

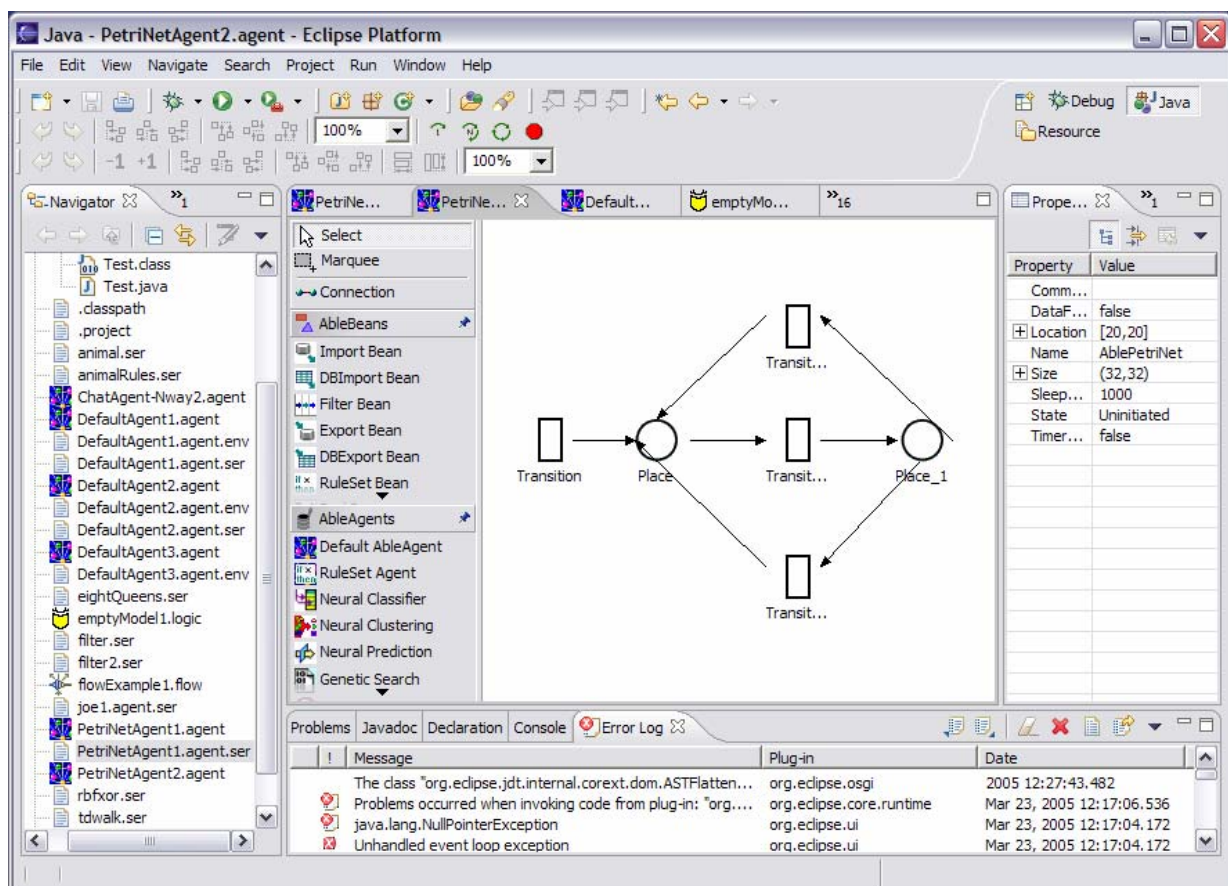
ABLE Petri Net Agent

The ABLE PetriNet agent allows users to define a high-level PetriNet with an arbitrary number of Place/Transition nodes and associated Arcs. You can use the ABLE Eclipse agent editor to graphically construct PetriNets. The PetriNet agent customizer (GUI) allows specification of variables and one-time initialization logic. Places (states) can have optional initialization expressions to set initial marking of the Petri Net.

Transitions can specify arbitrary sequence of actions (which reference PetriNet variables), an optional guard clause (Boolean expression), an optional firing probability, and an optional delay between input token removal and transition actions. PetriNet arcs, which are inputs to Transitions, can be normal or inhibitory. Tokens can be arbitrary Java Objects.

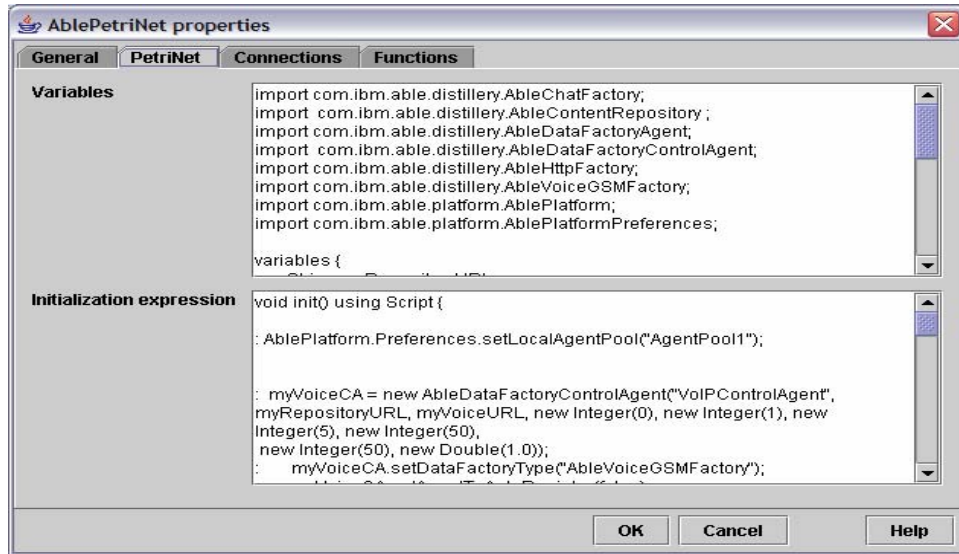
Overview

The screen shot below shows a simple PetriNet agent model in the Eclipse agent editor. *Places* are depicted by circles, while *Transitions* are depicted by vertical rectangles. *Arcs* are represented by lines with arrows showing *Token* flow.



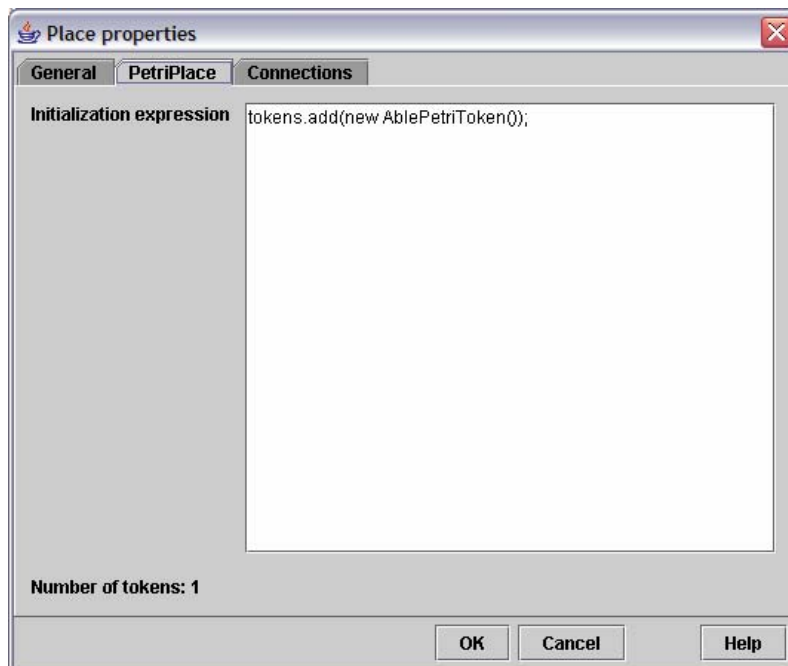
Variables

Variables can be defined for the PetriNet using the PetriNet agent Customizer. Any Java classes can be imported and variables declared which can then be referenced in Places/Transition node expressions. The Initialization Expression text area allows an `init()` ruleblock to be defined for one-time initialization of PetriNet variables.



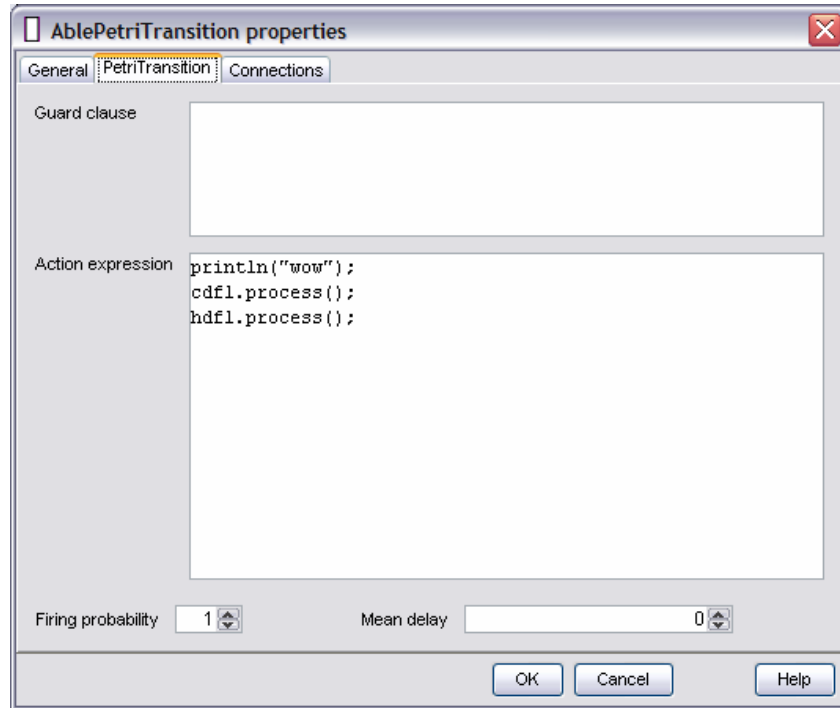
Places

Places allow initialization expression to be defined which can set the initial marking (contents) of the Place node. A single if/then rule is created for each Place node and if defined, the rule actions are executed when the PetriNet is `init()`ed.



Transitions

Transitions allow the definition of Boolean guard clause along with a set of actions which are performed when the Transition fires. The guard clause logic is over and above the standard PetriNet logic which requires that the correct number of Tokens exist at each input Place node. If not defined, the guard clause is **true**.



Arcs

Petri Net arcs are directed links from a **Place** node to a **Transition** node or from a **Transition** node to a Place node. Arcs can have integer valued multiplicity (shown when the arc multiplicity is > 1). Arcs can be normal (depicted by an arrow) or inhibitory (depicted by a circle).

Tokens

Petri Net tokens can be any Java Object. You must declare two global variables: **tokens** and **token**. The **tokens** variable is defined as a Vector as such `java.util.Vector tokens ;` while the **token** variable can be any Java object, typically a **Boolean** for simple Petri Nets or an application class such as **Person** or **Customer** in higher-level Petri Nets. For example:

```
Vector tokens;  
Boolean token; // or Person token;
```

You can create/add initial tokens in Place initialization expressions. Both **tokens** and **token** variables can be referenced in place/transition logic.