

Plugin jGPD (java Grafical Process Designer)

Sistema Timoneiro Manual do Usuário



#### Dezembro – 2005

#### Sumário de Informações do Documento

Tipo do Documento: Manual				
Título do Documento: Manual do usuário que utilizará o plugin jGPD				
Estado do Documento: Elaborado				
Responsáveis: Helio Silvio Piccinatto – Revisado por: Marlon da Cruz Volz, Thiago Meneghello				
Palavras-Chaves: Timoneiro, jGPD, workflow				
Resumo: Este documento contém orientações para o analista desenvolver workflow pela ferramenta jGPD no eclipse				
Número de páginas: 26				
Software utilizados: Open Office 2.0 / Open Office 1.1.3				
Versão	Data	Mudanças		
1.0	19/12/06	Elaboração do Manual		
	04/01/06	Conotação com maior nível de detalhamento aos temas relacionados no documento. Reestruturação de		
		Tópicos. Explicação de funcionalidades não contempladas pela versão atual do plugin jGPD.		
1,1	16/02/06	Explicação de funcionalidades não contempladas pela versão atual do plugin jGPD.		

# SUMÁRIO

1 INTRODUÇÃO	4
2 INICIANDO A UTILIZAÇÃO DO JGPD	5
2.1 Iniciando um Novo Projeto	5
2.2 Iniciando a Definição de um Processo 2 3 Tratamento de Incompatibil idades	
3 DEFININDO UM PROCESSO	
3.1 Barra de Ferramentas de Modelagem (Aba Diagram)	
3.2 Modelando Graficamente a Definição de um Processo	
3.3 Abas de Visualização	
3.4 Editando o Arquivo processdefinition.xml	
4 GLOSSÁRIO	25
5 REFERÊNCIAS	

### 1 INTRODUÇÃO

Este documento tem como finalidade prover o suporte inicial necessário para que o usuário, que ainda não teve contato com a ferramenta jGPD, possa entender sua funcionalidade e elaborar as definições de processos necessárias ao sistema workflow a ser desenvolvido.

O sistema Timoneiro necessita ser alimentado com arquivos de definição de processos (conforme padrão jPDL – *java Process Definition Language*), para poder proporcionar o controle desses processos no sistema hospedeiro. A ferramenta jGPD (*java Grafical Process Designer*) propicia a definição de processos de maneira gráfica através do ambiente eclipse, gerando os arquivos no padrão XML requerido pelo Timoneiro.

Maiores detalhes sobre a sintaxe e esquema da linguagem jPDL, utilizada para a definição de processos suportados pelo Timoneiro, podem ser encontrados no documento Timoneiro\_jbpm\_conceituacao.pdf (<u>http://www.frameworkpinhao.pr.gov.br</u>).

O plugin jGPD está disponível no pacote de desenvolvimento CELEPAR. Através do seguinte comando para ambiente linux debian: 'apt-get install celepar-dev'. Esse pacote contém a ferramenta Eclipse com o ambiente e todos plugins necessários (My Eclipse, por exemplo), inclusive o jGPD, já configurados.

OBS: Caso o usuário não conheça em detalhes o sistema Timoneiro, é necessário que leia primeiramente, os manuais de usuário, que se encontram no site <u>http://www.frameworkpinhao.pr.gov.br</u>, no menu produtos -> ferramentas -> timoneiro -> versão 1.0

### 2 INICIANDO A UTILIZAÇÃO DO JGPD

A iniciação ao jGPD descrita neste tópico, aborda os seguintes itens:

- Iniciando um Novo Projeto;
- Iniciando uma Definição de Processo e
- Tratamento de Incompatibilidades.

#### 2.1 Iniciando um Novo Projeto

Com o eclipse aberto, deverá ser acessado no menu o item File -> New -> Project. Será apresentada a tela ilustrada abaixo. Deve-se selecionar o item Jboss jBPM -> Process Project e em seguida clicar no botão Next.

New Project	×
Select a wizard A wizard that creates a new jBPM Proje	ct
<u>W</u> izards:	
<ul> <li>Java Project from Existing Ant Bui</li> <li>Plug-in Project</li> <li>CVS</li> <li>D CVS</li> <li>D Java</li> <li>Place iPDM</li> </ul>	ldfile
T Process Project	
<ul> <li>▷ ▷ Plug-in Development</li> <li>▷ ▷ Simple</li> <li>▷ ▷ Examples</li> </ul>	-
	Ø

Figura 2.1.1 - Tela de Seleção de Novo Projeto.

Caso o modelo de projeto jBoss jBPM não esteja disponível significa que o plugin jBPM não está incluído no eclipse, neste caso deve-se verificar a instalação do pacote celepar-dev conforme descrito no início deste documento.

Na próxima tela deve ser digitado o nome do projeto e o local para armazenamento do projeto. Para exemplificar, o projeto será chamado de Projeto\_Exemplo e a localização será a padrão. Após definidos o nome e a pasta, deve-se clicar sobre o botão Finish.

New Process Project	×
Process Project	
Create a new process project.	
Project name: Projeto_Exemplo	
Project contents	
🔽 Use default	
Directory: /home/heliopiccinatto/workspace/Prc Browse	
	_
< Back Mext > Furish Cancel	

Figura 2.1.2 – Tela de Criação do Projeto.

A versão atual do jGPD possui incompatibilidades com a perspectiva MyEclipse. Portanto, a visualização recomendada para o plugin jGPD é a perspectiva Java, conforme figura a seguir:



Figura 2.1.3 – Alterando a Visualização de Perspectiva.

Para alterar a perspectiva acesse o item Window -> Open Perspectiva -> Other (no menu), com isso será apresentada uma tela com todas as perspectivas disponíveis, deve-se selecionar a perspectiva Java (default) e, em seguida, clicar sobre o botão OK. Por vezes essa opção fica disponível diretamente em Window -> Open Perspectiva -> Java.

O assistente criará quatro pastas fontes (sources):

1) "src/process" - onde ficarão os processos do projeto;

- 2) "src/java" onde ficarão as classes Java,
- 3) "test/java" para os testes unitários e
- 4) "src/resources" para os recursos (jbpm.properties e hibernate.properties). Também será criada uma pasta contendo todas as bibliotecas núcleo do jBPM.

#### 2.2 Iniciando a Definição de um Processo

Com o projeto já criado, será necessário definir o processo responsável pelos fluxos do sistema hospedeiro. Para iniciar a definição do processo deve-se clicar em File -> New -> Other (no menu), selecionar a opção Process Definition e clicar no botão Next, conforme figura a seguir.

New	×
Select a wizard A wizard that creates a process diagram	
<u>W</u> izards:	
▽ 🗁 JBoss jBPM	<b>_</b>
😽 Process Definition	
🕸 Process Project	
Plug-in Development	
	0
< Back Next > Einish	Cancel

Figura 2.2.1 – Assistente para Nova Definição de Processo.

Na tela seguinte, deverá ser indicado o nome do processo a ser criado. A pasta a ser informada (Source Folder) deverá ser "Projeto\_Exemplo/src/java" (conforme padrão).

Create Process Definition         Create a new process definition         Choose a source folder and a process definition name.         Source folder :       Projeto_Exemplo/src/java         Process name :       Processo_Admissao_Funcionario	×				
Choose a source folder and a process definition name. Source folder : Projeto_Exemplo/src/java Browse Process name : Processo_Admissao_Funcionario	Create Process Definition				
Choose a source folder and a process definition name. Source folder : Projeto_Exemplo/src/java Browse Process name : Processo_Admissao_Funcionario	Create a new process delinition				
Source folder : Projeto_Exemplo/src/java Browse Process name : Processo_Admissao_Funcionario					
Process name : Processo_Admissao_Funcionario					
	1				
< <u>Back</u> <u>Next</u> > <u>Finish</u> Cancel					

Figura 2.2.2 – Tela de Criação da Definição do Processo.

A utilização da ferramenta jGPD será ilustrada através de um processo de admissão de um novo funcionário, com o intuito de demonstrar os recursos da ferramenta (não entrando no mérito dos detalhes do processo que pode não corresponder a um caso real).

A criação de uma definição de processo implica na criação de uma pasta com o nome do processo (Processo\_Admissão\_Funcionário, neste caso) agregado à extensão '.par'. Os arquivos "gpd.xml" e "processdefinition.xml" devem pertencer a essa pasta. O primeiro arquivo conterá informações gráficas, usadas pelo editor de definição de processo. O conteúdo do arquivo "gpd.xml" poderá ser visto graficamente, por intermédio da aba Design. O arquivo "processdefinition.xml" contém a definição do processo sem informações gráficas e pode ser verificado na aba Souce. O jGPD assume que esses dois arquivos fazem parte de uma mesma família.

Na tela seguinte é representada a visualização do projeto com o processo criado:



Figura 2.2.3 – Tela Eclipse com o Processo Criado

A tela do eclipse já será apresentada com a visualização do módulo de modelagem do processo "Processo\_Admissao\_Funcionario.par" (Figura 2.2.3 - em azul no centro da tela), permitindo a elaboração gráfica do processo.

#### 2.3 Tratamento de Incompatibilidades

Caso a tela acima (representada pela Figura 2.2.3) não esteja similar a tela de seu Eclipse, pode estar havendo alguma incompatibilidade entre o plugin jGPD e o plugin MyEclipse. Para solucionar essa incompatibilidade é necessário desabilitar o MyEclipse, clicando no menu Help -> Software Updates -> Manage Configuration.



Figura 2.3.1 – Selecionando Gerenciador de Configuração pelo Menu do Eclipse.

Com isso será aberta a tela de Configuração (ilustrada abaixo), deve-se clicar no item 'MyEclipse Enterprise Workbench' e, em seguida, na janela a direita deve-se clicar no link

Disable (Desabilitar). Para concretizar essa operação o Eclipse deverá ser reiniciado.



Figura 2.3.2 – Desabilitando o Plugin MyEclipse do Eclipse.

### **3 DEFININDO UM PROCESSO**

Este tópico conterá mais detalhes a respeito da utilização do jGPD. Subtópicos a serem abordados:

- Barra de Ferramentas de Modelagem (Aba Diagram);
- Modelando Graficamente a Definição de um Processo;
- Explorando as Demais Abas de Visualização;
- Editando o Arquivo processdefinition.xml.

#### 3.1 Barra de Ferramentas de Modelagem (Aba Diagram)

A aba Diagram contém a visualização e os componentes para a modelagem do processo. Abaixo, será apresentada a característica de cada componente representado na Figura 3.1.1.



Figura 3.1.1 – Visualização Inicial do Editor para Definição de Processo.

Select e Marque: O botão Select permite selecionar os objetos do diagrama, o botão
 Marque permite que seja feita uma seleção simultânea de objetos, clicando e arrastando o mouse sobre os objetos desejados.

- **Start:** Cria um estado inicial para o processo, esse estado é um nó obrigatório para toda definição de processo.

- State: Representa um estado de espera. É particularmente útil quando um processo deve

aguardar por algum sistema externo. Na entrada do nó, pode haver uma ação para mandar uma mensagem a um sistema externo, mantendo o processo em estado de espera. Quando o sistema externo envia uma mensagem de resposta, o processo segue a execução.

- **End:** Cria um estado final para o processo, esse estado é um nó obrigatório para toda definição de processo, podendo haver mais que um nó do tipo end em um processo.

- Fork: Divide um caminho de execução em múltiplos caminhos. A continuidade do processo, neste caso, depende da intervenção de um dos possíveis responsáveis pela execução de um nó. Sendo que, pode ser considerado encerrado, apenas se todos os caminhos (transitions) forem percorridos.

- Join: Componente que aguarda a execução dos nós emitidos através do Fork, dando continuidade ao processo, quando todas as condições forem atendidas. Por tanto, o Join é o componente que faz a junção de múltiplos caminhos (transitions).

- Decidion: É um nó que representa uma decisão a ser tomada para seguir determinado fluxo. Deverá existir no mínimo duas transições de saída do nó do tipo decision. É possível modelar decisões de fluxo de duas formas distintas, dependendo de quem será responsável pela decisão: o processo (especificado na definição do processo) ou uma entidade externa.

 Decisão feita pelo processo: Existem duas formas de se especificar os critérios de decisão. O mais simples é adicionar elementos condicionais (conditions) nas transições. Condições são expressões BeanShell script que retornam um valor booleano. Em tempo de execução, o nó de decisão irá verificar as transições de saída e executará cada condição, considerando como resultado a primeira transição que retornar condição verdadeira. Alternativamente, pode-se especificar um DecisionHandler. Então, a decisão de fluxo ficaria a cargo de uma classe Java.

• <u>Decisão feita por uma entidade externa</u> (não constando na definição de processo), não precisa ser utilizado um nó do tipo decision, deve-se usar múltiplas transições de saída deixando um nó do tipo state. Neste caso, a transição de saída escolhida pode ser provida a partir de uma trigger externa que resume a execução após o estado de espera terminar.

- Node: É um nó, utilizado para representar alguma lógica funcional importante. Este tipo de nó precisa de um sub elemento action, que é executado quando o nó é alcançado pelo fluxo. O código no action handler deve obrigatoriamente ser responsável por propagar a execução. O componente Node é caracterizado por executar uma ação sem a necessidade de intervenção externa ou do usuário.

- Task Node: Refere-se a uma ou mais tarefas a serem realizadas por indivíduos. Um nó do tipo task pode ter zero ou mais tarefas, que são descrições estáticas na definição do processo. Quando o Task Node é executado, cada tarefa definida para o nó, é instanciada. Então, o nó fica em estado de espera, até o término de todas as tarefas e assim o fluxo de execução pode prosseguir.

- **Transition:** Cria uma ligação entre os nós de origem e de destino, possibilitando o fluxo de execução do processo.

#### 3.2 Modelando Graficamente a Definição de um Processo

Dando continuidade ao exemplo "Processo\_Admissao\_Funcionario", serão demonstradas as demais funcionalidades do plugin para realizar a definição de um processo.

Para criar os elementos do diagrama de processo, basta clicar sobre o componente desejado (task-node, decision, etc.) na barra de ferramentas de modelagem do jGPD (abordada no tópico anterior); e clicar na posição da tela em que se deseja representar o elemento. No caso das transições deve-se clicar no componente Transition (da barra de ferramentas), clicar no nó de origem e clicar no nó de destino. Pode-se definir o nome dos nós e transições diretamente nos elementos criados, até chegar ao diagrama representado a seguir:



Figura 3.2.1 Diagrama da Definição do Processo\_Admissao\_Funcionario.par

A figura acima representa exatamente o arquivo processimage.jpg, gerado automaticamente pelo jGPD. A medida em que a definição do processo for sendo incrementada, a imagem é automaticamente atualizada. A imagem acima pode ser encontrada na seguinte estrutura do projeto: Projeto\_Exemplo/src/process/Processo\_Admissao\_Funcionario.par/processimage.jpg.

#### 3.2.1 Janela de Visualização Outline

Ao lado direito da visualização gráfica do Processo\_Admissao\_Funcionario.par, se encontra a janela de visualização de nome Outline (ilustrada na figura abaixo), onde consegue-se ver em formato de árvore toda a estrutura do processo.

Por essa janela também consegue-se atribuir as ações e tarefas dentro dos devidos nós ou transições. Para isso, deve-se clicar com o botão direito do mouse sobre os nós criados e escolher a opção desejada.



*Figura 3.2.1.1 – Visualização da Janela Outline do Processo\_Admissao\_Funcionario.par* É uma forma de se acrescentar recursos ao processo de uma maneira amigável. Ao adicionar esses recursos, pode-se fazer verificações na aba Source (tópico 3.3.2) de como está sendo montado o código jDPL (padrão XML para jBPM) e quais tags estão sendo acrescentadas. Desse modo é possível familiarizar-se com a linguagem, para editar o código manualmente, se necessário. É importante considerar essa observação, pois algumas tags do jDPL não são geradas a partir do modo gráfico e será necessário editá-las manualmente.

Abaixo seguem relacionadas as opções que podem ser utilizadas pelos nó do processo:

- Start

- Delete Exclui o nó e todos os elementos definidos dentro deste nó.
- Add Task Adiciona uma tarefa a um nó. É possível acrescentar mais de uma tarefa a um nó, pois pode haver casos em que o responsável por algum nó precise executar mais de uma tarefa (paralelamente ou seqüêncialmente), para que o processo siga para o próximo nó.
- Add Actions Adiciona uma ação a um nó. As ações são usadas para acionar operações que poderão ser executadas em quatro momentos, na ordem a seguir:
  (1)Before Signal antes de enviar o sinal para a transição acionar outro nó;
  (2)Node Leave ao sair do nó de origem; (3)Node Enter ao entrar no nó de destino; (3)After Signal após sinal de que a transição já foi encaminhada ao nó

de destino. Após definidas as ações, será gerado no source do jDPL (códigofonte do XML) as tags "event" seguidas das tags "action" dentro da tag "tasknode".

- State

- Delete Idem ao Delete do Start.
- Add Actions Idem ao Add Actions do Start.
- End
  - **Delete** Idem ao **Delete** do **Start**.
  - Add Actions Idem ao Add Actions do Start.
- Fork
  - Delete Idem ao Delete do Start.
  - Add Actions Idem ao Add Actions do Start.

– Join

- Delete Idem ao Delete do Start.
- Add Actions Idem ao Add Actions do Start.
- Decision
  - Delete Idem ao Delete do Start.
  - Add Actions Idem ao Add Actions do Start.
- Node
  - Delete Idem ao Delete do Start.
  - Properties Exibe uma tela que possibilita o cadastro de uma ação a ser executada pelo nó, para isso basta marcar a caixa que contém no nome *Configure action handler* e especificar a classe que implementa a interface ActionHandler no campo *Class*.

- Add Actions Idem ao Add Actions do Start.
- Task Node
  - **Delete** Idem ao **Delete** do **Start**.
  - Add Task Idem ao Add Task do Start.
  - Add Actions Idem ao Add Actions do Start.
- Transition
  - Delete Idem ao Delete do Start.
  - Add Action Insere uma ação para ser executado quando o processo passar pela transição selecionada.
- Task
  - Delete Idem ao Delete do Start.
  - Properties Exibe uma tela que possibilita alterar algumas configurações da Task, para isso pode-se navegar através do menu de seleção lateral que possui as opções Assignment e General.

Na opção Assignment pode-se selecionar a forma que a tarefa será atribuída para o usuário, que pode ser através de uma swimlane, expressão ou uma classe que implementa a interface AssignmentHandler.

A opção General possibilita a alteração do nome da tarefa pelo campo *Name*, e também o cadastro de uma data de validade para a tarefa que deve ser preenchida no campo *Due date*, este campo deve seguir as seguintes regras de formulação:

A duração é especificada em horas absolutas seguindo a sintaxe:

<quantidade> <unidade>

Onde a <quantidade> é um texto que será interpretado pela classe Double.parseDouble(quantidade). A <unidade> deve ser qualquer um desses {second, seconds, minute, minutes, hour, hours, day, days, week, weeks, month, months, year, years}.

- Action
  - Delete Idem ao Delete do Start.
  - Properties Exibe uma tela para se cadastrar o nome da ação pelo campo Name, e também pode-se cadastrar uma classe em Java que implementa a interface ActionHandler para ser executada quando essa ação for escolhida.

– Event

- Delete Idem ao Delete do Start.
- Add Action Insere uma ação para ser executado quando o processo executar este evento.

**Nota**: A versão atual do plugin jGPD, através da aba Diagram, contempla a adição de ações (*Add Action*) para os componentes "Task Node" e "Event", mas não permite a adição de ações ao componentes "Task" na aba Diagram. Para efetuar essa operação será necessário utilizar a aba Design ou a aba Source.

#### 3.3 Abas de Visualização

#### 3.3.1 Aba Design

A aba Design representa o processo hierarquicamente, através de estrutura de árvores. Cada tag pode ser aberta ou fechada como uma árvore de diretórios comum. As tags também podem ser editadas nesta aba. Pode-se alterar o nome dos atributos das tags, remover esses atributos, remover as tags, entre outras ações. A aba Design trata-se de outra opção gráfica para realizar a definição do processo, porém ao invés da definição ser visualizada através de um diagrama do processo é visualizada através das tags organizadas hierarquicamente.

Para navegar nas tags da definição basta clicar nos ícones mais (+) ou menos (-), conforme a necessidade. Para adicionar, editar ou remover tags, atributos e outros elementos, basta clicar com o botão direito do mouse sobre o elemento desejado e selecionar a opção desejada.

A utilização deste recurso é mais adequada para fazer pequenos ajustes na definição do processo. Para realizar a elaboração da definição é mais produtivo utilizar a aba Diagram, devido à gama de componentes gráficos disponíveis. Segue abaixo uma ilustração da aba Design:

🛿 Processo_Admissao_Funcionario.par 🗙		
?=? xml	version="1.0" encoding="UTF-8"	
🗢 🖻 process-definition		
@ xmlns	http://jbpm.org/3/jpdl	
@ xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance	
③ xsi:schemaLocation	http://jbpm.org/3/jpdl http://jbpm.org/xsd/jpdl-3.0.xsd	
(a) name	Processo_Admissao_Funcionario	
▽ e start-state		
© name	inicio_admissao	
(® name	entrevista	
l to	entrevista_individual_decisiva	
▽ e end-state		
® name	final_admitido	
▽ e task-node		
(a) name	entrevista_individual_decisiva	
Iname	encaminhar_responsavel	
@ to	analise_entrevista	
▽ e node		
(a) name	documentação_pessoal	
(a) name	analise_situacao	
@ to	analise_objecoes	
Image: State		
▷ e state		
▽ e fork		
@ name	forkl	
E transition		
E transition		
E transition		
▷ 🖻 task-node		
Diagram Swimlanes Design Source		

Figura 3.3.1.1 – Visualização Aba Design do Processo\_Admissao\_Funcionario.par

**Nota**: Conforme mencionado no tópico anterior, a aba Design possibilita a adição de ações às tarefas. Toda ação (Action) deve sempre estar associada a algum evento, portanto para efetuar essa operação na aba Design basta seguir as seguintes instruções: (1)clicar com o botão direito sobre a tag desejada (Task, por exemplo); (2)selecionar a opção "Add Child -> event", indicando que a tag "Task" terá uma tag filha "Event"; (3)fazer o mesmo procedimento dos passos 1 e 2, a partir da tag "Event" criando um tag filha "Action"; (4)definir os atributos "name" e "class" da tag "Action".

#### 3.3.2 Aba Source

Contém o código jPDL do arquivo processdefinition.xml referente ao processo modelado na

aba Diagram. As alterações que se fizerem necessárias para o processo, e que não possam ser realizadas através da ferramenta gráfica, poderão ser feitas diretamente no código por esta aba. A figura a seguir indica, parcialmente, os componentes do gráfico expressados em jPDL (relacionados pelo traço em amarelo):



Figura 3.3.2.1 – Comparando Componentes do Diagrama com Código do Arquivo processdefinition.xml

**Nota**: Conforme mencionado no Tópico 3.2.1, a aba Source possibilita a adição de ações às tarefas. Na realidade a aba Source possibilita a edição total da definição do processo, porém requer maior conhecimento técnico por parte do usuário. Para suprir a necessidade de adicionar ações às tarefas, basta adicionar o código abaixo à tag "Task" em questão e renomear os atributos conforme a realidade do usuário:

O código acima trata-se de um exemplo real, portanto assim que o código for adicionado ao source da definição do processo, deve-se alterar os atributos para a realidade do sistema hospedeiro.

#### 3.3.3 Aba Swimlanes

Nesta aba podem ser feitas as declarações dos swimlanes (papéis) que serão utilizados para atribuição das tarefas do processo aos grupos/usuários cadastrados no Timoneiro SA.



*Figura 3.3.3.1 – Visualização Aba Swimlane do Processo\_Admissao\_Funcionario.par* No campo Assignment Type existem duas opções:

 Expression: Indica que a atribuição da tarefa será feita diretamente pela expressão indicada no campo Expression, as regras de nomenclaturas para se cadastrar as expressões são as seguintes:

primeiro --> proximo --> proximo --> . . . --> proximo

Uma expressão é resolvida da esquerda para a direita. O primeiro termo especifica um **Usuário** ou um **Grupo**, os termos seguintes calculam o próximo termo a partir do usuário ou grupo, os termos são separados pelo sinal -->.

Como primeiro termo pode-se utilizar os seguintes nomes:

- previous Significa o usuário que está atualmente autenticado no sistema. Ou seja, a pessoa que executou o passo anterior no processo.
- swimlane(nome-do-swimlane) Significa que o usuário ou grupo é escolhido a partir da instância do swimlane utilizado no processo.
- variable(nome-da-variavel) Significa que o usuário ou grupo é escolhido a partir da informação cadastrada na instância de variável. Esta variável deve conter um java.lang.String, que deverá conter uma expressão igual a um user(nome-do-usuário) ou group(nome-do-grupo).
- user(nome-do-usuário) Significa que o usuário é escolhido a partir do login passado como parâmetro.
- group(nome-do-grupo) Significa que o grupo é escolhido a partir do nome do

grupo.

Os termos seguintes podem conter os seguintes nomes:

- group(tipo-do-grupo) Retorna um grupo de um usuário, significa que o termo anterior deve ter resultado em um *Usuário*. Então ele procura em todos os grupos daquele usuário por um grupo que seja do tipo passado como parâmetro.
- member(papel) Retorna o usuário que realiza o papel dentro do grupo anterior, o termo anterior deve ter resultado em um *Grupo*. Então este termo procura entre os usuários do grupo por aquele que combina com o papel passado como parâmetro.
- Handler: Indica que a atribuição deverá ser feita via código lógico externo, a ser definido nos campos Class (nome da classe externa, esta classe deve implementar a classe AssignmentHandler) e Type.

Crie os swinlanes conforme demonstrado na figura acima, para exemplificar pode-se criar o swinlanes abaixo:

- Name: medico\_trabalho; Assignment Type: Expression; Expression: group(grupo\_medico).
- Name: candidato; Assignment Type: Expression; Expression: group(grupo\_candidatos).

Após criados os swinlanes, verifique na aba Source que o código foi alterado para constar os mesmos.

#### 3.4 Editando o Arquivo processdefinition.xml

A ferramenta jGPD está sendo constantemente melhorada, e na medida em que os releases forem feitos estarão disponíveis no pacote debian celepar-dev.

Como já citado anteriormente em outros tópicos, alguns elementos são impossíveis de implementar graficamente na versão 3.0.5. Isto pode ser verificado, por exemplo, no caso de se incluir uma tarefa (task) dentro de um nó task, e necessitar que seja disparada alguma ação

em algum evento desta tarefa. Se notar a opção de menu que abre a partir deste item não contempla incluir referência a uma ação dentro de algum evento (o que é suportado por um nó/elemento do tipo task).

Conforme observa-se na documentação de apresentação do jbpm (Timoneiro\_jbpm\_conceituacao.pdf, cap 2.1.12 JPDL xml schema), por exemplo, dentro da tag task (pg 28) existem vários parâmetros que podem ser configurados dentro dela e que ainda não são automaticamente feitos através da ferramenta, um deles é o elemento "event" (pg 21), e no caso de se desejar acionar alguma action dentro de um dos eventos suportados pelo elemento task, deve ser inserido o mesmo de forma manual diretamente na aba Source, veja na figura a seguir:



Figura 3.4.1 – Edição do Arquivo processdefinition.xml do Processo\_Admissao\_Funcionario.par

Note na figura que existem algumas tags sublinhadas em vermelho, isso é uma validação feita pela ferramenta, deixando o mouse sobre essa tag será mostrado o motivo da marcação. Também deixando o mouse sobre qualquer palavra surgirá um hint sobre a mesma.

No trecho acima, dentro do nó task "entrevista\_individual\_decisiva", foi então acrescentado manualmente a chamada a action "envio\_email" no evento "task-create" que ocorre a partir do elemento task "verificar\_interesse\_experiencia", veja ele transcrito em seqüência e destacado em amarelo:

<task-node name="entrevista\_individual\_decisiva">

<task name="verificar\_interesse\_experiencia" swimlane="analista\_rh">

<controller> <variable access="read,write,required" name="data_ag&lt;br&gt;&lt;/controller&gt;&lt;/th&gt;&lt;th&gt;enda"></variable></controller>	
<event type="task-create"></event>	
<action name="envio_en&lt;/td&gt;&lt;td&gt;nail'&gt;&lt;/action&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/event&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/task&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;pre&gt;&lt;event type=" node-enter"=""></action>	
<action class="gov.pr.celepar.pacote.ClasseMail" name="envio_emai&lt;br&gt;&lt;/event&gt;&lt;/td&gt;&lt;td&gt;l"></action>	
<transition name="encaminha&lt;br&gt;&lt;/task-node&gt;&lt;/td&gt;&lt;td&gt;&lt;pre&gt;ar_responsavel" to="analise_entrevista"></transition>	

Como essa action já é referenciada em outro evento do processo, não foi necessário informar a classe que a implementa na aplicação.

No trecho acima foi inserido o tag controller pelo visualizador Outline, clicando com o botão direito do mouse sobre a tarefa "verificar\_interesse\_experiencia" e clicando em Properties, com isso abre a tela de propriedades possibilitando configurar o Assignment (neste caso configurado com o swinlane "analista\_rh") e Controller, de onde se incluiu a variável "data\_agenda" para o controller padrão. Poderia ter sido indicado um controller externo definido como uma classe que ficaria no código da aplicação. Essas variáveis são visíveis dentro da instância do processo e persistidas, podendo ser utilizadas pela aplicação.

Existe uma restrição na edição manual pela aba Source, que ocorre quando os nós são colocados diretamente ou se alteram as posições das tags de referência aos nós dentro do código, com isto a localização do desenho dos nós no diagrama poderá ser modificado e até travar. Esse é um detalhe que deverá ser resolvido nas próximas versões do plugin jGPD.

O conselho então é que somente sejam efetuadas adições ou alterações que não sejam referentes ao nó diretamente, mas sim relacionadas a seus parâmetros e elementos internos.

Enfim, após a definição do processo ter sido gerada através do plugin, o processo estará pronto para ser cadastrado no Timoneiro SA. Para interagir a definição gerada ao Timoneiro S.A. deve-se verificar o documento Timoneiro\_manual\_usuario\_sa.pdf.

## 4 GLOSSÁRIO

- Plugin: Um plugin ou plug-in é um programa que serve normalmente para adicionar funções a outros programas para prover alguma função particular ou muito específica.
- Workflow: Conjunto organizado de atividades e passos relacionados, que coletivamente atingem um objetivo.
- Token: Elemento sinalizador interno de uma instância de um processo que indica a passagem de estados no fluxo do processo.

# **5 REFERÊNCIAS**

- http://www.jboss.com/products/jbpm