Diseñador Gráfico de Proceso JBoss jBPM

Cómo Empezar: Cómo crear su primera definición de proceso

3.0 alpha2

Tabla de Contenidos

Público Objetivo

Todos los desarrolladores y analistas de proceso que están iniciándose en el uso de JBoss jBPM deben leer la presente guía de Cómo Empezar. Les entrega un inicio rápido mostrando cómo crear una definición de proceso.

Prólogo

El presente documento presenta el uso del Diseñador Gráfico de Proceso JBoss jBPM (GPD) para crear procesos de workflow. Le ayuda a quienes lo usan por primera vez en las siguientes tareas :

Traducción al Español en el contexto del proyecto Met@logo y auspiciada con fondos de la Unión Europea
www.metalogo.org

- Instalar el plugin Eclipse JBoss jBPM GPD disponible en el area de descargas de JBoss jBPM.
- Configurar un proyecto Java en Eclipse y lo prepara para llevar a cabo un desarrollo de proceso controlado por pruebas.
- Utilizar el asistente de creación para crear una definición de proceso vacía.
- Utilizar la paleta de diseño para darle forma a la primera definición de proceso.
- Mostrar cómo se puede inspeccionar la definición de proceso xml como archivo xml.
- Configurar un proyecto Java en Eclipse y prepararlo para llevar a cabo un desarrollo controlado con pruebas
- Escribir un ejemplo de caso de prueba de proceso.

Si tiene consultas, le rogamos contactar a Koen Aers o Tom Baeyens para obtener información adicional.

Capítulo 1. Instalación del Diseñador Gráfico de Proceso JBoss jBPM

El JBoss jBPM GPD viene en forma de plugin para el ampliamente conocido Eclipse IDE. De modo que un requisito para instalar el Diseñador es, por supuesto, contar con una versión de Eclipse operativa. La version de jBoss jBPM GPD en la que está enfocado el presente documento es la versión beta. Dicha version debería funcionar con todas las versiones de Eclipse 3.1, a partir de M7. Si no cuenta con la copia correcta de Eclipse, es posible descargarla desde http://www.eclipse.org. La instalación del proyecto webtools es opcional dado que los plugins necesarios están incluidos en la descarga de JBoss jBPM GPD. La descarga viene en forma de sitio de actualización Eclipse comprimido en zip. Descargue la versión 3.0 desde http://www.jbpm.org. La versión 3.0 del diseñador viene incluido en las versiones 3.0 correspondientes de proyecto jBPM y de proyecto de base de datos jBPM.

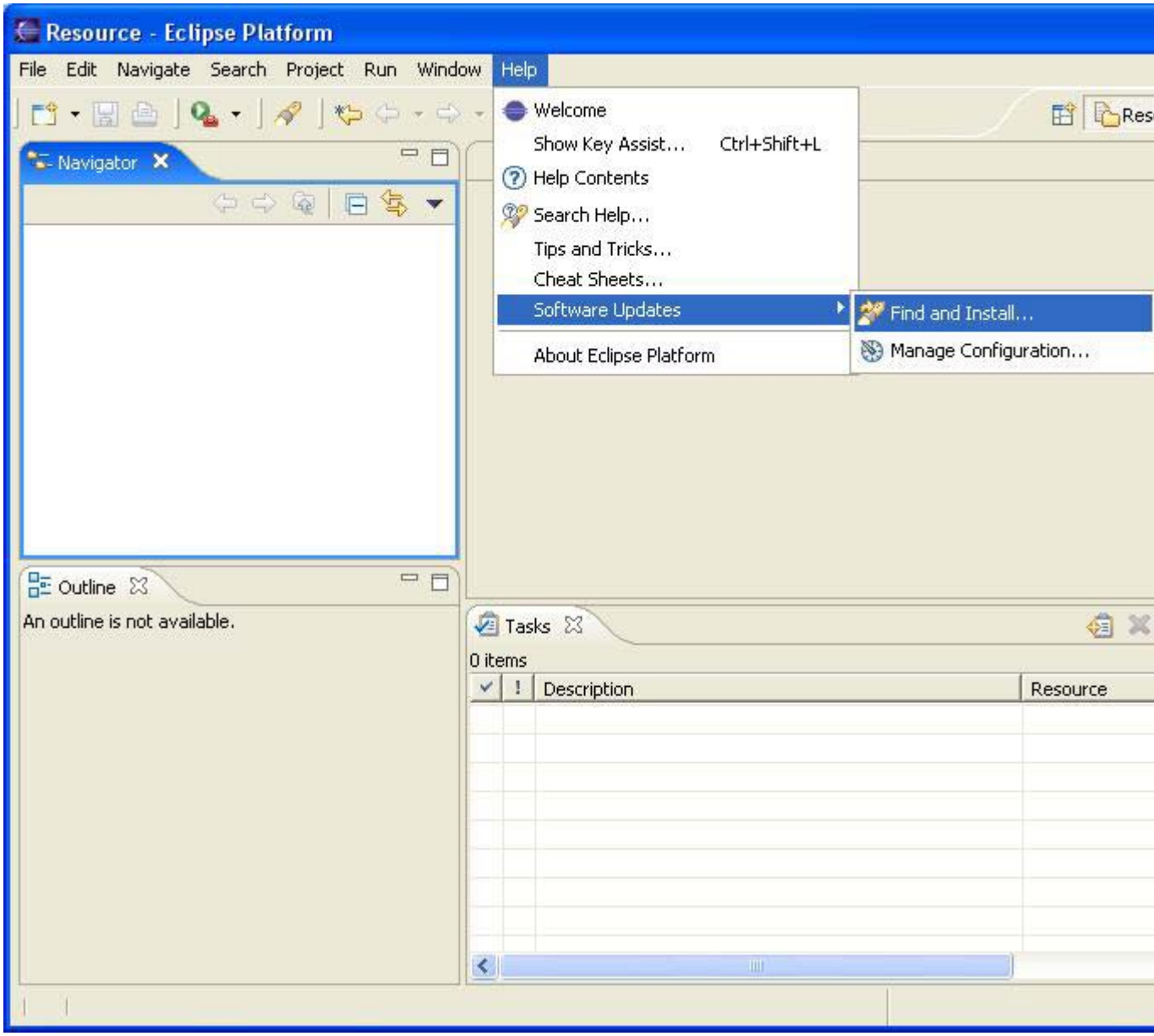

Figura 1.1. Buscar e Instalar

Ahora que ya Eclipse está operativo y que se ha descargado el paquete JBoss jBPM GPM en algún lugar de la unidad de disco duro, vamos a describir el proceso de instalar el plugin. Los usuarios de Eclipse experimentados se sentirán muy cómodos con la presente descripción, dado que se trata del mecanismo normal de instalación de un plugin para sitios de actualización de archivos. Según se ilustra en la *Figura 1.1, "Find and Install "* es necesario navegar hasta el submenú *'Find and Install...'* del menú *Help*.

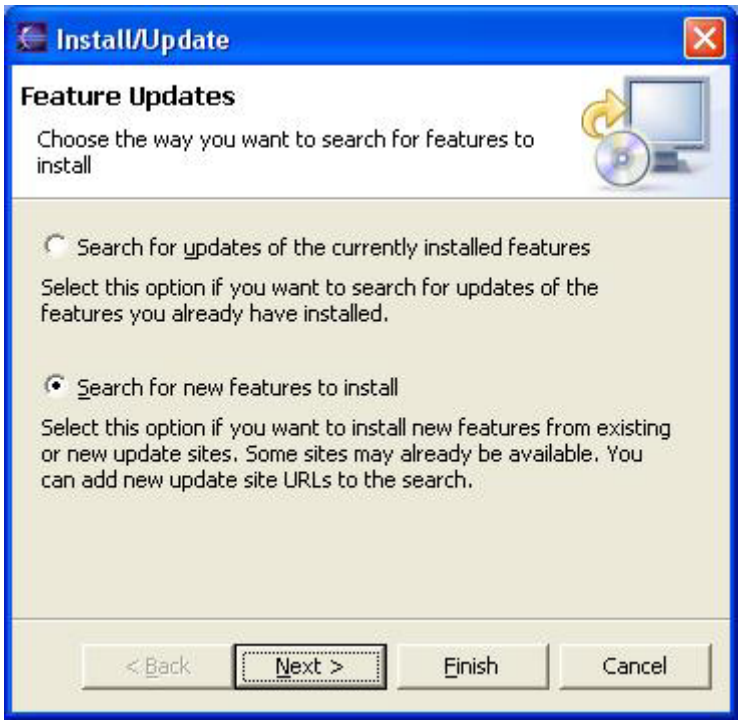

Figura 1.2. Buscar Nuevas Características

Figura 1.3. Sitios de Actualización

Al hacer clic en el submenú 'Find *and Install...'* que se muestra anteriormente se abre un asistente, tal como se muestra en la *Figura 1.2, "Search New Features "*. Dicho asistente permite elegir entre buscar nuevas actualizaciones de plugins actualmente instalados y buscar nuevas características. Esta es la opción que necesitamos, de modo

que la seleccionamos y presionamos el botón 'Next'. En la página siguiente del asistente, que se muestra en la *Figura 1.3, "Update Sites to Visit "*, hacemos clic en el botón *'New Archived Site...'*. Se abre el diálogo *'Select Local Archive Site'* (*Figura 1.4, "Select Local Archive Site "*), navegamos hasta una actualización comprimida recientemente descargada y la seleccionamos.

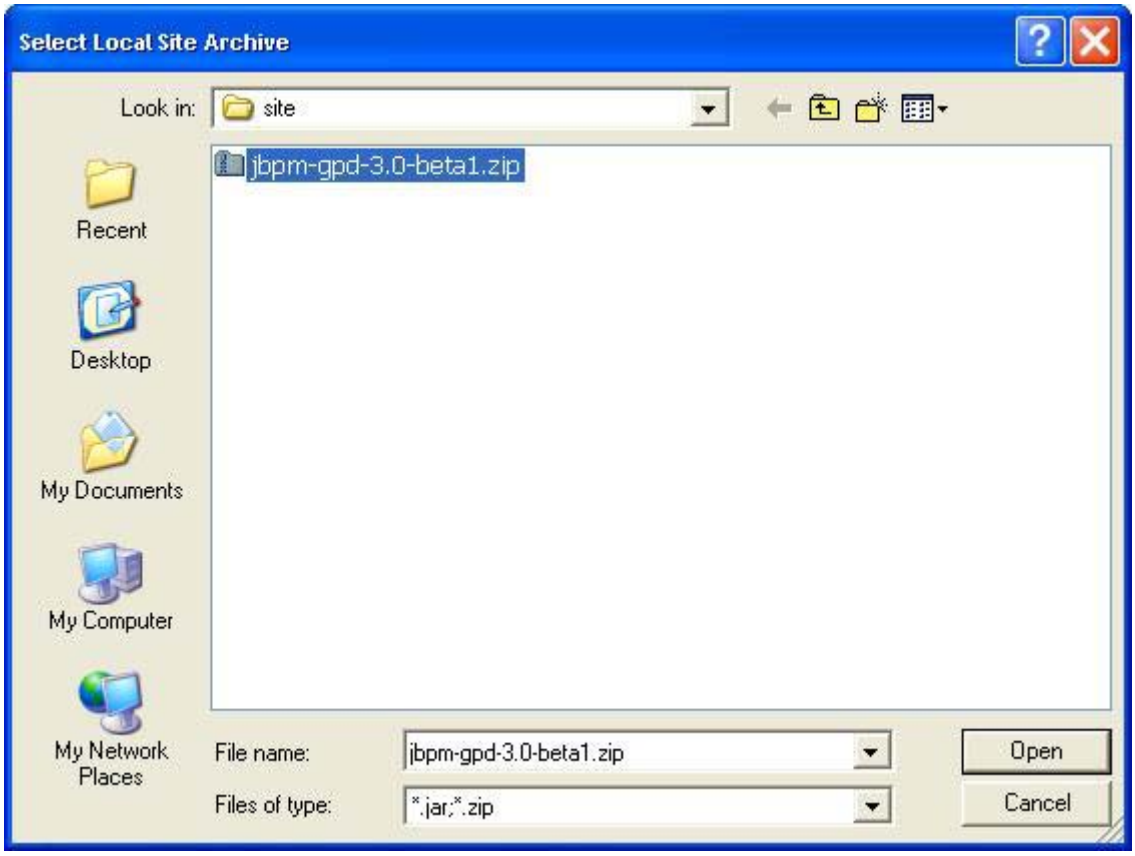

Figura 1.4. Seleccionar Sitio de Almacenamiento Local

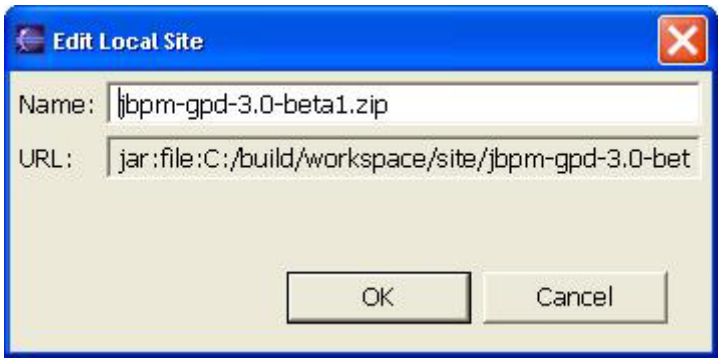

Figura 1.5. Editar Nombre de Sitio Local

Al seleccionar el archivo de almacenamiento y al hacer clic en el boton 'Open' llegamos a otro diálogo que nos permite editar el nombre del sitio de actualización (*Figura 1.5, "Edit Local Site Name "*). Aceptamos los valores predeterminados y esto nos lleva nuevamente al asistente de instalación (*Figura 1.6, "Select Update Site to Visit "*). Debe

aparecer el archivo seleccionado que contiene el sitio del plugin del jBPM Designer. Asegúrese que esté seleccionada la casilla de verificación junto a éste antes de hacer clic en el botón *'Finish'*.

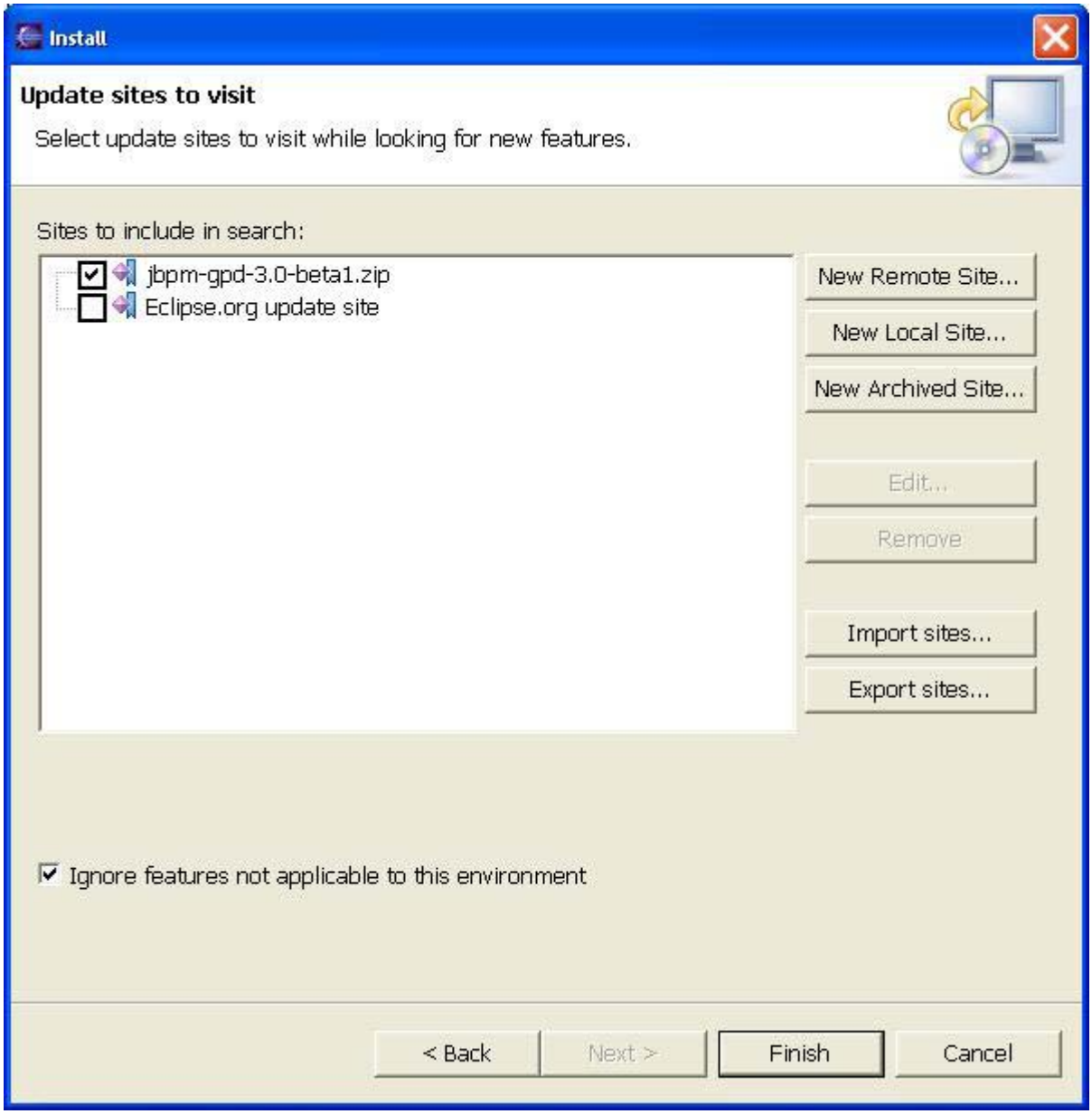

Figura 1.6. Seleccionar Sitio de Actualización

El asistente *'Select Update* Site' se cierra, pero todavía no hemos terminado dado que inmediatamente se abre el asistente 'Updates' (*Figura 1.7, "Select the Feature to Install "*). En este asistente se deben expandir los tres sitios de actualización recientemente agregados, seleccione la casilla de verificación junto a la característica *'org.jbpm.ide.feature feature 3.0.0'* y haga clic en el botón *'Next'*. Llegamos a la página *'Accept Feature License'* del asistente (*Figura 1.8, "Accept Feature License "*). Seleccione *'I accept the terms in the license agreement'* si es así y haga clic nuevamente en el botón *'Next'*.

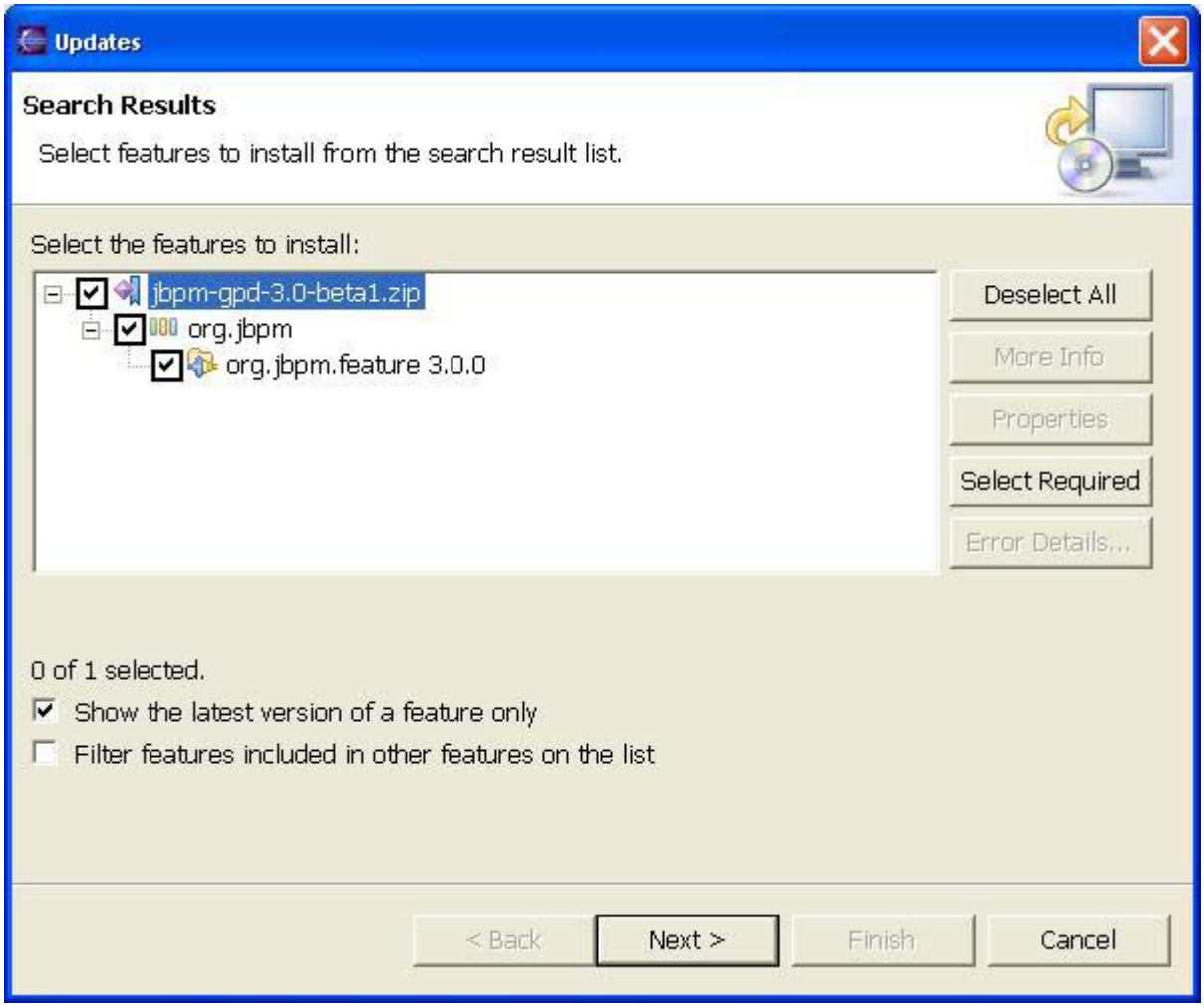

Figura 1.7. Seleccione la Característica que se va a Instalar

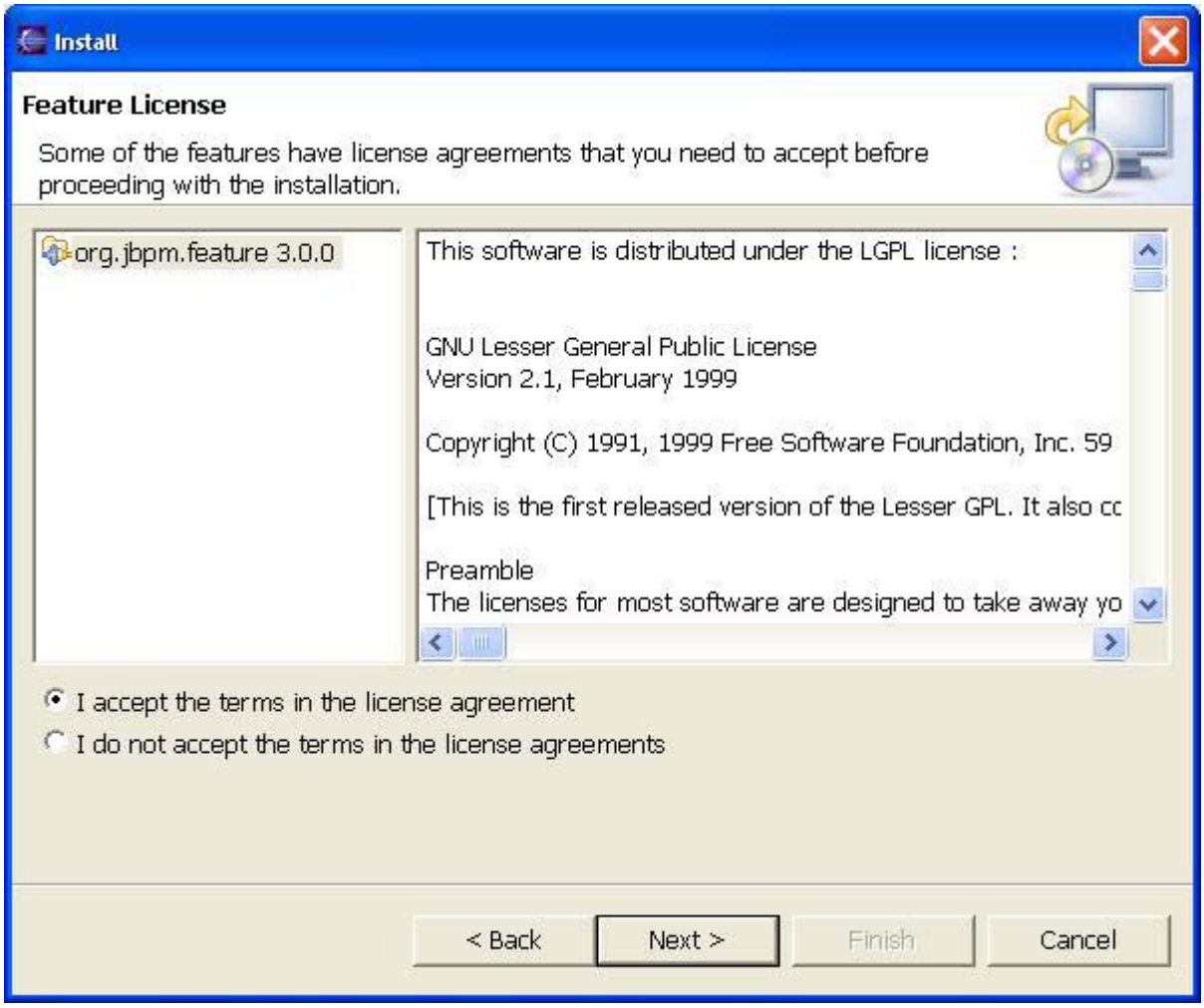

Figura 1.8. Aceptar Licencia de la Característica

Al hacer clic en el botón *'Next'* en la página *'Accept Feature License'* llegamos a la última página del asistente de instalación (*Figura 1.9, "Install the Selected Feature "*) : la página *'Installation'* . Haga clic en *'Finish'* para finalizar este asistente.

Durante un momento aparecerá un monitor de progreso (*Figura 1.10, "Progress of the Download "*), el cual indica que Eclipse está ocupado extrayendo información necesaria desde el archivo de almacenamiento para poder instalar los plugins contenidos. Después de un momento aparece un nuevo asistente (*Figura 1.11, "Feature Verification "*).

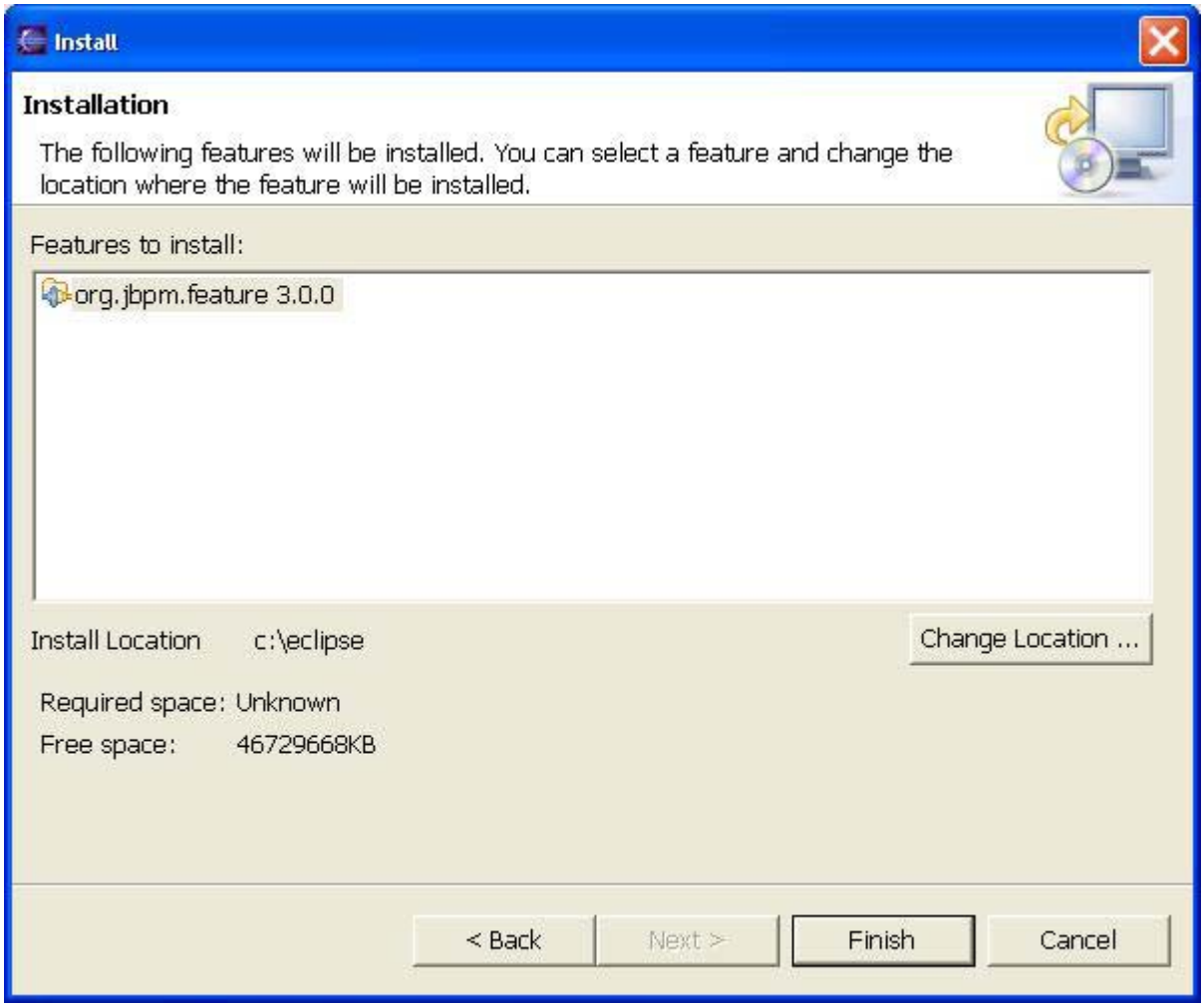

Figura 1.9. Instalar la Característica Seleccionada

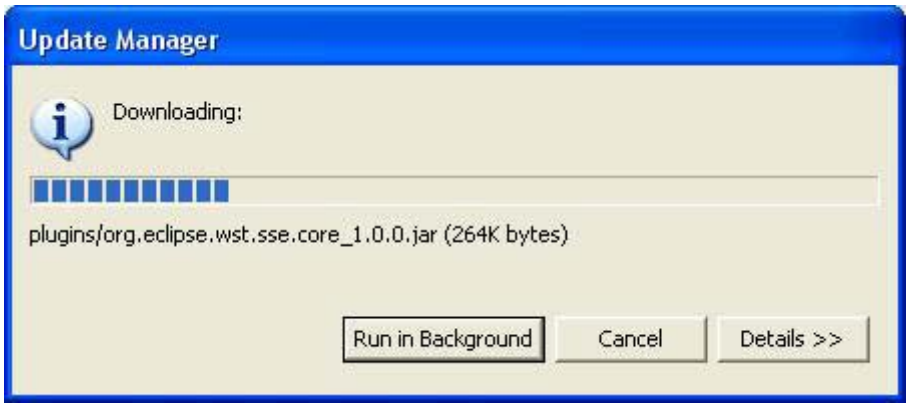

Figura 1.10. Progreso de la Descarga

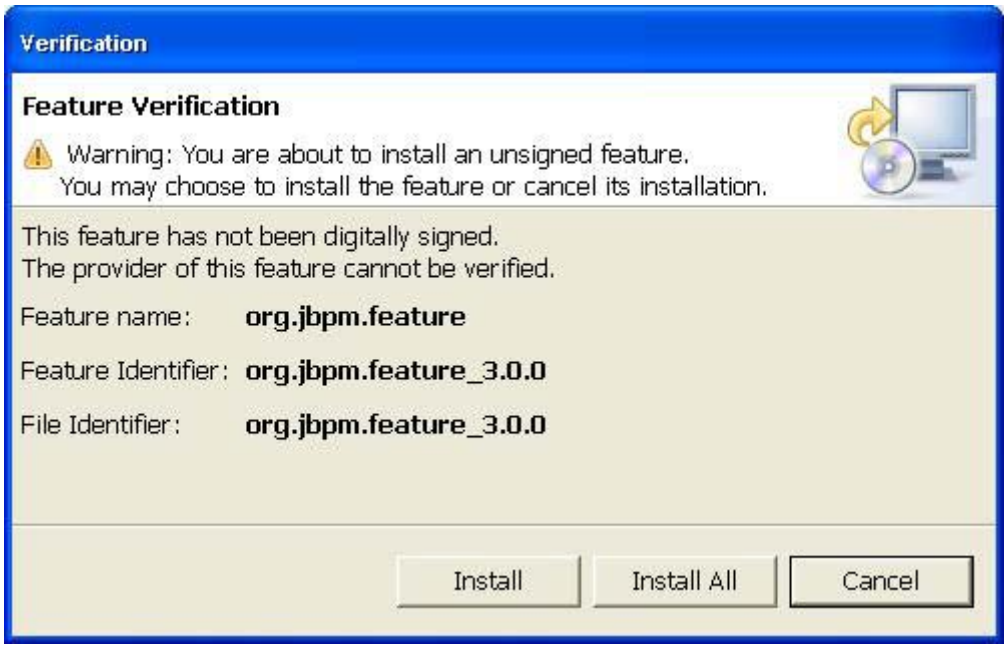

Figura 1.11. Verificación de la Característica

Al hacer clic en el botón *'Install All'* aparece un diálogo de progreso (*Figura 1.12, "Installation Progress "*). Después que termina la instalación otro diálogo nos pregunta si deseamos reiniciar el workbench en forma inmediata (*Figura 1.13, "Restart the Workbench "*. Queremos poder utilizar los plugins recientemente instalados inmediatamente, así que hacemos clic en *'Yes'*.

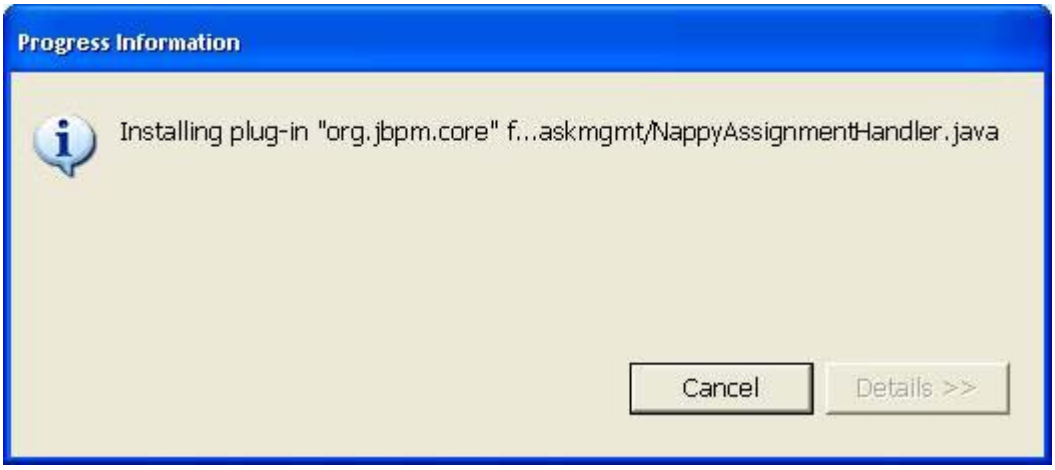

Figura 1.12. Progreso de la Instalación

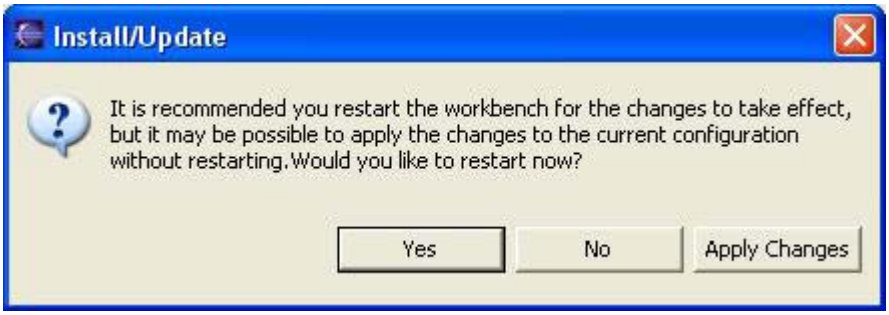

Figura 1.13. Reiniciar el Workbench

Felicitaciones. El plugin JBoss jBPM GPD está instalado y listo para usarse. En el capítulo siguiente vamos a ver cómo podemos configurar un proyecto y crear una definición de proceso.

Capítulo 2. Tour Guiado por JBoss jBPM GPD

2.1. Cómo crear un Proyecto jBPM

Hemos incluido un asistente en el plugin GPD para crear un proyecto jBPM. Hemos optado por crear un proyecto que ya contenga una serie de artefactos avanzados que ignoraremos durante esta sección. En el futuro mejoraremos este asistente y ofreceremos la posibilidad de crear un proyecto jBPM vacío, así como proyectos basados en plantillas obtenidas del turorial de jBPM.

Los proyectos jBPM en Eclipse son en realidad proyectos Java con una cantidad de opciones adicionales, de modo que cambiamos a la perspctiva Eclipse Java. Para crear un nuevo proyecto jBPM utilizando el asistente de creación de proyecto, seleccionamos *'New->Project...'* y en el diálogo New Project seleccionamos *'JBoss jBPM -> Process Project'* (*Figura 2.1, “New Project Dialog ”*). Al hacer clic en *'Next'* vamos a la página del asistente donde debemos especificar el nombre y ubicación del proyecto. Elegimos por ejemplo *'Hello jBPM'* como nombre y aceptamos la ubicación predeterminada (*Figura 2.2, “Process Name and Location ”*).

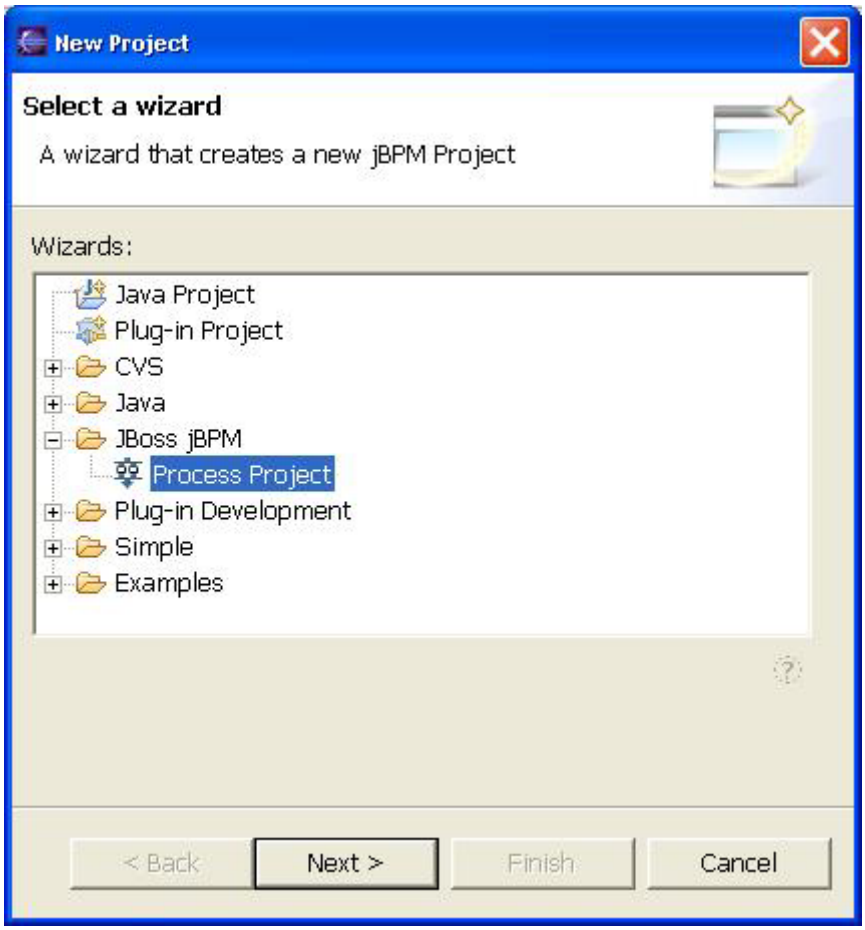

Figura 2.1. Diálogo de Nuevo Proyecto

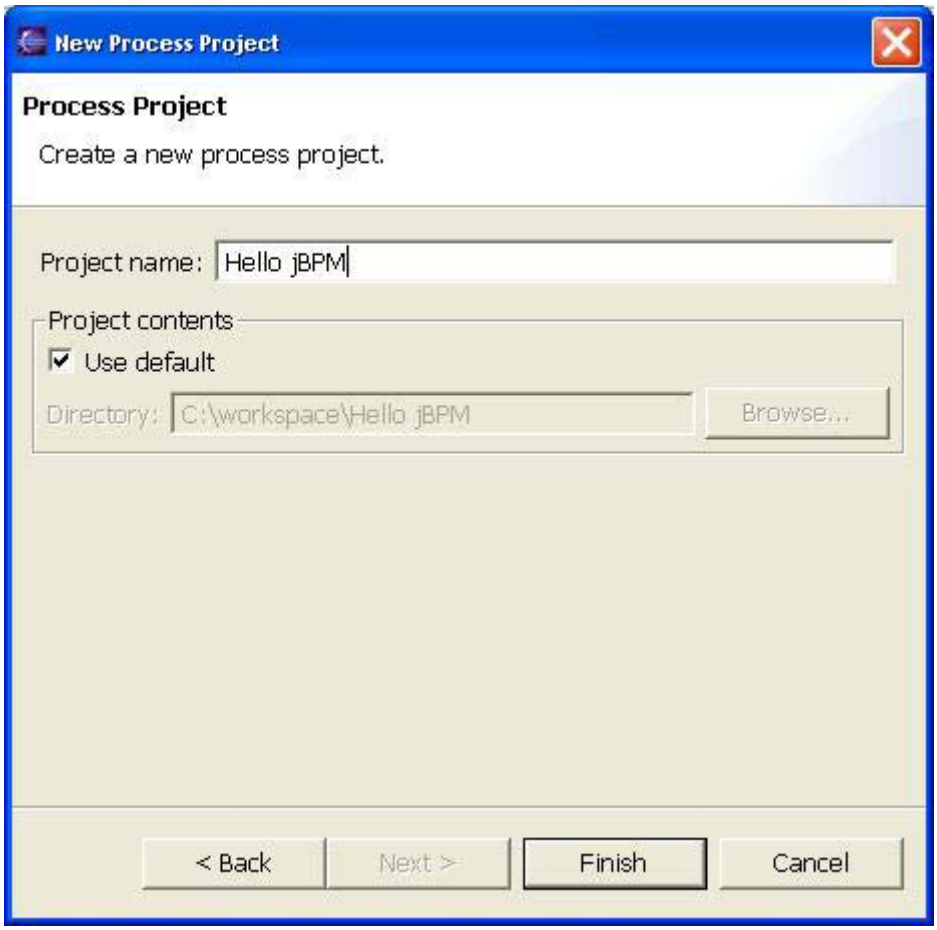

Figura 2.2. Nombre y Ubicación del Proceso

Al hacer clic en *Finish* se crea el proyecto. El asistente crea cuatro carpetas de origen: una para los procesos *('src/process'),* una para los java sources *('src/java'),* una para las pruebas de unidad *('test/java')* y una para los recursos como *jbpm.properties* y los archivos *hibernate.properties* ('src/resources'). Además se agrega al proyecto un contenedor de classpath con todas las bibliotecas esenciales jBPM (*Figura 2.3, "Layout of the Process Project "*

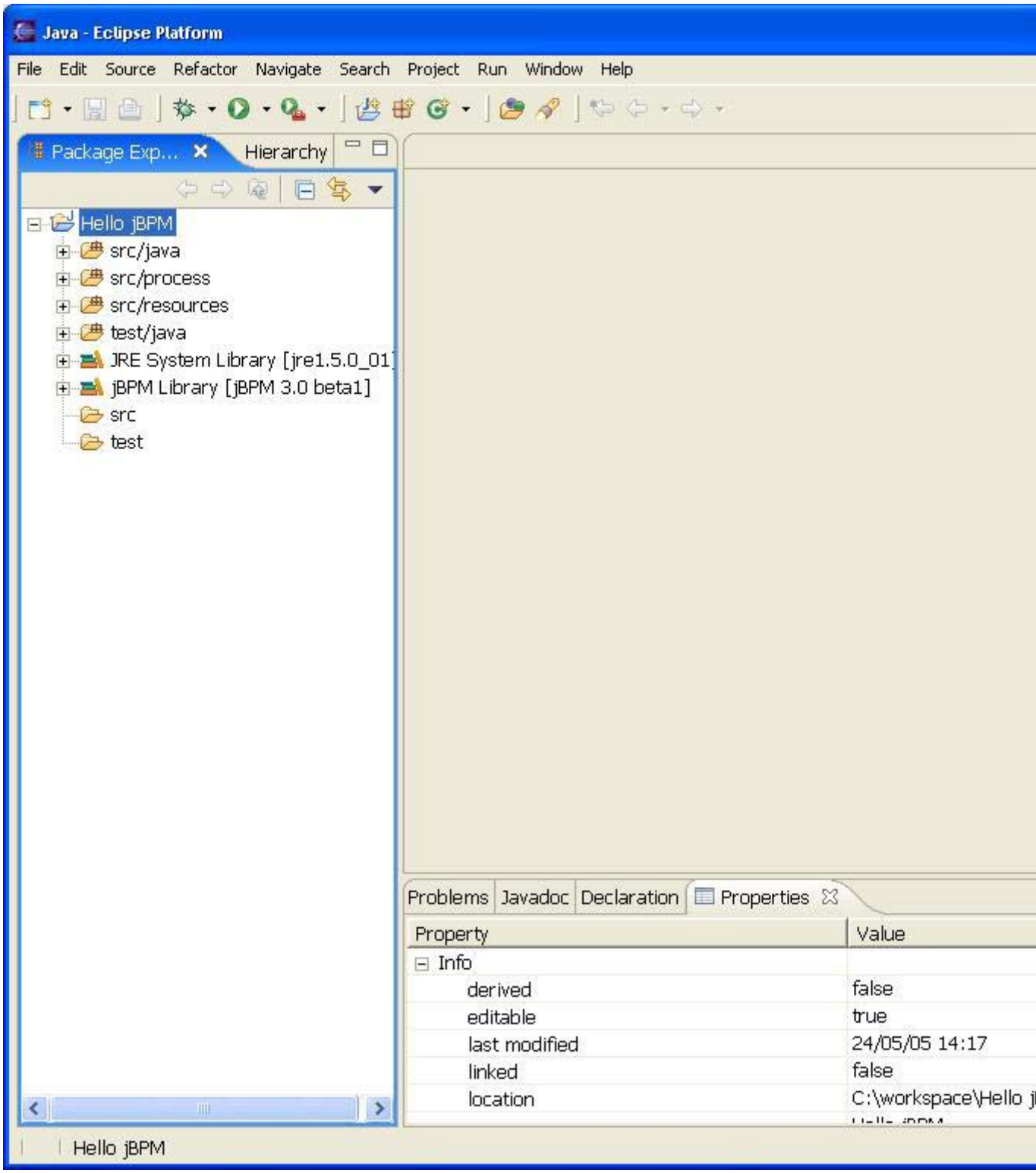

Figura 2.3. Diseño del Proyecto de Proceso

Al mirar el interior de cada una de las diferentes carpetas de origen se revelarán una serie de artefactos generados, pero los dejaremos intactos por el momento. En cambio, trabajaremos con otro asistente que nos permite crear una definición de proceso vacía.

2.2. Cómo crear una Definición de Proceso Vacía

Cuando se crea el proyecto podemos utilizar un asistente de creación para crear una definición de proceso vacía. Al hacer clic en el ítem de menú *'File->New->Other...'* obtendrá el asistente 'New'. El asistente se abre en la página *'Select Wizard'* (*Figura 2.4, "The Select Wizard Page "*).

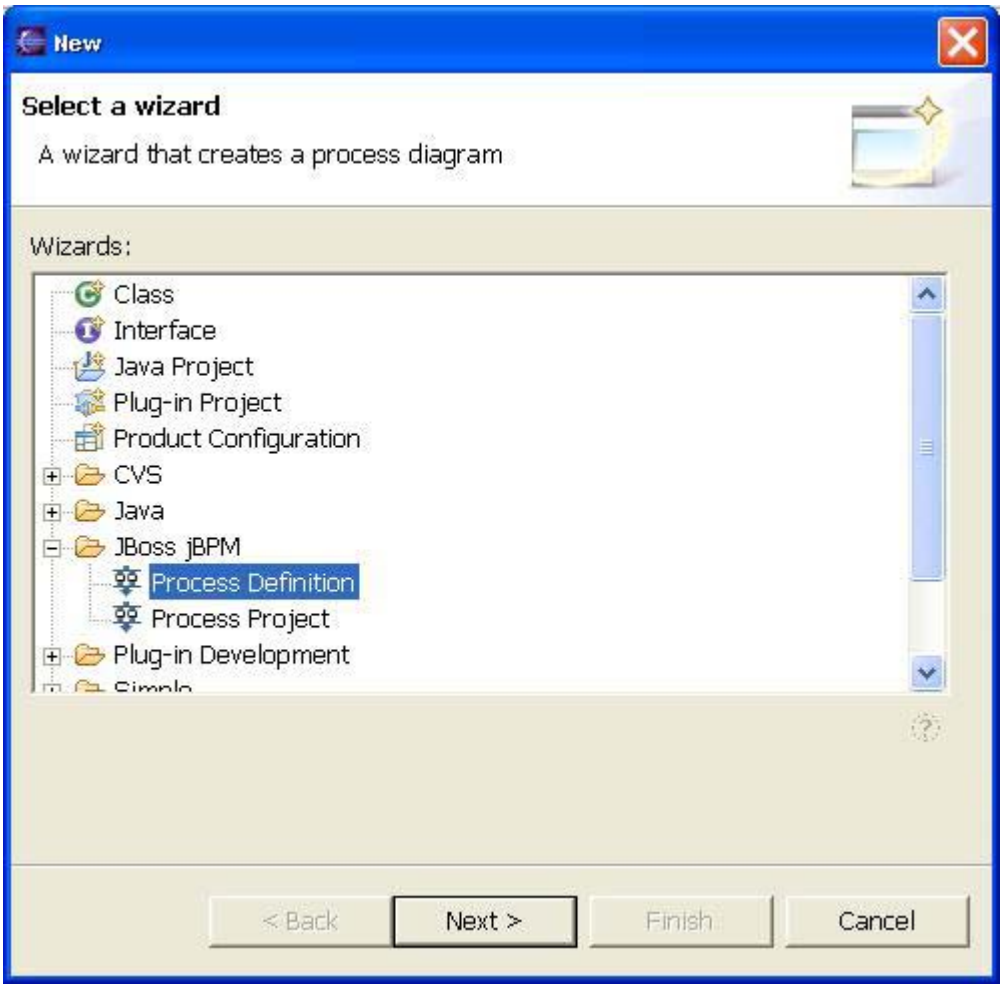

Figura 2.4. Página de Selección de Asistente

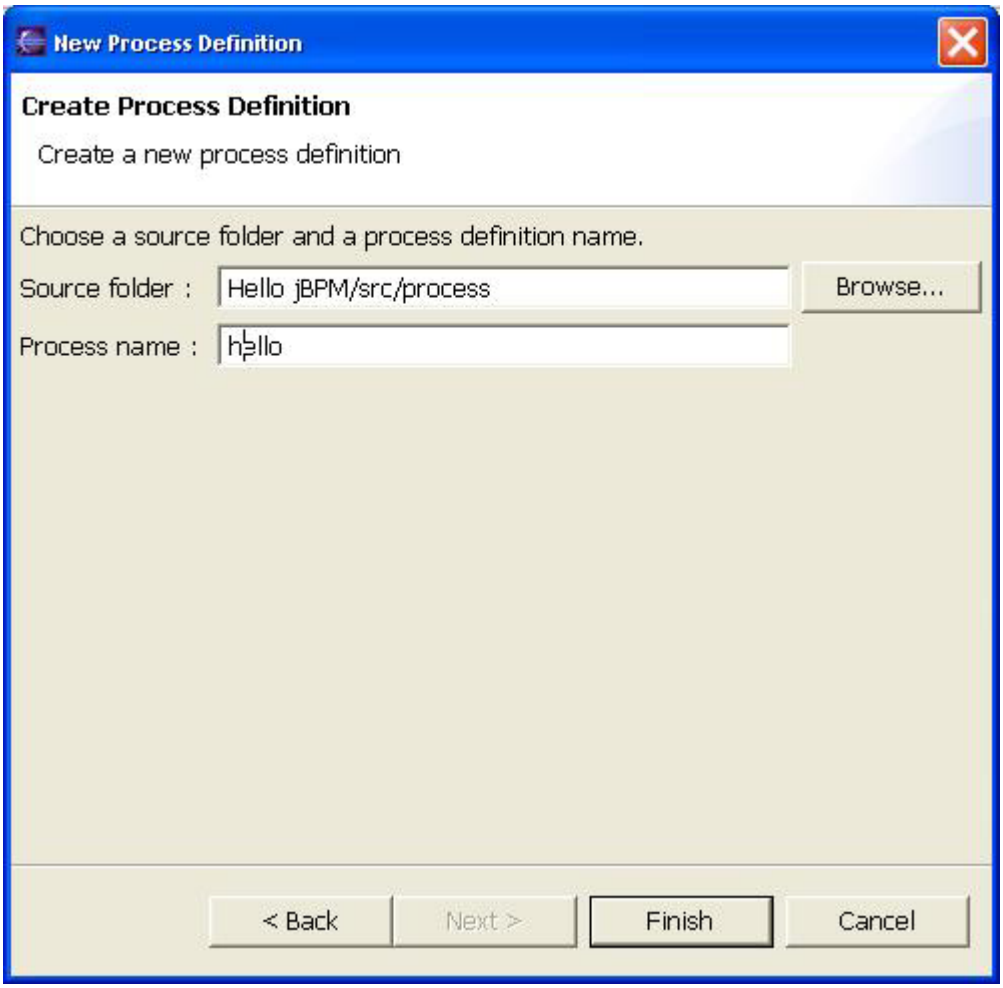

Figura 2.5. Página de Creación de Nueva Definición de Proceso

Al seleccionar la categoría 'JBoss jBPM', luego el ítem 'Process Definition' y al hacer clic en el botón 'Next' nos lleva a la página 'Create Process Definition' (*Figura 2.5, "The Create New Process Definion Page "*). Elegimos 'hello' como nombre del archivo de almacenamiento de proceso. Haga clic en el botón 'Finish' para finalizar el asistente y abrir el editor de difinición de proceso (*Figura 2.6, "The Process Definition Editor "*). Como se puede ver en el explorador del paquete crear una definición de proceso involucra la creación de una carpeta con el nombre de la definición de proceso y una extensión *'.par'* y poblar esta carpeta con dos archivos *.xml : gpd.xml* y processdefinition.xml. El primero de estos dos contiene la informacíón gráfica utilizada por el editor gráfico de proceso. Aunque es posible ver los contenidos con un editor xml genérico, el editor predeterminado que abre este archivo es el editor de definición de proceso. El archivo *processdefinition.xml* contiene la información de la definición de proceso sin la información de representación gráfica. Actualmente, el GPD asume que estos dos archivos son secundarios. Más adelante habrá soporte para configuraciones más sofisticadas.

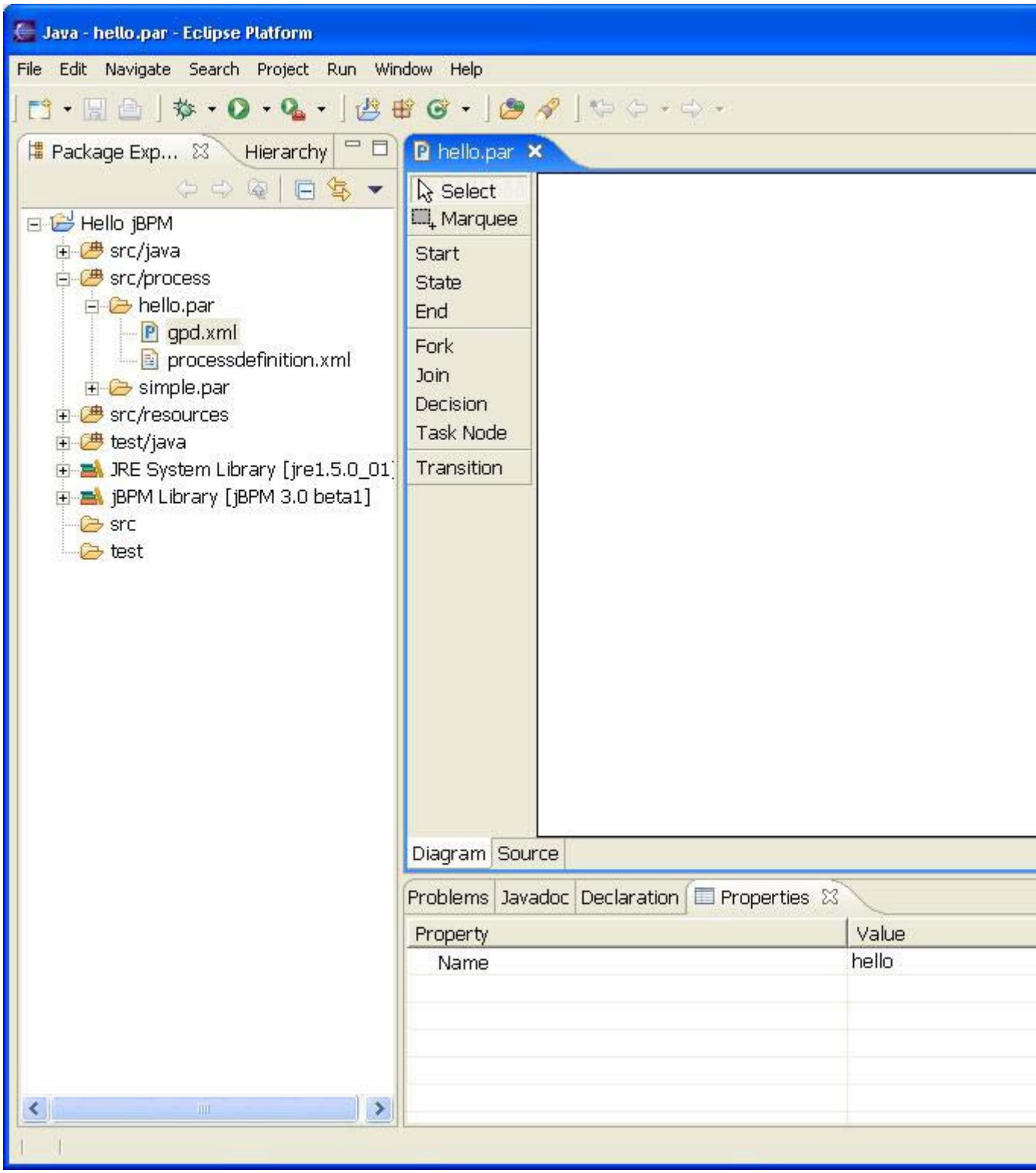

Figura 2.6. Editor de Definción de Proceso

2.3. Una Definición Mínima de Proceso

Crearemos una definición de proceso muy simple, que consta de un estado de inicio, un estado intermedio y un estado de término.

2.3.1. Cómo agregar nodos

Seleccione *'Start', 'State'* y *'End'* respectivamente en la paleta de herramientas y haga clic sobre el lienzo para agregar estos nodos a la definición de proceso. El resultado debería verse similar a la *Figura 2.7, "A Simple Process With Three Nodes "*

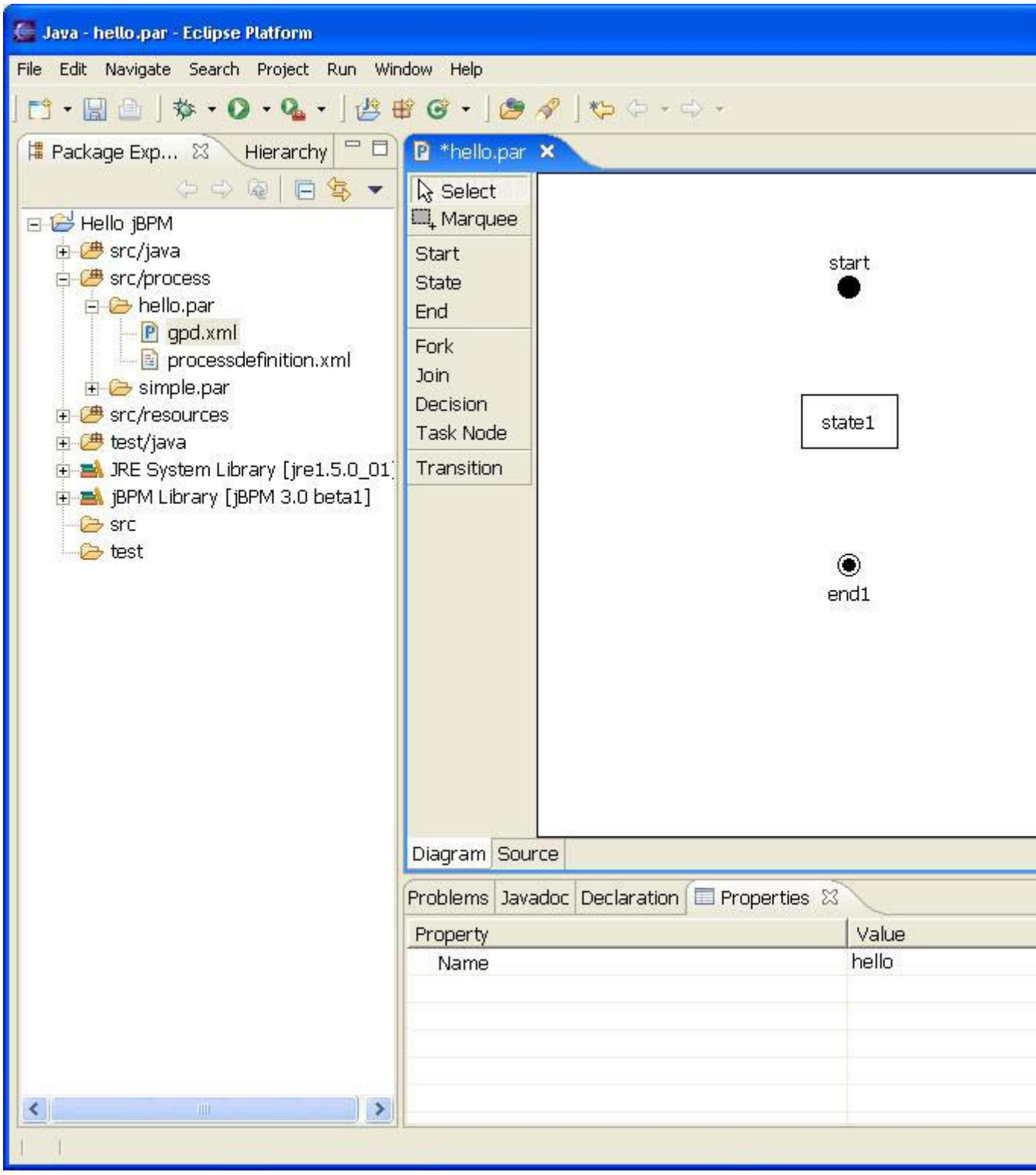

Figura 2.7. Proceso Simple con Tres Nodos

2.3.2. Cómo agregar Transiciones

Conectaremos los nodos con transiciones. Selecione la herramienta *'Transition'* en la paleta de herramientas y haga clic en el nodo *'Start'*, luego vaya al nodo *'State'* y haga

clic nuevamente para ver cómo se hace la transición. Siga los mismos pasos para crear una transición desde el nodo *'State'* al nodo *'End'*. El resultado se ve como la *Figura 2.8, "A Simple Process With Transitions "*.

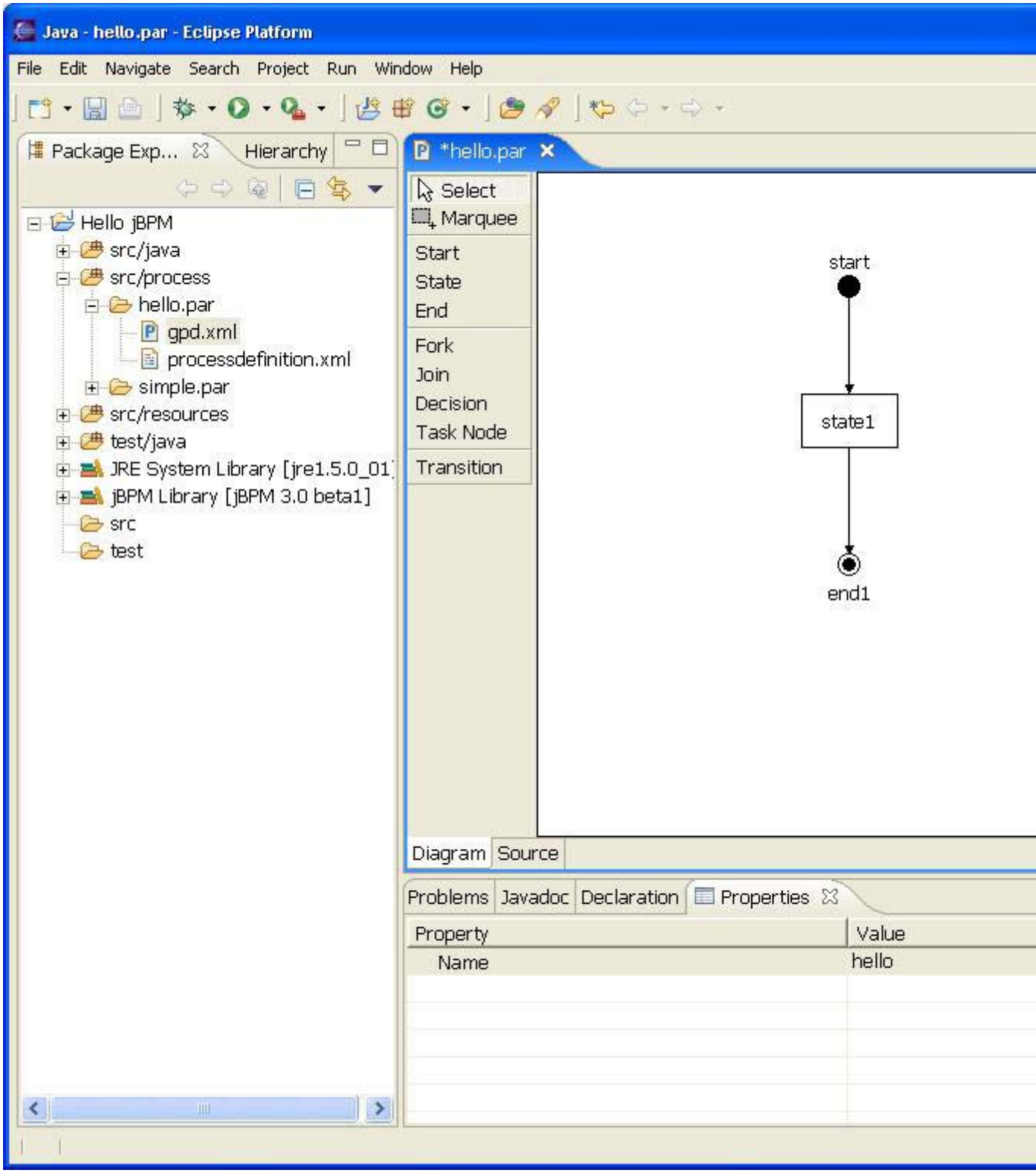

Figura 2.8. Un Proceso Simple con Transiciones

2.4. Vista de Diagrama

Se puede ver cómo se traza un diagrama del proceso en la vista de diagrama Eclipse, si está visible. El diagama de proceso viene en dos tipos diferentes. Uno de ellos es la clásica vista de árbol. Se puede ver el diagrama de árbol de nuestro proceso mínimo en la *Figura 2.8, "A Simple Process With Transitions "*. La otra posibilidad es ver el diagrama como una vista en miniatura desplazable. Esta posibilidad se ilustra en la *Figura 2.9, "The Outline as Thumbnail "*. Dado que la definición de proceso aun es bastante simple dado que sólo tiene tres nodos, esta vista en miniatura por el momento es de mayor tamaño que el gráfico real, pero en el caso de gráficos complejos esta característica es particularmente interesante.

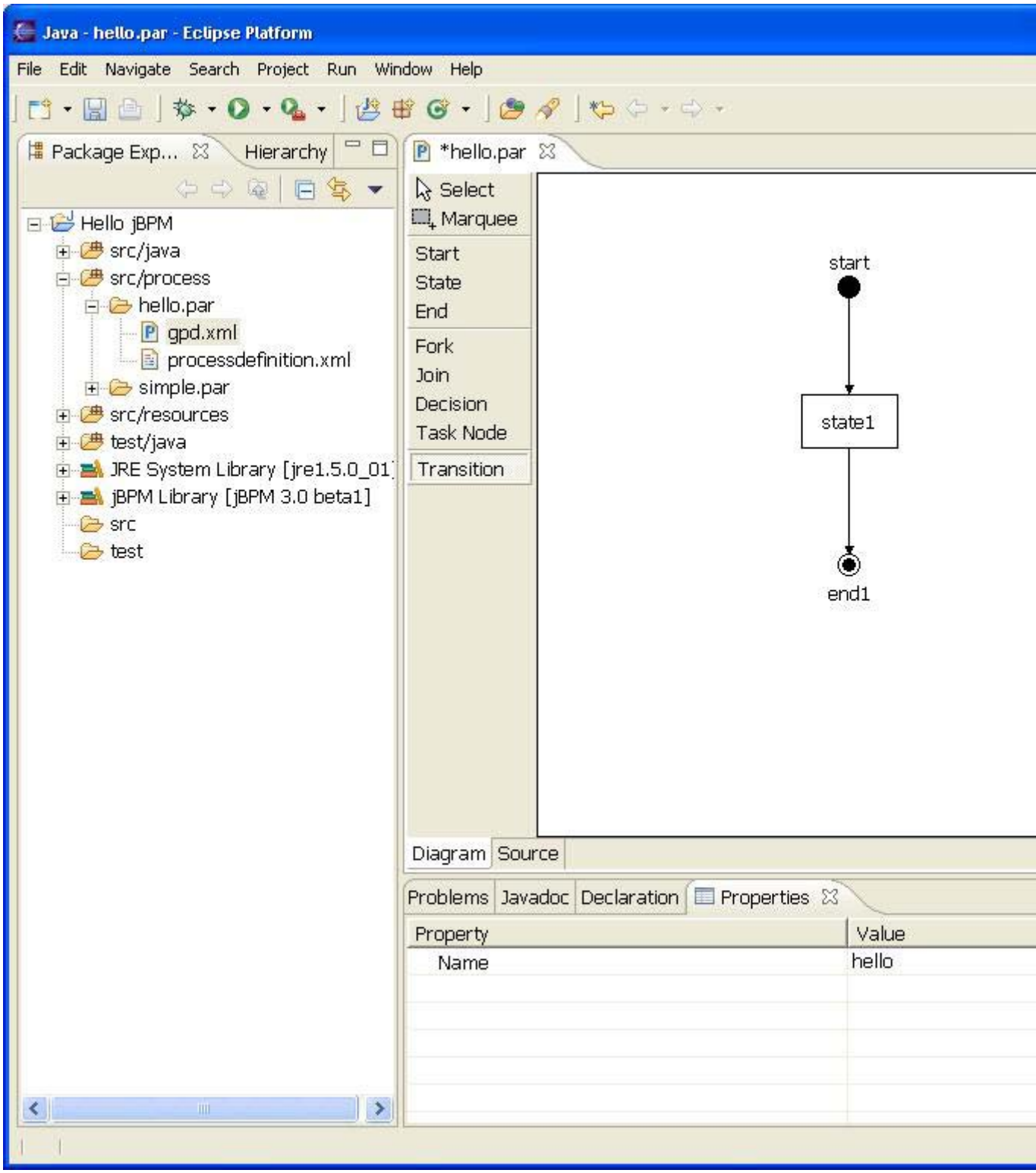

Figura 2.9. El Diagrama como Vista en Miniatura

Usted puede alternar entre estas dos vistas del diagrama utilizando los botones de la barra de herramientas del diagrama.

2.5. La Vista Propiedades

En la vista de Propiedades Eclipse es posible ver las propiedades pertinentes del ítem seleccionado. Algunas de estas propiedades se pueden editar directamente en la vista propiedades. Un ejemplo de una propiedad directamente editable es la propiedad de nombre de la definición de proceso. Como se puede ver en la *Figura 2.10, "The Properties of a Process Definition "*, la propiedad de nombre de la definición de proceso se puede cambiar a *'jbay'*.

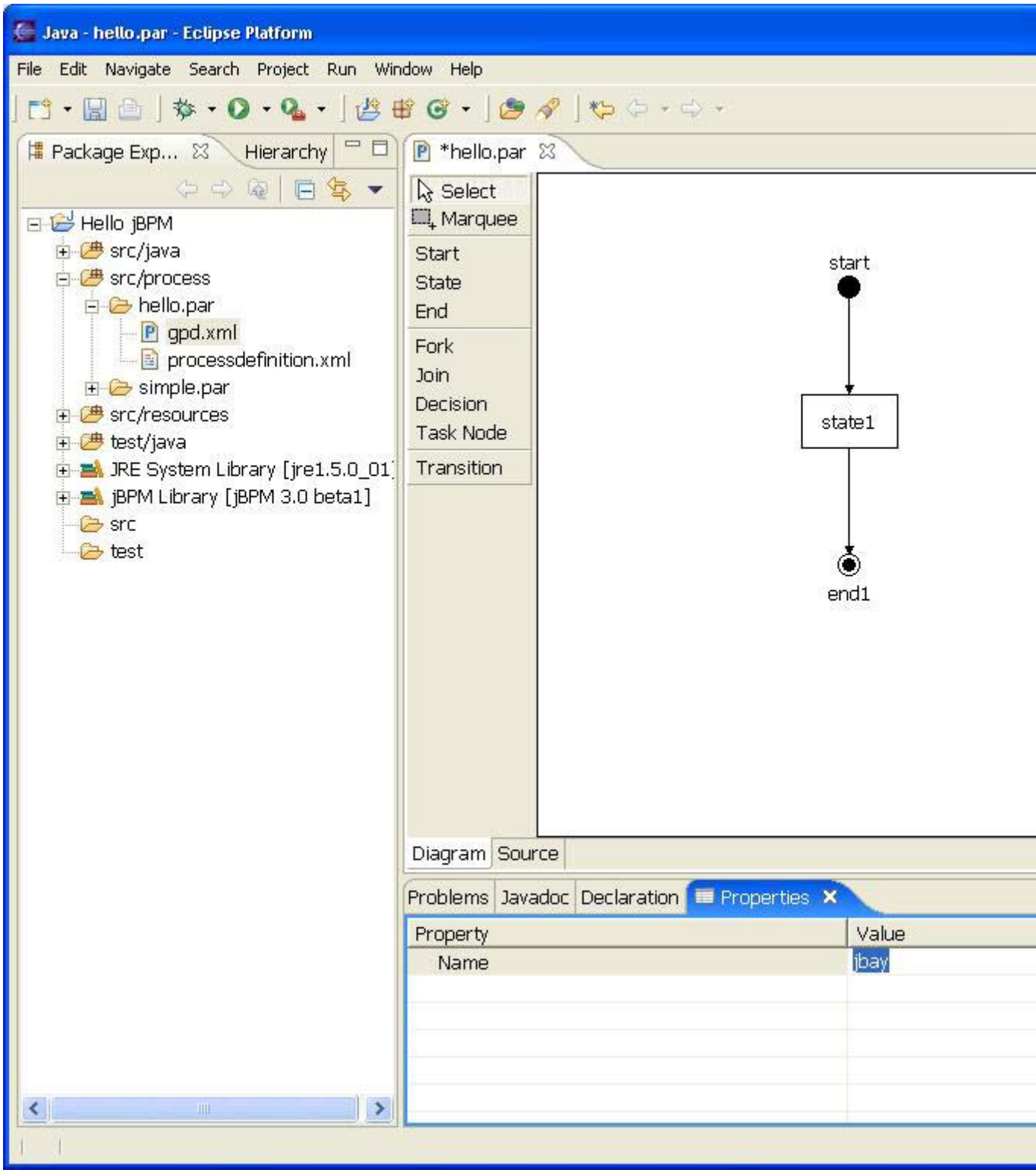

Figura 2.10. Propiedades de una Definición de Proceso

Cuando seleccionemos la primera transición, ya sea haciendo clic sobre ella en el lienzo o haciendo clic sobre su nodo en la vista de árbol del diagrama, veremos las propiedades de esta transición en la vista de propiedades (*Figura 2.11, "The Properties of a Transition "*). Podemos editar el nombre de la transición, pero las propiedades *'Source'*

y *'Target'* son sólo de lectura. Cambiamos el nombre de la primera transición a *'to_auction'*. Repetimos este mismo cambio para la segunda transición y la denominamos *'to_end'*.

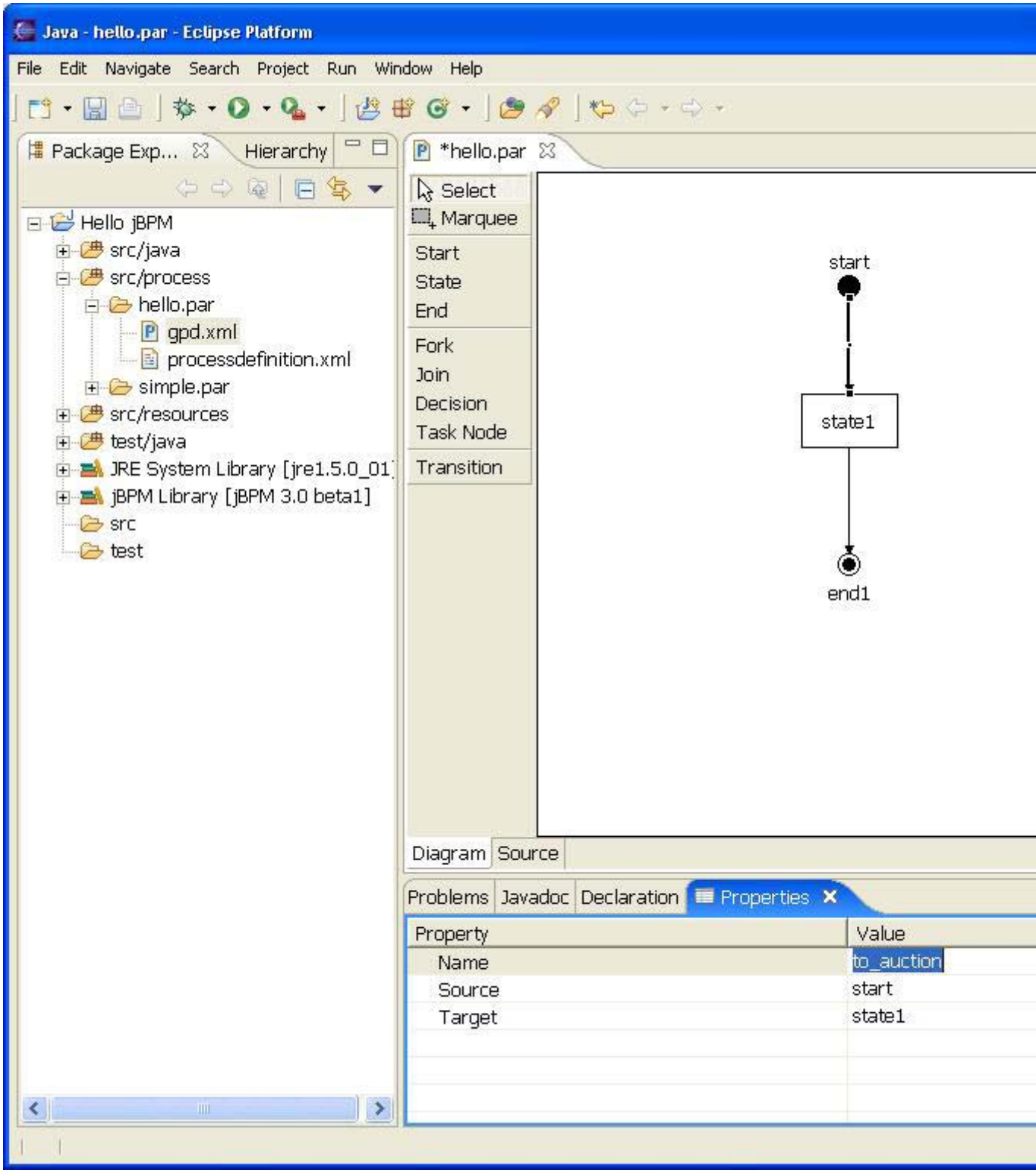

Figura 2.11. Propiedades de una Transición

2.6. Edición Directa

Algunas propiedades se pueden editar en forma directa en el editor gráfico. Un ejemplo de esto es la propiedad *'Name'* de los nodos. Esto se puede editar directamente al seleccionar el nodo al cual se quiere cambiar el nombre y luego hacer clic una vez dentro del nodo. Esto habilita el editor en el nodo según se muestra en la *Figura 2.12, "Directly Editing the Node Name "*. Cambiamos el nombre del nodo a *'auction'*.

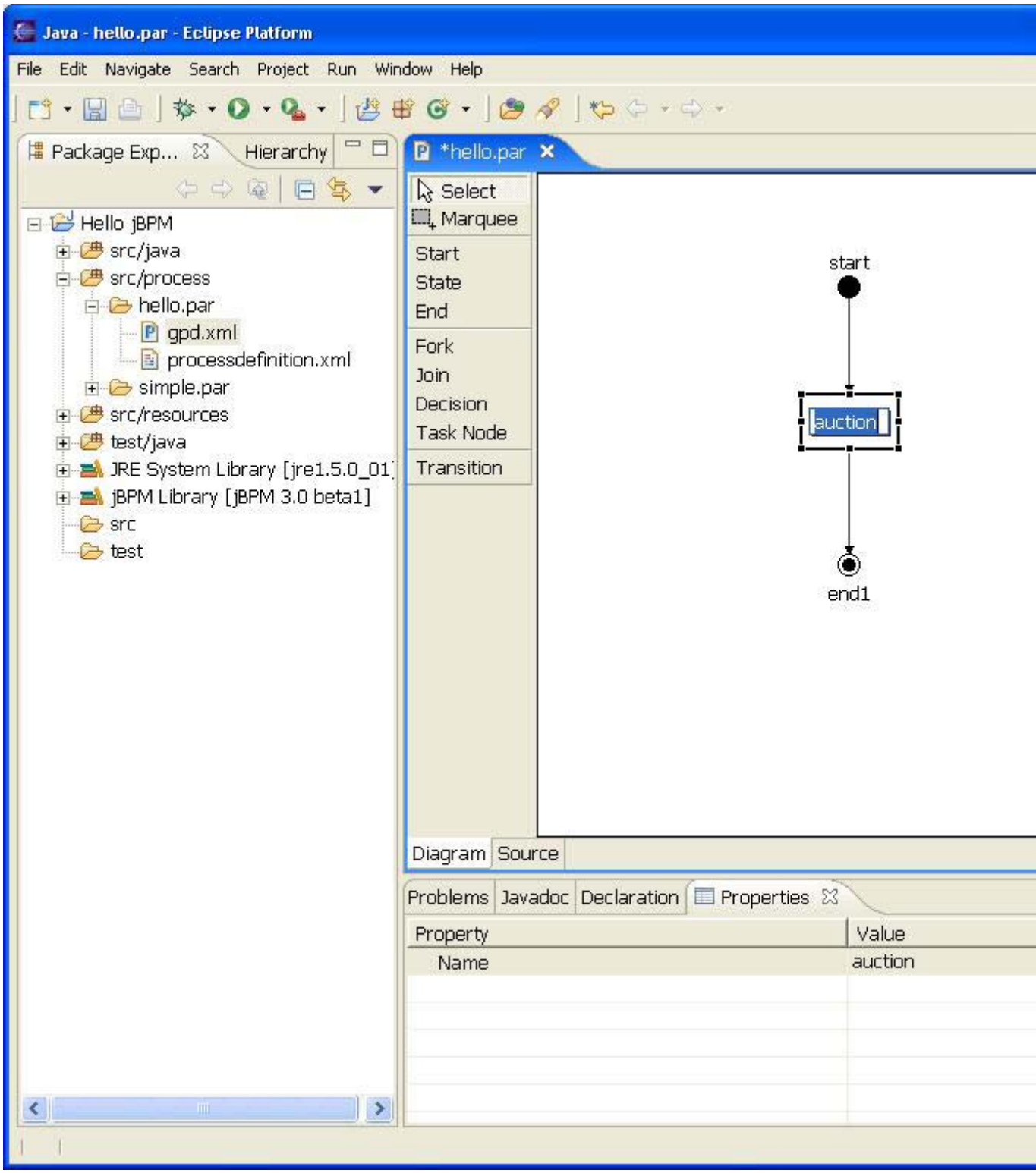

Figura 2.12. Editar Directametne el Nombre del Nodo

2.7. La Vista de Origen

Ahora que nemos definido una definición de proceso simple, podemos dar un vistazo al xml que se está generando por debajo. Para ver este xml haga clic en la ficha origen en el editor de definición de proceso. (*Figura 2.13, "The Source View "*).

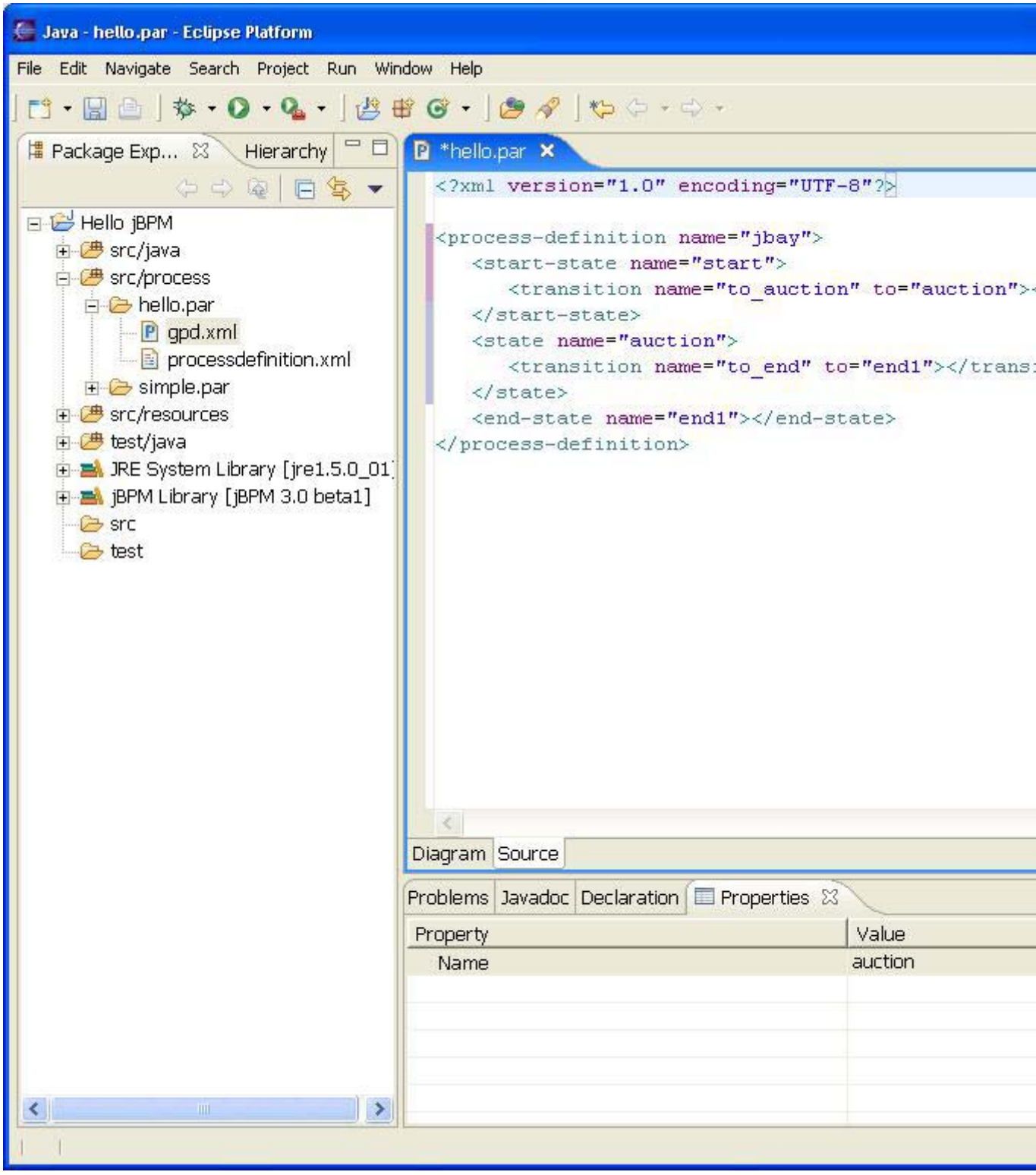

Figura 2.13. La vista origen

Esta ficha origen es editable, así que si conoce cómo se maneja jpdl va a poder crear o manipular sus definiciones de proceso directamente en el xml de origen. Pero si desea hacerlo, cabe mencionar que en este momento el diseño del dibujo se puede desorganizar. Quizá en una etapa posterior se agregarán algoritmos inteligentes de diseño. También, la validez del xml todavía no está vigente. Esto está en la lista de tareas por hacer y será agregado en el futuro cercano.

Capítulo 3. Desarrollo de Proceso Controlado por Pruebas

Una de las ventaja más importantes del enfoque liviano de JBoss jBPM hacia BPM y manejo de workflow es que los desarrolladores puedan aprovechar sus habilidades y técnicas de programación más conocidas. Una de estas técnicas es la prueba de unidades y el desarrollo impulsado por pruebas. En este capítulo mostraremos de qué forma los programadores que utilizan JBoss jBPM GPD pueden utilizar una técnica que hemos bautizado como *'Test Driven Process Development'* para crear definiciones de proceso y probar su corrección.

Al crear el proyecto 'Hello jBPM', el asistente de creación de proyecto ya ha implementado todos los requerimientos de biblioteca que necesitamos para comenzar a escribir pruebas unitarias de jBPM. Estos están contenidos en el contenedor de bibliotecas de jBPM y, sorprendentemente, el más importante es el archivo jar que contiene las clases esenciales de jBPM. Se debe mencionar que is posible cambiar la ubicación de la instalación principal de jBPM al cambiar la configuración de preferencias. Veremos más sobre esto más adelante en el manual.

Con ese conocimiento adicional sobre la configuración del proyecto, es posible crear su primera prueba. Para hacerlo creamos el paquete 'com.jbay' en la carpeta de origen 'test/java'. Luego se muestra el menú de contexto en este paquete y seleccionamos *'New->JUnit Test Case'* (*Figura3.1, "Create a Test "* y *Figura 3.2, "Create Test Dialog"*). Nosotros denominamos esta clase de prueba como *'HelloTest'*.

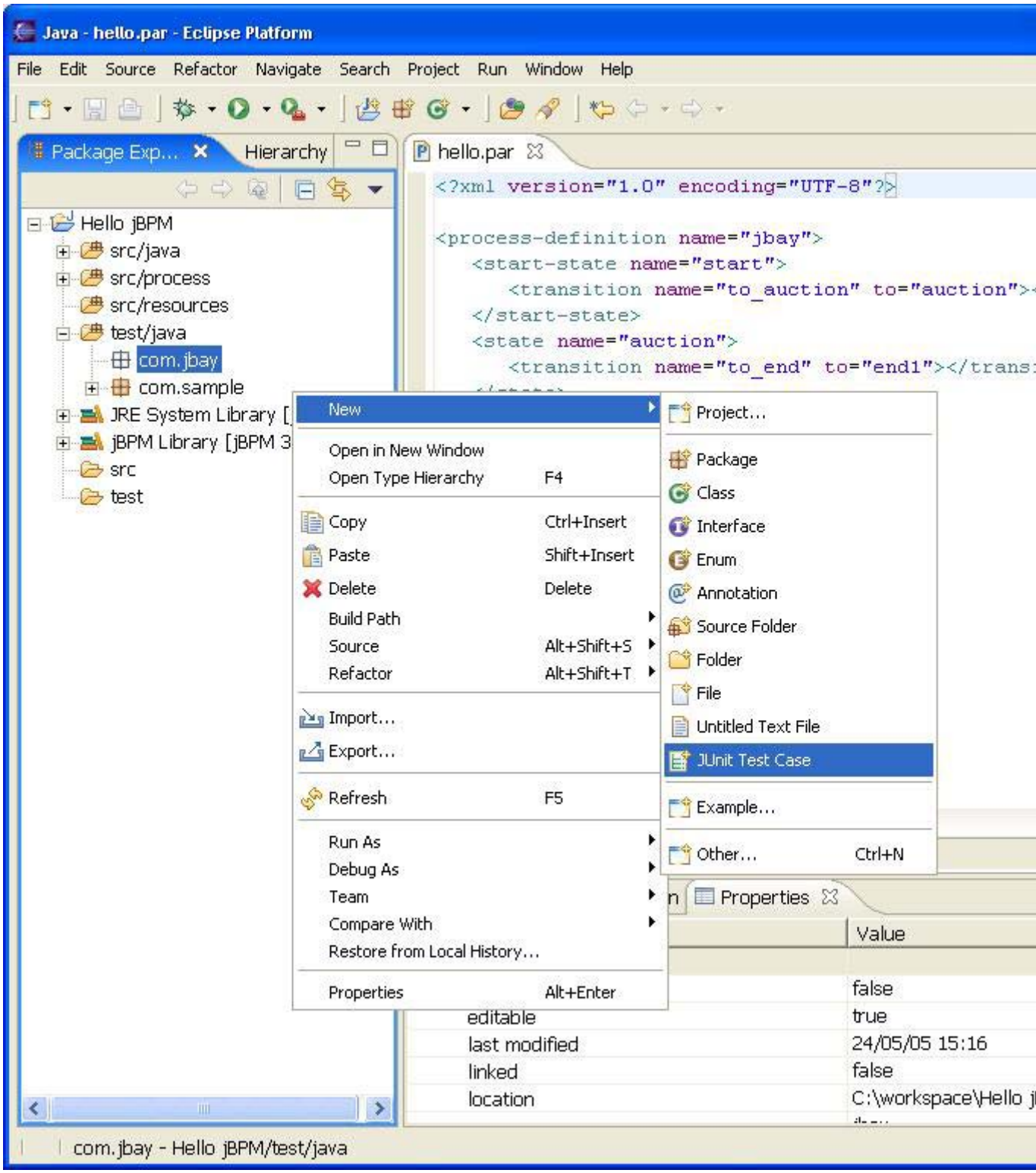

Figura 3.1. Crear una Prueba

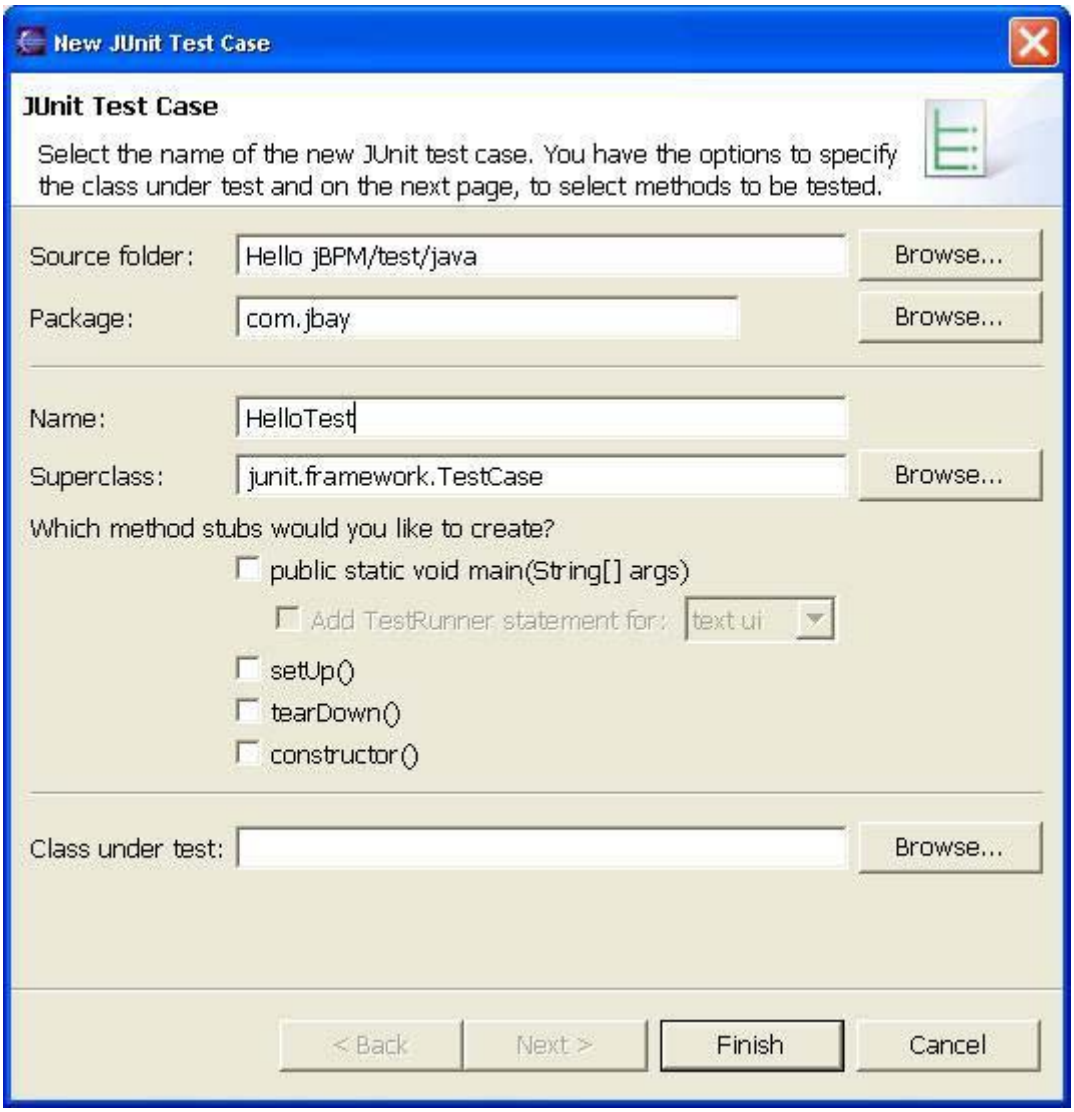

Figura 3.2. Crear Diálogo de Prueba

Escribimos un simple escenario de prueba según se muestra en la *Figura 3.3, "A First Test Scenario "*. Estudiemos el código de este caso de prueba. En la primera línea del método se crea un objeto de archivo de proceso jBPM. Nosotros utilizamos un constructor que acepta el nombre de archivo del achivo de almacenamiento. En nuestro caso estamos hablando del archivo 'hello.par' que creamos anteriormente y que reside en la carpeta *'src'* de nuestro proyecto. Después de confirmar la creación de este objeto cramos un objeto de definición de proceso a partir de este. Este objeto se alimenta al constructor de un objeto de instancia de proceso. Tenemos un objeto de instancia de proceso, pero este proceso todavía no se inicia, de modo que podemos confirmar con seguridad que su token de raíz aun reside en el nodo de inicio. Después de señalizar el token se mueve al estado siguiente y el proceso quedará en estado 'auction'. Finalmente otra señal se encarga de finalizar el proceso.

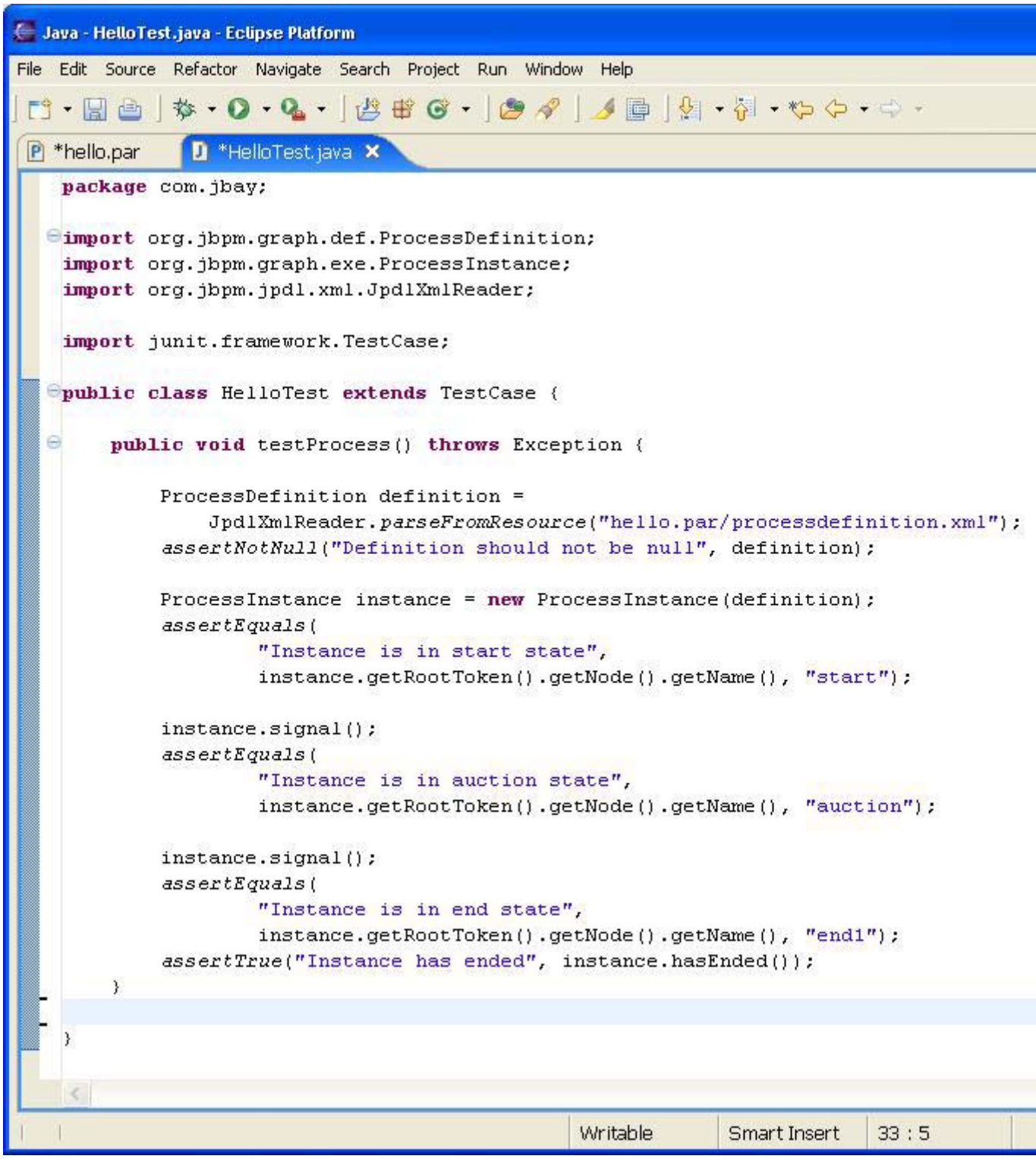

Figura 3.3. Primer Escenario de Prueba

Después de escribir esta prueba podermos verificar si funciona como se espera ejecutándolo (*Figura 3.4, "Running the Process Test "* y *Figura 3.5, "Successful Test Run "*) si todo salió bien, tenemos luz verde.

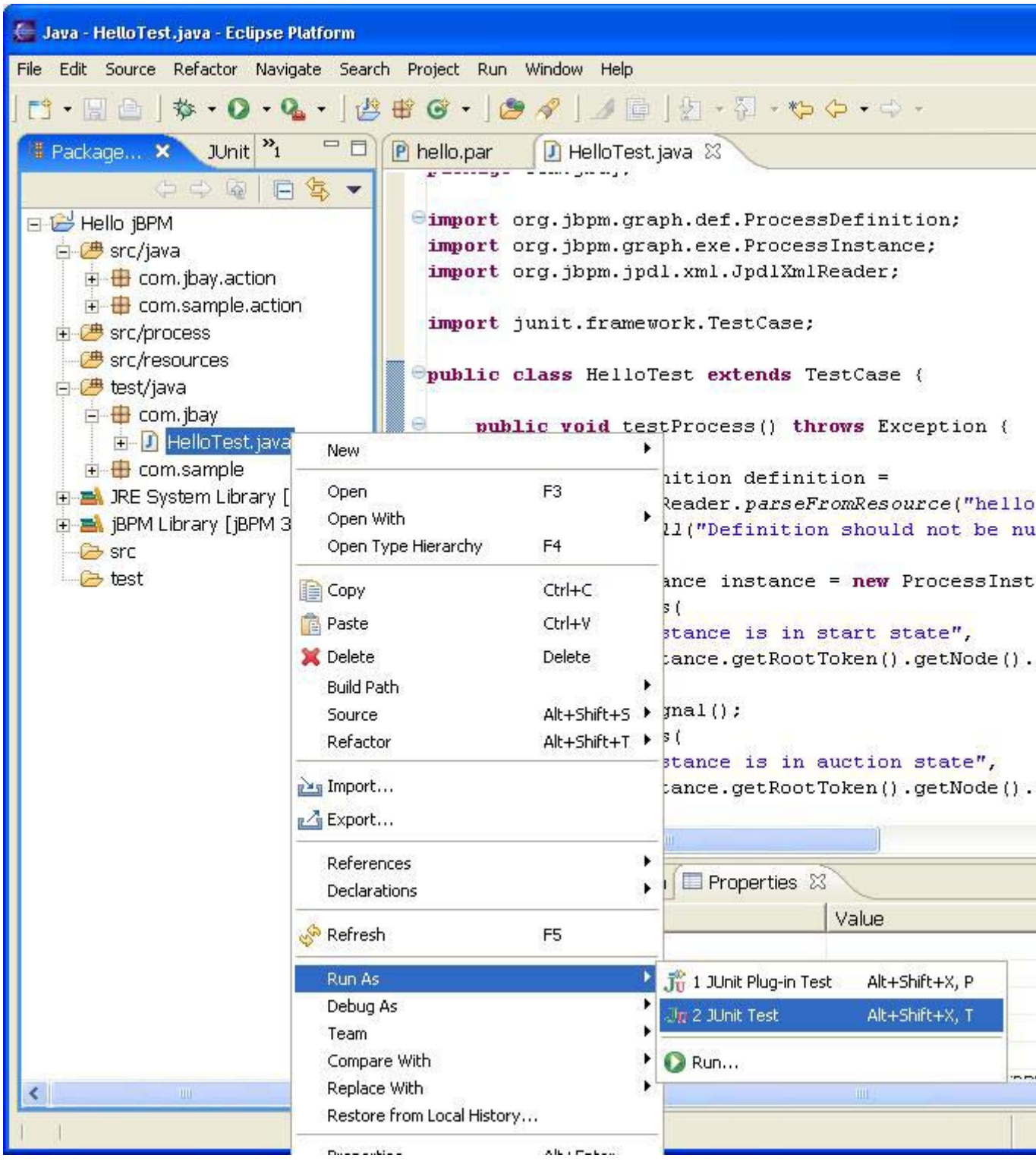

Figura 3.4. Cómo hacer la Prueba del Proceso

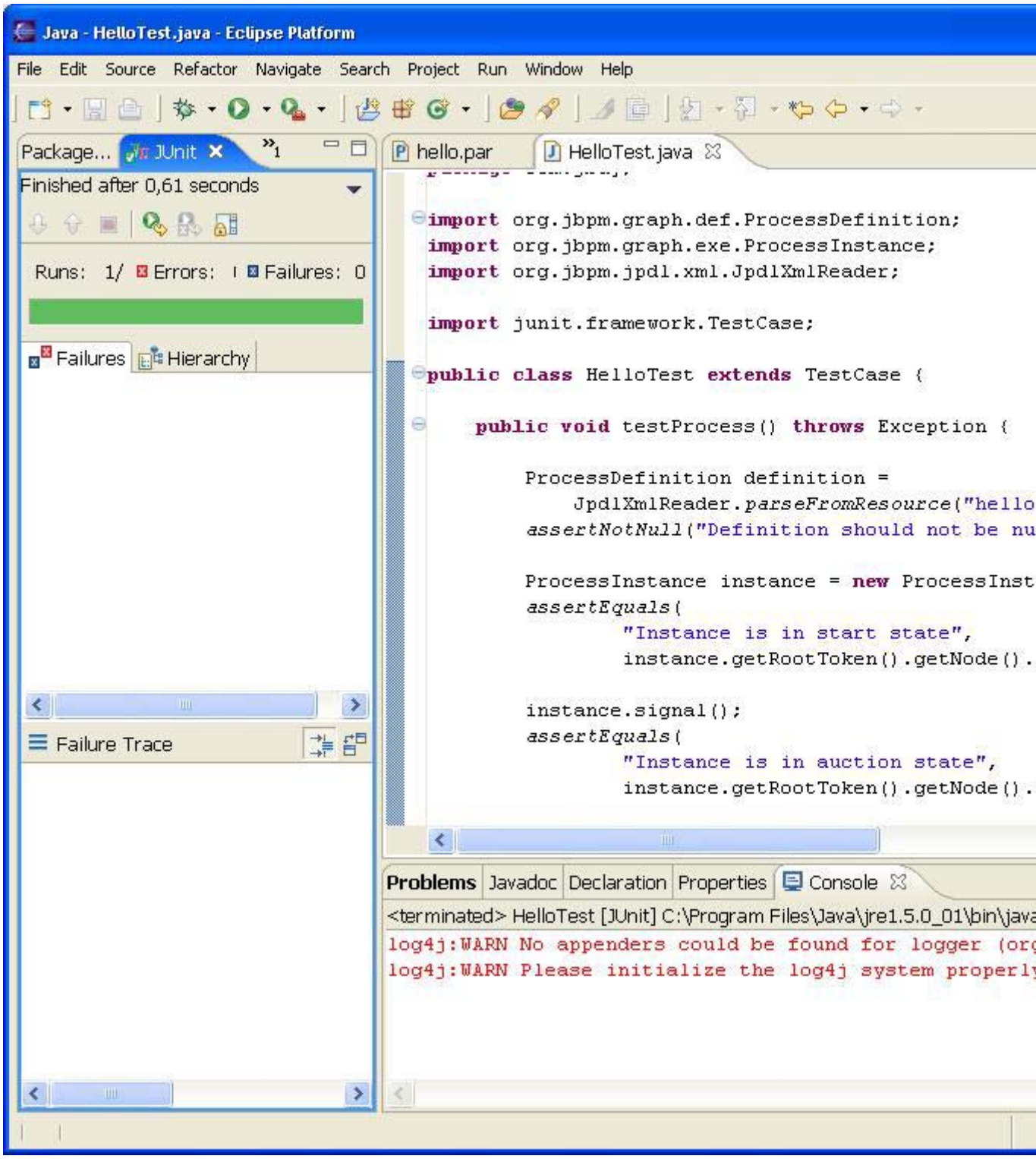

Figura 3.5. Ejecución Exitosa de la Prueba

Por supuesto este escenario de muestra no fue muy interesante, pero el propósito fue mostrar cómo se pueden volver a utilizar sus habilidades de desarrollo en una forma muy sencilla al hacer desarrollo de proceso. Para ver cómo se pueden implementar procesos y escenarios de pruebas de proceso más interesantes sugerimos leer la Guía de

Usuario de JBoss jBPM y estudiar la referencia API. Más adelante en este manual aparecen más ejemplos.

Capítulo 4. Acciones : Mecanismo de Integración JBoss jBPM

En este capítulo mostraremos cómo hacer integración de software con JBoss jBPM. El mecanismo estándar para hacer esto es incluir la funcionalidad que se desea integrar en una clase que implemente la interfaz ActionHandler.

4.1. Cómo crear una Acción Hello World

Cada proceso Hello World debe integrar una o mas acciones Hello World, de modo que esto es lo que vamos a hacer. Podemos integrar código en diferentes puntos en la definición de proceso. Para hacerlo debemos especificar un manipulador de acciones, representado por una implemetacion de la interfaz ActionHandler y adjuntar este segmento de código a un evento particular. Estos eventos son, entre otros, pasar por una transición, salir o entrar en nodos, después y antes de la señalización.

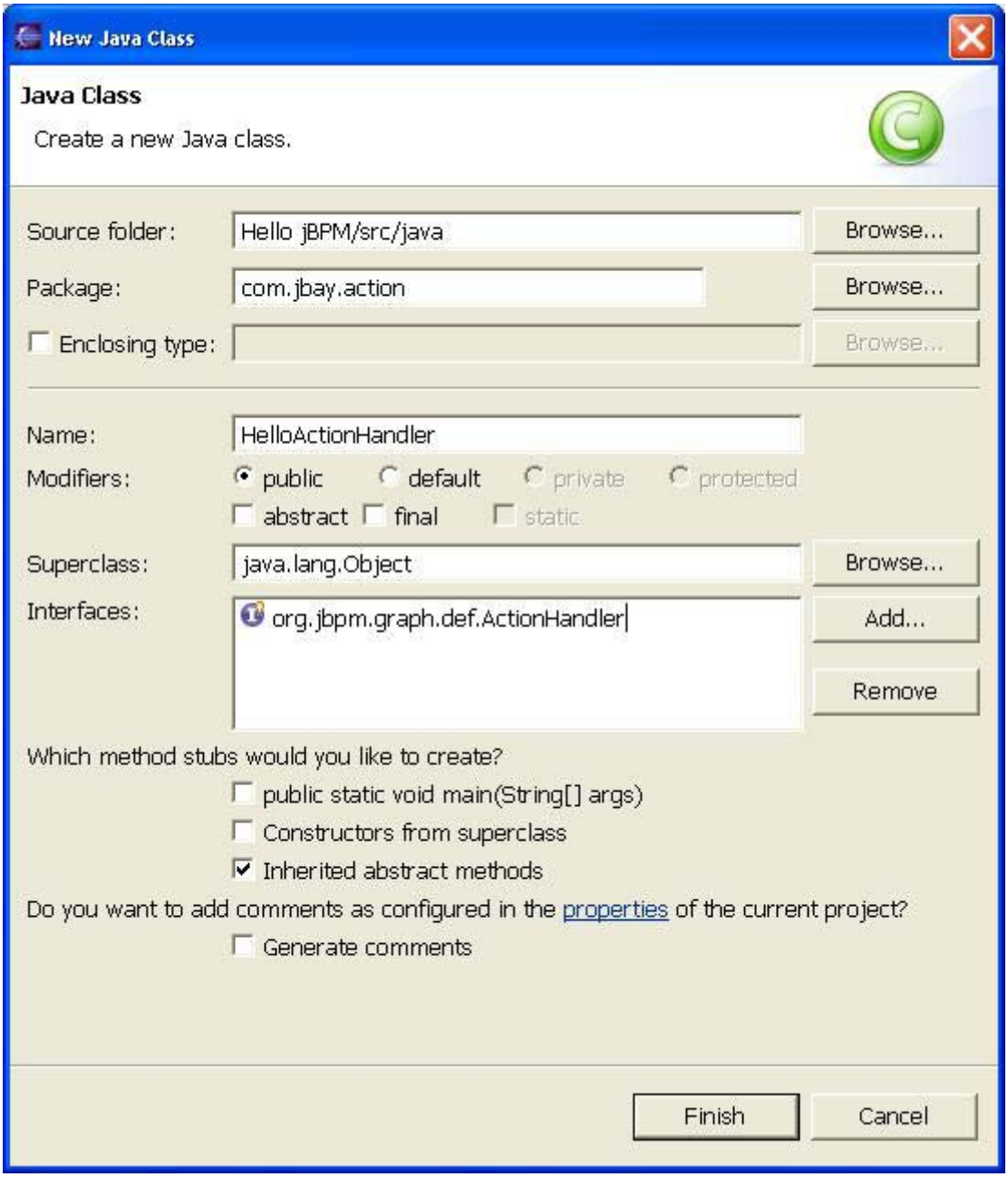

Figura 4.1. Una Acción Hello Simple

Para que este ejemplo sea un poco más concreto implementaremos un manipulador de acciones. Para hacerlo se debe crear una nueva clase llamada HelloActionHandler, la cual implementa la interfaz ActionHandler e implemente el método de ejecución en la *Figura 4.1, "A Simple Hello Action "* y *Figure 4.2, "A Simple Hello Action "*. Esta prueba agregará una variable denominada *'greeting'* a la colección de variables de proceso y le pone un mensaje: *"Hello from ActionHandler"*.

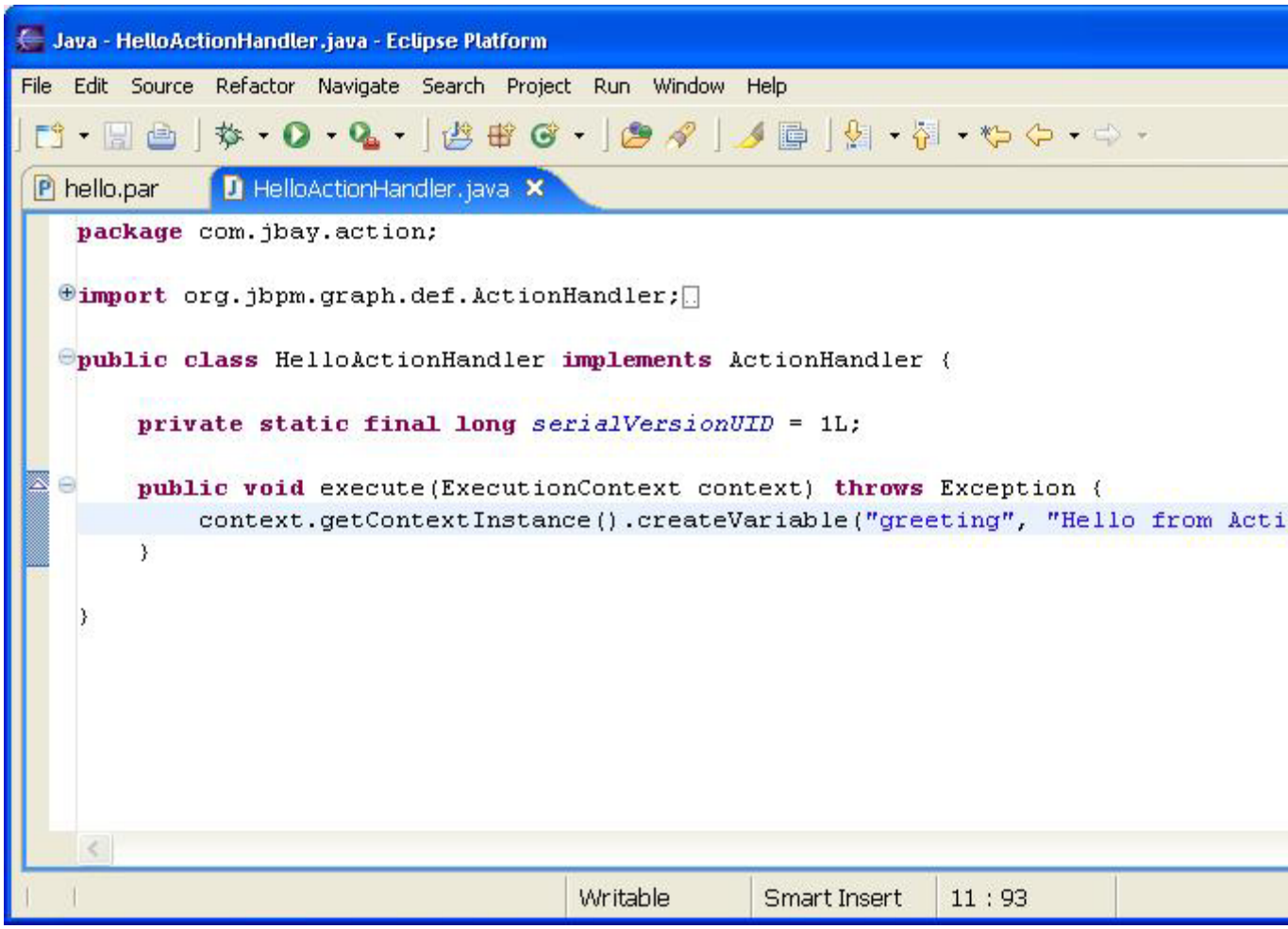

Figura 4.2. Una Acción Hello Simple

4.2. Cómo Integrar la Acción Hello World

Como buenos ciudadanos Testcity primero crearemos una prueba de unidad que compruebe el comportamiento que queremos lograr al agregar el manipulador de acciones al proceso. De modo que implementaremos otra prueba. La creación de la instancia de proceso corresponde a código que ya hemos visto en el capítulo anterior. Nos aseguramos de que no exista ninguna variable llamada greeting. Luego le entregamos una señal al sistema para que pase al primer estado. Queremos asociar la ejecución de la acción con el evento Pasar por una Transición desde el estado de inicio al primer estado. De modo que después de la señal el proceso debería estar en el primer estado, como en el escenario anterior. Sin embargo, la variable 'greeting' debe existir y contener la cadena *"Hello from ActionHandler"*. Eso es lo que garantizamos en las últimas líneas del método de pruea que se muestra en la *Figura 4.3, "Create the Hello Action Test "*

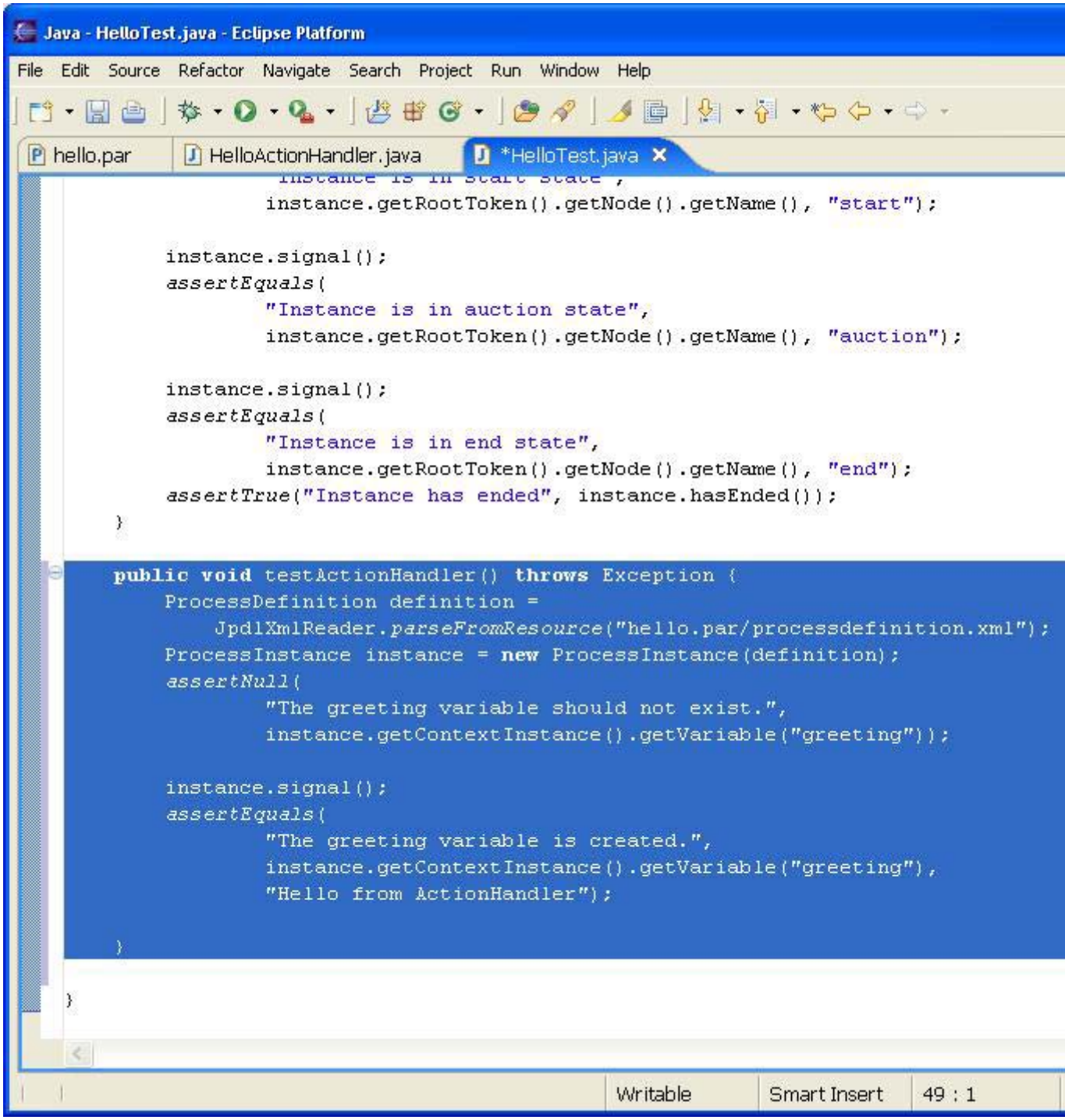

Figura 4.3. Creación de la Prueba Hello Action

La ejecución de las pruebas ahora provoca una falla. Esto se muestra en la *Figura 4.4, "Test Results Before Integration"* De hecho, no asociamos la acción con ningún evento en particular en la definición de proceso, de modo que no se estableció la variable de proceso.

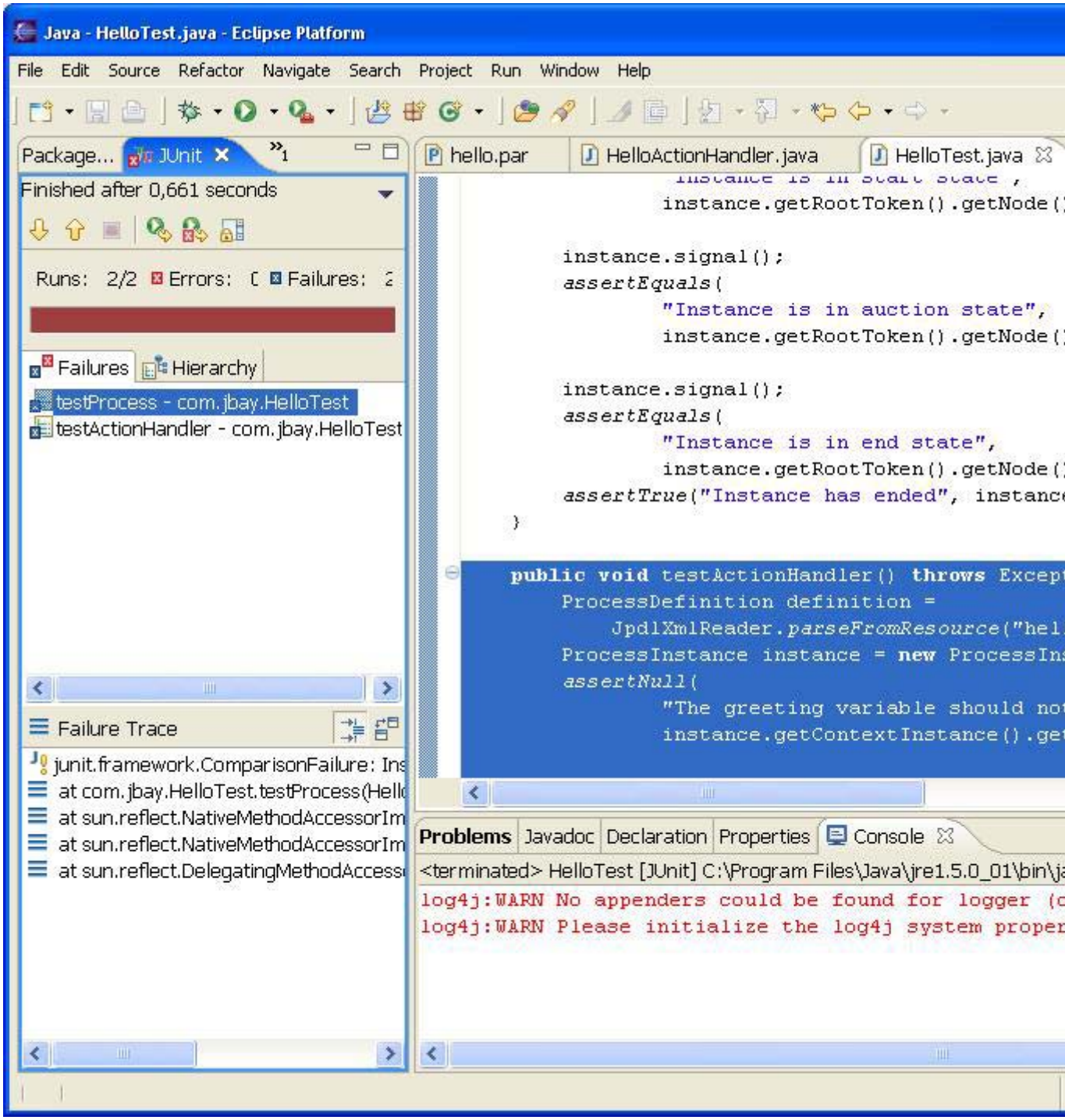

Figura 4.4. Resultados de Pruebas Antes de la Integración

Hagamos algo al respecto y agreguemos una acción a la primera transición de nuestro proceso de muestra. Esto se hace al mostrar el menú de contexto de la transición en la página de árbol de la vista de diagrama, según se muestra en la *Figura 4.5, "Adding an Action to a Transition "*.

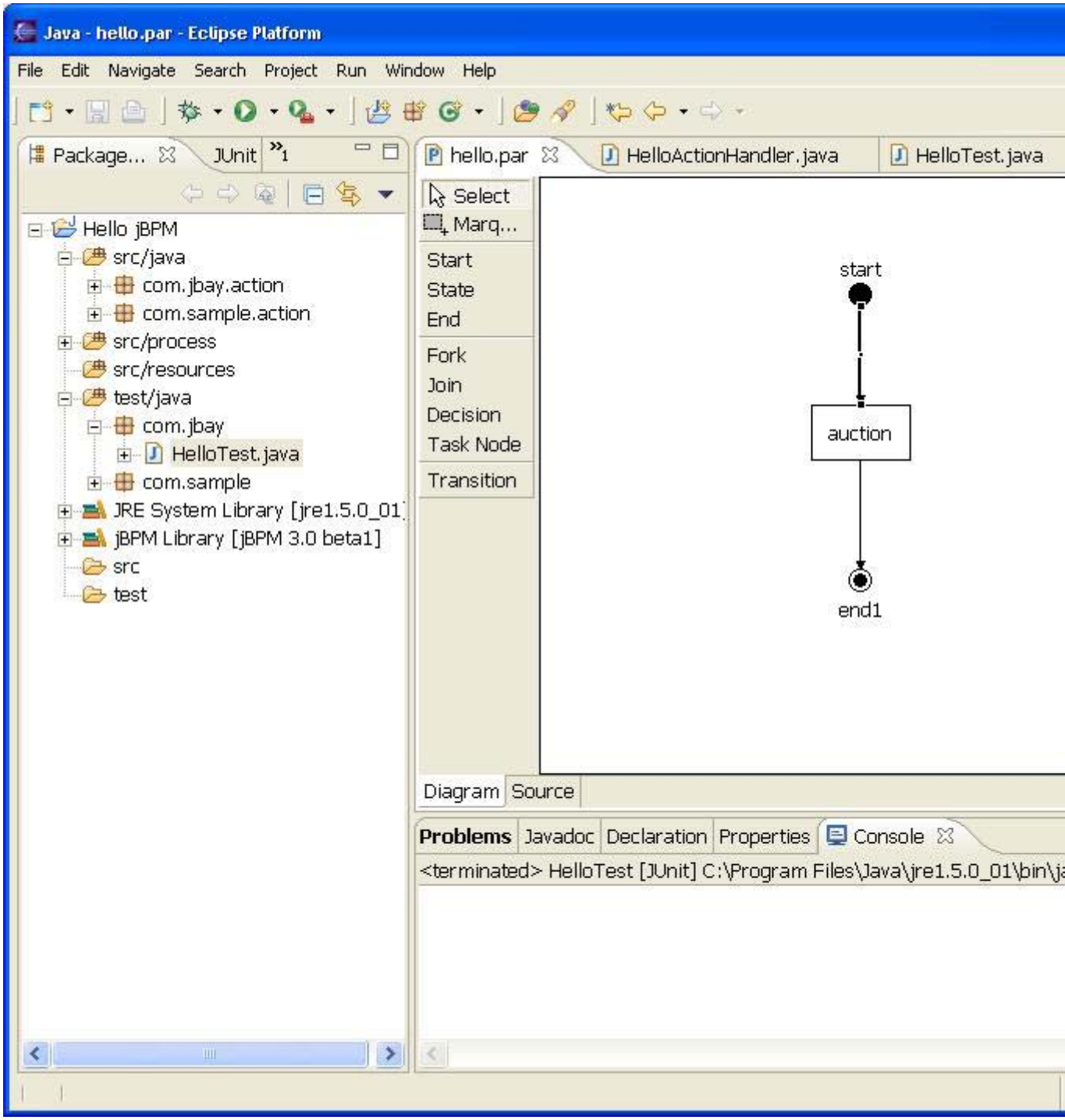

Figura 4.5. Cómo agregar una Acción a una Transición

Al hacer clic con el botón derecho se obtiene un menú emergente con un campo para ingresar las propiedades de la acciión seleccionada. Al hacer clic sobre este menú se obtiene un diálogo de configuración. Por ahora sólo vamos a configurar las propiedades de nombre y clase de esta acción. El resto se analizará más adelante. Ingresamos *'hello'* como nombre de la acción y hacermos clic en el botón *'Browse...'* para abrir un diálogo de selección de clase, donde podemos buscar las clases que implementan la interfaz

ActionHandler en el classpath de nuestro proyecto (*Figura 4.6, "The Choose Action Handler Dialog "*).

Figure 4.6. Diálogo de Selección de Manipulador de Acción

Figura 4.7. Diálogo de Configuración de Acción

Seleccionamos nuestra clase *'HelloActionHandler'* previamente creada y presionamos el botón *'OK'* (*Figura 4.7, "The Action Configuration Dialog "*). Después de seleccionar el manipulador de acción para la acción, podemos llevar a cabo la pueba y verificar que nos de una luz verde (*Figura 4.8, "The Action Configuration Dialog "*).

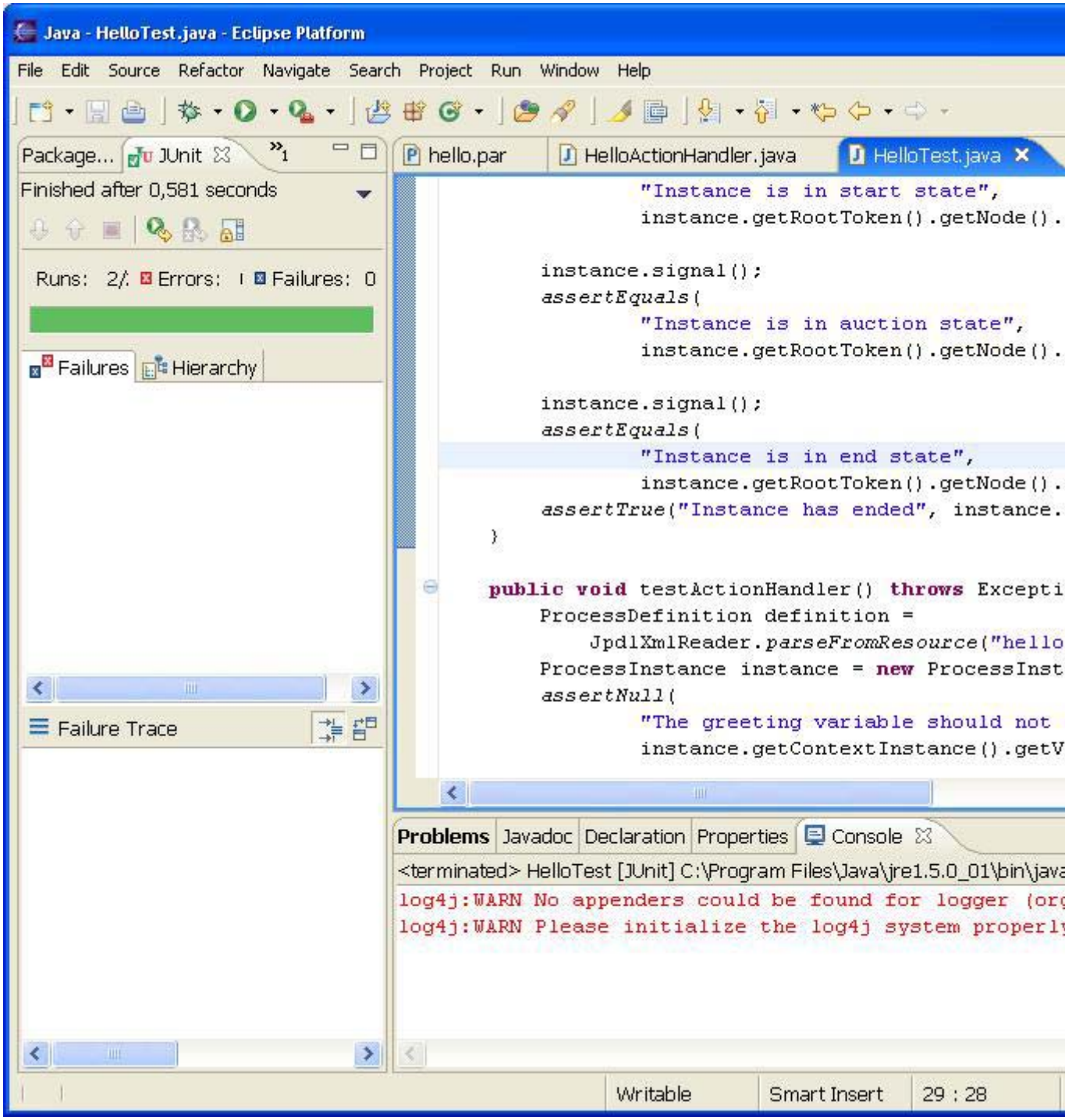

Figura 4.8. Diálogo de Configuración de Acción

4.3. Puntos de Integración

Los diferentes puntos de integración en una definición de proceso están completamente documentados en la Guía de Usuario de JBoss jBPM. Como se puede ver en la *Figura 4.9, "Adding an Event Action "* es posible agregar diferentes tipos de acciones n el casos de nodos de instancia. Agregar una acción de este tipo creará un objeto de

evento en la vista de diseño y agregará una acción como secundario de este evento recientemente creado. De la misma forma, las acciones se pueden agrega al objeto de definición de proceso. Más aun, las acciones se pueden agregar directamente a los eventos.

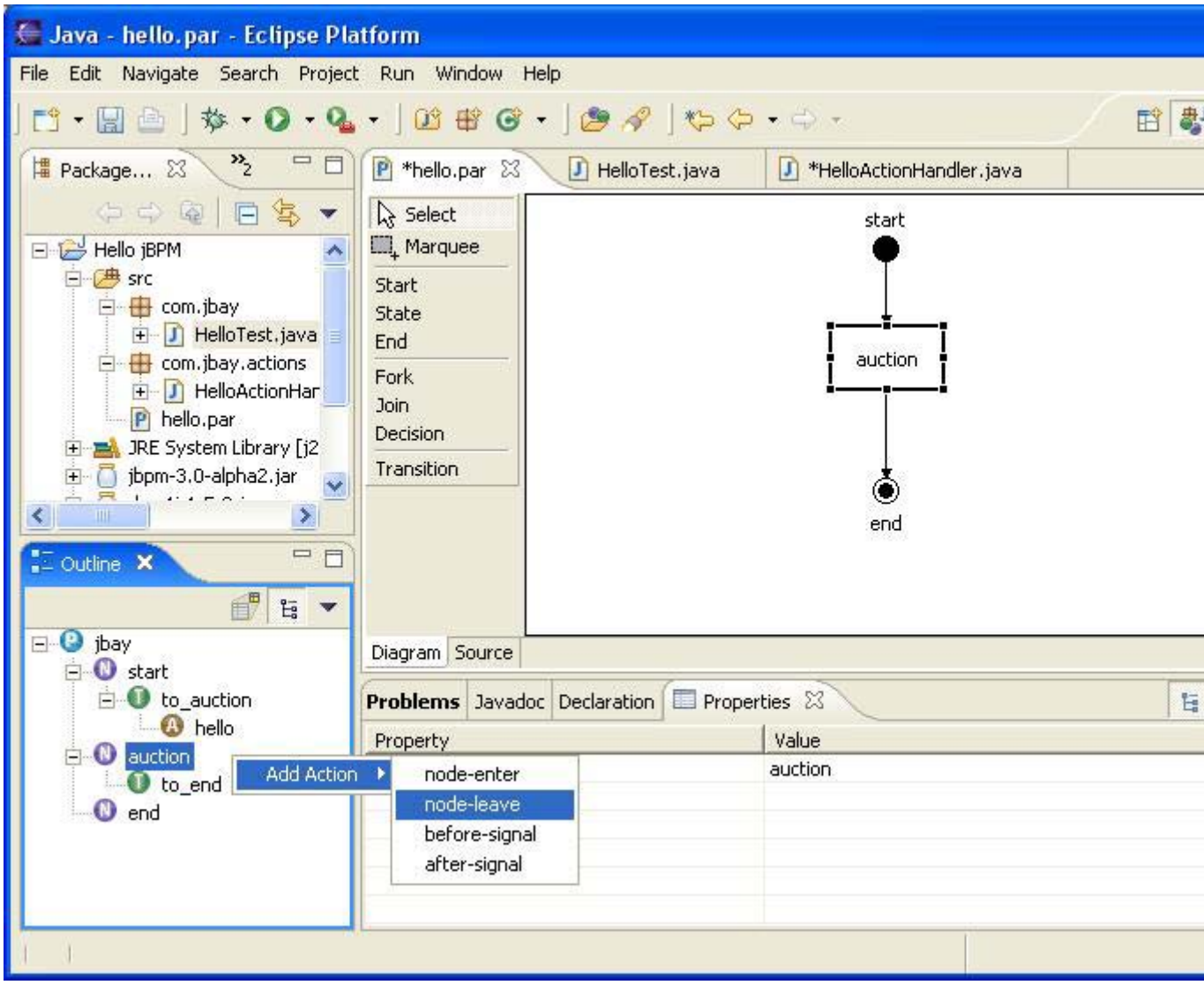

Figura 4.9. Cómo Agregar una Acción de Evento

Capítulo 5. Guía Rápida Howto

5.1. Cambiar la Instalación Predeterminada de Core jBPM

Es posible cambiar la instalación predeterminada de jBPM mediante el mecanismo de preferencia Eclipse. Abra el diálogo Preferencias seleccionando *'Window->Preferences'* y seleccione la categoría *'JBoss jBPM'* (???). Gracias a esta página se pueden agregar múltiples ubicaciones de instalación jBPM y cambiar la predeterminada. La instalaicón

predeterminada se utiliza para la configuración de classpath al crear un nuevo Proyecto de Proceso. Cambiar las preferencias no tiene ningun efecto sobre los proyecto anteriormente creados. Sin embargo, eliminar una instalación de jBPM que sirve de referencia para un proyecto sí provoca que el classpath contenga errores.

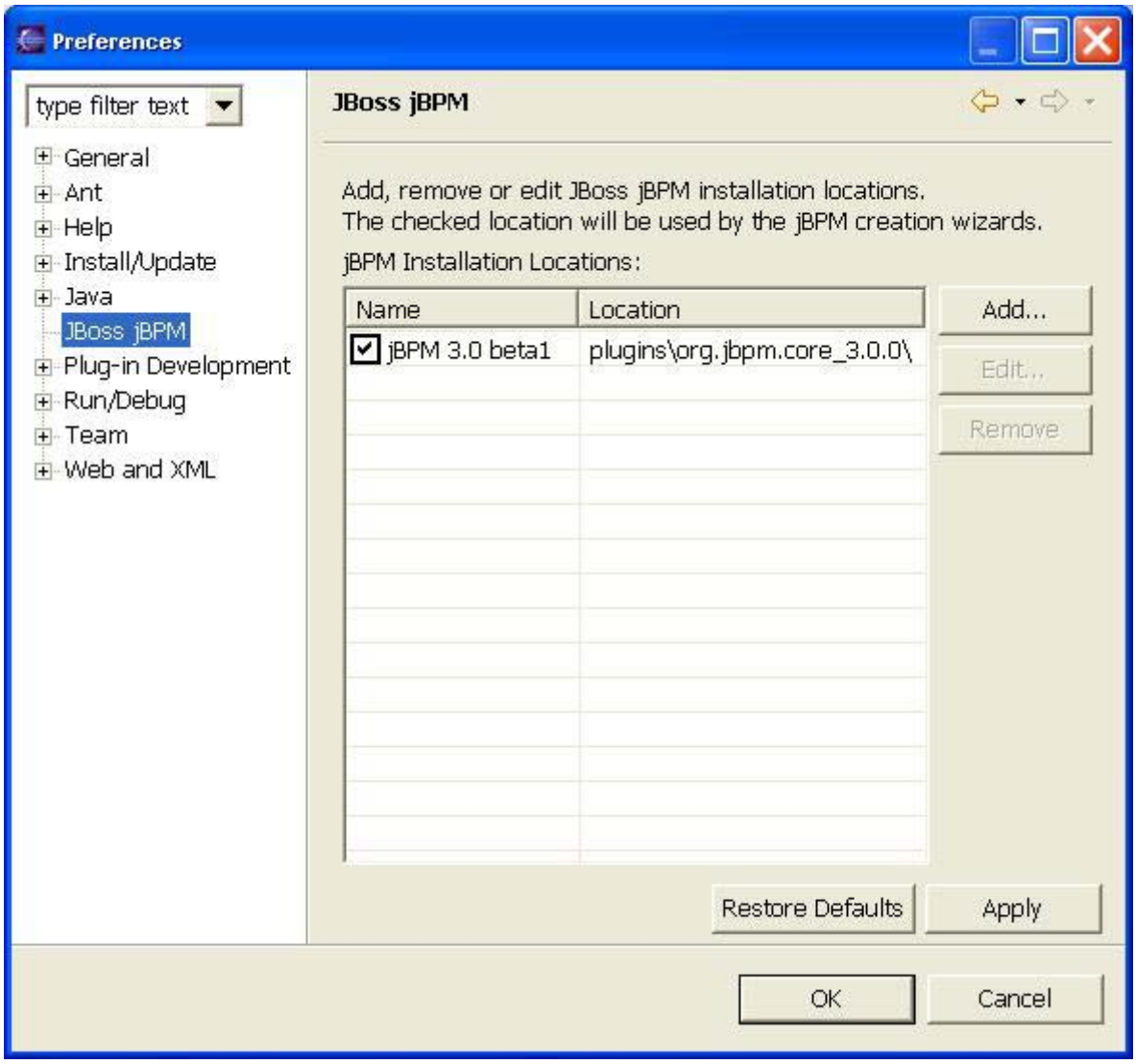

Figura 5.1. Página de Preferencias jBPM

5.2. Configurar Nodos de Tarea

Es posible agregar tareas a nodos de tarea y luego configurar éstos últimos en una forma parecida al mecanismo de configuración de Acción. El menú de contexto de las tareas contiene una entrada denominada *'Properties'* que abre un diálogo de configuración (???).

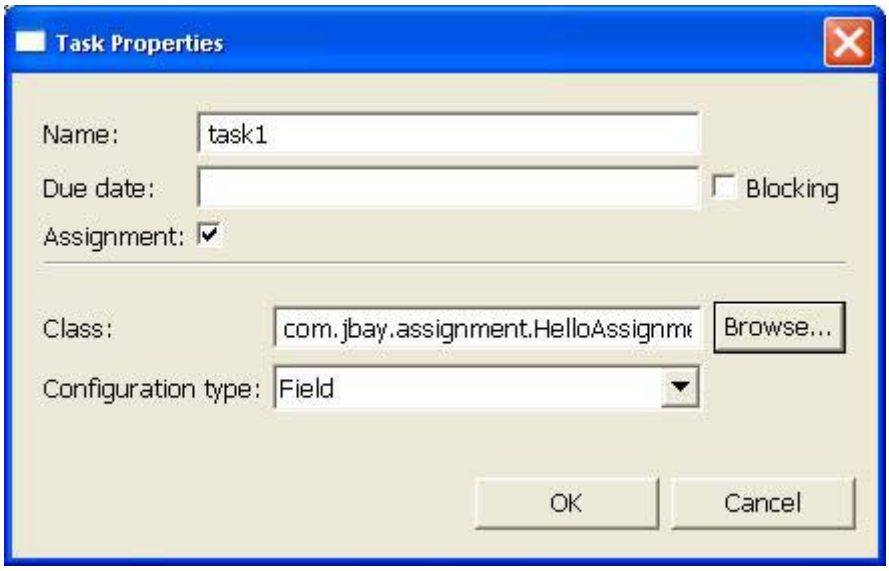

Figura 5.2. Diálogo de Configuración de Tarea