



Módulo de Ampliação de Expressividade para a Ferramenta jBPM

Aluno: Diorbert Corrêa Pereira
Orientador: João Paulo A. Almeida

Agenda



- Processos de Negócio
- Visão geral: Workflow Resource Patterns
- A Ferramenta jBPM
 - Arquitetura
 - Módulos
 - Perspectiva de Regras
- Motivação
- Objetivo
- Resultados Parciais

“A sequence of tasks that happen in a
repeatable order ...

executed by humans and/or systems ...

to achieve a business goal”

Processos de Negócio

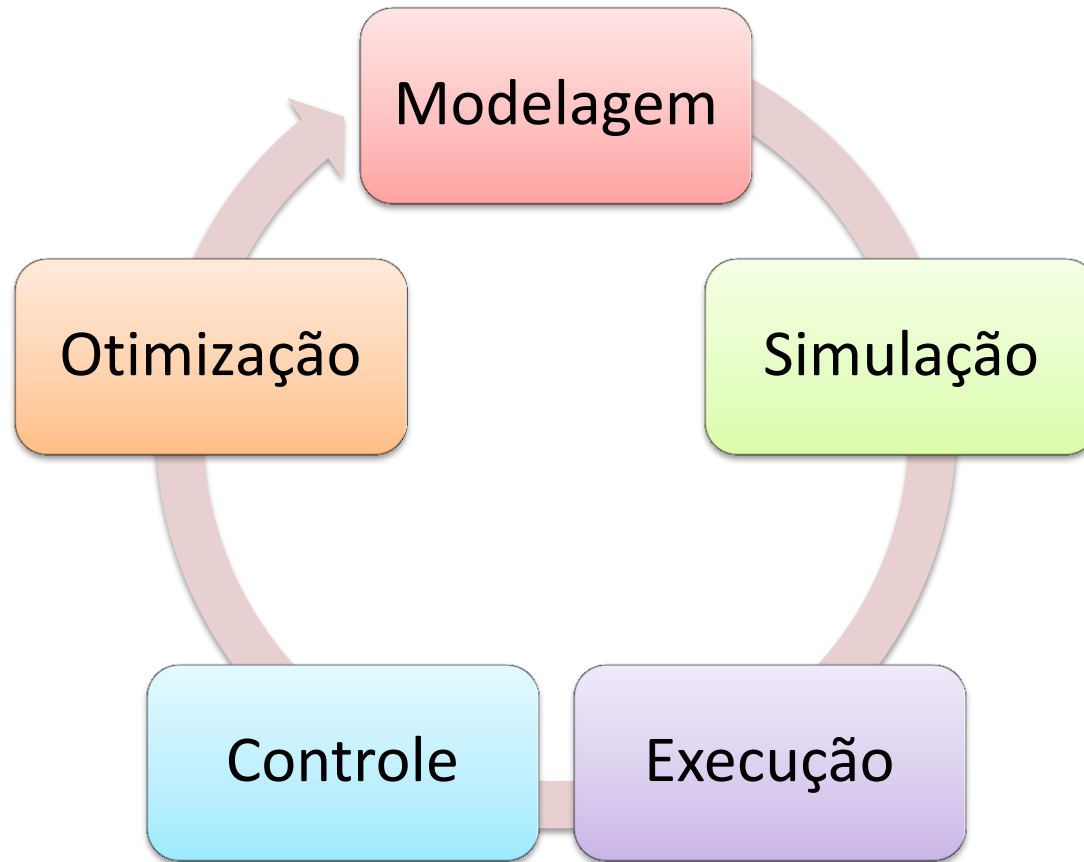


Como automatizar os processos de negócio ?

Como automatizar os processos de negócio ?

BPMS

Processos de Negócio

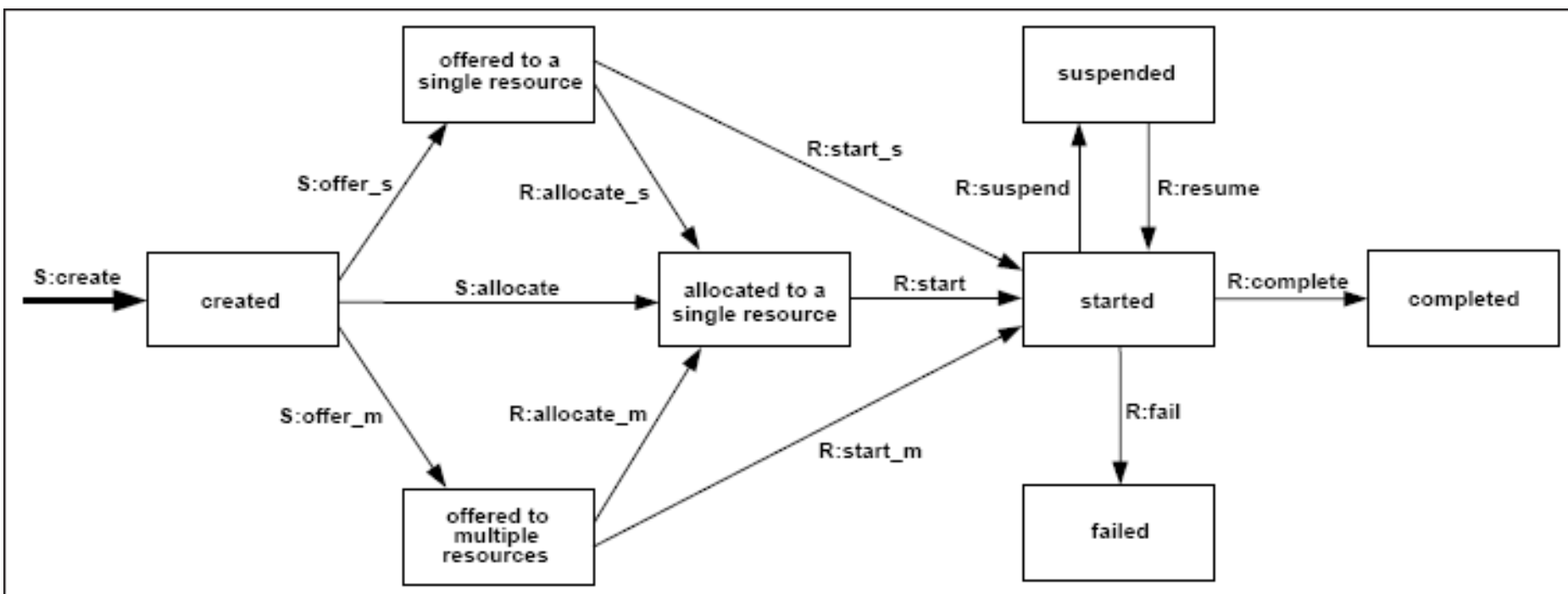


Workflow Resource Patterns

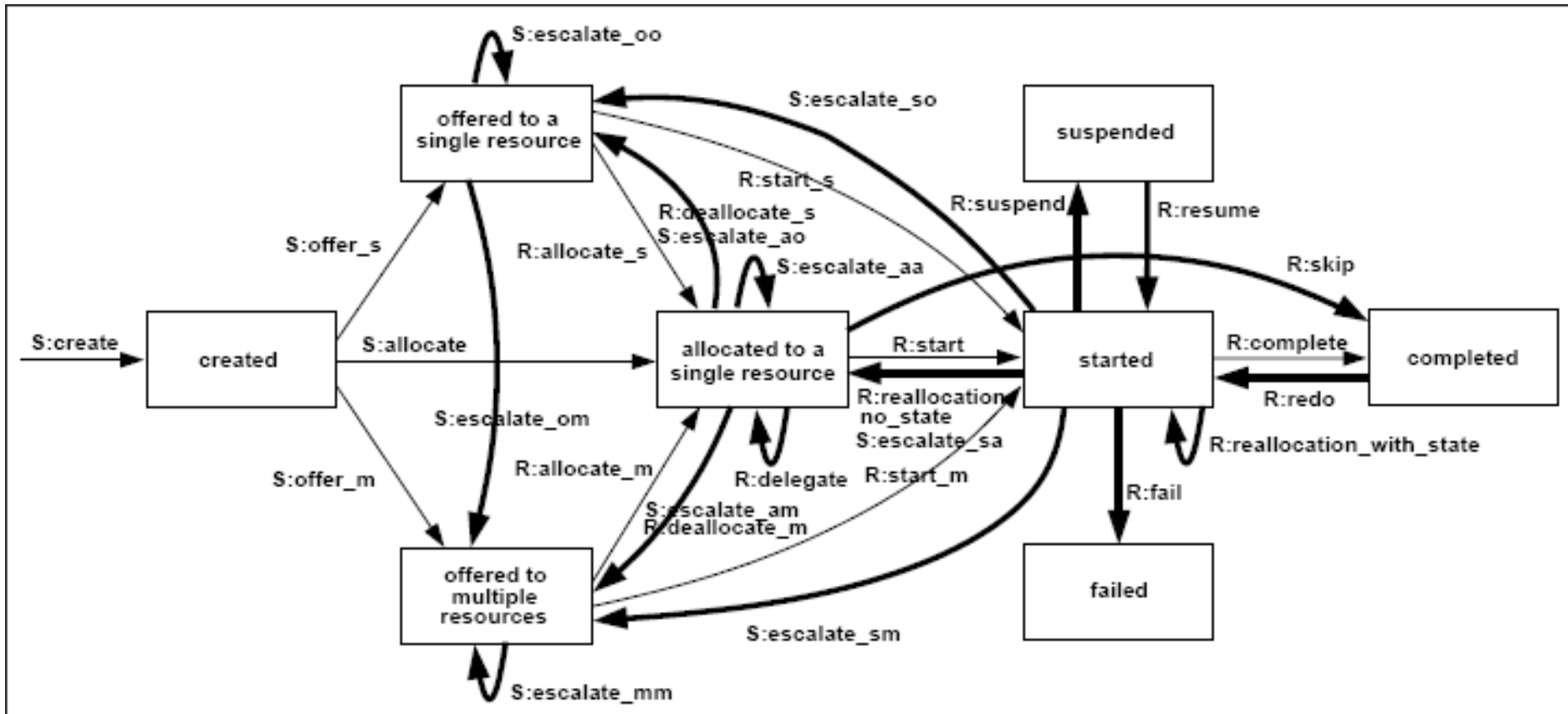


- Criados pela iniciativa Workflow Patterns
- 43 padrões
- Subdivididos em 7 grupos: Creation, Push, Pull, Detour, Auto-Start, Visibility e Multiple Resource.

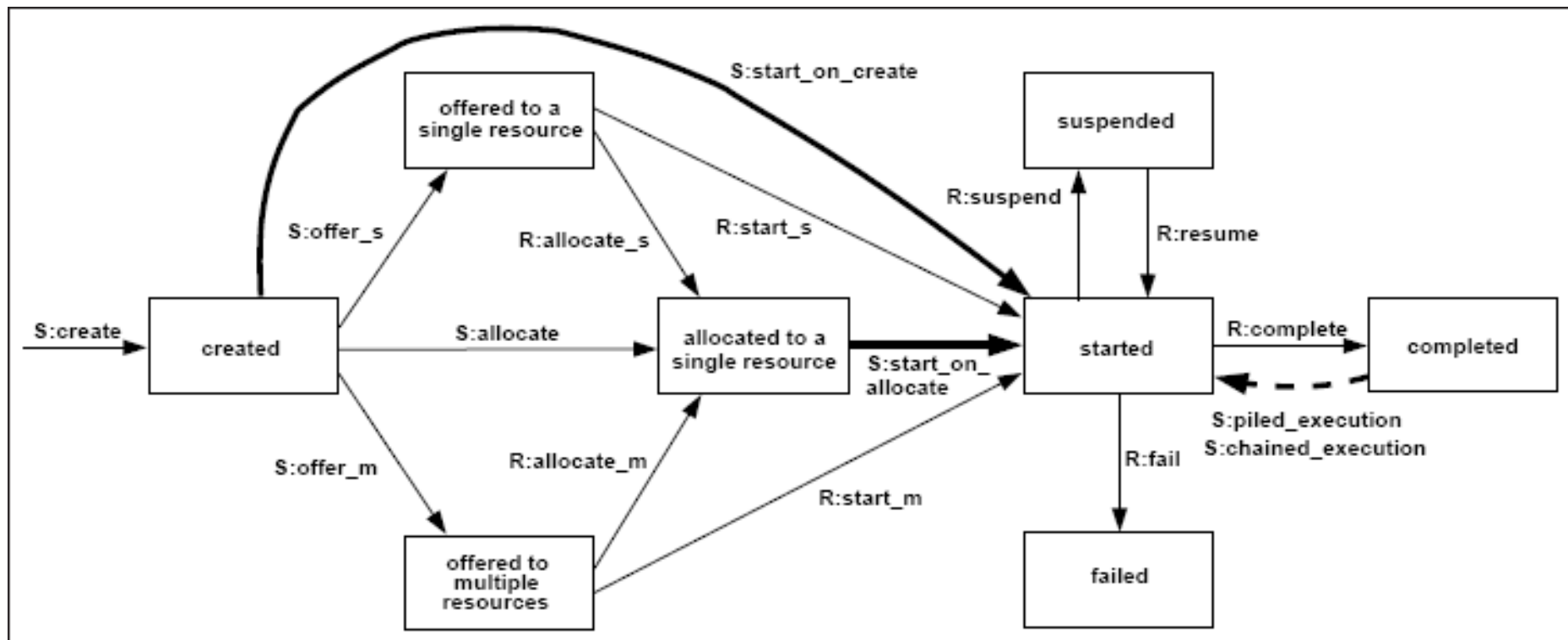
Workflow Resource Patterns



Workflow Resource Patterns



Workflow Resource Patterns



Workflow Resource Patterns

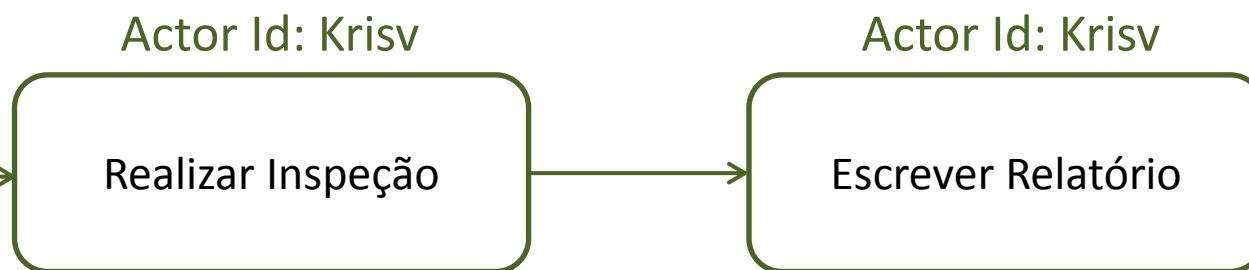


- Criados pela iniciativa Workflow Patterns
- 43 padrões
- Subdivididos em 7 grupos: Creation, Push, Pull, Detour, Auto-Start, Visibility e Multiple Resource.
- Representam a expressividade relacionada à alocação de recursos humanos às tarefas

Workflow Resource Patterns



Exemplo de Expressividade 1



Retain Familiar:

A capacidade de alocar um item de trabalho dentro de um dado caso para o mesmo recurso que executou o item de trabalho anterior.

Workflow Resource Patterns



Exemplo de Expressividade 2

Actor Id: Mary

Realizar Procedimento
de Ponte de Safena

Shortest Queue:

A capacidade de alocar um item de trabalho para um recurso escolhido, dentre um determinado grupo, baseado na quantidade de itens de trabalho pendentes (menor fila de itens de trabalho).

Workflow Resource Patterns



Exemplo de Expressividade 3

Actor Id: Mary

Analisar Relatórios de
Desempenho

Delegation:

A capacidade de um recurso alocar um item de trabalho (não iniciado), anteriormente alocado para ele, para outro recurso.

Workflow Resource Patterns



Exemplo de Expressividade 3

Actor Id: Krisv

Analisar Relatórios de
Desempenho

Delegation:

A capacidade de um recurso alocar um item de trabalho (não iniciado), anteriormente alocado para ele, para outro recurso.

Workflow Resource Patterns



- Criados pela iniciativa Workflow Patterns
- 43 padrões
- Subdivididos em 7 grupos: Creation, Push, Pull, Detour, Auto-Start, Visibility e Multiple Resource.
- Representam a expressividade relacionada à alocação de recursos humanos às tarefas
- Fornecem uma base de avaliação de ferramentas BPMS

Workflow Resource Patterns



Avaliação das Ferramentas Open Source

jBPM 3.1.4	9 / 43
OpenWFE 1.7.3	13 / 43
Enhydra Shark 2	10 / 43

Avaliação das Ferramentas Comerciais

Staffware 10	18 / 43
WebSphere MQ Workflow 3.4	20 / 43
FLOWer 3.51	24 / 43
COSA 5.1	30 / 43
iPlanet 3.0	20 / 43

Workflow Resource Patterns



Avaliação das Ferramentas Open Source

jBPM 3.1.4	9 / 43
OpenWFE 1.7.3	13 / 43
Enhydra Shark 2	10 / 43

Avaliação das Ferramentas Comerciais

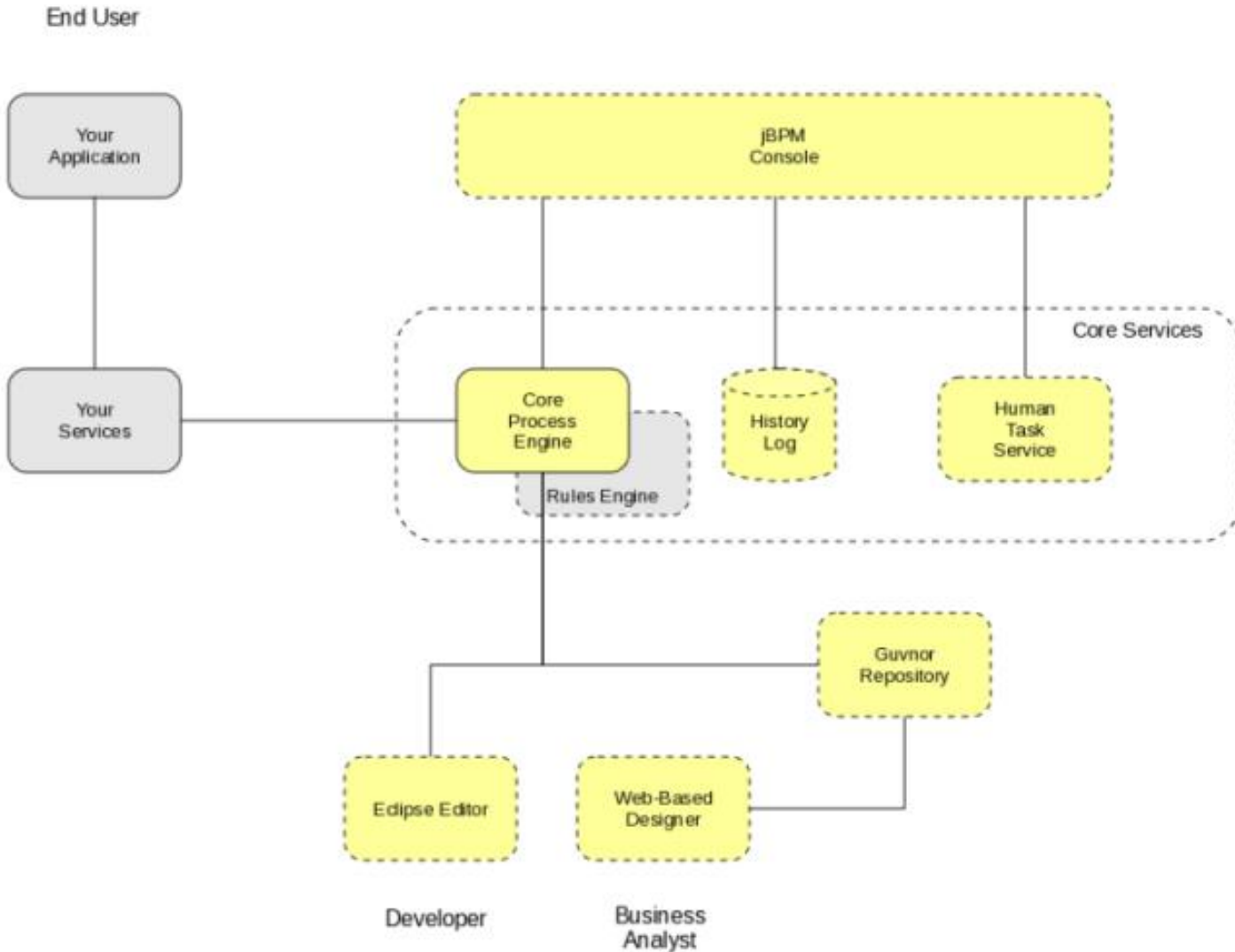
Staffware 10	18 / 43
WebSphere MQ Workflow 3.4	20 / 43
FLOWer 3.51	24 / 43
COSA 5.1	30 / 43
iPlanet 3.0	20 / 43

jBPM

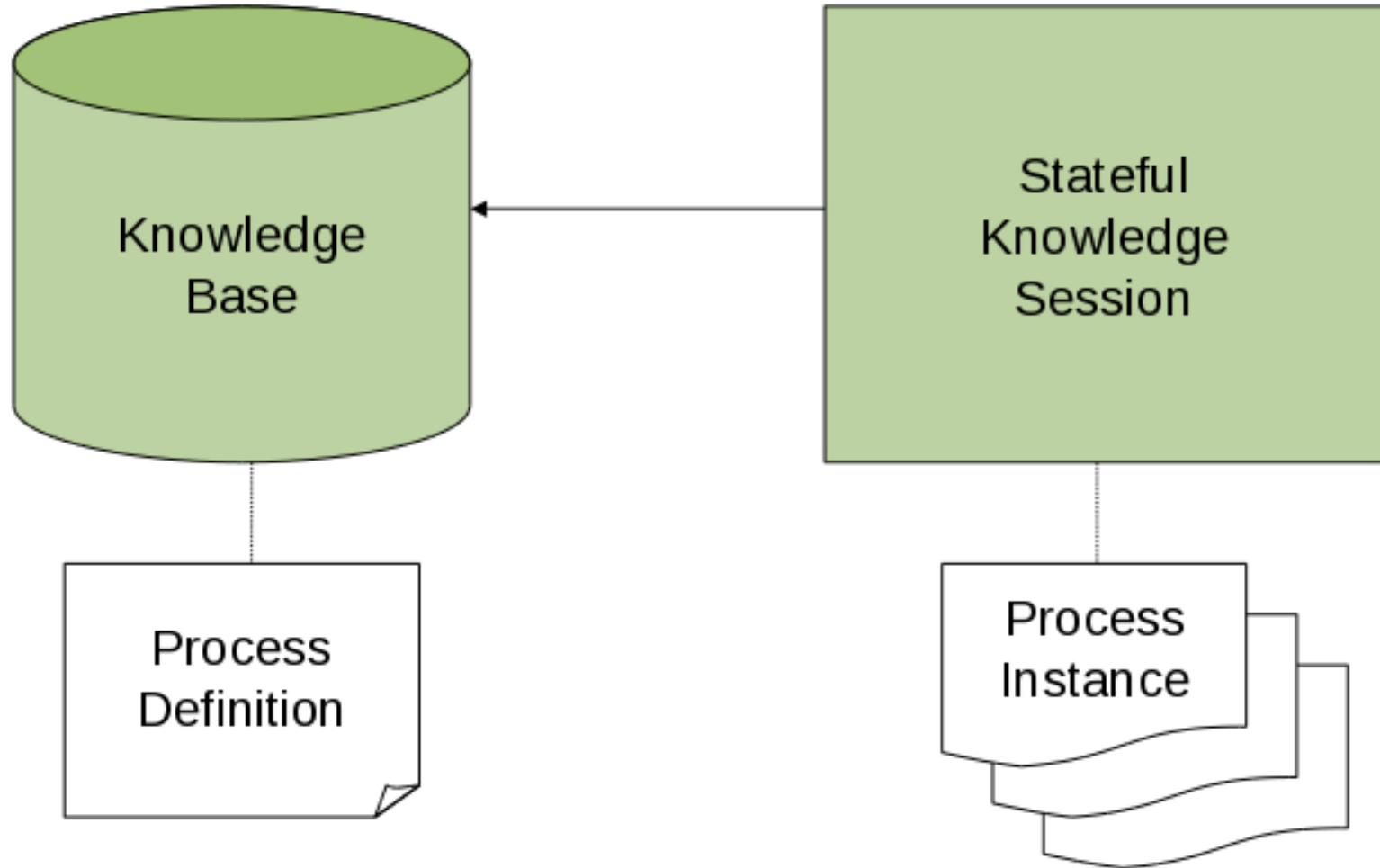


- Suíte BPM open-source
- Mantida pela comunidade jBoss
- Conjunto de ferramentas para o gerenciamento de processos de negócio

jBPM: Arquitetura



jBPM: Arquitetura



jBPM Console

- Gerenciamento das instâncias dos processos de negócio

jBPM: Módulos



Tasks

Processes

- Execution History
- Process Overview

Reporting

Settings

Process Overview

Refresh All Start Signal Delete Terminate

Process	v.	Instance	State	Start Date
AlocacaoSeletiva	0	1	RUNNING	2012-02-23 14:22:24
RoleAlocacionProcess	0			
Evaluation	0			

Execution details

Process:	Evaluation	Diagram
Instance ID:	1	Instance Data
Key:		
State	RUNNING	
Start Date:	2012-02-23 14:22:24	
Activity:		

jBPM: Módulos



Tasks

Processes

- Execution History
- Process Overview

Reporting

Settings

Process Overview

Refresh All Start Signal Delete Terminate

Process	v.	Instance	State	Start Date
Evaluation	0	1	RUNNING	2010-11-22 16:46:59

Process Instance Activity

Instance: 1

The diagram shows a process flow starting with a green circle (Start), followed by a task labeled 'Self Evaluation'. This leads to a yellow diamond with a plus sign (Exclusive Split), which branches into two parallel tasks: 'HR Evaluation' and 'PM Evaluation'. Both tasks merge at a second yellow diamond with a plus sign (Exclusive Join), which then leads to a red circle with a white center (End).

jBPM Console

- Gerenciamento das instâncias dos processos de negócio
- Visualização de relatórios de desempenho

jBPM Console

- Gerenciamento das instâncias dos processos de negócio
- Visualização de relatórios de desempenho
- Gerenciamento dos itens de trabalho (pessoais e de grupo)

jBPM: Módulos



Tasks

- Personal Tasks
- Group Tasks

Group Tasks

Refresh Claim

Priority	Process	Task Name
0		TaskOne

Tasks

- Personal Tasks
- Group Tasks

Group Tasks **Personal Tasks**

Refresh View Release

Priority	Process	Task Name
0		TaskOne

jBPM: Módulos



The screenshot displays the jBPM Personal Tasks interface. At the top right, the user is logged in as "krisv" with a "Logout" button. The main content area is titled "Personal Tasks" and contains a table with the following data:

Priority	Process	Task Name	Due Date
0		Performance Evaluation	

Below the table, a "Task Form: Performance Evaluation" window is open. The form contains the following text and controls:

- Employee evaluation**
- Please perform a self-evaluation.
- Please fill in the following evaluation form:
- Rate the overall performance:
- Check any that apply:
 - Displaying initiative
 - Thriving on change
 - Good communication skills
-

At the bottom left of the task details section, there are navigation arrows and a "Task details" section with the following fields:

- ID:
- Process:
- Name:
- Assignee:
- Description:

On the left side of the interface, there is a "Tasks" sidebar with "Personal Tasks" and "Group Tasks" options. At the bottom left, there are links for "Processes", "Reporting", and "Settings".

Eclipse Editor

- Permite a modelagem de processos de negócio

jBPM: Módulos



The screenshot displays the Eclipse IDE interface for editing a BPMN diagram. The main canvas shows a process flow starting with a Start Event, followed by a 'Self Evaluation' task, a Diverge Gateway, two parallel tasks ('HR Evaluation' and 'PM Evaluation'), a Converge Gateway, and finally an End Event. The left sidebar shows the Package Explorer with the project structure. The bottom pane shows the Properties view for the selected 'Self Evaluation' task.

Property	Value
ActorId	# {employee}
Comment	Please perform a self-evaluation.
Content	
Id	2
MetaData	{width=135, height=48, y=56, UniqueId=_2, x=96}
Name	Self Evaluation
On Entry Actions	
On Exit Actions	

Guvnor

- Repositório de Processos de Negócio
- Repositório de Task Forms
- Repositório de Regras
- Repositório de POJOs
- Permite a modelagem de processos de negócio

jBPM: Módulos



The screenshot shows the Drools web interface. The top navigation bar includes the Drools logo and a user status indicator: "Welcome: guest [Sign Out]". The main content area is divided into a left sidebar and a right main panel.

Left Sidebar:

- Browse**
- Knowledge Bases**
- Create New ▾
- Packages
 - defaultPackage
 - Business rule assets
 - Technical rule assets
 - Functions
 - DSL configurations
 - Model
 - Processes
 - Enumerations
 - Test Scenarios
 - XML, Properties
 - Other assets, documentati
- QA
- Package snapshots

Main Panel:

Find **defaultPackage** **Processes [defaultPackage]**

[refresh list] [open selected] [open selected to single tab] [RSS]

	Format	Name	Description	Status	Last modified	Open
<input type="checkbox"/>		HelloProcess		Draft	2011 Jan 13 11:53:14	<input type="button" value="Open"/>

1-1 of 1

jBPM: Módulos



Drools Welcome: guest [Sign Out!](#)

Browse
Knowledge Bases
Create New ▶
Packages
defaultPackage
Business rule assets
Technical rule assets
Functions
DSL configurations
Model
Processes
Enumerations
Test Scenarios
XML, Properties
Other assets, documentation
WorkingSets
SpringContext
WorkItemDefinition
Global Area

Find **Processes [defaultPackage]** **Evaluation**

File Edit Source Status: 'Draft'

Attributes **Edit**

Shape Repository
jBPM BPMN2
Activities
Task
Collapsed Subprocess
Embedded Subprocess
AdHoc Subprocess
Artifacts
Catching Intermediate Events
Connecting Objects
Data Objects
End Events
Gateways
Service Tasks
Start Events
Swimlanes
Throwing Intermediate Events

```
graph LR; Start((Start)) --> SelfEval[Self Evaluation]; SelfEval --> G1{Gateway}; G1 --> HREval[HR Evaluation]; G1 --> PMEval[PM Evaluation]; HREval --> G2{Gateway}; PMEval --> G2; G2 --> End(((End)))
```

Properties (BPMN-Diagram)
Name Value
Often used
Documenta...
Name
More Properties

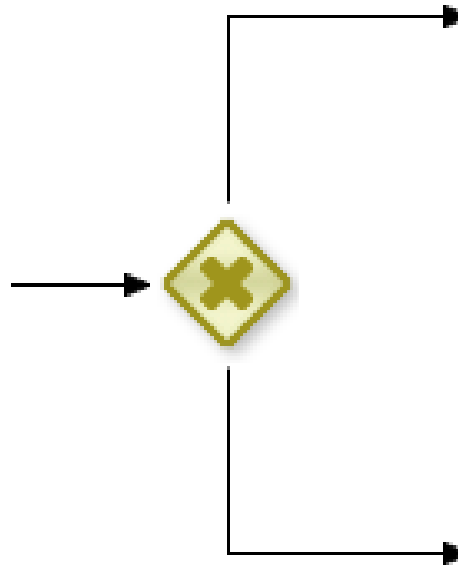
Integração com Drools

- Rule Constraints
- Rule Tasks



Rule Constraints

- LHS de um regra Drools



jBPM: Perspectiva de Regras



Exemplo:

```
$processInstance : WorkflowProcessInstance()  
$p : Person( name == ( processInstance.getVariable("name") ) )  
eval( $p.age > 18 )
```

Exemplo:

```
$processInstance : WorkflowProcessInstance()
```

```
$p : Person( name == ( processInstance.getVariable("name") ) )
```

```
eval( $p.age > 18 )
```

Rule Tasks

- Executa um conjunto de regras ao ser ativado
- O conjunto de regras é definido através do atributo “*RuleFlowGroup*” da tarefa



Rule Tasks

- **`ksession.fireAllRules()`**: Permite que as regras sejam disparadas. Deve ser chamada antes de uma Rule Task.
- **`ksession.fireUntilHalt()`**: Permite que as regras sejam disparadas até que seja ordenado o contrário. Uma única chamada permite que Rule Tasks sejam ativadas sem a necessidade da chamada do método `fireAllRules()`.

Rule Tasks

ATENÇÃO: O uso de `fireUntilHalt()` gera um bug conhecido quando utilizado em processos com persistência.

- jBPM possui baixo suporte aos padrões
- A ampliação do suporte aos padrões representa o aumento da expressividade da ferramenta
- A integração da plataforma orientada a regras (Drools) a partir da versão 5.0 de jBPM fornece novas possibilidades de uso da ferramenta

Objetivo



- Ampliar o suporte através do uso de regras

Resultados Parciais



- A avaliação da ferramenta jBPM realizada pela iniciativa Workflow Patterns se tornou obsoleta
- Reavaliação realizada (jBPM 5.2) constatando suporte a 17 dos 43 padrões
- Estudo de viabilidade da aplicação de regras para ampliação do suporte apontou 8 padrões em potencial

Resultados Parciais



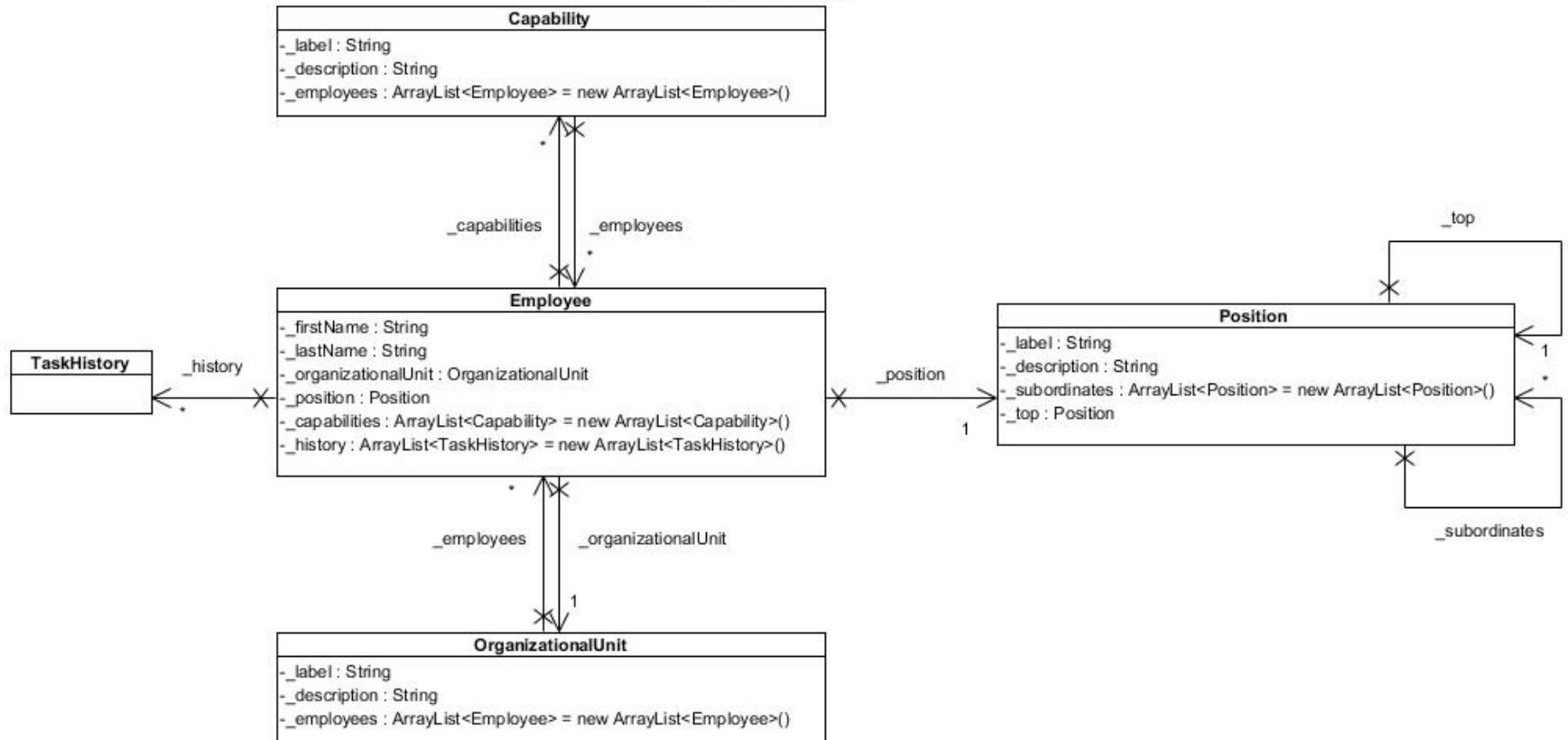
- Foram necessárias classes adicionais para serem utilizadas como interface de “configuração” das Rule Tasks que executam os padrões
- 7 padrões já foram implementados

Detalhes de Implementação

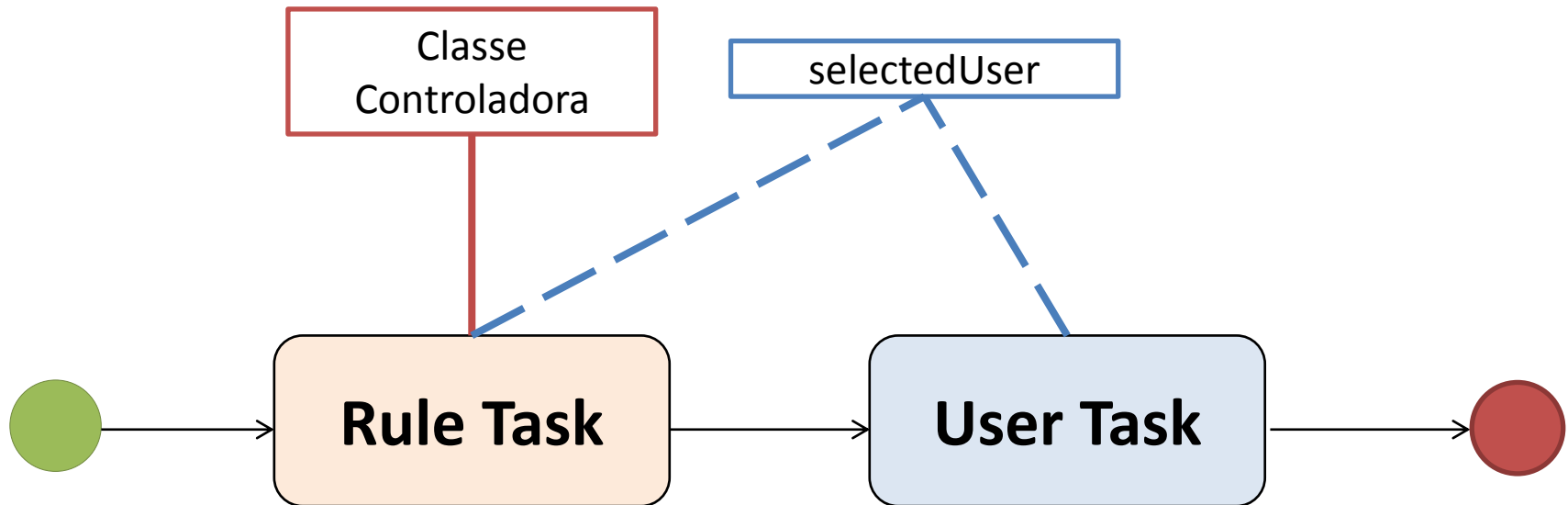


resourcePatternsManager

resourceDescriptor



Resultados Parciais



Resultados Parciais



```
rule "Random Allocation"  
  ruleflow-group "RandomAllocation"  
  when  
    $employees : ArrayList( size >= 1 ) from collect( Employee() )  
    $processInstance : WorkflowProcessInstance()  
  then  
    Random randomGenerator = new Random();  
    int index = randomGenerator.nextInt($employees.size());  
    Employee employee = (Employee)$employees.get(index);  
  
    $processInstance.setVariable("selectedUser",employee.getUserId());  
end
```

Resultados Parciais



rule "Capability Based Distribution"

ruleflow-group "CapabilityBasedDistribution"

when

\$processInstance : WorkflowProcessInstance()

\$manager : PatternsManager()

\$controller : CapabilityBasedDistributionController(

compareTaskId(

\$manager.getCurrentTaskId(\$processInstance.getNodeInstances())))

\$employees : ArrayList(size >= 1) **from**

collect(Employee(existsCapability(\$controller.getCapabilities())))

Resultados Parciais



```
then
    Employee employee = null;
    String resourcesList = "";

    for(int i = 0; i < $employees.size(); i++){
        employee = (Employee)$employees.get(i);

        if(i != 0)
            resourcesList += ", "+employee.getUserId();
        else
            resourcesList += employee.getUserId();
        }

    $processInstance.setVariable("selectedUser",resourcesList);
end
```

Resultados Parciais



```
rule "Shortest Queue"
```

```
    ruleflow-group "ShortestQueue"
```

```
when
```

```
    $processInstance : WorkflowProcessInstance()
```

```
    $employees : ArrayList( size >= 1) from collect( Employee() )
```

Resultados Parciais



then

```
TaskManager client = new TaskManager("localhost",9123);
```

```
Employee employee = null;
```

```
Employee employeeSelected = null;
```

```
int shortestCount = 2147483647;
```

```
int count = 0;
```

Resultados Parciais



```
for(int index = 0; index < $employees.size(); index++){  
    employee = (Employee)$employees.get(index);  
    count = client.getActiveTasksCount(employee.getUserId());  
  
    if(count < shortestCount){  
        shortestCount = count;  
        employeeSelected = employee;  
    }  
}  
  
$processInstance.setVariable("selectedUser",employeeSelected.getUserId());  
end
```



Obrigado pela atenção