



An Introduction to **WEASEL**

Web sErvice for Analyzing queueing networkS with multiple soLvers

And

PMIF Editor User Manual

by

Samuel Zallocco

Copyright © by

Zallocco.NET

Weasel:

The main goal of the WEASEL tool is to offer a Web Service that solves queueing network models specified in PMIF format, using multiple external solution tools. Actually the WEASEL tool provide support for QNAP, SHARPE, PDQ, PEPSY, MQNA1, MQNA2, PMVA, MVA-QFP, OPENQN and CLOSEDQN tools, but it provide a simple way to add support for other solution tools. The WEASEL tool follow a simple client-server approach to accomplish this scope, based on the standard XML based web service protocols SOAP and WSDL. The WEASEL tool provide a four level stack oriented architecture, where each level can use or not the result of the previous one. Clients application are keep free to use or not some functional level in order enhance the Web Service flexibility. This functional levels are shown in Figure 1 and the functions are described below

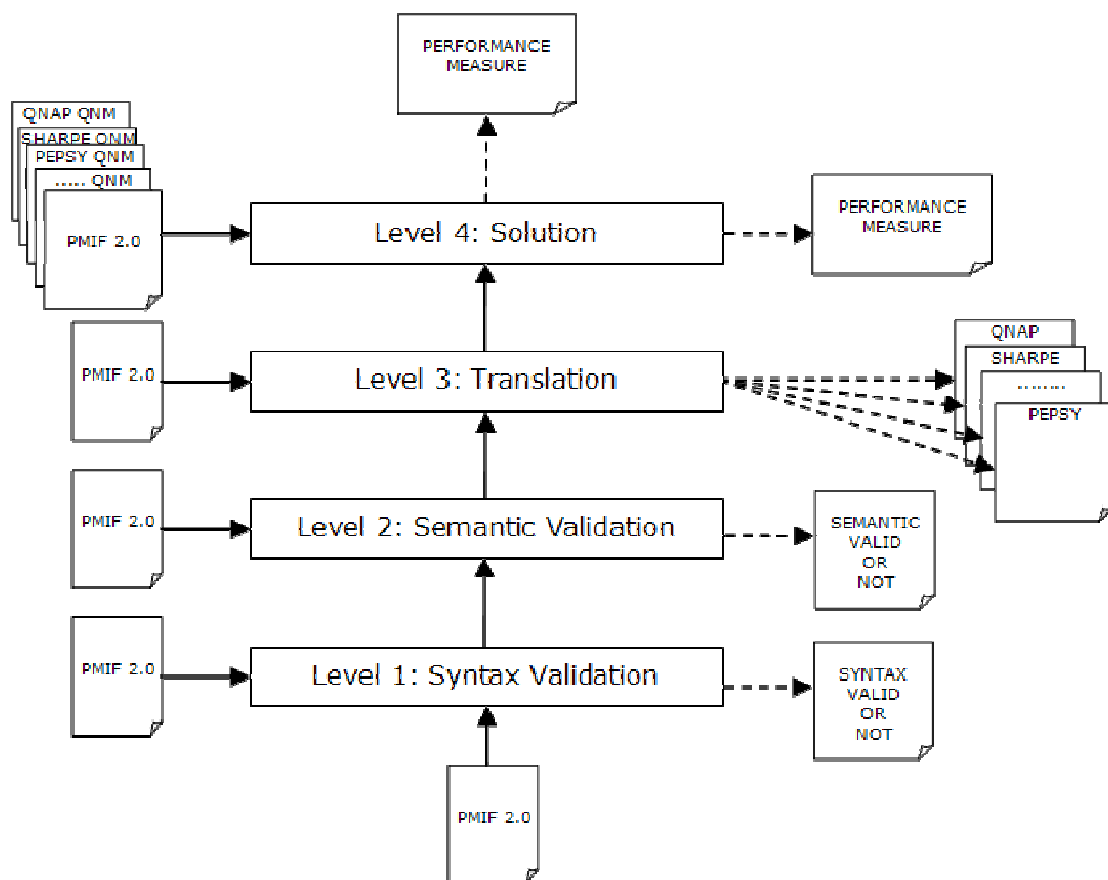


Figure 1: Weasel web service functional levels

- Level 1: PMIF 2.0 syntax validation against official PMIF 2.0 Schema definition.
- Level 2: PMIF 2.0 semantic validation against “PMIF Semantic Analyzer” external tool.
- Level 3: Translation of the PMIF 2.0 model into a file in one of the integrated tools format, using an XSLT transformation.
- Level 4: Solution tool execution on the PMIF 2.0 translated model or on the tool native language model.

We have preferred to leave the clients application free to select the functional level to use without the limit of a sequential invocation of the four levels. This, in some measure, follow the Web Service philosophy of providing

“simple” services that can be combined to perform a more complex task. This approach allow client users to decide if use our web service and which functionality of it. For example, some user may decide to use our web service only for the solution of a QN model, but to use a third party web service for the model semantic or syntax validation. The performance measure are presented to the client as a text file in the original output format of the selected tool. Actually the web service do not provide a method to format the tools output in XML, but these feature can be implemented as an extension in a future version. In order to implement our Web Service we have chosen the Apache, PHP 5, PEAR::SOAP trilogy. This approach offer a free of charge implementation that can be widely implemented in a various range of HW and SW architecture. Apache is a secure and efficient HTTP server available in the Microsoft™ Windows©, Mac™ OS/X© and Linux operating systems. In order to provide a standard access method to our web service we have to implement the SOAP protocol and we have to provide a standard interface to the web service methods via a WSDL file. The simplest way to do that is to using the PHP5 scripting language with the PEAR::SOAP extension. Using PHP5 will simplify our life providing us the support to DOM (Document Object Model) and XSLT (eXtensible Stylesheet Language for Transformations) extension used respectively for the PMIF 2.0 syntax validation and for PMIF 2.0 to tool models translation. The PEAR (PHP Extension and Application Repository) SOAP extension is a PHP5 module that provide support for the SOAP protocol and for the automatic generation of the WSDL file.

The PHP5 implementation of the Web Service consist essentially in a PHP5 class named QNSolver (see Figure 2) with the following public method:

- **Copyright:** this method have no parameter. The returned string can be customized by the Web Service mantainer and can contain the following information: web service name and version, guest operating system, PHP version, web service physical location, mantainer name and e-mail address.
- **GetToolsList:** like the previous one this method needs no parameters. It returns an XML string containing the description of the integrated tools. This method can be used by clients application to build a runtime menu to be able to use multiple web service that integrate different solvers. Information provided by this method are: tool name, description, author, supported client operation (translation, solution, native tool language solution) and supported solution methods and algorithms.
- **ValidateSyntax:** this method receive an input string with a QN model and is type (actually can be only “PMIF 2.0”) and returns an output string containing an XML response with the value of the syntax validation (True or False). The validation is made against the PMIF 2.0 XML Schema provided by www.perfeng.com/pmif/pmifschema.xsd.
- **ValidateSemantic:** this method receive an input string with a QN model and a string with is type (actually can be only “PMIF 2.0”) and returns an output string containing an XML response with the value of the semantic validation (True or False). The validation is made against the PMIF Semantic Analyzer external tool of D. G. Cousillas and C. M. Lladó.
- **Transform:** this method receive an input string containing a QN model, a string with is type (actually can be only “PMIF 2.0”), a string containing the target tool ID, an optional string containing the solution

method (for those tools that support more than one solution algorithms) and one more optional string containing some extra params needed by some tools (for example number of jobs).

- **Solve:** this method receive an input string containing a QN model, a string with is type (here the type can be any of the integrated tools ID), a string containing the target tool ID (if the type of model is different from “PMIF 2.0” then type and tool ID must be equal), an optional string containing the solution method (for those tools that support more than one solution algorithms) and one more optional string containing some extra params needed by some tools (for example number of jobs).
- **GetModelDescription:** this method receive an input string with QN model and an input string with is type (actually can be only “PMIF 2.0”) and return an output string containing an english human readable description of the model.

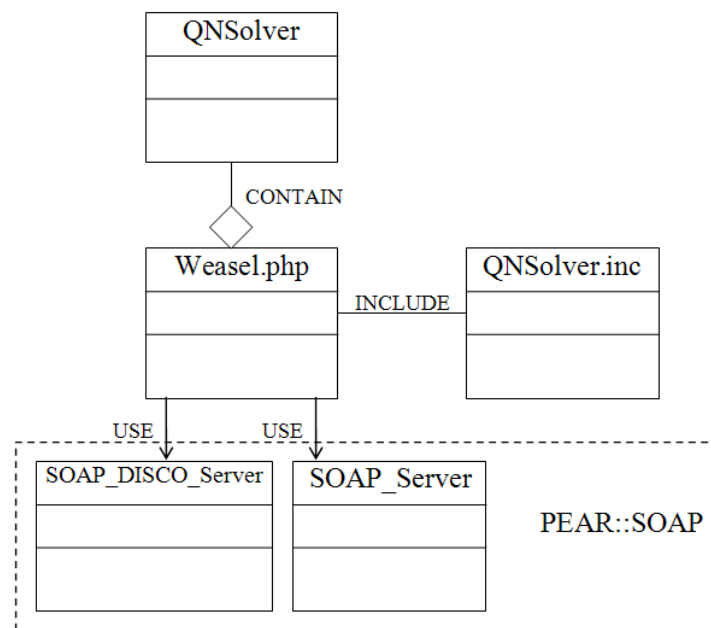


Figure 2: WEASEL web service UML Class Diagram

The WEASEL tool is fully customizable and extensible, we have split the configurable part of the web service in an external file “QNSolver.inc” where user can add new tool and change the web service behaviour.

End user is free to develop his own web service client and implement it with any programming language that provide standard SOAP support. In order to test our web service we have developed three clients application. The first was been written in PHP5 and have a simple command line interface. The second was a web based application using the emerging AJAX (Asynchronous Javascript And Xml) technology. The third was a real Microsoft™ Windows© application written in Borland™ Delphi© 2005 Architect Edition, that we have named “PMIF Editor” due to the fact that integrate a graphical PMIF 2.0 visual editor (see Figure 3) able to assist the analysts from the visual design phase of the queueing networks, to the translation of the graphic model in PMIF 2.0 and finally to the resolution invoking the remote web service.

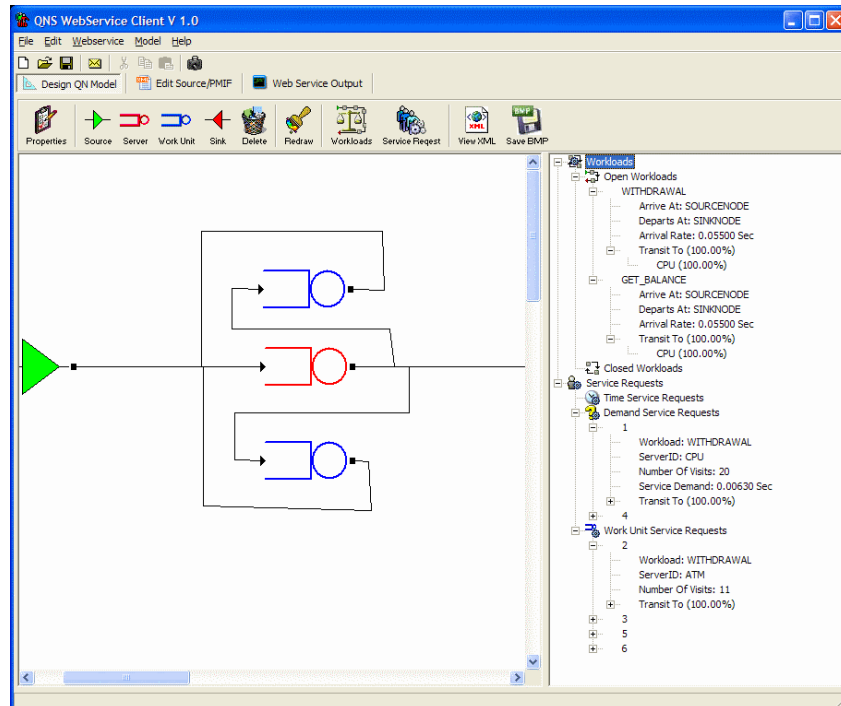


Figure 3: The Weasel Client and PMIF Editor

Given a standard PMIF 2.0 queuing network model, the WEASEL tool can actually translate the model into the following tools/languages:

Tool ID	XSLConversion File	Supported QN Type	Guest OS
PDQ	PMIF2_to_PDQ.xsl	Open, Closed	Windows, Linux
QNAP	PMIF2_to_QNAP.xsl	Open, Closed, Mixed?	Windows, Linux?
SHARPE	PMIF2_to_SHARPE_PFQN.xsl	Closed	Windows, Linux
OPENQN	PMIF2_to_OPENQN.xsl	Open	Windows
CLOSEDQN	PMIF2_to_CLOSEDQN.xsl	Closed	Windows
MVAQFP	PMIF2_to_MVAQFP.xsl	Closed	Windows, Linux
PMVA	PMIF2_to_PMVA.xsl	Closed	Windows
PEPSY	PMIF2_to_PEPSY.xsl	Open, Closed, Mixed?	Linux
MQNA1	PMIF2_to_MQNA1.xsl	Open, Closed	Windows
MQNA2	PMIF2_to_MQNA2.xsl	Open, Closed, Mixed, Finite Capacity Queue with Blocking and Lost	Windows

PMIF Editor:

The PMIF Editor is a visual application running under the Microsoft Windows Operating System. The idea is those to define an application that will allow user to define a graphical queueing networks model using the typical PMIF 2.0 queueing network theory formalism (Source Node, Server, Work Unit Server, Sink Node and Arc), with the choice to define Closed or Open Workload and related service requests (Work Unit Service Requests, Demand Service Requests and Time Service Requests).

After the model has been defined, user can ask to the application to convert it in the PMIF 2.0 format and than can send the model to a remote WEASEL web service to perform a conversion, a syntax validation, a semantic validation or the solution via one of the web service supported external tools.

The application may be downloaded from Zallocco.NET (<http://www.zallocco.net>) or from the University of L'Aquila SeaLab Quality Group Web Site (<http://sealabtools.di.univaq.it>). The downloaded package is a Microsoft Windows MSI installer that will create a folder named "Zallocco.NET" in the Windows Programs directory containing a subfolder named "PMIF Editor" with two file inside:



The first is a Win32 application and the second is a start-up configuration file with the following content:

```
[WebService]
LastURL=http://sealabtools.di.univaq.it/SeaLab/Weasel/Queueing_Network_Solver_Service.php?wsdl
URLCount=1
URL0=http://sealabtools.di.univaq.it/SeaLab/Weasel/Queueing_Network_Solver_Service.php?wsdl
```


Due to some server consolidation and reorganization the reported web service URL is no more valid. At the time this text was written there are two valid URL where you can find a functional web service that are:

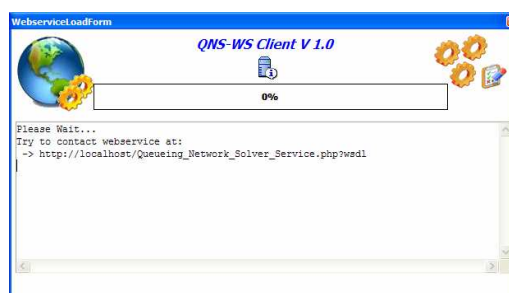
- http://weaselwin.univaq.it/Queueing_Network_Solver_Service.php?wsdl
- http://weaselinux.univaq.it/Queueing_Network_Solver_Service.php?wsdl

To check the availability of an updated web service URL list visiting Zallocco.NET or the SeaLab Quality Group Web Sites.

Application start-up:



To start the application simply double click on the  icon present on your desktop. The application will present a splash screen like this:



In this phase the application try to contact the latest valid web service used showing a progress bar with some message in the message notification area.

A typical message you can see in the message notification area look like this:

```
Please Wait...
Try to contact web service at:
-> http://localhost/Queueing_Network_Solver_Service.php?wsdl
-----
Queueing Network Solver Web service (QNS-WS)
WS Version : 1.0
Guest OS   : WINNT
PHP Version: 5.1.1
Location  : University of L'Aquila - Science Faculty
Contact   : Samuel Zallocco <samuel.zallocco@univaq.it>
Created by Samuel Zallocco.
(CC) Some right reserved.
-----
Web service seems to be active...:
Retrieving tools...
Web service integrate 7 tools
-----
Tool ID: PDQ
Version: 3.0
Description: Pretty Damn Quick Solver
Author: Prof. Neil J. Gunther
Option: SAVE
Method: EXACT
Method: APPROX
Method: CANON
Option: SOLVE
Method: EXACT
Method: APPROX
Method: CANON
Option: SOLVE AS IS
...
-----
Done. Web service exploration completed.
```

As you can see, during this phase the application call the web service invoking the Copyright_ and GetToolsList_ method. If all goes ok, in the message notification area you can find a list of integrated tools and the option you can use with them:

- **SAVE**: You can translate a PMIF 2.0 model in the native language of the integrated tool;
- **SOLVE**: You can use the remote tool to solve the model;
- **SOLVE AS IS**: You can send a native tool language model to the web service for the solution.

While if something goes wrong or if the web service is not reachable you will see the following text in the message notification area:

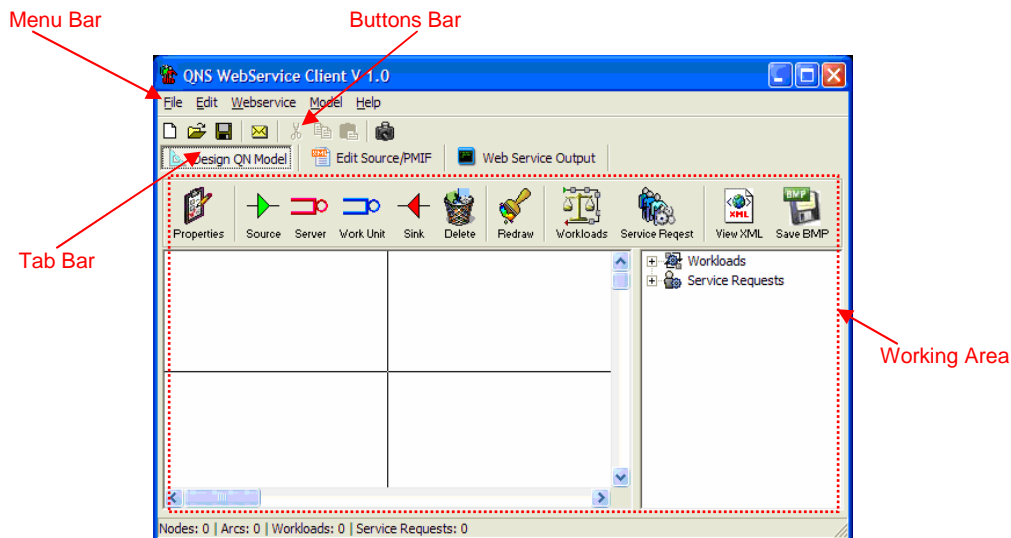
```
Please Wait...
Try to contact webservice at:
-> http://sealabtools.di.univaq.it/SeaLab/Weasel/Queueing_Network_Solver_Service.php?wsdl
Error connecting to webservice at:
-> http://sealabtools.di.univaq.it/SeaLab/Weasel/Queueing_Network_Solver_Service.php?wsdl
Please verify webservice URL and retry.
Sorry!! The webservice return an empty tools list!!
-----
Done. Webservice exploration completed.
```

But don't worry the application will work fine also in off-line mode!

Click on the button having the label “Click here to continue” to start working:

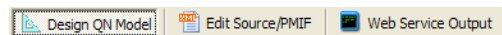


When started, the application will appear like in figure:



You can identify four main areas, some of these are typical of Windows Applications (Menu Bar, Buttons Bar), the other are application specific.

With the Tab Bar you have access to the three section of the application that are:



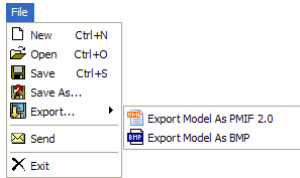
- **Design QN Model (design section):** Here you can visually define the queueing network model;
- **Edit Source/PMIF (edit section):** Here you can create, edit or modify the queueing network model in PMIF 2.0 or in some other language;
- **Web Service Output (output section):** This area contain the output of the web service invocation.

This three section match with the three phases of a WEASEL/PMIF 2.0 queueing network model design, that are:

- Paint the network;
- Review the automatically generated PMIF/XML Code;
- Query the remote Web Service for model resolution or other services (validation, translation, etc).

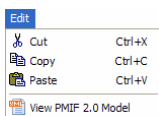
PMIF Editor Menu structure:

Get a look at the menu option present in each application menu:

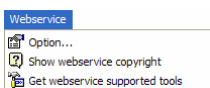


Note that the option present in the File Menu are “section sensitive” and have different behavior referring to the current section selected in the Tab Bar.

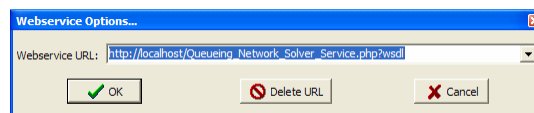
- **New:** As the name hint, with this option you can start designing a new queueing network model, discarding all the previous loaded model (both in design and edit sections) or clearing the web service output area.
- **Open:** Allow to load a graphic model (if used in design section) or to load a PMIF 2.0/XML source (if used in edit section). Note that PMIF Editor use an extended version of PMIF 2.0 to store graphic model that with modesty I've named PMIF 3.0. PMIF Editor make some check to verify if the loaded model are PMIF 3.0 compliant before load it and if not so, the model will not be loaded. While there are no check to be made when you load a model directly in the edit section.
- **Save:** Save the current model (graphical or textual) or the web service output depending on the tab bar section active;
- **Save As...:** Allow to save a model (graphical or textual) with a different name;
- **Export...:** Allow to save a graphical model in PMIF 2.0 or as a BMP image;
- **Send:** Allow to send the PMIF 2.0 model via e-mail;
- **Exit:** Close the application.



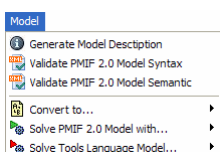
- **Cut, Copy and Paste:** Are the typical windows functions, but work only in the edit and output sections;
- **View PMIF 2.0 Model:** if you are in design section then the graphic model will be converted in PMIF 2.0 and the visualization will switch to the edit section.



- **Option:** Allow to set the web service URL. You also have an history of all previously used URL's;



- **Show webservice copyright:** Invoke the remote Copyright_ web service method and the visualization will automatically switch to the output section containing the web service output;
- **Get webservice supported tools:** Invoke the remote GetToolsList_ web service method and the visualization will automatically switch to the output section containing the web service output.

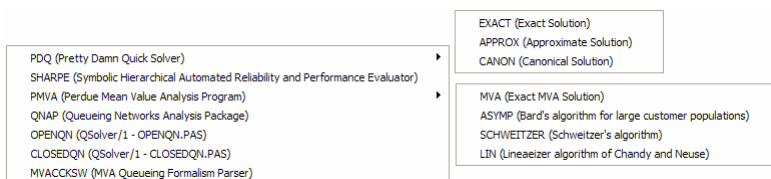


- **Generate Model Description:** This option invoke the remote GetModelDescription_(PMIF_MODEL) web service method, providing it the PMIF 2.0 model present in the text area of the edit section as argument of the invocation. The result of invocation is presented in the output section and it is an English description of the model in natural language;

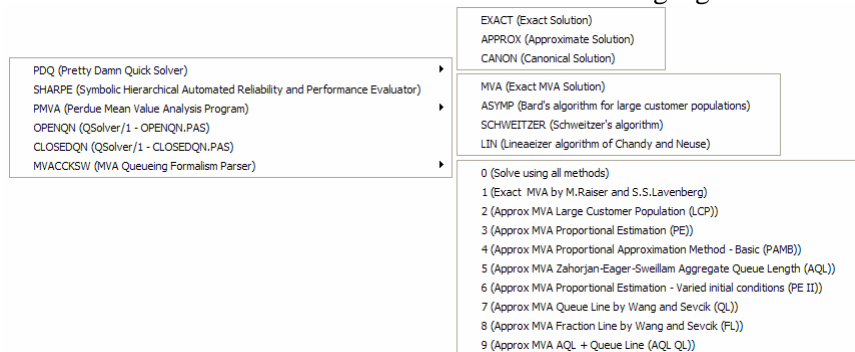
- **Validate PMIF 2.0 Model Syntax:** Invoke the remote ValidateSyntax_(PMIF_ MODEL) web service method, providing as input the PMIF model present in the edit section. The result of invocation is an XML file in the output section with the result of the syntax validation process;
- **Validate PMIF 2.0 Model Semantic:** Invoke the remote ValidateSemantic_(PMIF_ MODEL) web service method, providing as input the PMIF model present in the edit section. The result of invocation is an XML file in the output section with the result of the syntax and semantic validation process;
- **Convert to...:** Invoke the remote Transform_(PMIF_ MODEL, TOOLID) web service method. The content of the related sub menu options varying depending on the web service supported tools and is building up during the application start up in the initial phase of web service exploration or when the web service URL was changed (Menu: Webservice→Option...). This method require two parameters that are the PMIF model present in the edit section and a tool ID that was automatically passed to the web service by the selected sub menu option chosen. The result of invocation is the same queueing network model passed to the web service but described in the language of the selected tool;
- **Solve PMIF 2.0 Model with...:** Allow to invoke the remote Solve_(PMIF_ MODEL, TOOLID) web service method. As the previous option the sub menu content varying according to the web service capabilities. The result of the invocation is the output of the solution of the model with the selected tool. At this time the tools outputs are those that the tools authors have implemented in their tools but in future version we will implement a common output format for all the integrated tools;
- **Solve Tools Language Model...:** Allow to invoke the remote Solve_(MODEL,TOOLID) web service method. As the previous two options the sub menu content varying according to the web service capabilities. The model passed to the web service is that present in the edit section. But this time the model must be describer in the specific tool language and not in PMIF.

View an example of how the sub menus of the last three option can appear:

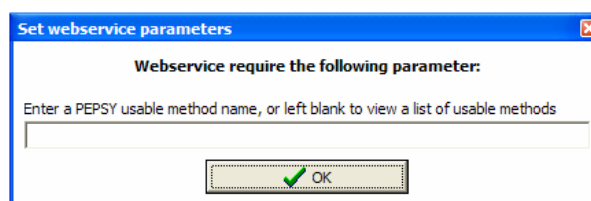
“Convert to...” sub menus example:



“Solve PMIF 2.0 Model with...” and “Solve Tools Language Model...” sub menus example:



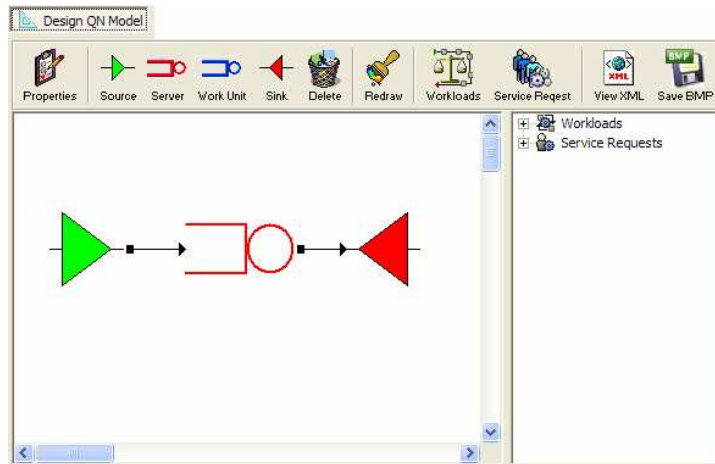
Some tools need extra parameters to work. In this case application will ask to input this extra parameters with a dialog window like those in the following figure:



As you can see there are also a brief description of the requested parameter.












Queueing networks graphic design:

The first working area, the graphic one, present itself like in the following figure:

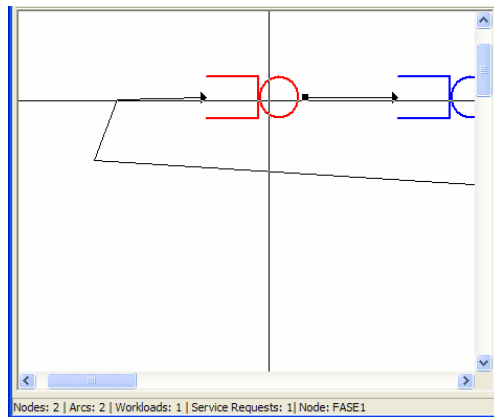


As you can see there are three main visual area: in the top there was an icon bar, in the bottom left centre there was the design area and on the bottom right side there was an area where you can define the non graphic properties of the queueing network model (Workloads and Service Requests).

Icons in the icon bar are arranged by functionalities. Their meaning is the following:

- 
Properties Allow to define the model name and description.
- 
Source Allow to define the source node of an open queueing network model.
- 
Server Allow to define a server node.
- 
Work Unit Allow to define a work unit server.
- 
Sink Allow to define the sink node of an open queueing network model.
- 
Delete Allow to delete a node or arc of the graphic model.
- 
Redraw Useful when you have deleted several node to remove artefact from the design area. The entire queueing network was repainted and visual artefact removed.
- 
Workloads Allow to define a new workload.
- 
Service Request Allow to define a new service request.
- 
View XML Convert the queueing network in XML/PMIF and the visualization will be automatically switched to the source edit section.
- 
Save BMP Take a snapshot of the graphic model and allow to export it in BMP image file format.

The queuing network design area:



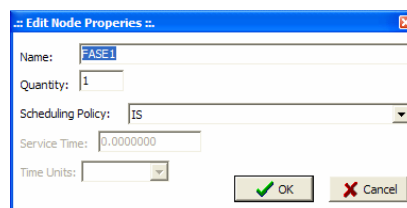
have horizontal and vertical ruler that follow the mouse movement to help to position the graphic elements and arcs. When you hold down the [CTRL] key the ruler disappear.

To position one of the four available node type:

- Source Node,
- Server Node,
- Work Unit Server Node,
- Sink Node,

simply click on the associated button on the icon bar and then click with the left mouse key in the position you want to draw it.

When a node was placed in the drawing area you can edit it's properties double clicking on it. It will appear a dialog box that, depending on the selected node type, will allow to change some node parameters:

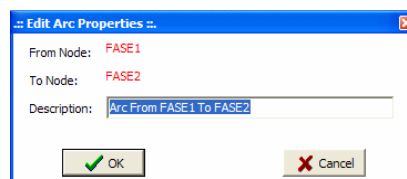


When all nodes are placed you can connect them with an arc using the [CTRL] key. Positioning the mouse over the source node and clicking on it holding down the [CTRL] key it will appear an arc; at this point, always holding down the [CTRL] key, it is possible to define the path the arc should do and when you finally make the last click on the destination node, it will appear the desired arc connecting source to destination node.


If you release the [CTRL] key before you have reached the destination node the arc will disappear and should be

necessary a refresh using the Redraw icon  to remove some residual invalid arc artefact.


The same procedure used to edit the node properties is valid also for arcs. Making a double click on an arc it will appear a dialog window that allow editing of some arc properties like arc description. Please pay attention to the fact that Pmif Editor provide an automatic auto-refreshing (related to nodes name) description for the arcs, but if you provide you personal arc description then if you change the nodes name than you have to manually update the arc description too:

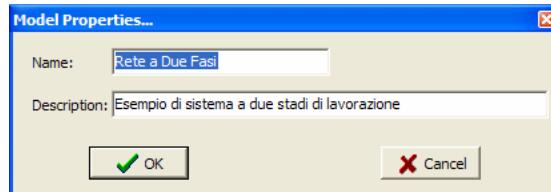




To delete a node or arc you have first to click first on the Delete icon  and then you have to click with mouse over the item you want to delete. If you delete a node, then all the connected arc are deleted too.



To name a model and to provide a brief description you can use the Properties icon  and a dialog box will appear with two available field for entering model name and model description as in the following figure:

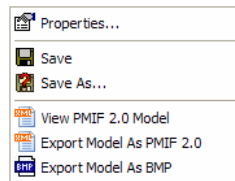


On the bottom of the application you can note a status bar reporting some useful information like: the total number of defined nodes, arcs, workload and service request. When you position the mouse pointer over a node or arc in the right most position of the status bar it will appear its name:

Nodes: 2 | Arcs: 2 | Workloads: 1 | Service Requests: 1 | Node: FASE1

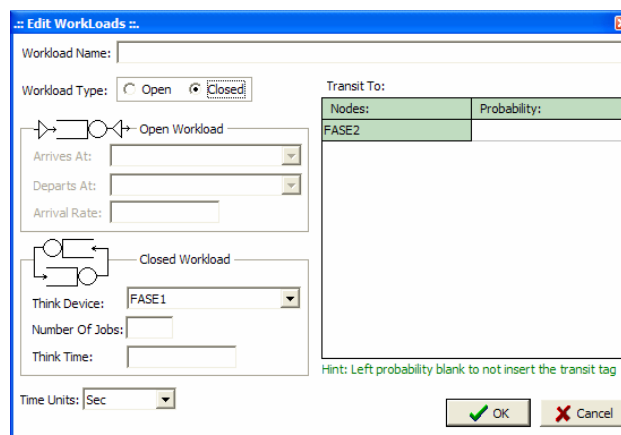
Node name (or arc description) are also displayed using standard Microsoft Windows “tool tip”, by positioning the mouse pointer over an arc or node and waiting for few seconds without making a click.

Clicking with the right mouse key in an empty point of the design area it will appear the following context menu:



that allow quick access to some common use function, like model properties, model saving, PMIF or BMP model export.

When the network with nodes and arcs is fully defined, it is possible to proceed defining Workloads and Service Requests by clicking on the related icons. Workload definitions take place with the following dialog window:



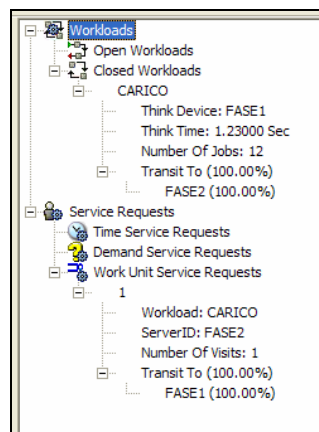
As you can see from the image above, the first thing to do is to define a name for the new workload, then we can choose the workload type (Open or Closed). According to the previous selection some control in the window may be active or not.

Nodes selection take place using a combo box. It is not possible to choose every defined nodes but only that keep to some constrain (see “Performance Model Interchange Format: Semantic Validation Technical Report”, May 2006, D. García, C. M. Lladó, C. U. Smith, R. Puigjaner.)

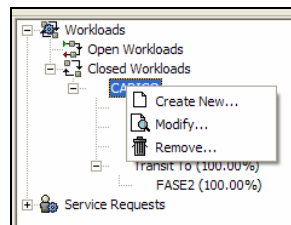
On the right side of the dialog window there was an area dedicated to transit probability definition (in percentage using % symbol or as number between 0 and 1).

Time units is a choice to do with care. Actually Pmif Editor do not provide security check against different time units used to define different workloads.

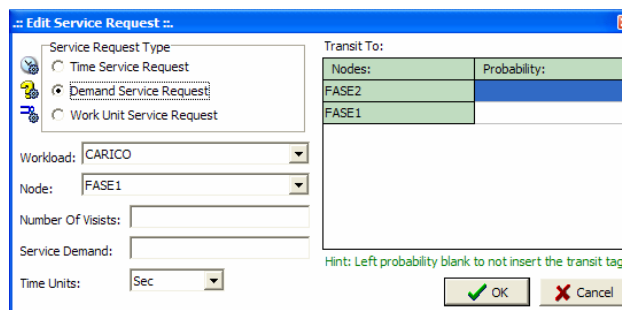
Clicking on the [OK] button the workload ware confirmed and it will appear under the “Workloads” tree of the working area in the right side of the application:



To modify or delete a workload, you have to select it's name in the workload tree and then using the right mouse key it will appear a context menu with Create New, Modify and Remove options, so simply click on the option you need:

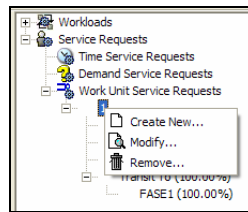


Procedure to define service request is similar to the previous one, but dialog window vary a bit!



PMIF 2.0 standard do not specify name convention for service request nor require it should be explicitly defined, so in the application development we choose to assign progressive numeration to defined service requests.

So, to modify or delete a service request select it's number and, using right mouse key, activate the context menu and follow the same procedure used for Workloads:



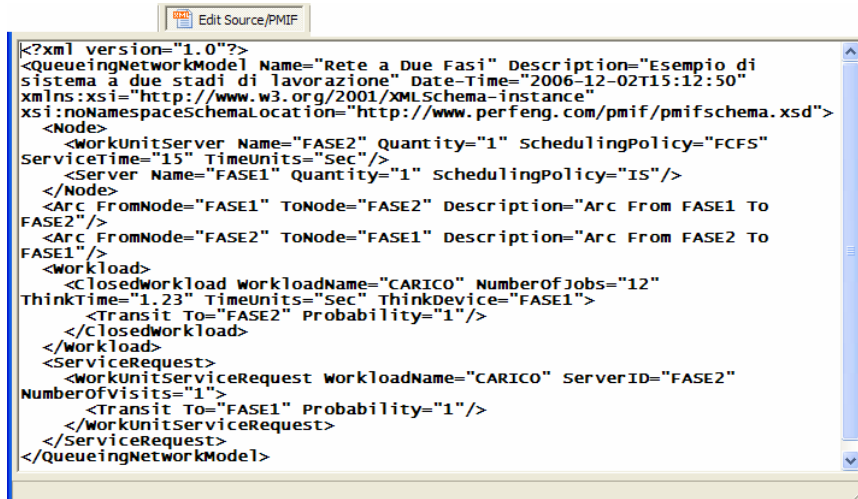
Now the model can be saved and reloaded using the classic FILE menu common in all Windows application.

XML/PMIF 2.0 source editing:



When the graphic design phase is completed, using the  icon you can transform the graphic queueing network model directly in PMIF 2.0 source code and automatically switch to the editing section.

Editing section present it self like a classic and simple text editor:



```

<?xml version="1.0"?>
<QueueingNetworkModel Name="Rete a Due Fasi" Description="Esempio di
sistema a due stadi di lavorazione" Date-Time="2006-12-02T15:12:50"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.perfeng.com/pmif/pmifschema.xsd">
  <Node>
    <WorkunitServer Name="FASE2" Quantity="1" SchedulingPolicy="FCFS"
ServiceTime="15" TimeUnits="sec"/>
    <Server Name="FASE1" Quantity="1" SchedulingPolicy="IS"/>
  </Node>
  <Arc FromNode="FASE1" ToNode="FASE2" Description="Arc From FASE1 To
FASE2"/>
  <Arc FromNode="FASE2" ToNode="FASE1" Description="Arc From FASE2 To
FASE1"/>
  <Workload>
    <Closedworkload workloadName="CARICO" NumberOfJobs="12"
ThinkTime="1.23" TimeUnits="Sec" ThinkDevice="FASE1">
      <Transit To="FASE2" Probability="1"/>
    </Closedworkload>
  </Workload>
  <ServiceRequest>
    <workunitServiceRequest workloadName="CARICO" ServerID="FASE2"
NumberOfVisits="1">
      <Transit To="FASE1" Probability="1"/>
    </workunitServiceRequest>
  </ServiceRequest>
</QueueingNetworkModel>
  
```

It is possible to edit, modify, delete, copy, cut and paste text like if you are using the classic Windows Notepad application.

This section is totally independent from previous one, in the meaning that if you own a hand made PMIF 2.0 model it can be loaded with the FILE menu and it is possible to modify, save or solve the model sending it to the remote webservice, without using the visual editing feature of PMIF Editor.

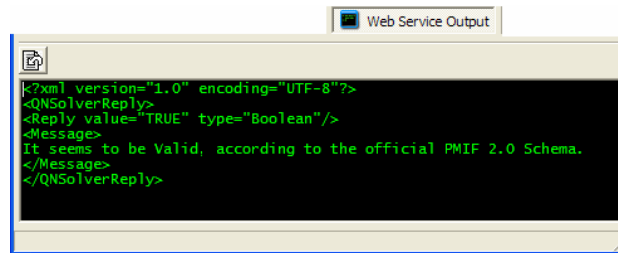
This application section is intended for expert user, all who are familiar to PMIF 2.0 syntax and semantic and that prefer to have full control on generated code and that prefer to manually fine tune the source code of the model.

Web Service Invocation:

When the queueing network model was been translated in XML/PMIF 2.0 and ready in the editing section editor, it is possible to switch to the webservice output section to observe the result of the remote tool invocation.

If the editor in the editing section is empty then you will receive an empty model error from the webservice invocation.

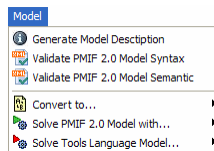
This section contain only a not editable text box that will show the remote webservice reply:



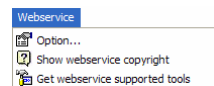
```


<?xml version="1.0" encoding="UTF-8"?>
<QNSolverReply>
  <Reply value="TRUE" type="Boolean"/>
  <Message>
    It seems to be valid, according to the official PMIF 2.0 Schema.
  </Message>
</QNSolverReply>
  
```

You will be automatically positioned in the webservice output section on every webservice invocation made using the “Model” menu:



Or when you click the “Show webservice copyright” and “Get webservice supported tools” list from the “Webservice” menu:



The only key present in this section is  that can be useful to copy the webservice output in the editor of the editing section. This feature can be useful when you request a translation of the PMIF 2.0 model in some other formats to fine tune the tool model specific source code before invoke the webservice sending, this time, not the PMIF model but the tool specific language model using the “Solve Tools Language Model...” option of the “Model” menu .

Selecting an option of the Webservice and Model menu it will appear the output of the webservice method in the output section. As previously seen, outputs are in XML format except those of model resolution and model conversion that are in textual format.

To save the output of conversion or resolution you can use the “Save” or “Save As...” option fo the “File” menu.

For more information on weasel please visit:

- <http://www.zallocco.net>
- <http://sealabtools.di.univaq.it>
- <http://weaselwin.univaq.it>
- <http://weaselinux.univaq.it>

Or contact:

Dr. Samuel Zallocco

Area Gestione Laboratori Scientifici e Didattici ad Indirizzo Informatico del Polo di Coppito

Facoltà di Scienze MM. FF. NN.

Università degli Studi dell'Aquila

Via Vetoio snc, 67010 COPPITO (AQ) – ITALY

email: samuel.zallocco@univaq.it , samuel@zallocco.net

<http://www.zallocco.net/>

Vittorio Cortellessa, Ph.D.

Dipartimento di Informatica

Facoltà di Scienze MM. FF. NN.

Università degli Studi dell'Aquila

Via Vetoio snc, 67010 Coppito (AQ) - ITALY

ph. (+39) 0862433165

email : cortelle@di.univaq.it

<http://www.di.univaq.it/~cortelle/>