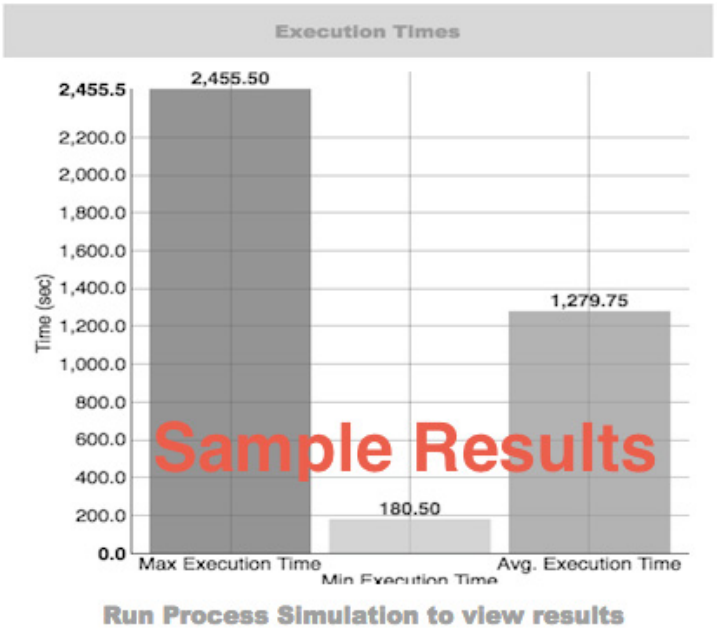

Figure 1.7. The simulation output report can be a bar chart, like shown here, or changed into one of several other types to give a visual representation of the simulation data.

After setting up the simulation parameters, you can determine the process instance loads you want to run, first hundred, then thousands and more. Each simulation load runs, giving the tester a chance to watch the server log output and provide a simulation report upon completion.

Back to the retail process and hiring decisions that management think might help with the Christmas rush. The process developers and testers were able to simulate the process with two extra employees on the user task and determined three more employees were required on the user task. They were also able to determine the user tasks would have to be worked on for

sixteen hours a day, double the norm, for the expected work load to be processed in time for the holiday. They were able to make the right hiring decisions and adjust the process workflow by extending the user task working hours based on valid test data provided by the process simulation tools.

1.4.4 Execution management made easy

After completing process development and deploying the process project, there are a few tools that are integrated into JBoss BPM Suite to help start and run through a process instance. A process manager provides a list of available process deployments from which you can select one to start. To start a process, process data must be submitted, and the task manager component provides a user form for entering data. The task manager keeps track of any user tasks that might be in the process, providing tasks lists and their status for users to claim, work on, and complete during the process instance lifecycle.

1.4.5 Providing the necessary reporting and monitoring tools

A key element of any BPM suite is the ability to monitor and generate reports based on what's happened during process execution. JBoss BPM Suite has an extensive set of tooling to provide Business Activity Monitoring (BAM). Out of the box there are reports designed to show historical data on process and task execution. Along with these reports there is a dashboard that allow reports to be designed based on any data source, both internal and external to the BPM suite. The dashboard modeler allows drag-and-drop creation of web based reports, allows for business owners to measure based on their own Key Performance Indicators (KPI) and ensure their process audit trails are shown according to their own business defined needs. More on this component is available in chapter 9.

1.4.6 The supporting components

Several components shown in figure 1.5 support the development, deployment, and execution of BPM projects. These are repositories, developer IDE tooling and system administration components that round out the suite and are based on industry wide standard components. The system administration tooling is used to configure various aspects behind the scenes when using the suite, such as:

- Users
- Roles
- repository locations
- and more

The development repository is where the BPM project artifacts reside and is Git (https://git-scm.com) based. The business central web console integrates all the various modelers with the developer repository to preserve versions and history, as does the developer tooling. The deployment repository is a Maven (https://maven.apache.org) repository where the project

build is deployed and can be tied into standard enterprise continuous build systems. There are also several programmable interfaces (APIs) available, but these are not covered here, instead see the product documentation. You will find an example in chapter 2 that leverages one of the APIs to start a process from a web application. Let's take a look at a real BPM application and see how all this works together to provide an online travel agency booking experience.

1.5 Summary

- Business process management (BPM) orchestrates all manner of systems, people and services, while allowing you to structure the timeline of their execution.
- BPM starts with a process, where a piece of business value is automated either fully or partially to improve the business as a whole.
- A fully automated business process is referred to as Straight Through Processing (STP).
- When unable to fully automate a business process, there is a need to deal with wait states, such as human interaction through user tasks.
- Capturing and maintaining state in a process is a main characteristic of BPM that ensures efficient use of resources.
- Some processes are not possible to automate, identified by the need for pure human intelligence for completion.
- Reporting and monitoring provides insights into the current and past conditions of processes, giving business owners insights into process improvements that might help their business.
- Rules, events and processes are the key building blocks for BPM projects.
- Rules are often found embedded in applications, making it difficult and costly to update. This is alleviated by using a business rules engine to capture and externalize business logic for application to call when needed.
- Business events are rules with a temporal element.
- Open Source products like JBoss BPM Suite leverage upstream community projects like Drools and jBPM.
- These community projects are created and worked on by a vast array of coders to deliver functionality that can be used together. It is often difficult to tie together all the components in the community to achieve a BPM suite.
- JBoss BPM Suite is a product from Red Hat that brings together a set of the community projects that are supported with official service level agreements.
- Various components make up the suite, such as runtimes, modeling tools, simulation tools, process and task execution tooling, business activity monitoring tooling and the background repositories.