YAWL Editor 1.1
User Manual

Release 1.0

QUT Faculty of Information Technology

Document Control

Date	Author	Version	Change
2 Sep 2004	Sean Kneipp	0.1	Initial Draft
27 Sep 2004	Sean Kneipp	0.2	Data Perspective
17 Oct 2004	Sean Kneipp	0.3	Updated Editor Version
26 Oct 2004	Sean Kneipp	0.4	Updated Engine Integration
25 Nov 2004	Lindsay Bradford	0.5	Sync with Editor 1.1
04 Feb 2005	Lindsay Bradford	1	Finalised Release

Contents

Document Control ii

Welcome to YAWL 1

- What is YAWL? 1
- Obtaining the Latest Version of the YAWL Editor 1
- The YAWL Project 2
- YAWL Architecture 2

Getting Started 3

- Copyright Notice 3
- Launching the YAWL Editor 3

The YAWL Editor Workspace 4

- Menu Toolbar 4
- Palette Bar 7
- Status Bar 8
- Canvas 8
- Menus Overview 8

Creating Your First Specification 10

- Overview 10
- The Scenario 10
- Creating Your First Specification 10
- Atomic Tasks 11
- Decorating a Task 12
- Composite Tasks 14
- Multiple Atomic Tasks 15
- Multiple Composite Tasks 17
- Conditions 20
- Changing the Starting Net 21

Changing the Appearance of Your Specification 22

- Improving the Look of Flow Relations 22
- Editing Objects 23
- Changing Font Size 23

Advanced Specification Features 24

- Cancellation Sets 24
- Updating Data Type Definitions 26
- Net Decomposition Detail / Updating Net Variables 27
- Task Decomposition Detail / Updating Task Variables 30
- Adding / Updating Task Parameters 32
- Updating Flow Detail 37
- End of Scenario 38
- Set YAWL Engine Detail 38
- Setting Multiple Instance Queries 39
- Validating and Exporting 41

Illustrative Examples 42

- Multiple Instance Task Example 42
- Time Service Example 44

Known Issues 45

Troubleshooting 45

Acknowledgements 47

Welcome to YAWL

What is YAWL?

Based on a rigorous analysis of existing workflow management systems and workflow languages, we have developed a new workflow language: YAWL (Yet Another Workflow Language). To identify the differences between the various languages, we have collected a fairly complete set of workflow patterns. Based on these patterns we have evaluated several workflow products and detected considerable differences in expressive power. Languages based on Petri nets perform better when it comes to state-based workflow patterns. However, some patterns (e.g. involving multiple instances, complex synchronizations or non-local withdrawals) are not easy to map onto (high-level) Petri nets. This inspired us to develop a new language by taking Petri nets as a starting point and adding mechanisms to allow for a more direct and intuitive support of the workflow patterns identified.

YAWL can be considered a very powerful workflow language, built upon experiences with languages supported by contemporary workflow management systems. While not a commercial language itself it encompasses these languages, and, in addition, has a formal semantics. Such an approach is in contrast with e.g. WfMC's XPDL which takes commonalities between various languages as a starting point and does not have formal semantics. Its design hopefully allows YAWL to be used for the purposes of the study of expressiveness and interoperability issues.

At this stage YAWL supports the control-flow perspective and the data perspective. However, it is important that relations with the resource perspective are investigated and formalized. For YAWL to be more applicable in the area of web services and Enterprise Application Integration, it is also desirable that support for communication patterns is built-in.

Obtaining the Latest Version of the YAWL Editor

As new versions of the YAWL Editor are released to the public, they will be available for download at the YAWL website:

http://www.citi.qut.edu.au/yawl.

Developers interested in obtaining the YAWL Engine, can download the files from the YAWL website above. The source code for all of the YAWL components can be downloaded through the SourceForge Website:

http://sourceforge.net/projects/yawl .

The YAWL Project

For more information and progress on the YAWL project, visit the YAWL Homepage on Queensland University of Technology's Centre for Information Technology Innovation website:

http://www.citi.qut.edu.au/yawl.

YAWL Architecture

The following image, **Fig. 1.**, depicts the interaction between components of the YAWL Engine.

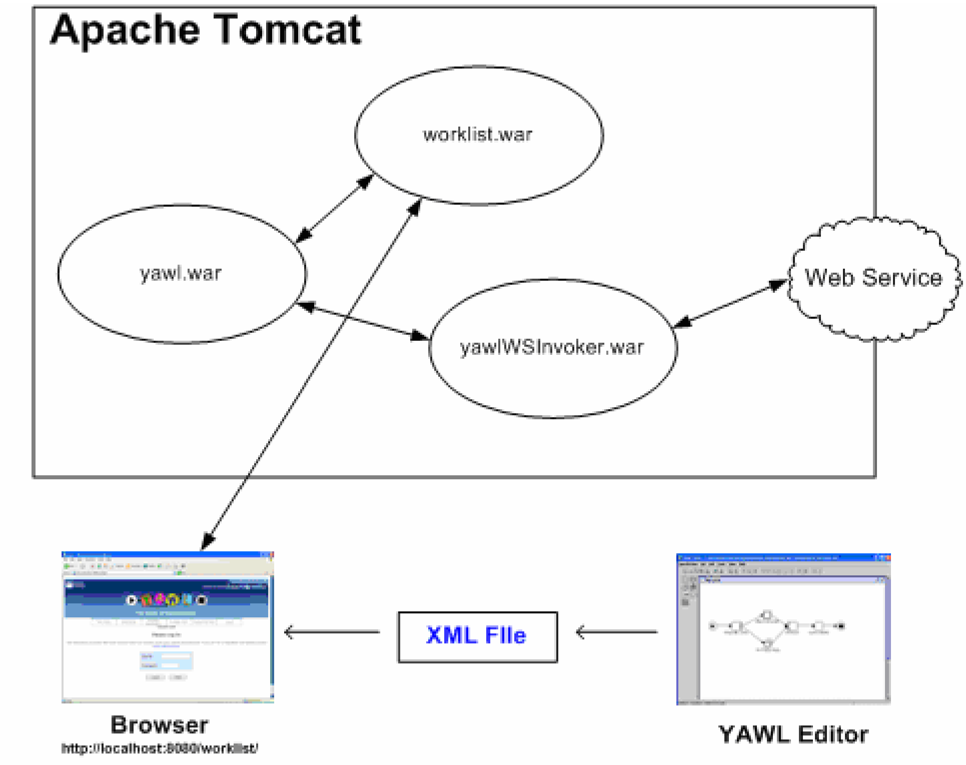


Fig. 1. The YAWL Components

Getting Started

Copyright Notice

Copyright to this software and its source code is granted under the GNU Lesser General Public Licence (v2.1).

For detail on the permissions this licence grants you, please refer to http://www.gnu.org/copyleft/lesser.html.

This editor makes use of JGraph 3.1. This version of the JGraph library is covered under a BSD-style licence and is available separately from :

http://www.jgraph.com/.

This editor also makes use of code from the YAWL Engine and its support libraries. The engine and its support libraries are covered under their own individual copyright licences, detail of which may be found at the YAWL website:

http://www.citi.qut.edu.au/yawl.

In accordance with this editor's copyright licence, the source code may be obtained from the YAWL SourceForge website:

http://sourceforge.net/projects/yawl/.

Launching the YAWL Editor

1. Download the latest version from the YAWL SourceForge website:

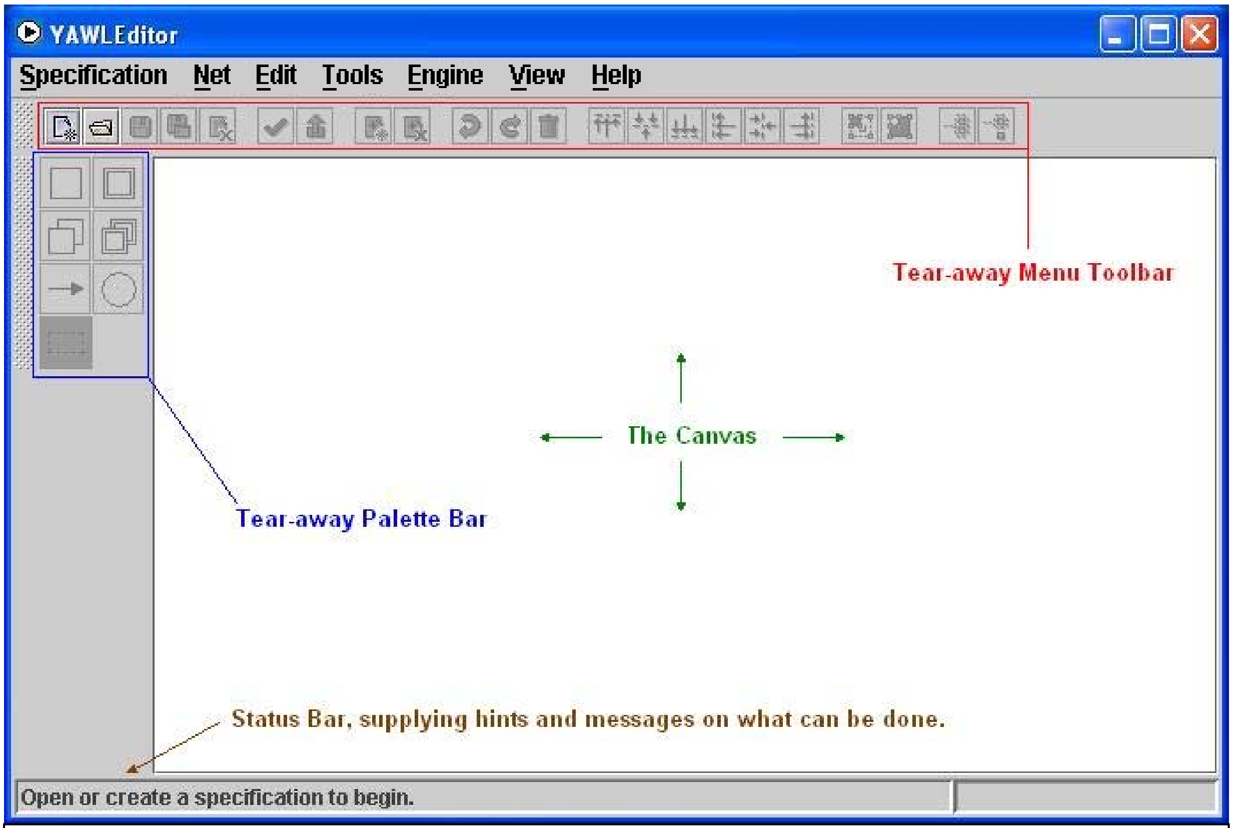
   http://sourceforge.net/projects/yawl/.

2. Double click on the YAWLEditor.jar file to start the application and away you go.

The YAWL Editor Workspace

The first time you start the YAWL Editor, you will be presented with a blank canvas, with the instructions in the Status Bar asking you to open or create specification to begin.

Before you create your first specification, let us take a brief tour of the Editor's workspace and the elements within.



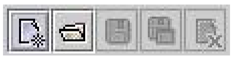
Open or create a specification to begin.

Fig. 2. The YAWLEditor Workspace

Menu Toolbar

The Menu Toolbar contains six groups of buttons to assist you in maintaining your YAWL specification. The menu can be repositioned by dragging the left-hand anchor bar.

Specification Maintenance

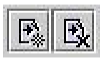


This group of buttons provides you the standard options to create, save, open and close YAWL specification files.

Specification Verification & Exporting

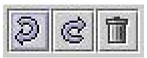
These two buttons allow you to validate your specification for any workflow inconsistencies and then export your workflow diagram into an XML file for uploading to the YAWL Engine.

Net Maintenance



Your workflow diagrams are captured within Nets, of which a specification could contain many. You can use these buttons to create a new Net or remove existing Nets within your specification.

Edit Options



These group of buttons provides the standard Undo and Redo options as well as the option to delete the currently selected objects.

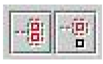
Alignment Options

These buttons can be used to assist with the alignment of objects within your specification, when multiple objects have been selected.

Object Sizes

To increase the size of an object within your specification, select the object(s) and then use these buttons.

Cancellation Sets



These buttons allow you to include in and/or exclude elements from the cancellation set of a task.

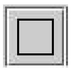
Palette Bar

The Palette Bar contains seven selector buttons that assist with creation, selection and positioning of objects within your specification. This menu can be repositioned by dragging the left-hand anchor bar.

The Palette bar is also accessible by right-clicking anywhere on a net that does not contain a net element.

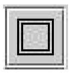
Once an element is selected, it is possible to drop objects in the canvas by left-clicking the mouse button.

Atomic Task



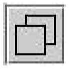
Select this button to create an Atomic Task, which represent a single task to be performed by a human or external application.

Composite Task



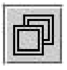
Select this button to create a Composite Task, which is a container for another YAWL Net - with its own set of YAWL elements constrained by the same syntax.

Multiple Atomic Task



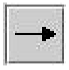
Select this button to create a Multiple Atomic Task, which allows you to run multiple instances of a task concurrently.

Multiple Composite Tasks



Select this button to create a Multiple Composite Task, which allows you to run multiple instances of a composite task concurrently.

Flow Relation



Select this button to create a Flow from one task to another.

Condition

○

Select this button to create a Condition, which is a way to represent state for the Net.

Marquee Selection

⬚

Select this button to activate the Marquee Selector, which will allow you to select individual or multiple objects by clicking and dragging your left mouse button.

Status Bar

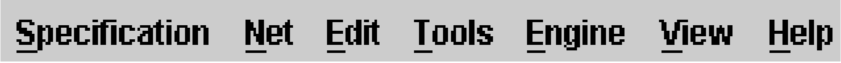
The Status Bar provides useful hints throughout the creation of your specification and depending upon which object you are using in the Palette Bar.

Canvas

The Canvas is where you will be creating and editing your workflow diagram(s).

Menus Overview

This section provides a brief overview of the YAWL Menus located along the top of the YAWL Editor.



Specification

The Specification Menu provides all the standard file options of Create, Open, Save, Close and Exit. It also contains options to validate and export the specification to the YAWL Engine (XML format) as well as updating the specification's data type definition (in XMLSchema format) and for printing the specification.

Net

The Net Menu provides options to Create, Remove, set the Starting Net and specify decomposition detail for the currently selected net. It also provides options to export a net to a PNG image file and for directly printing out the net.

Edit

The Edit Menu provides the standard options of Undo, Redo, Cut, Copy, Paste and Delete objects within your specification.

Tools

The Tools Menu allows you to align objects within your specification, modify their size or add or remove objects from your Cancellation Set.

Engine

The Engine Menu allows you to specify a running engine that the editor should connect to so you can tailor your workflow designs to specific modules and web services installed in the engine.

View

You can use this menu to turn off or on the Tooltips, which provide useful hints when your mouse is positioned over a toolbar button or an option in a pop-up dialog window, obtained by right-clicking on the canvas. It also allows you to change the font size used for element labels.

Help

The Help Menu provides details of the YAWL Editor copyright as well as Acknowledgements for all the dedicated YAWL contributors.

Creating Your First Specification

Overview

This next chapter will lead you through the process of creating a YAWL specification from beginning to end, through a series of brief lessons following a scenario.

You can either follow the all the instructions including the scenario provided, from beginning to end, or skip straight to the section that you are interested in and follow the instructions.

S Look for the **blue S** next to the instructions for specific details of the scenario.

The Scenario

S The scenario that we will be following throughout this manual is the workflow of a student who has just completed their secondary study and is now looking to start their career.

This scenario will follow the path of a student who either enrols in a University to complete their tertiary eduction, or undertakes privite study which will eventually lead them to getting a job and starting their new career.

Creating Your First Specification

1. Click on the Create a New Specification button, , at the top left of the Menu Toolbar, or click on **Specification** in the Menu and choose **Create Specification.**

   This will bring up a blank Net called "New Net 1" which will be, by default, the starting net of the workflow. For details on selecting a starting net, consult the *Changing the Starting Net* section in this manual.

2. Rename this Net by clicking on the **Net** Menu and choosing **Update Net Detail**.

3. Enter the new name of the Net in the "Decomposition Label" field, then click the **Done** button.

   Decomposition Variables will be explained later in the "Net Decomposition Detail" section of this manual.

Change the name of the Net in the Decomposition Label, to “My Career”. This Net will be the primary net for our scenario.

4. You are now ready to start drawing your specification.

Atomic Tasks

1. Click on the “Add an Atomic Task” button, □ , in the Palette Bar, or right click in the whitespace of the Net and choose **Atomic Task**.
2. Position your mouse just to the right of the Input Condition (the ▶ symbol), and click the left mouse button once to place an Atomic Task
3. Set the decomposition of this task by right clicking on the Atomic Task and choosing **Select Task Decomposition**.

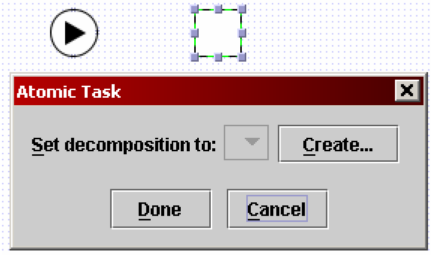


Fig. 3. The "Update Task Decomposition " dialog

4. Press the **Create…** button, and in the following window, “Update Task Decomposition”, enter the decomposition’s label.

   See the “Select Task Decomposition” section of this manual for a full explanation of its features.

Set the label to “Begin My Career”, and click the **Done** button.

5. Select the “Flow Relation” button, → , in the Palette Bar, or right click in the whitespace of the Net and choose **Flow Relation**.
6. Connect the Input Condition to your Atomic Task by finding the flow connectors, indicated by two parallel blue lines on the sides of the objects, and hold the left mouse button down and draw a line from the flow connector on the Input Condition to the one on the Atomic Task. The editor will indicate which connection points are valid by drawing a blue box around suitable connectors as your mouse passes over them.

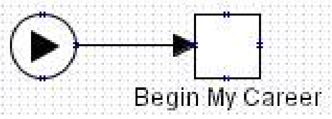


Fig. 4. An established Flow Relation

That’s it! Your Atomic Task is set.

7. Repeat the process for the following Atomic Tasks in order:
    - Go to University
    - Get A Job
    - Begin My Career
8. Link the Begin My Career task to the Output Condition (the ◉ symbol), as per **Fig. 5**.
9. Finally check the validity of specification by clicking on the Validate this Specification button, ☑ , in the Menu Toolbar or click on **Specification** in the Menu and choose **Validate Specification**. If all things are going to plan, then you should receive a confirmation saying that there were no errors detected.

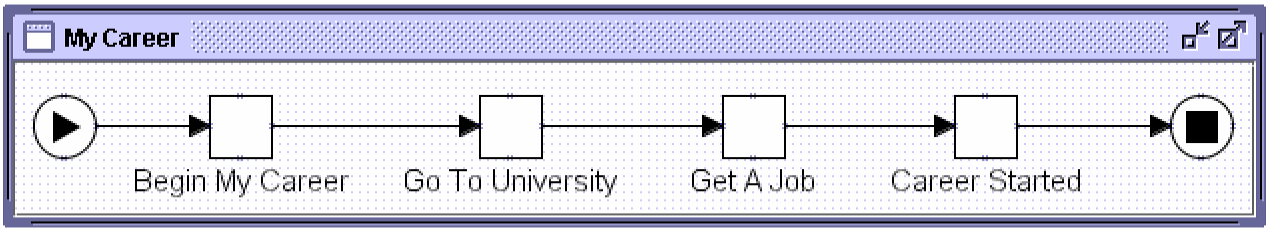


Fig. 5. The "My Career" Net

Decorating a Task

Decorating a task is the process of adding splits or join conditions on task.

By putting a split on a task, you are telling the task that when it has been completed, it's succeeding task could be one or more tasks. Here are the possible splits for a task:

- No split (no split on task)
- AND split
- OR split
- XOR split

By putting a join on a task, you are telling the task that it could become available through the completion of one or more preceding tasks. Below are some possible joins for a task:

- No join (no join on task)
- AND join
- OR join
- XOR join

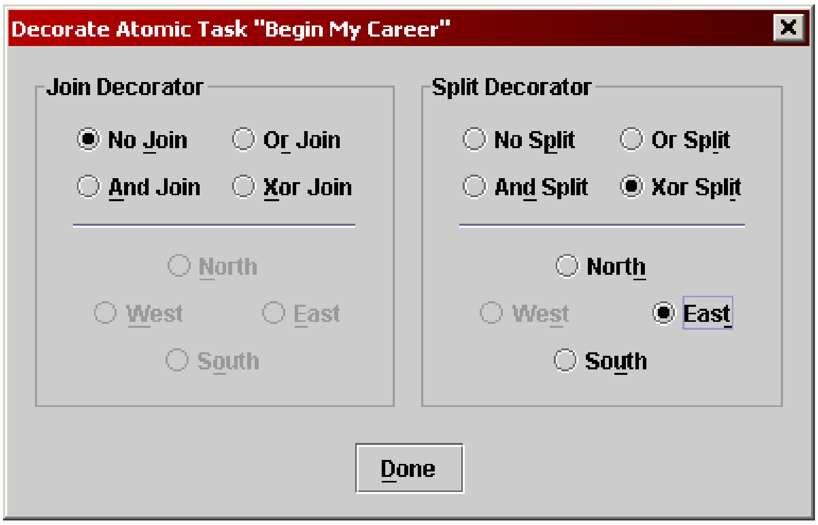
For more information on join and split types, please consult the YAWL technical papers on the YAWL website.

Creating Splits and Joins

To create a split or join:

1. Right click on your task and choose **Decorate**.

   You will be presented with the Decorate window for your task, which is divided between Join Decorator and Split Decorator options (see **Fig. 6.**)



Decorate Atomic Task "Begin My Career"

Join Decorator

● No Join ○ Or Join
○ And Join ○ Xor Join

○ North
○ West ○ East
○ South

Split Decorator

○ No Split ○ Or Split
○ And Split ● Xor Split

○ North
○ West ● East
○ South

Fig. 6. Decorating a Task

2. Choose the required split or join and the orientation for the split or join to appear and then click **Done**.

   Right click on your "Begin My Career" task and change the split to **Xor Split**. Then set the orientation to **East** , as per **Fig. 6.**

   Create a new Atomic task called "Do Private Study". This task will represent those students that choose not to go to University.

   Finally, right click on your "Get A Job" task and decorate it with an **Xor Join**. Then set the orientation to **West**.

3. Split and Join decorators allow you to connect several Flow Relations from and to your task respectively.

   Create a flow relation from "Begin My Career" to "Do Private Study", then create another flow relation from "Do Private Study" to "Get A Job", as per **Fig. 7.**

4. Don't forget to check the validity of your specification.

Hint: If you are having troubles with positioning your tasks, the alignment tools are a big help.

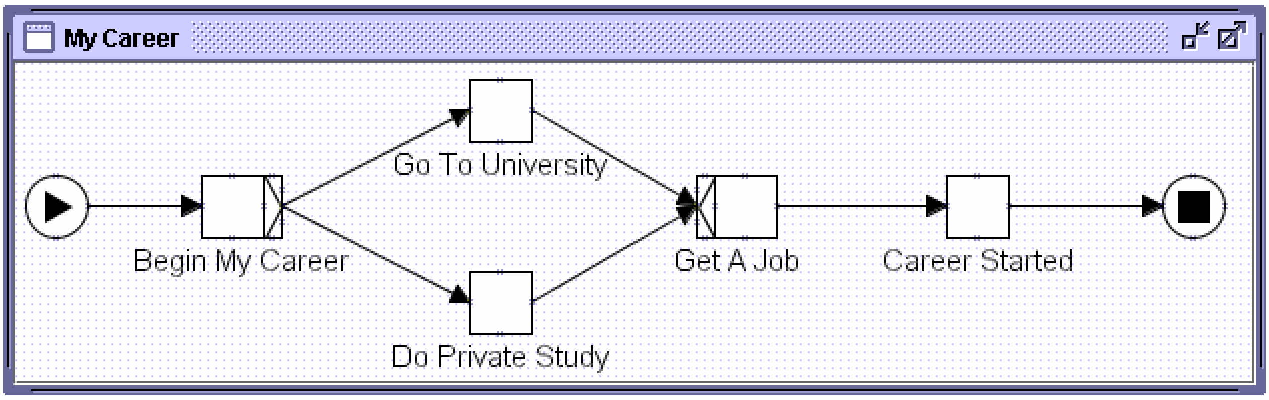


Fig. 7. XOR Split and Join

When "Begin My Career" has been completed, a choice must be made on which of the two tasks ("Go To University" or "Do Private Study") will be followed (XOR Split). "Get a Job" will become available after the completion of the task selected at the point of the XOR split.

Composite Tasks

Composite tasks are place holders for other YAWL Nets. That is, you can create another workflow in a separate Net, which is represented by the composite task in other nets.

To create a Composite Task:

1. Click on the Composite Task button, , in the Palette Bar or right click in the whitespace of the Net and choose **Composite Task**.

   We are going to replace our existing "Go to University" Atomic Task, with a composite task, so choose the Enter Marquee Selection Mode button, , on the Palette Bar, click on the "Go to University Atomic Task" and press the Delete key on the keyboard. We will add in the new composite task next.

2. Place your Composite Task in your Net.

   Reconnect the Flow Relations from "Begin My Career" to the new Composite Task, and from new Composite Task to "Get a Job".

3. Create a new Net by clicking on the Create a new Net button, , on the Menu Toolbar, or click on **Net** in the Menu and choose **Create Net**.

4. Choose a name for this Net by clicking on the **Net** Menu and choosing **Update Net Detail**.

   We are going to call this new Net "Attend University".

5. Return to your original Net and right click on your Composite Task and choose **Unfold to net…**. You will then be given a drop-down list with all the Nets available – choose the Net this task is to represent and then click **Done**.

S Choose "Attend University".

6. You can now fill out the detail of your new ``Attend University'' Net.

S Create the following Atomic Tasks in order and then link them with Flow Relations as per .,.,., and don't forget to check for validity:

- Enrol
- Do Subjects
- Pass All Subjects
- Get Degree

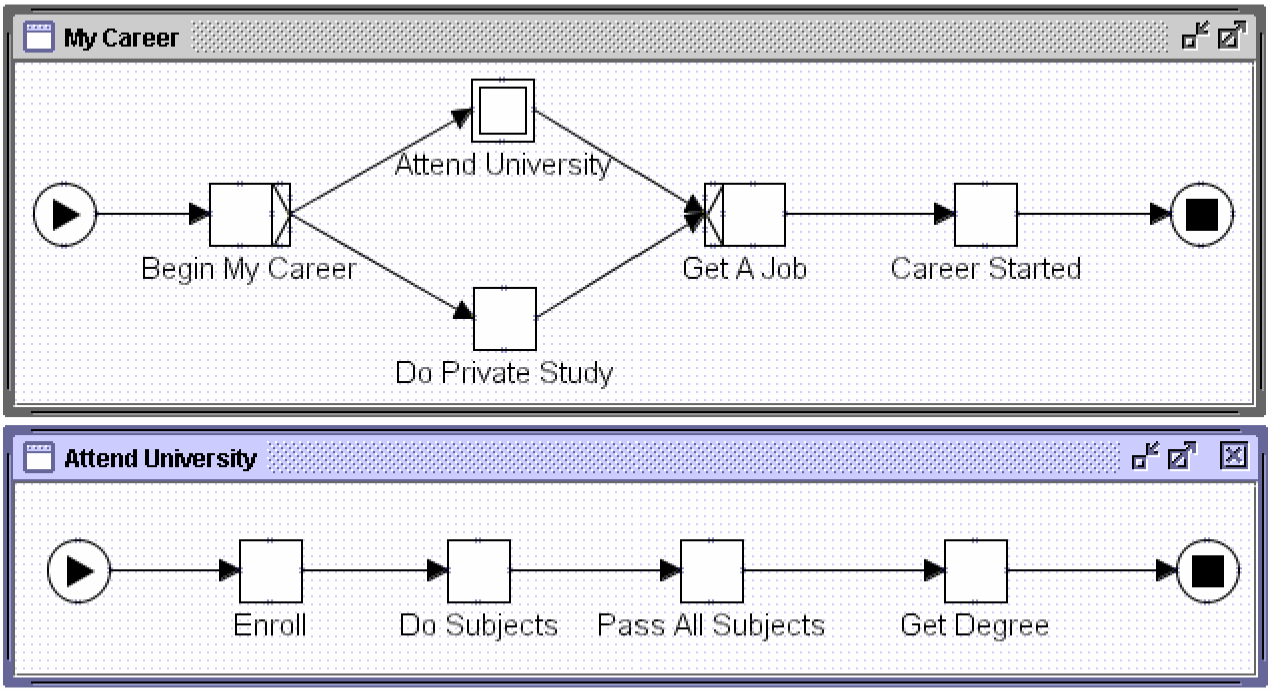


Fig. 8. "Attend University" Composite Task and its matching Net

Multiple Atomic Tasks

Multiple Atomic Tasks are allow you to run multiple instances of a task concurrently.

To create a Multiple Atomic Task:

1. Click on the Add Multiple Atomic Task button, ⧉ , in the Palette Bar or right click in the whitespace of the Net and choose **Multiple Atomic Task**.

Go back to the "My Career" Net. We are going to replace our existing "Do Private Study" Atomic Task, with a Multiple Atomic task, so choose the Enter Marquee Selection Mode button, , on the Palette Bar, click on the "Do Private Study" Atomic Task and press the Delete key on the keyboard. We will add in the new Multiple Atomic task next.

2. Place your Multiple Atomic Task in your Net and set the name of this task by right clicking on the task and choosing **Select Task Decomposition**.

Call this task the same name as before by selecting the "Do Private Study" decomposition from the drop-down list.

Reconnect the Flow Relations from "Begin My Career" to "Do Private Study", and from "Do Private Study" to "Get A Job", as per **Fig. 9.**

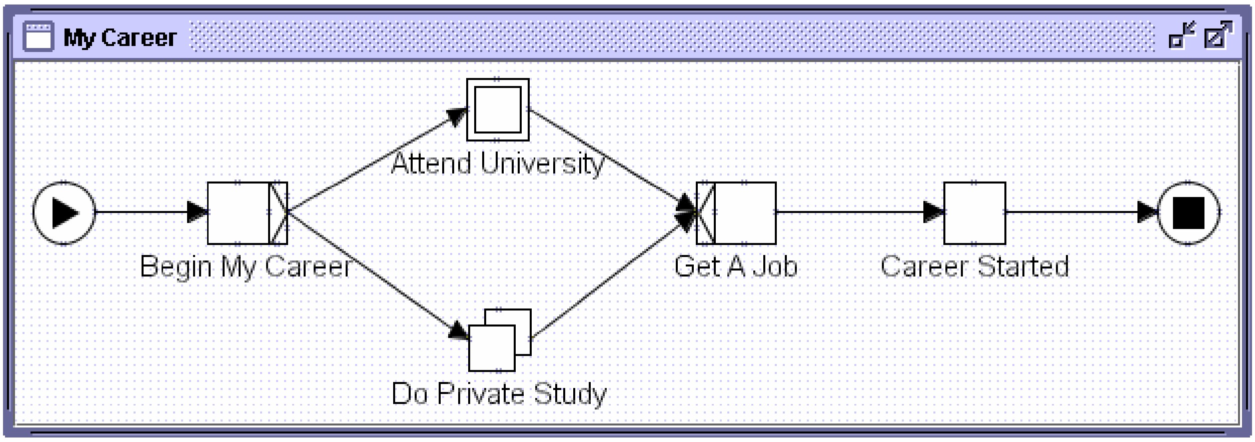


Fig. 9. Adding a Multiple Atomic Task

3. You will now need to set the parameters of the Multiple Atomic Task. Right click on the task and choose **Set Instance Detail**. Ensure that you are viewing the "Bounds" tab of the dialog, as per **Fig. 10.**

4. Set the Minimum Instances value. This is the minimum number of instances of this task that will be started when the task needs processing.

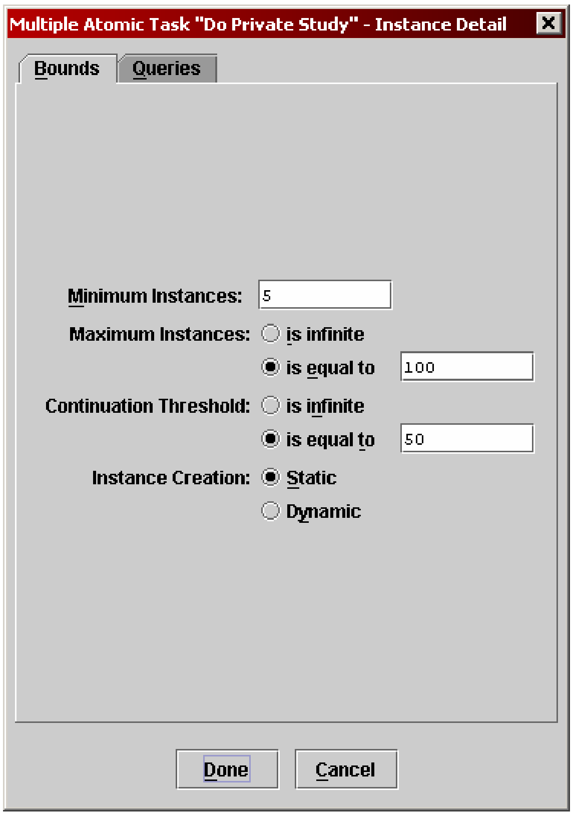
Set the Minimum Instances to 5.

5. Set the Maximum Instances value. This is the maximum number of instances of this task that can be created.

Set the Maximum Instances to 100

6. Set the Continuation Threshold value. The moment all instances created have completed or as many instances as the Continuation Threshold specified have completed, the multiple instance task itself is considered completed, which triggers relevant outgoing flow relations from this task.

Set the Continuation Threshold to 50.


Multiple Atomic Task "Do Private Study" - Instance Detail

Minimum Instances: 5

Maximum Instances: ○ is infinite
● is equal to 100

Continuation Threshold: ○ is infinite
● is equal to 50

Instance Creation: ● Static
○ Dynamic

Fig. 10. Instance Bounds on Multiple Instance Tasks

7. Choose the Instance Creation. If a multi-instance task has static creation mode, when the task is started the number of instances created is decided then, and cannot be altered during the execution of the task. Conversely, if a task has dynamic creation mode then new instances of the task can start once the initial minimum instance number of tasks has started, but before the maximum number has been received.

S Set the Instance Creation type to "Static".

8. Click **Done**.

With the values set in the scenario, the Do Private Study task is indicating that a maximum of 100 instances of this task can be executed before triggering the task's outgoing flow. , A minimum of five instances will be created, and once 50 instances have completed, the outgoing flow relations to ''Get A Job" will trigger.

The "Queries" tab of the Multiple Instance dialog will be explained later in the "Setting Multiple Instance Queries" section of this manual.

Multiple Composite Tasks

Multiple Composite Tasks allow you to run multiple instances of the Net represented by a multiple composite task, concurrently.

To create a Multiple Composite Task:

1. Click on the Add Multiple Composite Task button, , in the Palette Bar or right click in the whitespace of the Net and choose **Multiple Composite Task**.

   Go to the “My Career” Net. We are going to replace our existing “Do Private Study” Multiple Instance Task, with a Multiple Composite task, so choose the Enter Marquee Selection Mode button, , on the Palette Bar, click on the “Do Private Study” Task and press the Delete key on the keyboard. We will add in the new Multiple Composite task next.

2. Place your Multiple Composite Task in your Net.

   Reconnect the Flow Relations from “Begin My Career” to the new Multiple Composite Task, and from the new Multiple Composite Task to “Get a Job”.

3. You will now need to set the parameters of the Multiple Composite Task. Right click on the task and choose **Set Instance Detail**.

9. Set the Minimum Instances value. This is the minimum number of instances of this task that will be started when the task needs processing.

   Set the Minimum Instances to 5.

10. Set the Maximum Instances value. This is the maximum number of instances of this task that can be created.

    Set the Maximum Instances to 100

11. Set the Continuation Threshold value. The moment all instances created have completed or as many instances as the Continuation Threshold specified have completed, the multiple instance task itself is considered completed, which triggers relevant outgoing flow relations from this task.

    Set the Continuation Threshold to 50.

12. Choose the Instance Creation. If a multi-instance task has static creation mode, when the task is started the number of instances created is decided then, and cannot be altered during the execution of the task. Conversely, if a task has dynamic creation mode then new instances of the task can start once the initial minimum instance number of tasks has started, but before the maximum number has been received.

    Set the Instance Creation type to “Static”.

4. Click **Done**.

5. Create a new Net by clicking on the Create a new Net button, , on the Menu Toolbar, or click on **Net** in the Menu and choose **Create Net**.

6. Give the new Net a name by clicking on the **Net** Menu and choosing **Update Net Detail.**

We are going to call this new Net "Study Privately".

7. Return to your original Net and right click on your Multiple Composite Task and choose **Unfold to Net**. You will then be given a drop-down list with all the Nets Available – choose the Net for this task to initiate and then click **Done**.

Choose "Study Privately":

8. You can now complete your new "Study Privately" Net represented by your Composite Task.

Create the following Atomic Tasks in order and then link them with Flow Relations as per **Fig. 11.**

- Read a Book
- Feel Smarter

Don't forget to validate your specification.

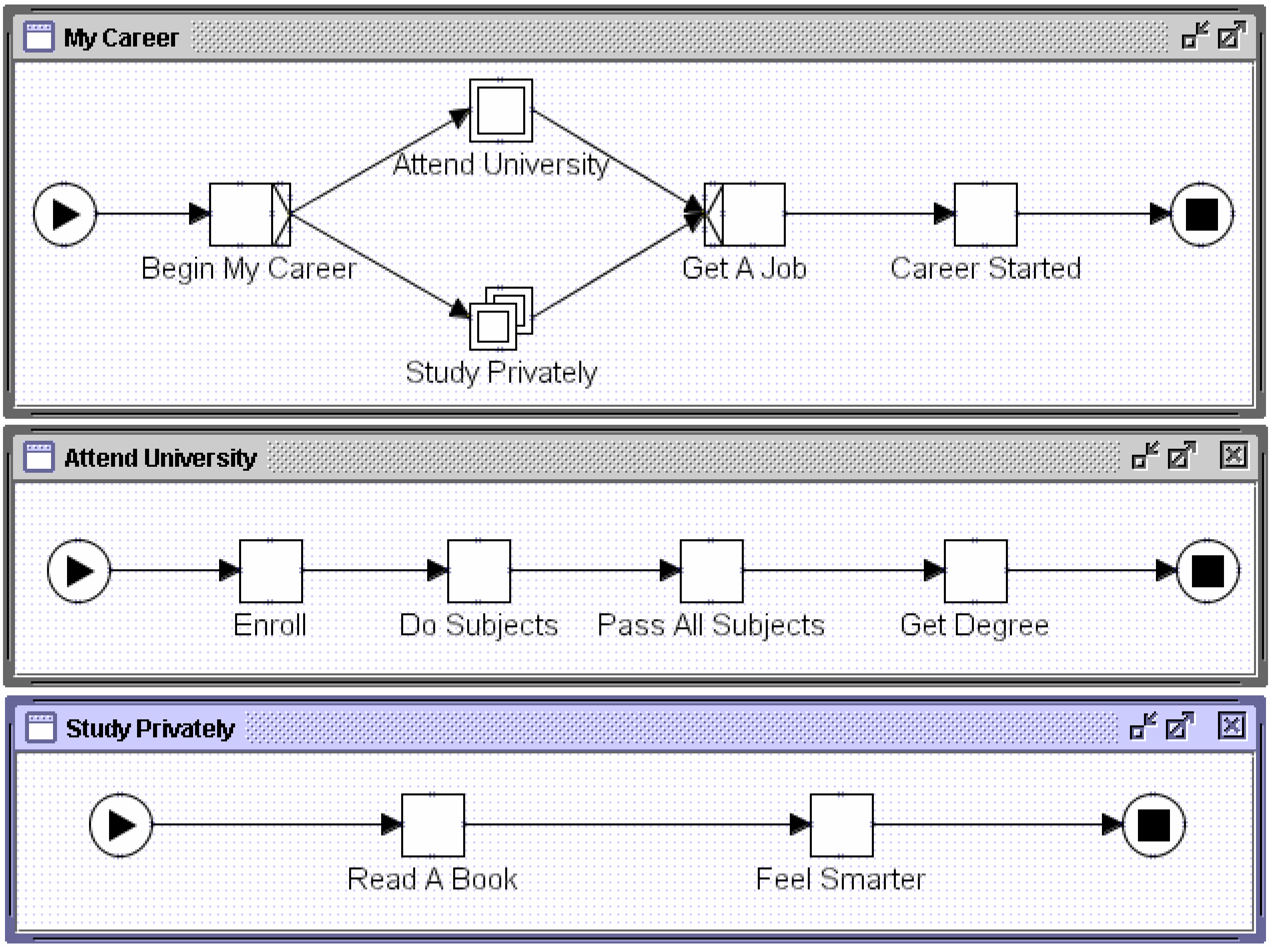


Fig. 11. "Privately Study" Multiple Composite Task and its Net

Conditions

Conditions are ways to represent a state, in between tasks, for the workflow.

To create a Condition:

1. Click on the Add a Condition button, ○ , in the Palette Bar or right click in the whitespace of the Net and choose **Condition**.

S Go to the Do Private Study Net. We are going to place a loop Condition after the Read a Book Atomic Task, to determine whether we gained any knowledge from the book. We will add the new Condition next.

2. Place your Condition in your Net and set the name by right clicking on the Condition and choosing **Set Label**.

S Call this Condition "Knowledge Gained?"

3. Now link to the Condition using the Flow Relations object.

S Select the Enter Marquee Selection Mode button, ⬚ , from the Palette Bar and click on the Flow Relation going from the Read a Book Atomic Task to the Feel Smarter Atomic Task and press the Delete button on the keyboard.

S Create a Flow Relation from "Read A Book" to "Knowledge Gained?"

4. Create a Flow Relation from your Condition to a task.

S Set the Flow Relation from "Knowledge Gained?" Condition to "Feel Smarter" Atomic Task.

5. Create another Flow Relation from your Condition to another task to signify the two possible flows from the Condition.

S Before we create our second flow relation from our Condition, create another Atomic Task and call it "Look for Another Book".

S Change the Join Decoration for Atomic Task "Read a Book" to an OR Join, with the orientation being West, by right clicking on Read a Book and choosing **Decorate**. Click **Done**.

S Finally create the Flow Relation from the "Knowledge Gained?" Condition back to the OR Join of the "Read A Book" Atomic Task, as per **Fig. 12.**

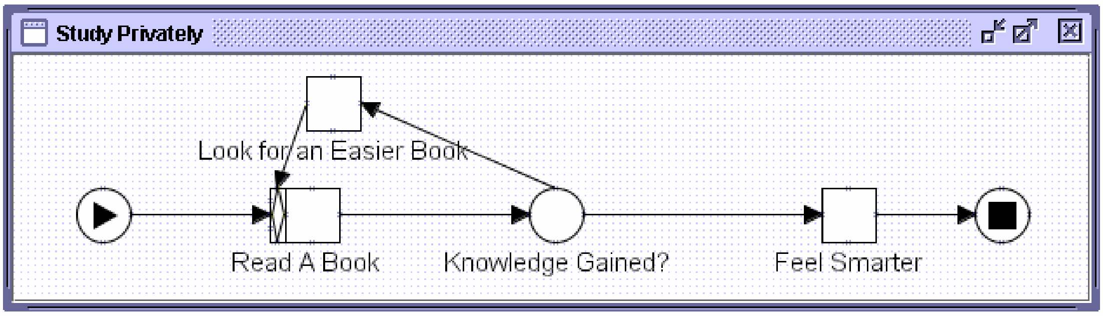


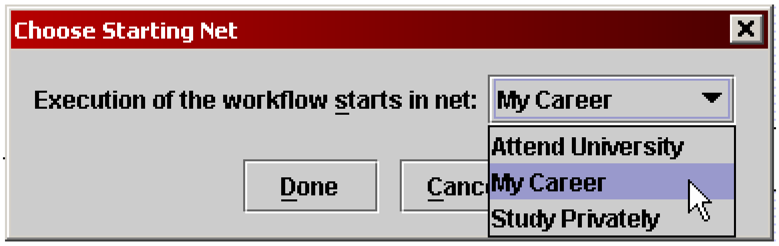
Fig. 12. Condition Loop for Do Private Study Net

S Last but not least, validate your specification.

Changing the Starting Net

At any stage you can change the starting Net of the specification. To change the starting Net:

1. Select **Net** from the Menu.
2. Choose **Set Starting Net**.
3. From the Choose Starting Net window, click on the drop-down list and select a new starting Net.
4. Click **Done**.



Choose Starting Net

Execution of the workflow starts in net: My Career

Fig. 13. Changing the Starting Net

Changing the Appearance of Your Specification

Improving the Look of Flow Relations

You can control and improve the look of the Flow Relations by adding in "knees".

1. Select the Enter Marquee Selection Mode button, ⬚, from the Palette Bar.

S Go to the Do Private Study Net.

2. Right click on the position in the Flow Relation where you want to add a "knee", signified by a small square, ─▪─, in the Flow Relation.

S Right click somewhere on the Flow Relation going from "Knowledge Gained?" Condition to "Look for an Easier Book" Atomic Task.

3. Then left click on the knee and drag it out to a more desirable location. You can add as many knees to a Flow Relation as you like.

S Repeat the process for the Flow Relation between "Look For an Easier Book" and "Read A Book" tasks (see **Fig. 14.**).

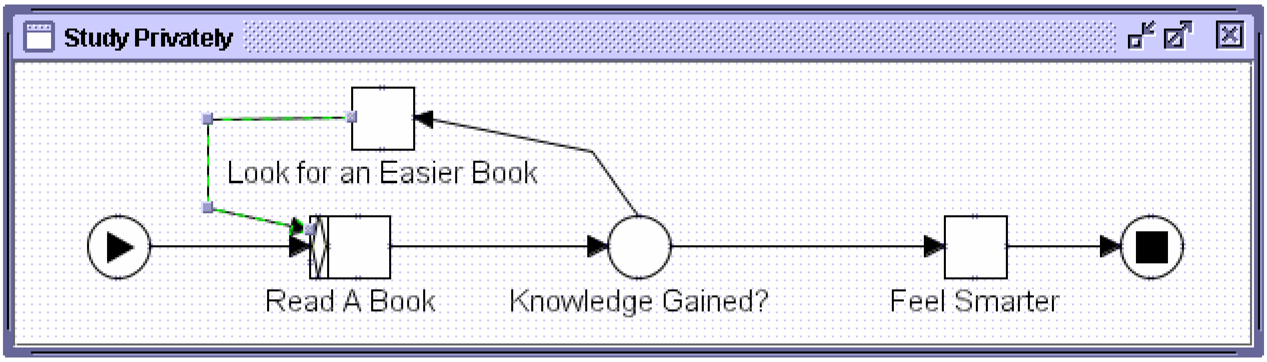


Fig. 14. Adding "knees" to a Flow Relation

Editing Objects

You can edit more than one object at a time by using the Enter Marquee Selection Mode tool. See **Fig. 15.**

1. Select the Enter Marquee Selection Mode button, ⬚ , from the Palette Bar.
2. Click on the first object that you want to edit.
3. Hold down the shift key and then click on the other objects that you want to edit.
4. Now choose the **Edit** option from the Menu or continue holding down the shift key and right click on the mouse button. Below are the edit options:
   - Cut, Copy, Delete
   - Align
   - Size Increase / Decrease to change the appearance of the objects. This can also be done using the CTRL key plus Up or Down arrow on your keyboard.

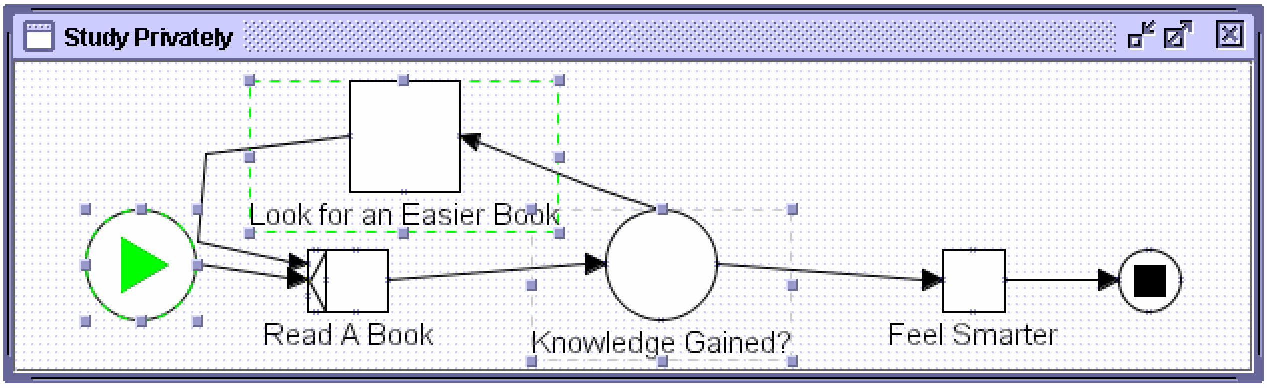


Fig. 15. Changing the Size of Multiple Objects

Changing Font Size

You can change the size of the font used to label tasks and conditions.

1. Change the font size by clicking on the **View** Menu and choosing the **Label Font Size… option.**

S Change the font size to 18.

Advanced Specification Features

Cancellation Sets

Cancellation Sets allow you to nominate any number of tasks, conditions or implicit conditions (which exist on flow relations between tasks, but are not drawn) for cancellation, upon the execution of a specified task. That is, once a specified task is executed in the workflow, all state within that task's cancellation set is removed.

To create a Cancellation Set:

1. First select the task that will initiate the Cancellation Set, using the Enter Marquee Selection Mode button, ⬚ , from the Palette Bar.
2. Right-click on the task, then choose **Viewing My Cancellation Set**.

   The task will be coloured grey to indicate that this task is the cancellable task.

Go to the "My Career" Net. We are going to make the "Attend University" task a cancellable task to demonstrate in the workflow, that if a person executes this task, then they don't have time to perform the Do Private Study task, therefore it should be removed as an option.

Right-click on "Attend University" and choose **Viewing My Cancellation Set** (as per **Fig. 16.**).

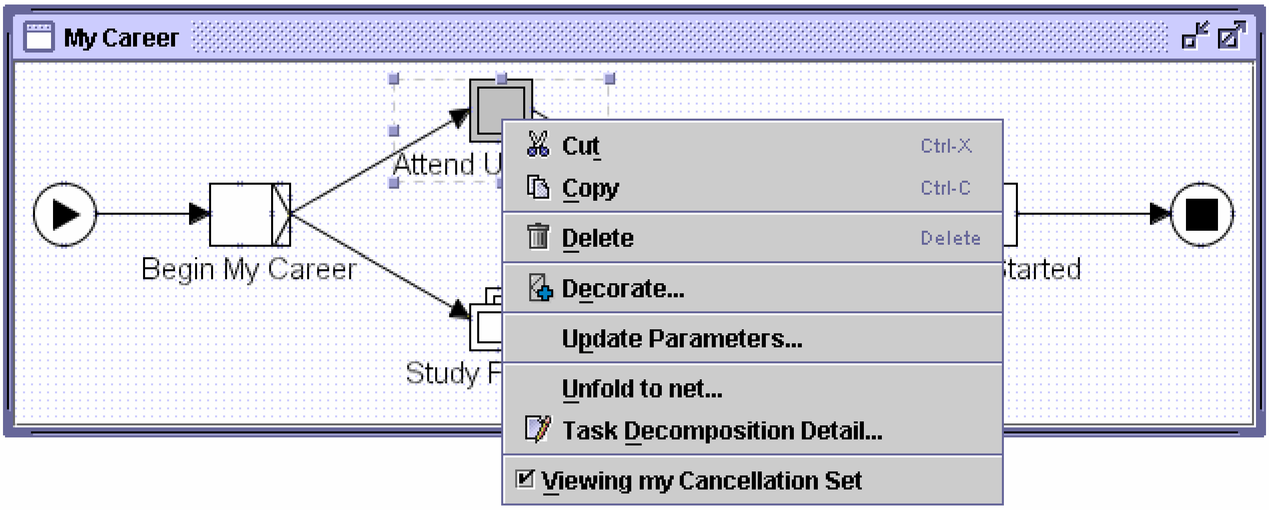


Fig. 16. Viewing a task's cancellation set

3. Next, using the Enter Marquee Selection Mode button, ⬚ , from the Palette Bar, choose a task, condition or implicit condition to add to the Cancellation set. Hold down the shift key to select more than one object for cancellation.

Select the “Study Privately” task and the Flow Relation preceding it.

4. Click on the Add Selected Items to Visible Cancellation Set button, ⬚ , on the Menu Toolbar.

   Items will be coloured red to indicate belonging to the cancellation set (see **Fig. 17.**).

Add the “Study Privately” task and the preceding Flow Relation to the cancellation set.

5. Once you have established the cancellation set, you can right-click on the cancellable task and tick-off **Viewing My Cancellation Set** option.

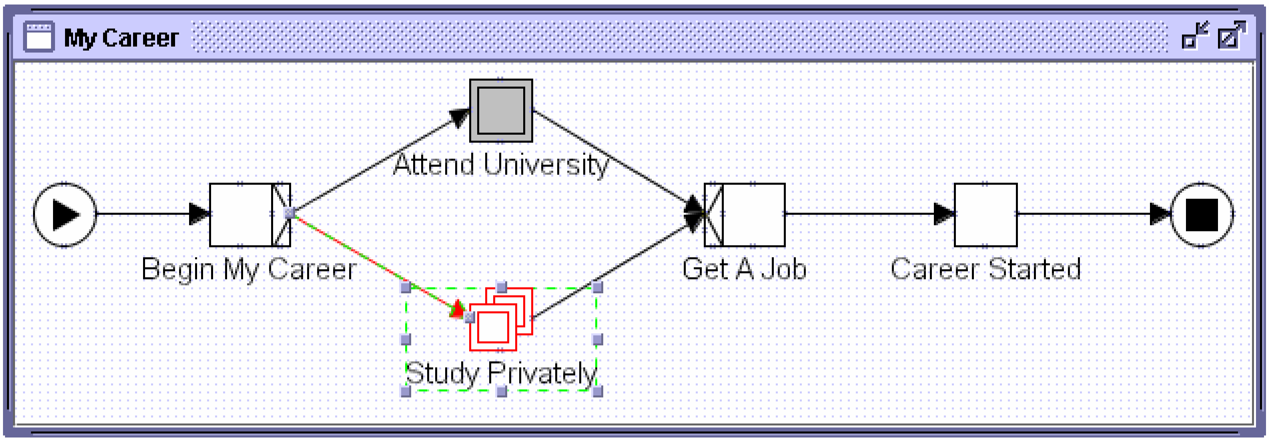


Fig. 17. Adding elements to a Cancellation Set

Notes about Cancellation Sets:

- A Cancellation Set that has been created will remain in the workflow, regardless of whether you have the **Viewing My Cancellation Set** option ticked.
- You can create multiple Cancellation Sets in your workflow, by selecting another task and choosing the **Viewing My Cancellation Set** option.
- All flows leading to or from conditions are not valid cancellation set members. Neither are the Input and Output conditions. The editor will ignore them if you select them for inclusion in a task’s cancellation set.

To remove an element from a Cancellation Set:

1. First, make sure you have the **Viewing My Cancellation Set** option ticked.

   If it isn't ticked, select the task that is initiating the Cancellation Set, using the Enter Marquee Selection Mode tool from the Palette Bar and right-click, then choose **Viewing My Cancellation Set**.

2. Select the element for removal, using the Enter Marquee Selection Mode button, .

3. Click on the Remove Selected Items from Visible Cancellation Set button , ,on the Menu Toolbar.

Updating Data Type Definitions

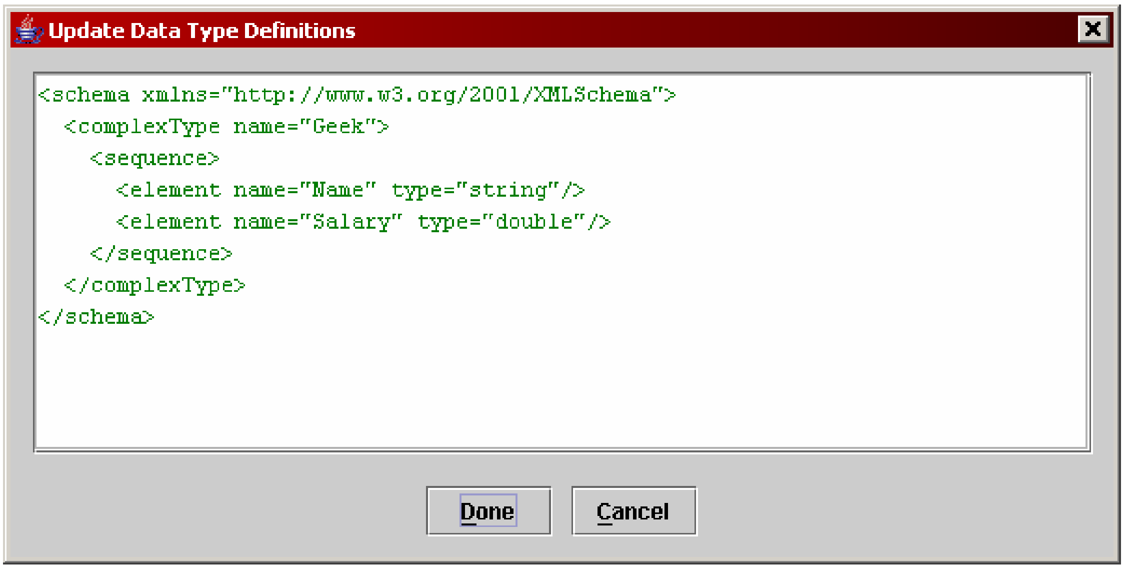
Version 1.1 of the YAWL Editor introduces the ability for workflow designers to specify data that is processed by a workflow. XMLSchema is used to describe the valid data types that are available.

By default, a number of simple XMLSchema data types are supplied for variable definition, but if you need more complex data types, you can supply your own XMLSchema definition to describe them.

1. Select **Update Data Type Definitions** from the **Specification** Menu.

2. Enter your XMLSchema Data Type Definition into the dialog box. **Fig.28.**

3. If the definition text is green, your new data type is now available for defining Net or Task variables in your specification. If the text is red, there is something wrong with your data type definition, and the data type will not be available.

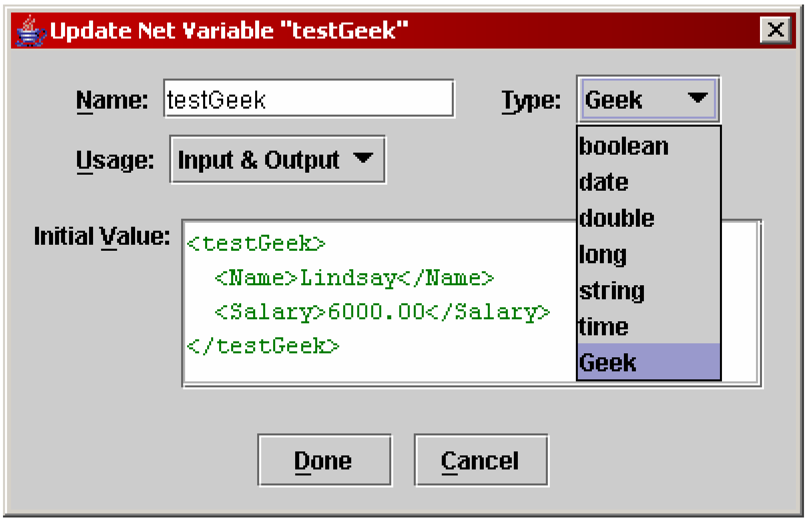


```
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <complexType name="Geek">
    <sequence>
      <element name="Name" type="string"/>
      <element name="Salary" type="double"/>
    </sequence>
  </complexType>
</schema>
```

Fig. 18. Adding the "Geek" complex data type

Open the Data Type Definitions dialog and type in the XML text that appears in **Fig. 18**.

The above example creates a complex data type called "Geek" that has two separate sub-components, "Name" and "Salary" of type "string" and "double" respectively. As depicted in **Fig. 19.,** *the new data type "Geek" is available to choose when creating a task or net variable.*



Name: testGeek

Type: Geek

Usage: Input & Output

Initial Value:

```
<testGeek>
  <Name>Lindsay</Name>
  <Salary>6000.00</Salary>
</testGeek>
```

Fig. 19. A "Geek" task variable

Net Decomposition Detail / Updating Net Variables

You can add variables to a net to store information relating to that net that tasks within the net may need to read or update.

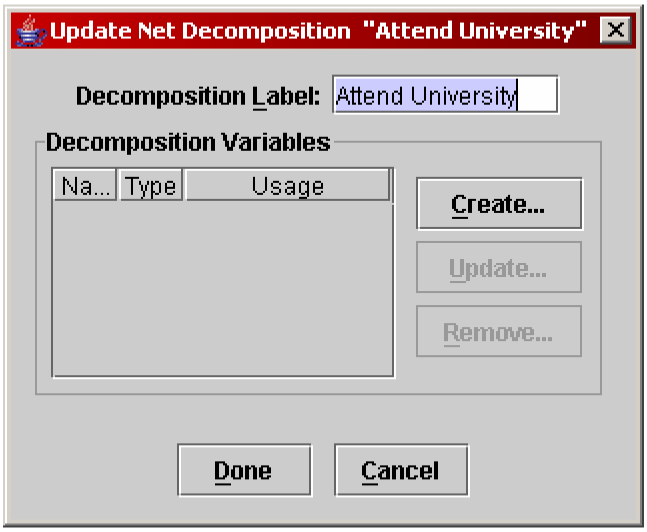
To add a variable to a Net:

1. Choose **Update Net Detail** from the **Net** Menu.

   We will be setting up Net variables in the "Attend University" net.

   Go to the "Attend University Net" and choose **Update Net Detail** from the **Net** Menu.

2. An Update Variables dialog box will appear. **Fig. 20.**
3. Click on **Create**. An Update Net Variable dialog box will appear.



Update Net Decomposition "Attend University"

Decomposition Label: Attend University

Decomposition Variables

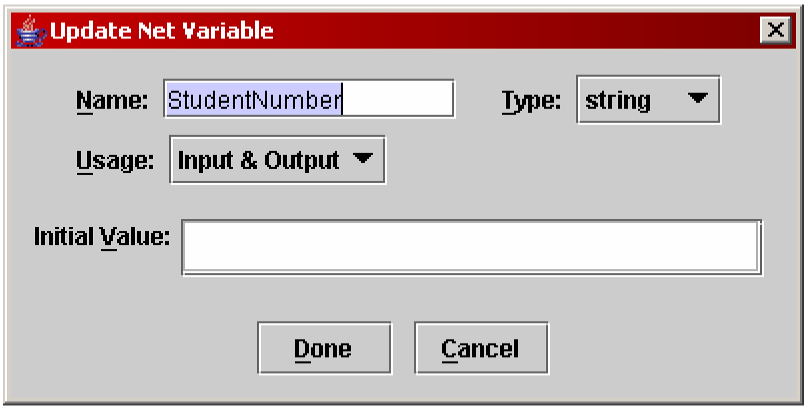
Na...	Type	Usage

Fig. 20. Updating "Attend University" Net Variables

4. Enter the Name of your variable, choose the Type of the variable from the list, optionally enter an Initial Value for the variable then click **Done**, then **Done** again to close the Net Decomposition dialog.

   Type in "StudentNumber" for the name of the variable and choose its type as **string**. Leave the Initial Value blank and Usage as "Input & Output". **Fig. 21.** Click **Done**.

   Create another Net Variable with the name "SubjectCode" and Type of **string**. Leave the Initial Value blank and Usage as "Input & Output". Click **Done**.



Update Net Variable

Name: StudentNumber

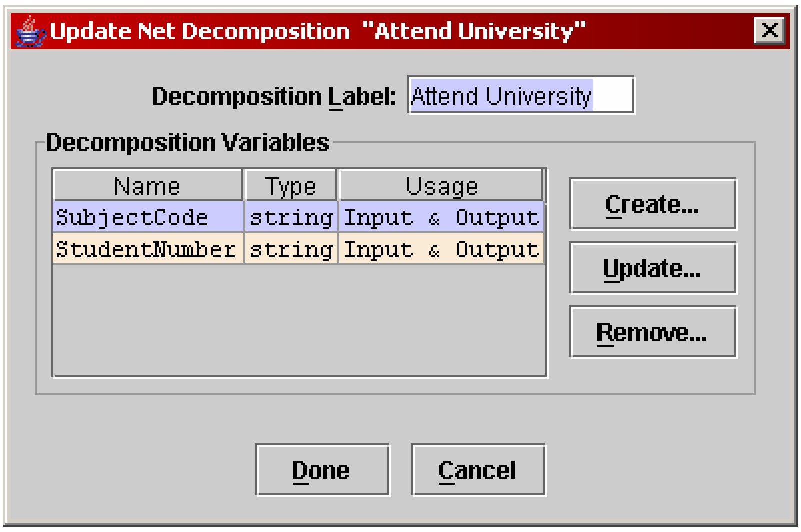
Type: string

Usage: Input & Output

Initial Value:

Fig. 21. The Net Variable "StudentNumber"

5. The Net Variables should now appear in the Update variables of Net "Attend University" dialog box. **Fig. 22.**



Update Net Decomposition "Attend University"

Decomposition Label: Attend University

Decomposition Variables

Name	Type	Usage
SubjectCode	string	Input & Output
StudentNumber	string	Input & Output

Fig. 22. Updated "Attend University" Net Variables

Select Task Decomposition

By choosing the **Select Task Decomposition…** option when you right click on a task, you have the ability to identify which decomposition this task represents.

Like nets, tasks have decompositions where you can specify variables and a label to associate with the task or net. Unlike nets, which cannot share net decompositions, there is a 1:N relationship between task decompositions (scoped to the entire specification) and their tasks (scoped to nets).

Besides variables and a label, task decompositions also allow the workflow designer to identify which web service the decomposition should invoke in a running workflow engine. When two tasks share the same decomposition, we are saying that the same activity is required in two different places in the workflow.

You can use the drop-down list to select the task decomposition of a task, or alternately you can press the **Create…** button and generate a new one that will automatically become the task's decomposition. **Fig. 23.**

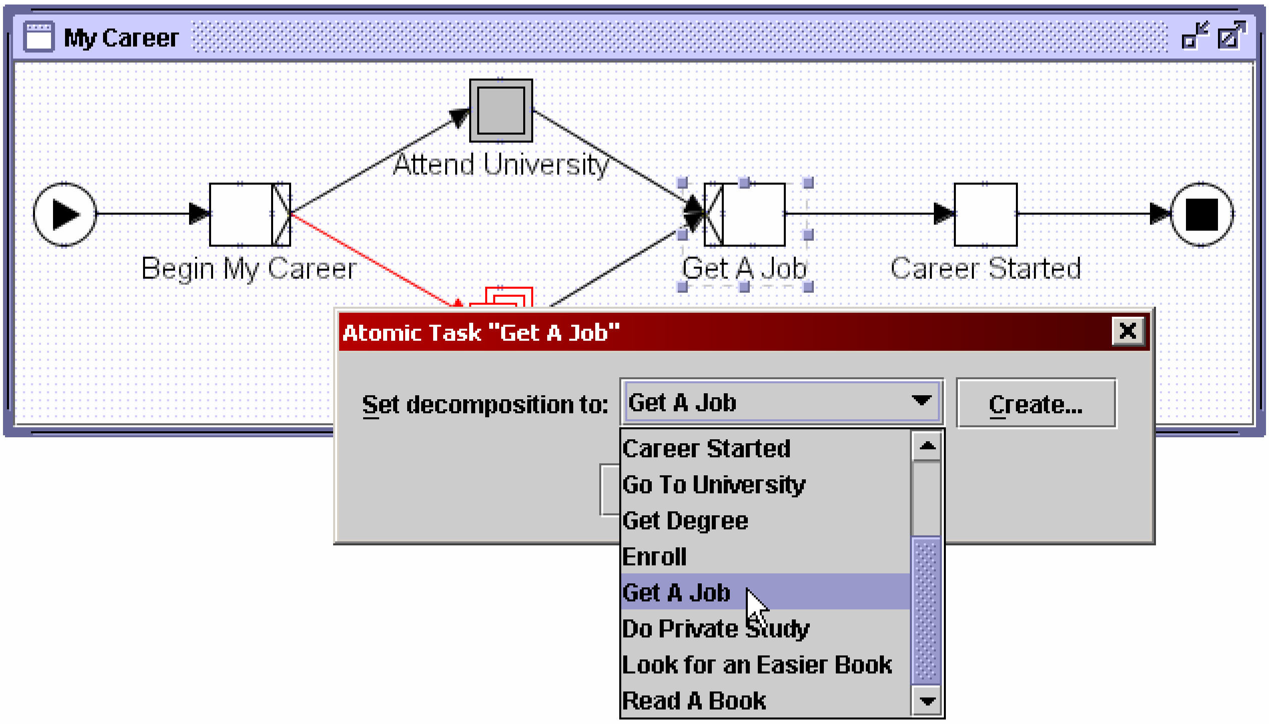


Fig. 23. Select Task Decomposition

Task Decomposition Detail / Updating Task Variables

You can add variables to a task to store specific information relating to that task. Task variables have several uses. One use is to transferring information between workflow users and the workflow engine. A second use is for passing data between web services that the running workflow engine invokes and the Net the task resides in.

For example, if your task is called ‘Place an Order’, you may want to store the name or identification number of the person placing the order or maybe store the items being ordered.

Adding a Variable to a Task

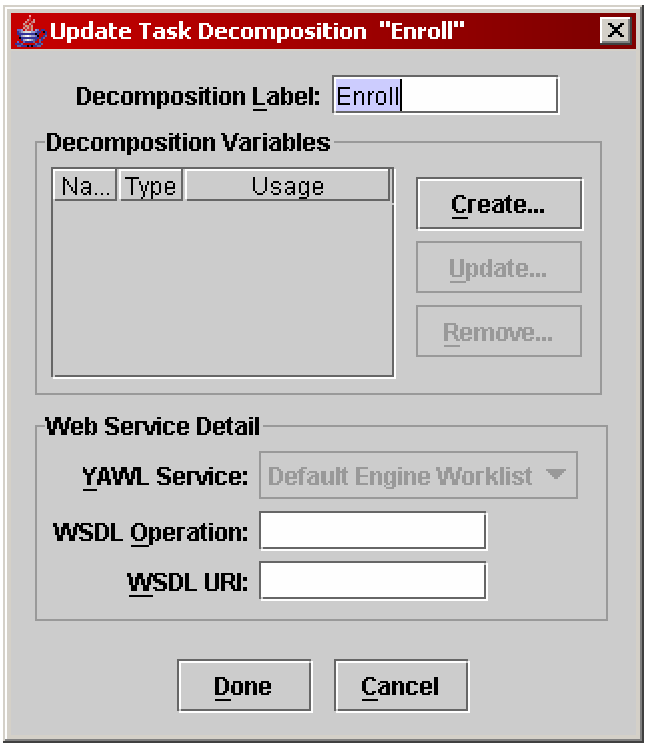
1. First select the task that will require the variable, using the Enter Marquee Selection Mode button, , from the Palette Bar.

We will be setting up variables for the “Enroll” task.

Go to the “Attend University” Net and select the “Enroll” task.

2. Right-click on the task and choose **Task Decomposition Detail…**. An Update Task Decomposition dialog box will appear. **Fig. 24.**

Retrieve the decomposition detail for the “Enroll” task.



Update Task Decomposition "Enroll"

Decomposition Label: Enroll

Decomposition Variables

Na...	Type	Usage

Web Service Detail

YAWL Service: Default Engine Worklist

WSDL Operation:

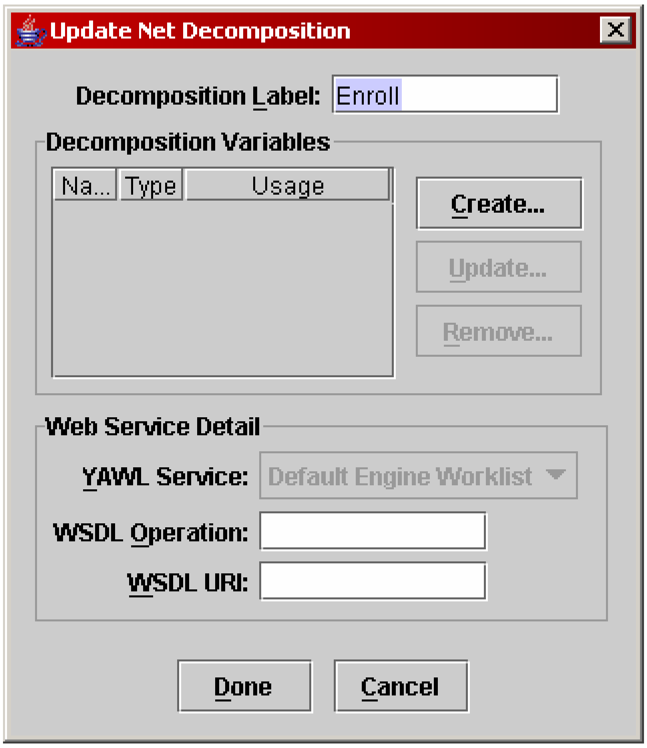
WSDL URI:

Fig. 24. Updating variables for a task

3. Click on the **Create…** button. An Update Task Variable dialog box will appear.

4. Enter the Name of your variable, choose the Type of the variable from the list, optionally enter an Initial Value for the variable then click **Done**, then **Done** again to exit the task decomposition detail dialog.

   Type in "StudentNumber" for the name of the variable and choose the type **string**. Leave the Initial Value blank. **Fig. 25.**



Update Net Decomposition

Decomposition Label: Enroll

Decomposition Variables

Na...	Type	Usage

Web Service Detail

YAWL Service: Default Engine Worklist

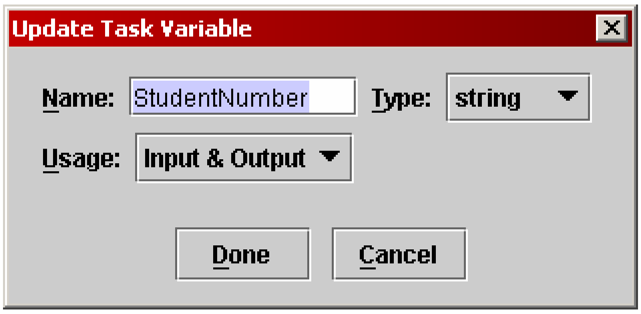
WSDL Operation:

WSDL URI:

Fig. 25. Updating the Task Variable

5. Create another variable for the same task, called “SubjectCode” with the type being **string** and usage being “**Input & Output**”**.** Click **Done**. **Fig. 26.**

   The “Enroll” task now has two variables, “StudentNumber” and “SubjectCode”.



Update Task Variable

Name: StudentNumber

Type: string

Usage: Input & Output

Fig. 26. Updating another Task Variable

Adding / Updating Task Parameters

Both Input and Output Parameters can be assigned to any tasks to allow the passing of state between nets and their tasks, and between tasks and workflow engine users and web services.

Input Parameters use an XQuery to massage net variable state (across possible several net variables) into a value that can be passed to a single selected task variable.

Output parameters use an XQuery to massage task variable state (across possible several task variables) into a value that can be passed to a single selected net variable.

For example, if a task is called 'Place an Order', then an Input Parameter could be the name of the person placing the order, whereas the Output Parameter could be the corresponding identification number of that person.

To add an Input Parameter:

1. First select the task that will require the parameter, using the Enter Marquee Selection Mode button, ⬚ , from the Palette Bar.

   We will be setting up Input Parameters for the variables that we created in the Adding / Updating Task Variables section previously.

   Go to the "Attend University" Net and select the "Enrol" task.

2. Right-click on the task and choose **Update Parameters…**. An Update Parameters dialog box will appear. **Fig. 27.**

   Update the Parameters for the "Enrol" task.

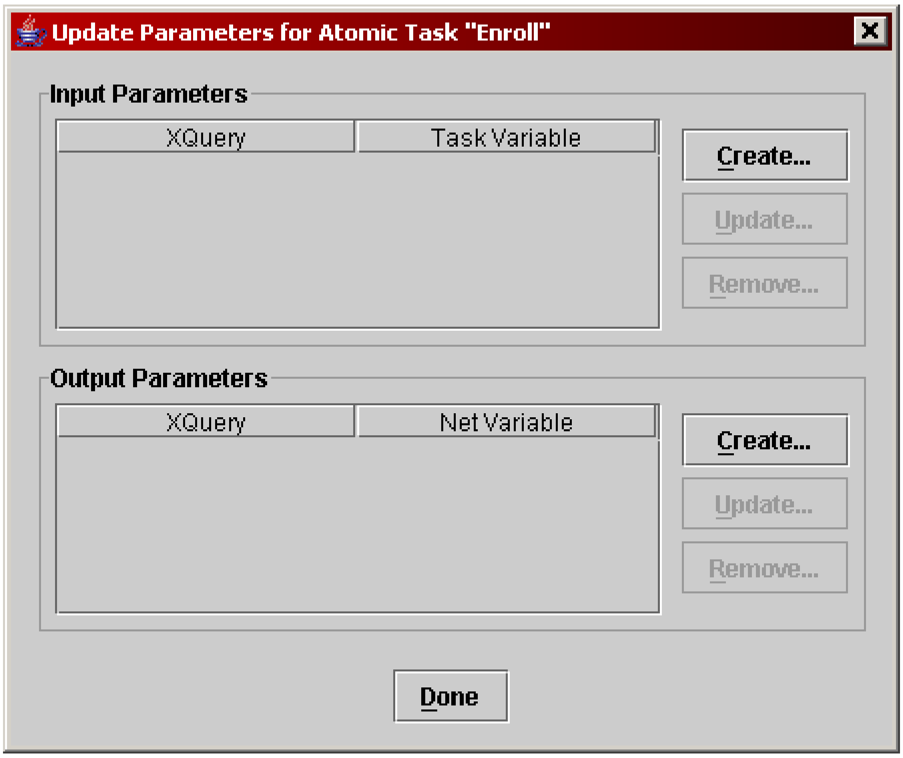


Fig. 27. Updating Parameters for a Task

3. In the Input Parameters section, click on **Create…**. An Update Task Parameter dialog box will appear.

4. If you have already set up a Task Variable for this task, then the Existing Task Variable option will be activated and there will be a list of task variables to choose from. Choose a variable from the list and click on **Done**, then **Done** again to close the task parameters dialog.

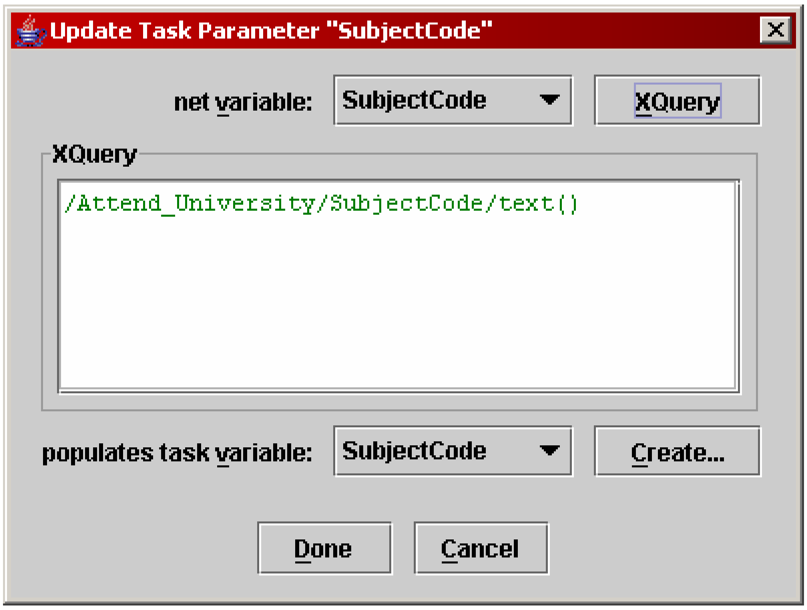
   If you haven't set up Task Variables, then click on **Create…** and return to the previous section in the manual for Adding a Task Variable.

   If you are familiar with XQuery syntax, then you can paste in an XQuery to allow manipulation of the Input Parameters. "Syntactically well formed" XQueries will be green, and badly formed ones will be red.

   As a convenience method, the **XQuery** button will generate an engine-compatible XQuery that supplies the net variable currently selected in the net variable drop-down box.

   Select the "SubjectCode" variable from the list of Existing Task Variables. Select the "SubjectCode" net variable and click **XQuery**. Alter the query to extract the net variable's content by adding "/text()" to the end of the query. **Fig. 28.** Click **Done.**

   Create another Task Parameter and map the net "StudentNumber" to the task variable of the same name using this technique. Click **Done** and **Done** again.


Update Task Parameter "SubjectCode"

net variable: SubjectCode

XQuery

```
/Attend_University/SubjectCode/text()
```

populates task variable: SubjectCode

Fig. 28. Passing a net's SubjectCode to a Task

To add an Output Parameter:

1. First select the task that will require the parameter, using Enter Marquee Selection Mode button, , from the Palette Bar.

   We will be setting up Output Parameters for the variables that we created in the Adding / Updating TaskVariables section.

Go to the “Attend University” Net and select the “Enroll” task.

2. Right-click on the task and choose **Update Parameters…**. An Update Parameters dialog box will appear. **Fig. 27.**

Update the Parameters for the “Enroll” task.

3. In the Output Parameters section, click on **Create**. An Update Net Parameter dialog box will appear (similar to **Fig. 28.**).

5. If you have already set up a Task Variable for this task, then the Existing Task Variable option will be activated and there will be a list of task variables to choose from. Choose a variable from the list and click on **Done**, then **Done** again to close the task parameters dialog.

   If you haven’t set up Net Variables, then click on **Create…** and return to the previous section in the manual for Adding a Net Variable.

   If you are familiar with XQuery syntax, then you can paste in an XQuery to allow manipulation of the Input Parameters. "Syntactically well formed" XQueries will be green, and badly formed ones will be red.

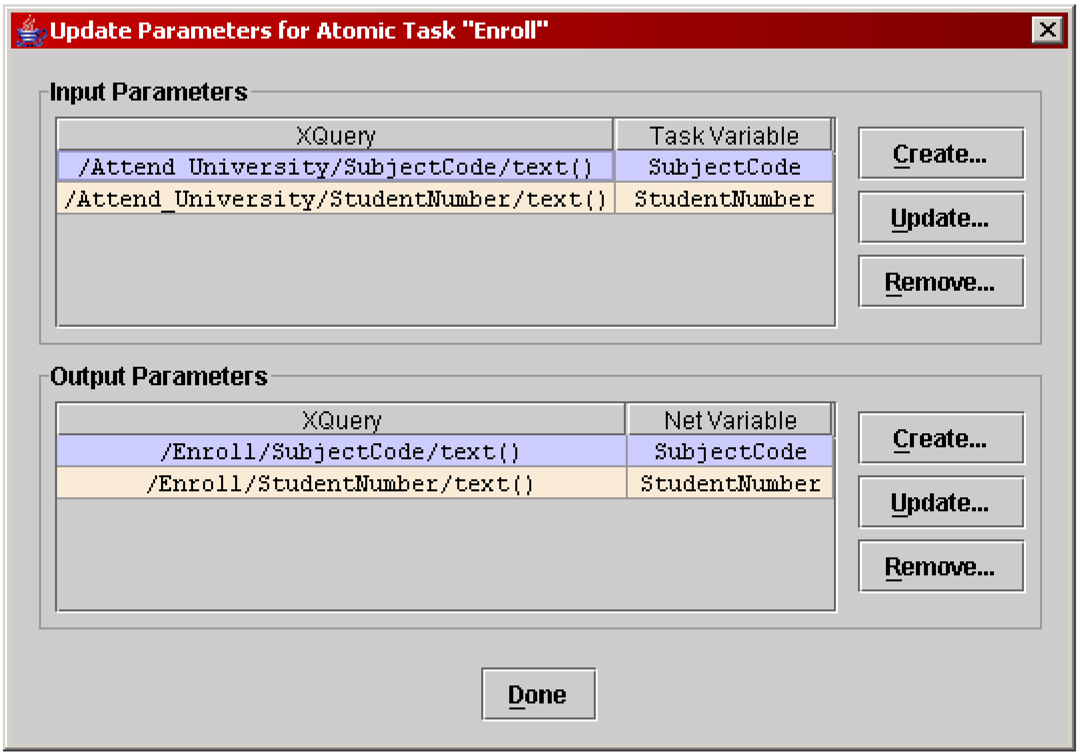
   As a convenience method, the **XQuery** button will generate an engine-compatible XQuery that supplies the task variable currently selected in the task variable drop-down box.

4. Select the “SubjectCode” variable from the list of Existing Net Variables. Select the “SubjectCode” task variable and click **XQuery**. Alter the query to extract the task variable’s content by adding “/text()” to the end of the query. Click **Done.**

   Create another Task Parameter and map the task “StudentNumber” to the net variable of the same name using this technique. Click **Done** and **Done** again.

   These Output Parameters were set up to demonstrate a simple transfer of state from a net to a task and back to the net. Perhaps the task would allow a user to change the values of one of the variables which would eventuate in the net’s values changing.

   The Update Parameters dialog box should appear as in **Fig. 29.**



Input Parameters

XQuery	Task Variable
/Attend University/SubjectCode/text()	SubjectCode
/Attend_University/StudentNumber/text()	StudentNumber

Output Parameters

XQuery	Net Variable
/Enroll/SubjectCode/text()	SubjectCode
/Enroll/StudentNumber/text()	StudentNumber

Fig. 29. Established task parameters

A brief introduction to XQuery:

- XQuery is a separate language that describes how to manipulate a given XML document to extract data from it. For example, the XQuery “/data/myTaskVariable/number() + 5" is saying that there is some XMLSchema element in an engine XML document with an element called myTaskVariable whose content is a number, and that the resultant XQuery expression is that element's value with 5 added to it.

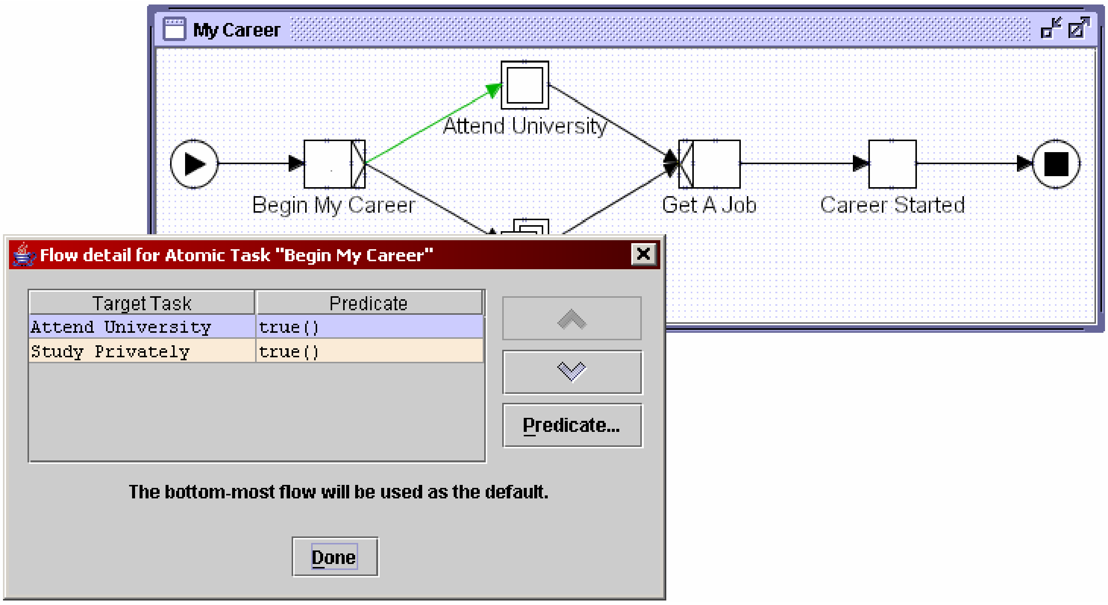
Updating Flow Detail

When dealing with tasks that have XOR and OR splits, we need some way of telling a running engine which flows should be activated. By updating the flow detail, you can specify the default flow path and also establish predicates for each flow direction. Flows whose predicates evaluate to true in a running engine will execute those flows.

To update the flow detail of a task that has a split:

1. Right click on the task and choose **Update Flow Detail…**.

   The number of splits will be the number of Target Tasks. The currently selected flow in the dialog will also be identified by being drawn green in the Net. **Fig. 30.**



Target Task	Predicate
Attend University	true()
Study Privately	true()

The bottom-most flow will be used as the default.

Fig. 30. Updating the Flow Detail

2. To choose the flow ordering, select the Target Task that you want as the change the order of, use the Up and Down arrows on the right to position the flow. The bottom-most flow will be used as the default if some flow needs to be followed, but no predicates evaluate to true.

3. To specify a predicate for a particular flow, select the flow from the list and click on **Predicate…**

   Enter a predicate as a boolean XQuery expression and choose **Done**.

End of Scenario

This is where our scenario ends. The My Career Scenario was designed to explain all the functions of the YAWL Editor and to provide you with a rudimentary understanding of designing a YAWL workflow specification.

But the complexity of the YAWL Editor does not stop with the current scenario. If you are after something more challenging, try adjusting your version of the scenario to expand into more sub-Nets and more complex situations.

Set YAWL Engine Detail

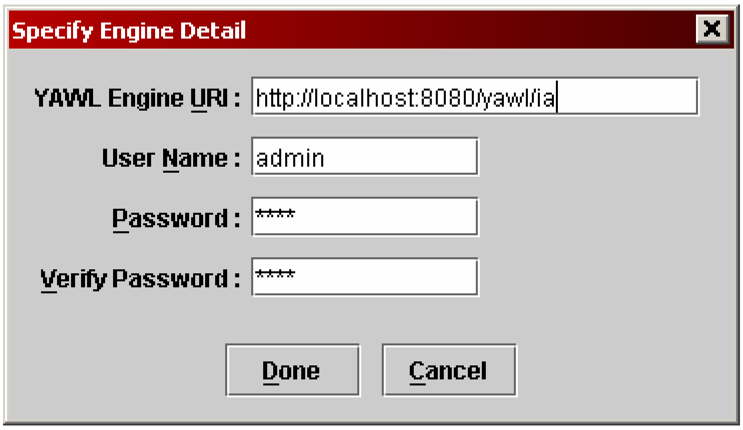
By specifying the location of a running instance of the YAWL Engine, it is possible to have the editor generate specifications that take advantages of that YAWL engine's specific setup.

For this release of the editor, the engine behaviour available is limited to a number of web services that the engine has been configured to interact with. Further detail on specifying web services to have the workflow interact with can be found in the next section.

To set the YAWL Engine details:

1. Click **Engine** on the Menu and choose the **Set Engine Detail…** item**.**
2. From the resulting dialog (**Fig. 31.**), accept the default values or enter the following engine details:
   - YAWL Engine URI
   - User Name
   - Password
   - Verify Password

The YAWL Engine URI value will default to a locally installed YAWL Engine administration URI. The User Name and Passwords will default to the engine default "admin" user.



YAWL Engine URI : http://localhost:8080/yawl/ia

User Name : admin

Password : ****

Verify Password : ****

Fig. 31. Specifying the YAWL Engine Details

Connecting a Decomposition to a Web Service

You can use task decompositions within your workflow to make a connection to a Web Service, when activated through the YAWL Engine.

For example, a decomposition may be set up to place an order with an external company. Upon execution of any task using this decomposition, data could be transmitted via a Web Service to this company.

To a decomposition at a Web Service, do the following:

1. Right-click on the task and choose **Task Decomposition Detail…**. An Update Task Decomposition dialog box will appear. (see the "Web Services Detail" box of **Fig. 24.).**

2. Enter the following details regarding the Web Service:

   - **WSDL URL** – the location of the Web Service WSDL specification.
   - **WSDL Operation** – the operating command for the Web Service
   - **YAWL Service** – the Service registered with the YAWL Engine

3. Click **Done** to finish.

If a valid running YAWL Engine instance can be connected to via the detail supplied in the section above, the YAWL Service Dropdown Box will contain entries for all the Web Services that Engine can interact with. Otherwise, only the default (manual) "Worklist" YAWL service will be available.

Setting Multiple Instance Queries

Multiple Instance Tasks need extra detail for data manipulation. The "Queries" tab, available by right-clicking on a multiple-instance task and selecting **Set Instance Detail…**. The "Queries" tab allows you to manipulate multiple instance data for the task.

Right click on the task and choose. Ensure that you are viewing the "Queries" tab of the dialog, as per **Fig. 32.**

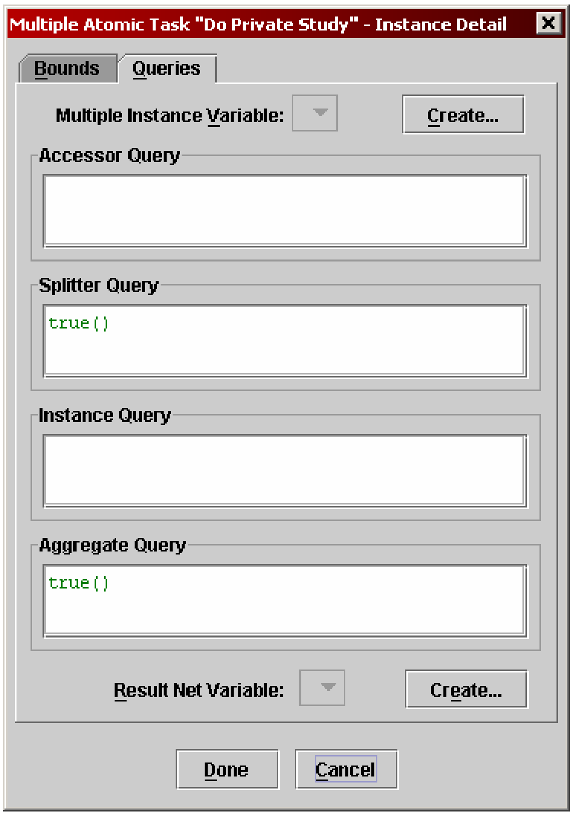


Fig. 32. Data Manipulation for Multiple Instance Tasks

To successfully specify multiple instance data manipulation, you first need to select which of the task's decomposition variables will be used as the "Multiple Instance Variable". This variable needs to contain (or can derive) a number of unique values that will be separated with the Splitter query to pass a unique value per each instance task. If the "Multiple Instance Variable" needs manipulation overall before the unique values are split out of it, use the Accessor Query to do so.

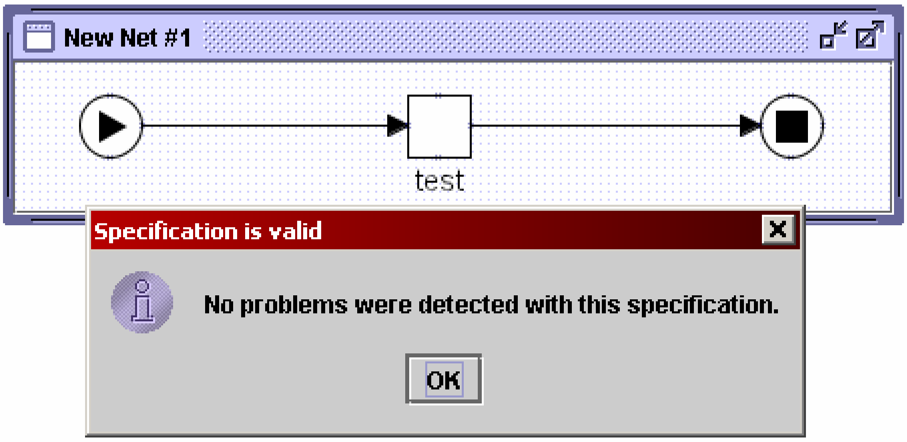
On the completion of an instance, the "Instance Query" can be used to transform the XML document returned to a form suitable for the Aggregate query to finally generate an overall result. The overall result will be assigned to the "Result Net Variable" on completion of the multiple instance task.

Validating and Exporting

At any stage you can validate or export your specification to a YAWL Engine executable XML file.

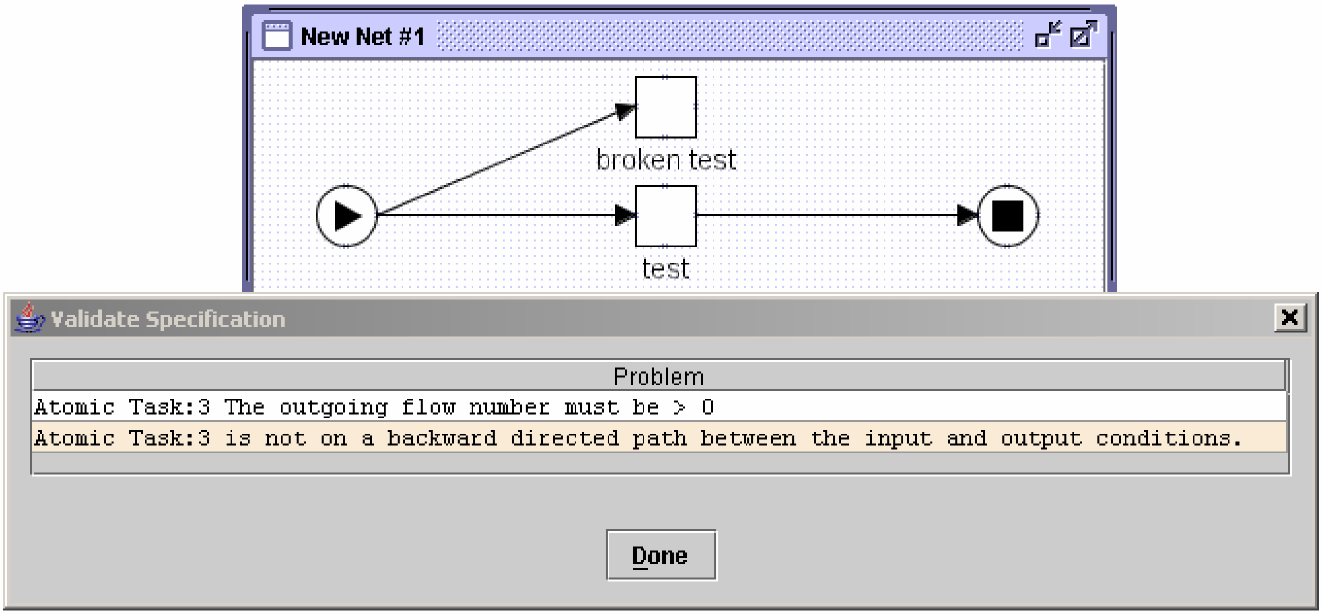
To Validate your specification:

1. Click on Validate this Specification button, ✓, on the Menu Toolbar or click **Specification** on the Menu and choose **Validate Specification…**.
2. A Validation Specification window will appear with details of any inconsistencies in the specification or you will get a window confirming that your specification is valid. **Fig. 33.** shows an example valid specification. **Fig. 34.** shows an example invalid specification.



No problems were detected with this specification.

Fig. 33. A Valid Specification



Problem
Atomic Task:3 The outgoing flow number must be > 0
Atomic Task:3 is not on a backward directed path between the input and output conditions.

Fig. 34. An Invalid Specification

To Export your specification to an XML file:

1. Click on "Export this specification to the YAWL Engine file format" button, , on the Menu Toolbar or click **Specification** on the Menu and choose **Export to YAWL Engine File…**.

2. A window will appear asking you wear to save the file.

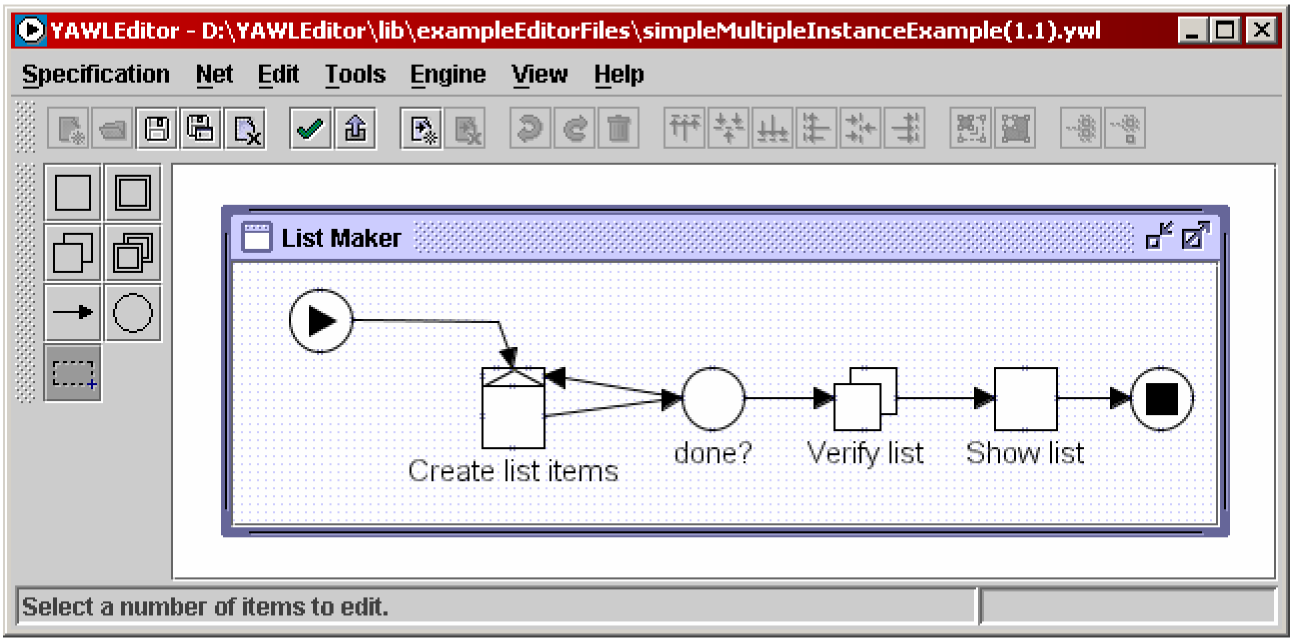
3. Select a location and choose **Save**.

This specification save file can now be loaded into a running YAWL Engine and executed.

Illustrative Examples

Multiple Instance Task Example

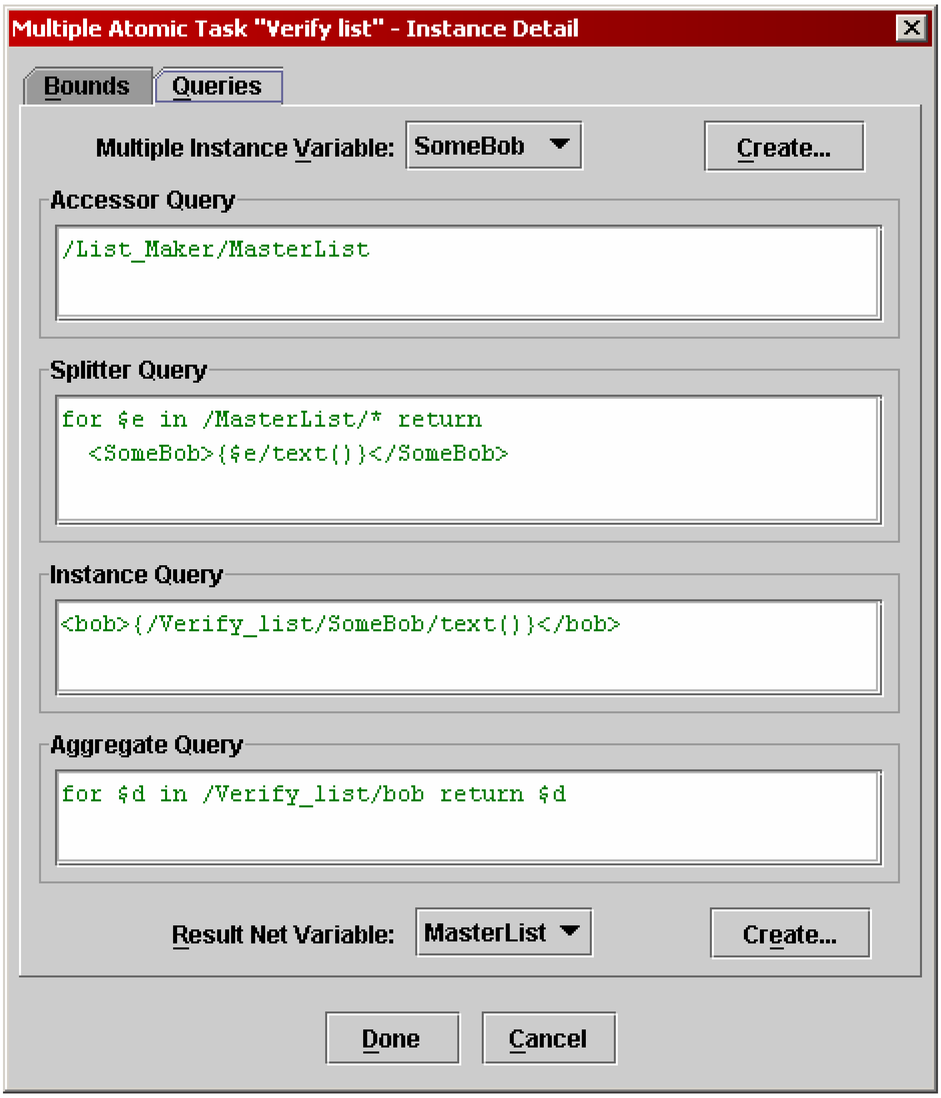
The example save file below, supplied with the editor source, gives you an example of how to process a simple multiple instance task, with its queries filled in (**Fig. 35**).



Select a number of items to edit.

Fig. 36 - A simple specification showing how multiple instance tasks are done.

Fig. 37 shows the core multiple instance queries used to manipulate a XMLSchema element defined for this specification. The element has a complex type of "BobList" which defines a sequence of "<bob>someText<\bob>" elements.



Multiple Instance Variable: SomeBob

Accessor Query

```
/List_Maker/MasterList
```

Splitter Query

```
for $e in /MasterList/* return
  <SomeBob>{$e/text()}</SomeBob>
```

Instance Query

```
<bob>{/Verify_list/SomeBob/text()}</bob>
```

Aggregate Query

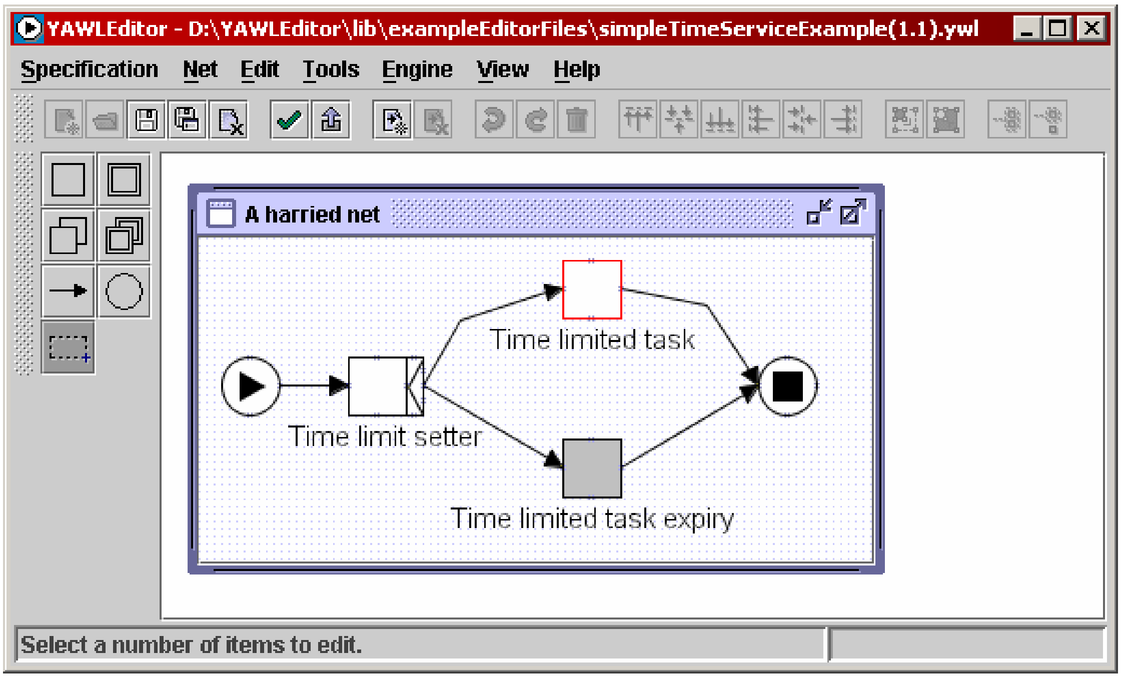
```
for $d in /Verify_list/bob return $d
```

Result Net Variable: MasterList

Fig. 38 – Example multiple instance queries

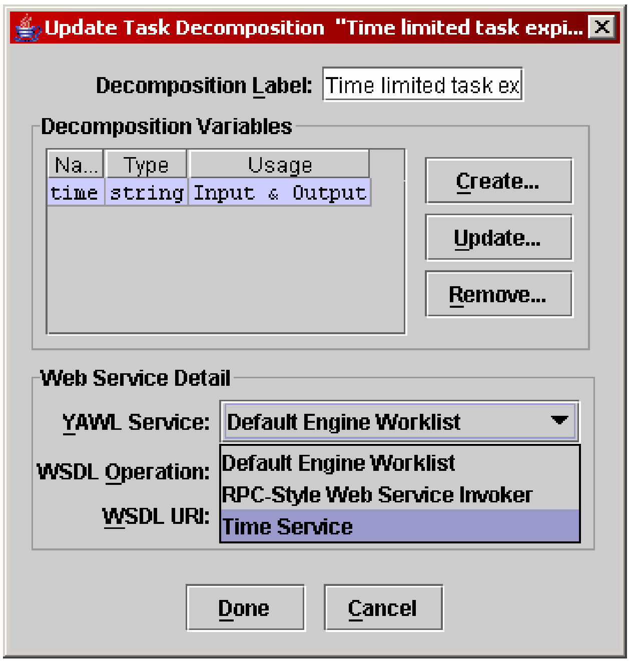
Time Service Example

The example save file below, supplied with the editor source, gives you an example of how to build a task that calls through to the optional timer service supplied with the engine (**Fig. 39**). **Fig. 40** shows a user selecting the time service currently registered with a running YAWL engine as they generate their specification.


YAWLEditor - D:\YAWLEditor\lib\exampleEditorFiles\simpleTimeServiceExample(1.1).ywl

Select a number of items to edit.

Fig. 41 - Example showing the usage of a custom YAWL service


Update Task Decomposition "Time limited task expi...

Decomposition Label: Time limited task ex

Decomposition Variables

Na...	Type	Usage
time	string	Input & Output

Web Service Detail

YAWL Service: Default Engine Worklist

WSDL Operation:

WSDL URI:

Fig. 42 - Assigning a registered YAWL service from a running engine

Known Issues

We need to find a way of getting the clipboard to notify listeners of updates. Currently, we kludge up whether the “Paste” action can be activated by simply turning it on when an editor user uses “Cut” or “Copy” in the same editor. Editor elements cannot currently be cut/paste across two running editors on the same machine.

The creation and deletion of nets need to be undoable edits. Until the deletion of nets becomes an undoable edit, deleting a net will cause odd undo behaviour with respect to composite tasks that used to reference the deleted net.

Having spaces in the file path of the editor will cause the XMLSchema validation to stop working. Validation uses temporary files that are getting confused by the spaces.

The menu item "Print Specification" freezes if not used for some time.

For variable name fields and decomposition label fields that drive the enabled state of their respective "Done" buttons, clicking on the "Done" button when these fields still have focus will have the first button press event ignored.

Cutting and pasting net elements involved in a currently viewed cancellation set will have the newly pasted elements rendered as if they were still in a cancellation set.

Currently, renaming a variable, net or task decomposition does not also change XQuery/XPath expressions that reference it.

Troubleshooting

The YAWL Editor is not responding to my mouse clicks and it is beeping every time I click the mouse.

Check to see if you have any YAWL Editor dialog windows open. These windows don’t appear in the Windows Taskbar. To check if you have any open dialog windows, hold down the ALT key and press the TAB key. With the ALT key held down, press TAB until you reach the dialog window and let go of the ALT key. Close the dialog window and you should be able use the YAWL Editor again.

Dialog windows appear as the Java icon (blue coffee cup).

I can't connect two elements with a flow. Why?

If the editor does not allow you to connect one element to another, it is steering you away from building an invalid net. Typical examples include:

- Trying to connect a second flow to an undecorated task.
- Trying to point an incoming flow to a split decorator (or an outgoing flow from a join decorator).

Se the troubleshooting entry "How do I find out more about elements and principles....." for more detail.

When I validate my Net, I get the following validation message, 'The net (. . .) may complete without any generated work. Should all atomic tasks in the net be unlabelled?'

This message appears if you have not labelled any of your tasks. To label a task (and thus ensure that the engine will get a user to handle the task at this point in the workflow), right click on the task and choose **Select Task Decomposition…**. Choose **Create…** and in the following window, "Update Task Decomposition", and enter the name of your task under Decomposition Label, then click **Done**.

To label a Condition, right-click on a task and choose **Set Label…**, then type in a name for the label and click **Done**.

My specification won't validate without any errors and I don't know what to do.

First, check the logic of your specification and exercise every branch of your workflow for correctness.

If you are sure of your workflow, consult the YAWL website for the proper use of the YAWL elements:

http://www.citi.qut.edu.au/yawl.

How do I find out more about the elements and principles of the YAWL workflow specifications?

For more information about the mechanics of the YAWL workflow elements, please consult the YAWL website:

http://www.citi.qut.edu.au/yawl.

Acknowledgements

Queensland University of Technology

Dr Marlon Dumas

Lindsay Bradford

Lachlan Aldred