Integrated Performance Modeling Environment (IPME): User Guide

A tool for simulating operator workload and performance

DLR ID IB-112-2014/5
Classification C/II

Institut für Flugführung
Director: Prof. Dr. Dirk Kügler

Document Control Information

Responsible Project Manager:	Dr.Norbert Fürstenau
Responsible Author:	Ghazal Mahmoudzadeh Vaziri, B.A.(Physik)
Additional Authors	Dr.Norbert Fürstenau
Project / research Task:	X-EPRAS
Classification:	C/II (im Institut für Flugführung unbegrenzt zugänglich)
Filename:	\\bsflsda\HMI-Labor\Intern\RAiCe\IPME
Version:	01
Save Date	2014-02-11
Total Pages:	86

Release Note:

Document is released in accordance to Document Control Sheet:

Change Log

Release	Date	Changed Pages or Chapters	Comments
0.01	15.10.2013		final version G.M.
0.02	31.1.2014	all	Full text review N.F.

Inhalt

1 Goal of the Document 8
2 IPME Overview 9
2.1 IPME MODES 9
2.1.1 IPME (VACP and W/Index) 10
2.1.2 POPIP 10
3 Getting started 11
4 IPME Interface 12
4.1 File menu 13
4.2 Edit menu 15
4.3 System menu 16
4.3.1 Execution Settings Parameters 18
4.4 NetModel menu 29
4.5 CrewModel menu 31
4.6 PSFModel menu 33
4.7 Environment Model menu 34
4.8 Measurement suite menu 35
4.9 Tools menu 36
4.10 Windows menu 37
4.11 Help menu 38
5 IPME System model 39
5.1 Task Network Model 39
5.1.1 Network Diagram 40
5.1.2 Snapshots 40
5.1.3 Scenario event 42
5.1.4 Function 43
5.1.5 Variables 45
5.2 Crew Models 46
5.3 Environment Models 48
5.4 Performance Shaping Function Models 49
5.5 Measurement Suite 51
5.6 External Models 51
6 Task 51
6.1 Main Tab 51
6.1.1 Timing and Effects Subtab 51
6.1.2 Repeating/Continuous Subtab 53
6.1.3 Failure Subtab 54
6.2 Description Tab 55
6.2.1 VACP and W/Index Subtab 55
6.2.2 Function Allocation Subtab 59
6.2.3 Task criticality 59
6.2.4 Report 60
6.2.5 Required Traits Subtab 60
6.3 POPIP Tab 61
6.3.1 Scheduling subtab 61
6.3.2 Workload Demands subtab 63
6.4 Display Properties 64

7 Queue 65
8 Decision 67
9 Operator 68
10 Example 70
10.1 Definition of the variables 70
10.2 Definition of the Operators 70
10.3 Network Diagram 70
10.4 Decision Type 71
10.5 Queue 73
10.6 Tasks 75
10.7 VACP and Workload chart 80
11 Appendix 84
11.1 Abbrevations 84
11.2 References 84
11.3 List of Figures 84
11.4 List of Tables 85

Zusammenfassung

Im Rahmen der WdV Studie „XEPRAS = extreme events predictor for aeronautical systems“ 2011-2013 wurde das Tool IPME [1] (= Integrated Performance Modeling Environment, Alion Science / Boulder/Co.) im Hinblick auf seinen Nutzen für die Modellierung und Analyse von Operateurverhalten beim Design neuartiger Mensch-Maschine-Systeme (z.B. Remote Tower Arbeitsplatz mit unterschiedlichem Automatisierungsgrad) untersucht. Hierfür wurde vom Hersteller eine Demoversion von IPME v. 4.5 zur Verfügung gestellt. Das Werkzeug basiert auf der Multiple Resources Theorie (z.B. Wickens 2000 [7]) und der Perceptual Control – Information Processing/Time Pressure Theorie ([2][3][8]). Der vorliegende Bericht ist eine Benutzer-Anleitung auf der Basis des sehr umfangreichen und detaillierten Handbuchs. In Form von Beschreibungen und Bedienanweisungen für sämtliche Menüs wird mittels screenshots der entsprechenden Bildschirme ein praktischer Überblick über die Struktur, Möglichkeiten und Handhabung des Werkzeugs gegeben.

1 Goal of the Document

This report is a manual to be used as a practical guide for initial use of the “Integrated Performance Modeling Envirenment (IPME)” of Alion Science, Boulder/Co. It does not replace the detailed IPME user guide. Instead it is intended as a quick reference to support the beginner by means of a short description of all relevant displays of the tol, supported by figures of the respective displays of menus, tabs and input fields. The descriptions are based on an IPME demo version (4.5) available for free download.

2 IPME Overview

IPME is an integrated modeling environment that consists of an environment model, a crew model, a task network of processes or procedures, a performance shaping function model, and optional external models. These component models are configured into a composite system model.
The task network modeling system is based on the Micro Saint Human Operator Simulator, developed as a commercial product by MA&D. The task network model allows a user to easily describe the processes used by a human operator to perform an activity. A library of micro model functions is used to calculate various types of human performance times for general classes of activities including cognitive, perceptual, and psychomotor activities. Micro model functions can be used to define human task performance times for micro actions such as walking, reading, or typing.
IPME can run in two different modes: IPME and POPIP (Prediction of Operator Performance Information Processing/Perceptual Control Theory) modes. The POPIP and IPME modes address workload with different algorithms. Both modes use a model based on Performance Shaping Functions (PSF), which globally adjust operator performance based on environmental factors and a taxonomy mapping between performance shaping functions and individual tasks. The IP/PCT component of POPIP mode uses the Information Processing/Perceptual Control Theory Model[1]. This algorithm adjusts operator performance based on task scheduling demands.

Results can be obtained from IPME through a measurement suite. A measurement suite is where you define the design of an experiment to run using state and object variables that form the system model. You can specify blocking variables, independent variables, and dependent variables. Data collection can be triggered by the end of the run, a specific simulation time, a task, or a queue. Data can be saved in a format to be used in other data analysis or statistical systems, also including MS-EXCEL. In the Execution Settings, you set the number of runs, a random number seed, and indicate whether you want workload and critical path data collected.
Interposed communication is provided by IPME through the use of Internet domain sockets. Communication takes place between endpoints known as sockets. Each socket can exchange information with other sockets of an appropriate type within the domain. Other sockets can use this protocol to provide data for simulations. IPME is also capable of communicating to other High Level Architecture (HLA) compliant applications.
The present report is based on a demo-version of IPME version 4.5.

2.1 IPME MODES

IPME can run in two different modes: IPME and POPIP. Features of each mode include the following.

• IPME – uses PSFs to modify operator performance. VACP and W/Index data are collected.

• POPIP – combines a workload algorithm with IPME based input, Central, and output, demands, and task scheduling demands. VACP and W/Index data are also collected.

2.1.1 IPME (VACP and W/Index)

IPME uses PSFs to modify operator performance. VACP (visual, auditive, cognitive, psychomotor) and W/Index data are collected.

The attentional demand algorithm measures the task loading for an operator within four separate channels and accounts for the demands on human resources. With this algorithm, each task is rated with regard to the weighted task demand specific requirements within the four independent channels: Visual, Auditory, Cognitive, and Psychomotor (VACP).

The format of these ratings was developed originally for an LHX mission function analysis performed by Aldrich and others [4], for the US Army Research Institute. The scales provide a subjective rating for various levels of attentional demand.

Additional work was later published by Bierbaum, Aldrich and others [5][6] providing enhanced descriptors and interval scale values. An additional descriptor was included entitled Visually Aided: Read in the Night Vision Goggles (NVG) scale by BAE Systems-Canada (formerly the Canadian Marconi Company).

The channels used for this model include visual perception, auditory perception, verbal cognition, spatial cognition, manual response, and speech response. Ratings are based on ordinal categories (integers from 1 to 7).

2.1.2 POPIP

POPIP (prediction of operator performance / information processing) combines a workload algorithm with IPME based on input.

Because POP and IP/PCT already existed in IPME 3.x, the two were combined to achieve a workload scheduling method which capitalises on the strengths of each scheduler while also recognising their similarities. The way that IP/PCT recognises and quantifies time pressure fills a void in the POP theory. POP recognises that time pressure is important, but has not yet quantified it. Also, the structural interference that is represented in IP/PCT has been captured to a greater level of detail than the structural interference representation implemented in the POP scheduler. IP/PCT allows operator tasks to be interrupted, delayed, and shed. POP allows tasks to be interrupted and delayed, but not shed. The inclusion of shed tasks increases the realism of the task management strategy. In POPIP, modellers can now designate tasks that can be shed due to memory size limitations or time pressure constraints, which the POP algorithm did not address. The goal of POPIP is to create a workload scheduler that addresses workload, time pressure, structural limitations, and task manipulation options (shed, delay, and interrupt) in one comprehensive workload model.

2.1.2.1 POP Contributions

POPIP incorporates most of POP. The Task Demand Modifiers (TDMs) have been retained, as have the operator workload overload and tasks interrupt and delay due to workload overload. For each task assigned to an operator, the POP inputs have been retained, with one minor exception. The user can enter Input, Central, and Output Demand Ratings, specify the sub-channel utilised (except for specific psychomotor designations), and categories each task as internally or externally paced. The specific psychomotor channel designation (such as left or right hand) now uses the more comprehensive structural interference implemented in the IP/PCT model.

2.1.2.2 IP/PCT Contributions

From IP/PCT, POPIP inherited some of its workload processing and scheduling capabilities. Users can enter task scheduling and priorities, as well as the consequences for shedding a task. POPIP includes a tab on the task information dialogue to enter these task components. The components available for selection are: visual (except for externally cued) including visual areas, peripheral or central focus required, the designated home area, and the visual subtense range limits; auditory, where the user selects whether the task is externally cued and selects from a difficulty category; and psychomotor (preferred and non-preferred), where the user details which psychomotor resources (for example, left or right hand) are used to complete the task. In POPIP, it is assumed that once a task is started with either the preferred channel or the non-preferred channel, it does not get reassigned unless it is restarted. Externally-cued visual detection events are not implemented in the POPIP scheduler. Because POPIP differs from POP by recognising shed tasks, the features from IP/PCT which address memory limits have been included. The IP/PCT Task Performance Modifier (TPM) of Forgetting is included.

The Preferred and Non-preferred psychomotor channel selections are used to determine whether two tasks have structural interference.

3 Getting started

The system specifications provided here indicate the minimum required to execute IPME. However, the recommended system specification will result in improved system performance. Complex or large models may require additionalsystem capabilities.

• Supported Operating Systems: RedHat Enterprise Linux 5.X, Microsoft Windows XP or Microsoft Windows 7

• Minimum 1GB free disk space; Recommended 1.5 GB free disk space The free space size does not include swap space for Linux or the virtual memory or paging file for Windows.

• Minimum 1GB RAM and 3GB combined RAM and swap space; Recommended 2GB RAM and 4GB combined RAM and swap space.

• CD-ROM drive

• Minimum CPU speed of 1.5 GHz; Recommended multi-core CPU speed of 1.8 GHz

The following are system requirements:

• For Microsoft Windows XP and Microsoft Windows 7: Microsoft .NET Runtime version 3.0 or newer

• For RedHat Enterprise Linux: Mono .NET Runtime version 1.2.5.1 or Newer

One of the following is required if the user wants to perform any database operations (such as saving or loading to a database). If flat file storage will be used exclusively, the following are optional.

•MySQL database (server, client, and shared libraries) version

3.23.X or 4.0.X (not compatible with newer versions).

•PostgreSQL 8.2.0

4 IPME Interface

Besides the usual PC window components (close box, title bar, scroll bars, etc.), a IPME window has other elements, as shown in the following figure.

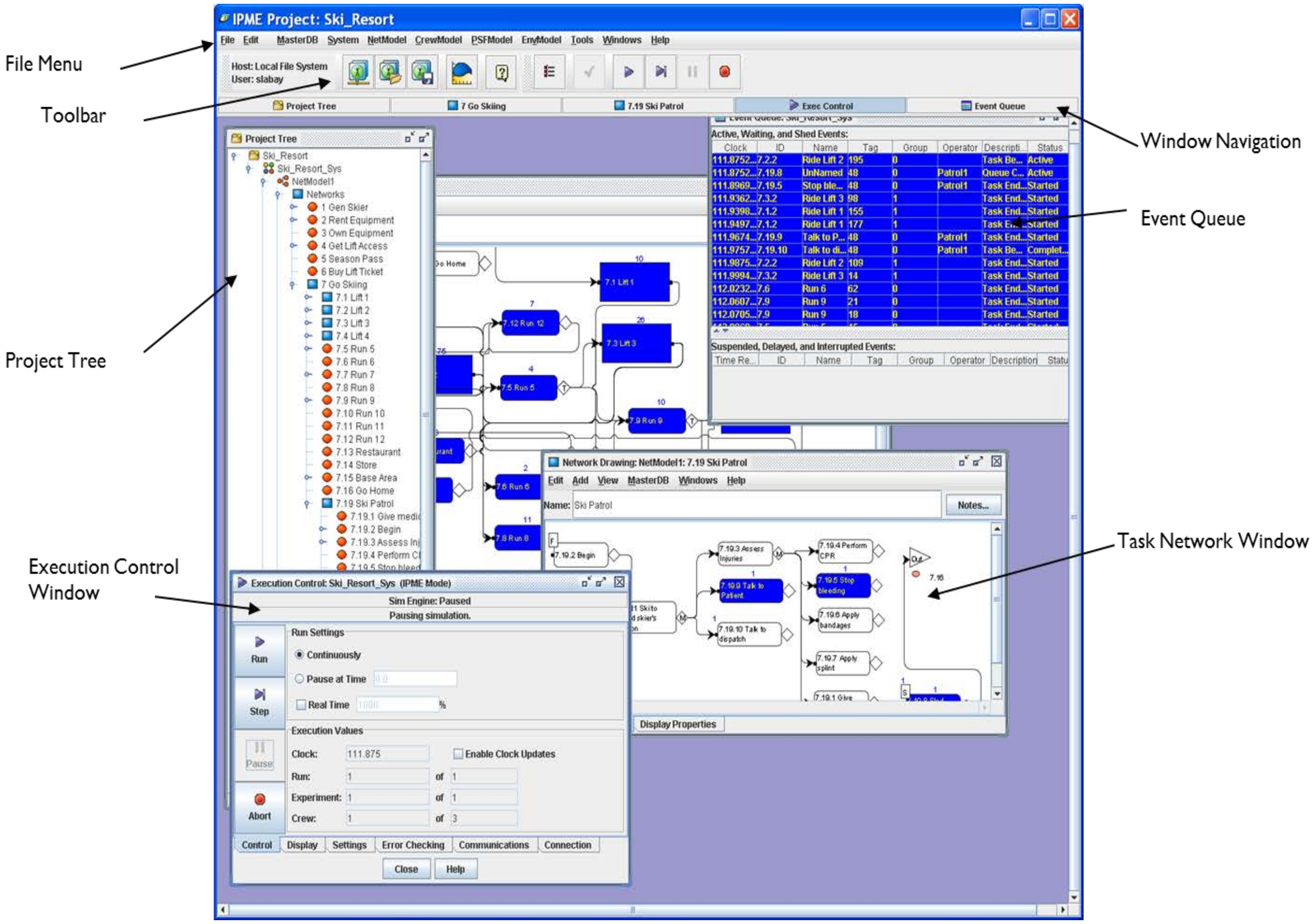


Figure 1: IPME-main window

Component	Functionality or Purpose of the Component
Menu Bar	Contains File ,Edit, system, NetModel, crewModel, PSFModel, EnvModel, Tools, Window and Help menus
Standard Toolbar	Contains icons for shortcuts to menu commands.
Project tree	Displays all current open Projects and four component models. They are the task network model, environment model, crew model, and performance shaping function model.
Execution Control Window	You can execute a simulation based only a task network (IPME mode only), or you can execute a simulation based on a system, which includes the task network model and the other component models. You are not required to have a measurement suite experiment defined in order to run the simulation.
Windows navigation	Contains all current open Project tree, Net Overview, Net Model, Execution Control
Event Queue	you can start the model from the System menu. As the model runs, variables can be displayed to monitor their values and the event queue can be displayed to watch events as they are waiting to execute. The trace file can also be enabled to record the time when each task begins and ends. These options help you to verify that the model is operating as intended and to make changes if it is not.
Task Network Window	Contains Network diagrams

Table 1: Components of main window

4.1 File menu

File menu functions also appear on the Standard Toolbar.

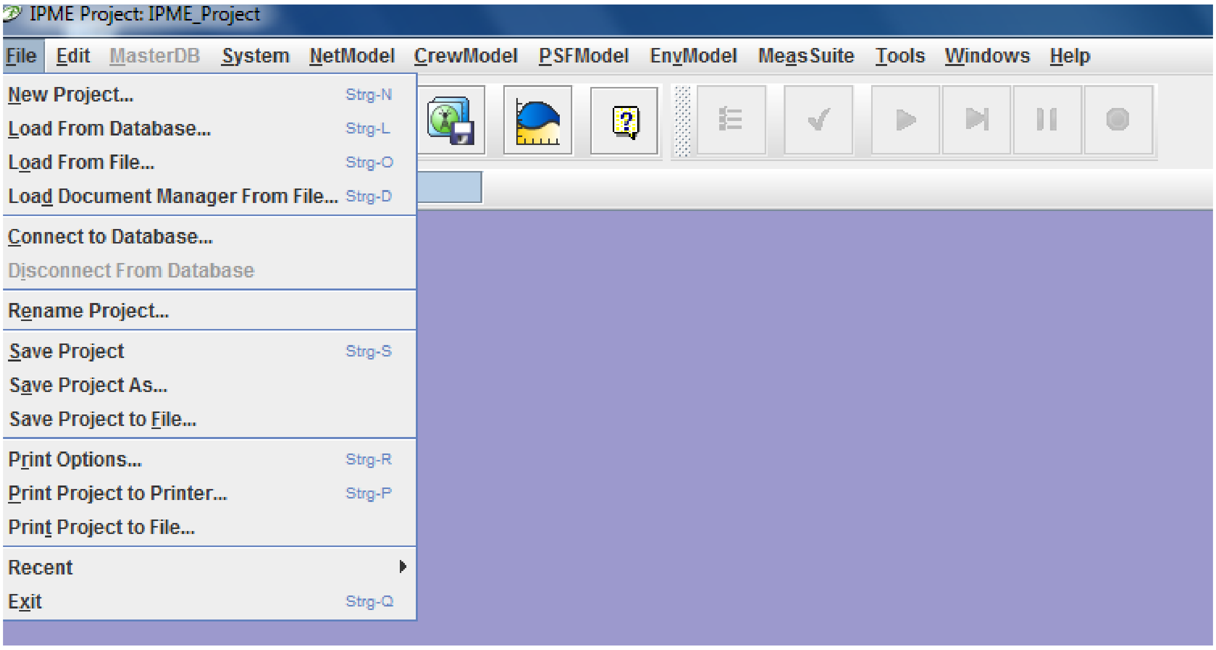


Figure 2: File menu

New Project	A project is a grouping of systems and component models that facilitates modelling. If multiple users are sharing a version of IPME, each user could have a separate project to keep the modelling efforts separate. When you add a new project to IPME, it replaces the currently open project. You can only have one project open at a time.
Load From Database 	The Selection of a Project, System, or Model to Load dialogue displays with a list of projects in the database. To expand a project and display its contents, click the project name or click the adjacent handle icon
Load from File	You can load projects, systems, and component models from a file. If you load a project, the currently loaded project is replaced with the new project. • Project files have the extension .prx. • System files have the extension .syx. • Crew model files have the extension .crx. • Task network model files have the extension .tnx. • Environment model files have the extension .enx. • Performance Shaping Function model files have the extension .psx.
Load Document Manager from File	You can load a Document Manager file with the file extension
Connecting to Database 	You can connect to any IPME database and load projects, systems, or models. Projects may be saved to a database. You must know the host name, user name, and password to successfully make a connection.
Disconnect From Database	With a connection established, select Disconnect from Database from the File menu.
Rename Project	You can rename projects if you have the appropriate permissions
Save Project 	Any project can be saved to a IPME database if a connection to the database has already been established. If no connection is established, you will be required to complete the connection prior to saving. You can save a project, system or component model to a file. The file is saved in XML format.
Save Project as	Saves by opening a window which gives the opportunity to change the file name, location or format.
Save project to File	If this project has been saved to a file previously, the project is automatically saved in the same location with the same name (Steps 2 and 3 are not required). The previously saved file will be overwritten. If this is the first time this project has been saved to a file, the Save Project to File dialogue displays for the selected project.
Print Option	Select this check box to show the Print Preview dialogue when printing to a file.
Print Project to Printer	You can print a project, system or component model to a printer. Before you Print the item, use the Print Options dialogue box to select printing options.

Print Project to File,	You can print a project, system or component model to a file. The file is saved in HTML format. Before you print the item, use the Print Options dialogue box to select printing options.
Recent	displays a list of files and documents that you most recently used.
Exit	Closes Program

Table 2: File menu

4.2 Edit menu

In the Edit menu, the following commands are available:

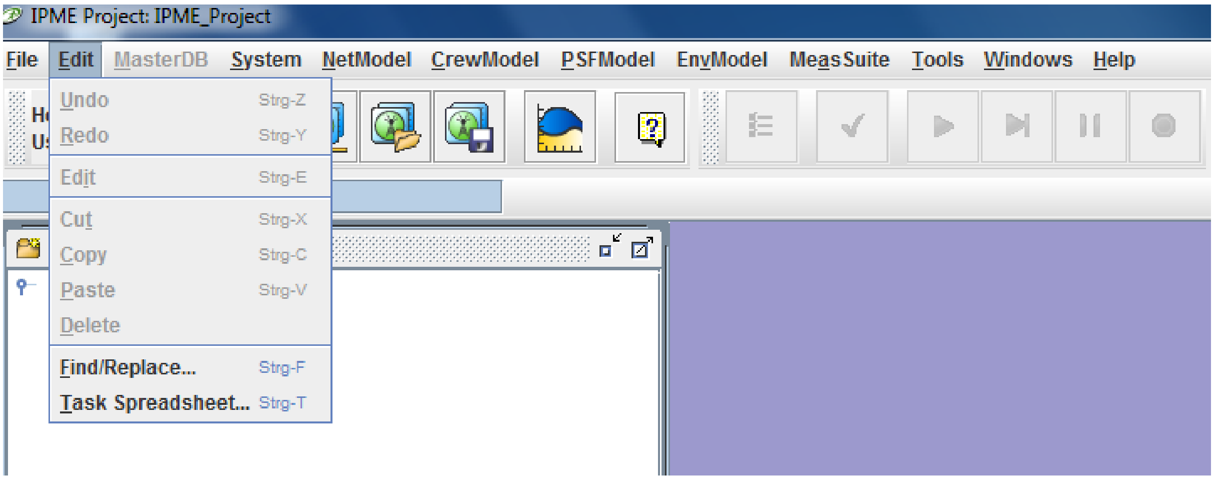


Figure 3: Edit menu

Undo	The actual entry of this item will depend on what you did last. In my example I had typed, so that was displayed. This selection can be repeated several times.
Redo	After an action has been undone, it can be reinstated in the document.
Edit Task	To edit a task
Cut	Removes the selection from the active document and places it on the clipboard
Copy	Copies the selection to the clipboard
Paste	Inserts the contents of the clipboard at the insertion point (cursor) or whatever is selected.
Delete	To delete a task
Find/Replace	Find - Searches for specified text in the active document. Replace - Searches for and replaces specified text and formatting.
Task Spreadsheed	The task spreadsheet viewer allows you to view and edit network and task information for all tasks and networks within a project from the same dialogue. This view is useful for copying and pasting the same information between tasks or networks. It also allows for comparison of task information. Each row contains information for a task or network contained in the project, with one column dedicated for each data field. Changes made in the spreadsheet viewer are reflected in the model. Each object's id displays at the far left of the row. Objects are indented according to their hierarchy in the project tree. Icons also display to indicate whether the row belongs to a task or a network.

Table 3: Edit menu

4.3 System menu

A system can consist of any combination of a task network model, crew model, environment model, performance shaping function model, measurement suite, and an external communications model. A task network model is required to execute the system. A system is an association of these individual component models. Each system also has a system check list to assist the analyst in tracking the progress of a modelling effort.

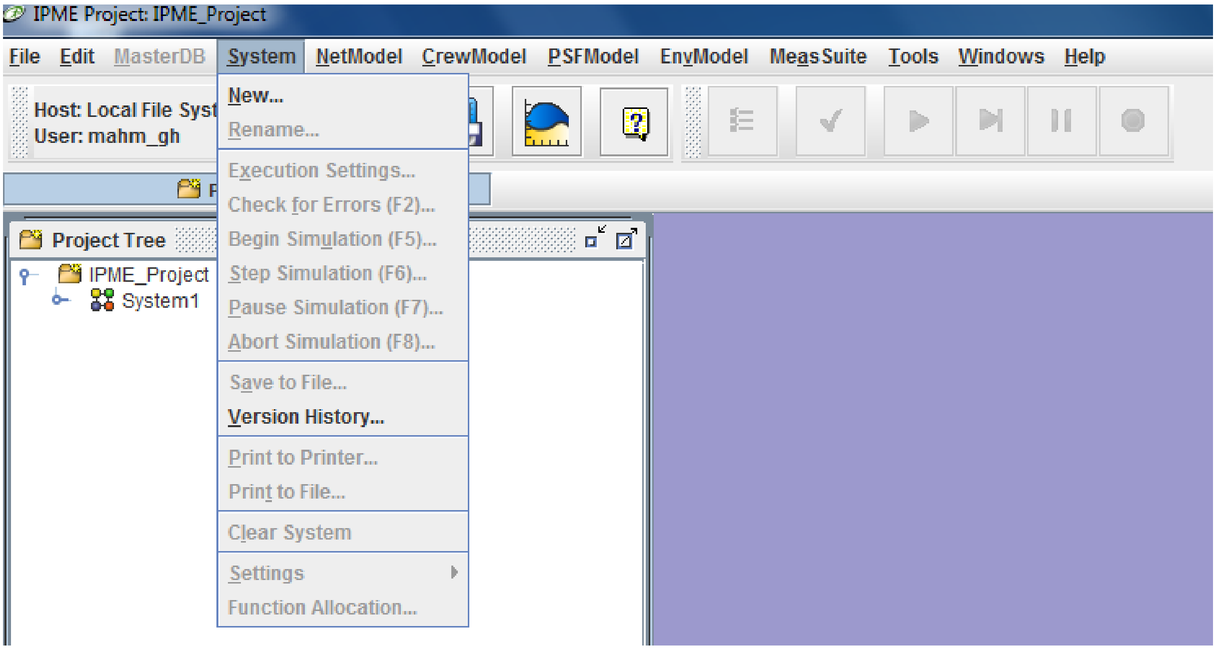


Figure 4: System menu

New	you can add systems to an IPME project. If you select Full System, the newly added system will contain the default component models: task network, crew, environment, and performance shaping models. An empty system will not contain any component models.
Rename	You can rename a system if you have the appropriate permissions.
Execution Setting	You can execute a simulation based only a task network (IPME mode only), or you can execute a simulation based on a system, which includes the task network model and the other component models. You are not required to have a measurement suite experiment defined in order to run the simulation. **Execution Settings Parameters:** Runtime Tab, Mission Tab, Output Tab, POPIP Tab(Subtab: Main, Compatible Tasks, Visual Areas, Global Operator Setting), Charts Tab
Check for Errors(F2)	Error checking begins automatically. If no errors are found, the Control tab displays. If errors are found, the Error Checking tab will display the errors found. You must

	resolve the errors before you can continue.
Begin Simulation (F5)	You can execute a simulation based only a task network (IPME mode only), or you can execute a simulation based on a system, which includes the task network model and the other component models. You are not required to have a measurement suite experiment defined in order to run the simulation.
Step Simulation (F6)	You can step through model execution one event at a time. This option is available only when the model has not yet begun execution, or when model execution is paused. Step execution used with the Watch Variables and Event Queue can be a useful debugging tool.
Pause Simulation (F7)	This button pauses model execution.
Abort Simulation (F8)	This button aborts model execution.
Save to File	You can save a project, system to a file. The file is saved in XML format.
Version History	Contains a list of all versions for the selected item. Created :Contains the date (dd/mm/yyyy) and time (24 hour clock) on which the selected version of the item was created. Created By: Contains the name of the user that created the selected version of the item. Note: Enter any additional information about this version (optional). Description: Contains a brief description of the selected version.
Print to Printer	You can print system to a printer. Before you print the item, use the Print Options dialogue box to select printing options.
Print to File	
Clear System	Clearing a system removes the task network, crew, environment, performance shaping function, measurement suite, and external communication models from a system. Clearing a system also restores the default System Checklist. Any changes made to the System Checklist are lost by clearing the system.
Setting	**VACP Descriptors:** The attentional demand analysis algorithm is a measure of task loading for an operator. It accounts for sensory information processing and decision making demands on human resources. Under VACP, each task is rated with regards to the attentional requirements within four independent channels: visual, auditory, cognitive, and psychomotor. VACP assignments are made in the Task dialogue box. **W/Index Conflict Matrix:** The W/Index algorithm measures the resource demand imposed upon the operator. W/Index decomposes a task into a set of channels, and establishes weights representing the amount of attention required for a task in each channel. Unlike the traditional workload models, W/Index also accommodates interference *between* channels. **External Comunications:** IPME provides an interface to allow the use of IPME in conjunction with other simulations (IPME-to-IPME or IPME-to-custom simulations) or federations. IPME can execute in one of the following modes: · If IPME is the server, the external simulations notify IPME of the time for their next event and IPME informs the simulations when it is okay to proceed. This is achieved by inserting events into IPME simulation. This mode may also be selected when IPME is a federate. · If IPME is the client, IPME notifies the external simulations of the time of its next event and waits for permission to advance to that time. IPME notifies the external simulation when it has completed the event. This mode may also be selected when IPME is a federate.

	· If IPME is a federate, it can send and receive values to and from the run-time infrastructure. This option is only included with special distributions of IPME. This mode can be selected with either the client or the server option. **System Checklist:** The System Check List allows you to take notes for the stages of analysis you are conducting. The default check list items are standard analysis steps in a human engineering program. Use the System Check List to help you manage the stages required to build your system. The System Check List can help ensure you complete critical tasks.
Function Allocation	IPME uses a design evaluation matrix to determine function allocation. In this method, candidate subsystem alternatives are listed and compared against selected criteria for allocation (response time, error rate, operability, cost, etc.). Each function/evaluation criteria relationship is assigned a numerical score, describing how well each function best meets the selected evaluation criteria. Use the Function Allocation subtab for each task to designate whether the functions in a task should be performed by operators, machines, or both. By default, the Function Allocation list contains 21 criteria used to determine whether a task should be performed by an operator, machine, or both. The analyst may add or remove criterion as necessary. Weighted values for the 21 criteria were developed by the Canadian Defence and Civil Institute for Environmental Medicine. Note that not all criteria require a non-zero value.

Table 4: Edit menu

4.3.1 Execution Settings Parameters

The execution settings parameters determine the details of Runtime processing, Mission, Output, POPIP (Main, compatible task pairs, visual areas, global operator settings, charts)

4.3.1.1 Runtime tab parameters

Simulation Mode

IPME can be executed in two different modes: IPME or POPIP. Execution in POPIP mode requires a complete system to be defined (Task Network, Crew, Environment, and PSF Models). The simulation mode is displayed in the Execution Control dialogue title during execution. Select one of the following:

a) **IPME** - The simulation will execute in IPME mode. PSFs can be used to modify task performance. VACP and W/Index data can be collected.

b) **POPIP** - The simulation will execute in POPIP scheduling mode. PSFs and the Forgetting TPM can be applied to modify task performance. Task scheduling can occur. POPIP, VACP, and W/Index data can be collected.

Scenario Goal

IPME simulation can be executed with one of four different goals selected: No Goal, Minimise Cost, Minimise Crew Size, or Balance Utilisation. The selected goal will determine which operator is returned from the *getQualifiedOpers()* function.

Scenario Goal contains the following parameters:

No Goal, Minimise Cost, Minimise Crew Size, Balance Utilisation

Number of Runs

Enter the number of times to execute the model. If the number of runs is greater than one, the execution times, paths of execution, and other probabilistic events are different for each run.

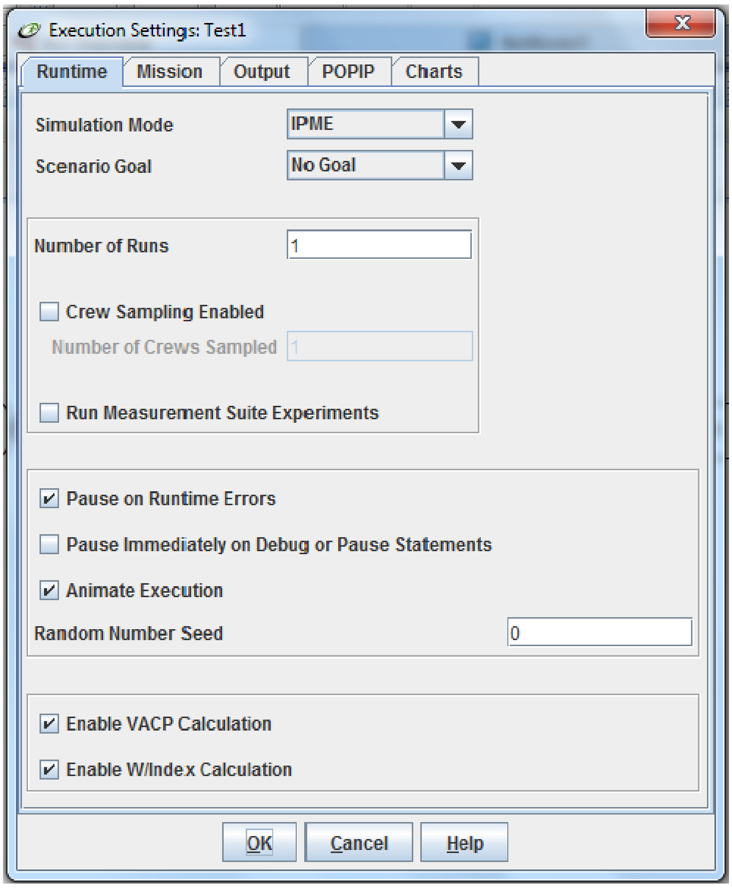


Figure 5: Execution Setting – Runtime Tab

If you execute the model multiple times with the Output Snapshot option selected, the results files contain data from all of the runs.

Crew Sampling Enabled

Controls whether sampled initial values are used for trait attributes and anthropometry values rather than the Initial Mean values entered in the Edit Trait Attribute dialogue box and the Anthropometry tab in the Edit Operator dialogue box.

Number of Crews Sampled

Specify the number of times to randomly sample initial values for trait attributes and anthropometry variables from the operator population

Run Measurement Suite Experiments

Controls whether the measurement experiment is run during system execution.

Pause on Runtime Errors

Controls whether the model will pause when a runtime error occurs.

Pause Immediately on Debug or Pause Statements

When selected, the simulation will pause immediately whenever a call to the built-in *pause()* function or a debug statement where the first argument evaluates to true is encountered.

Animate Execution

Runs the model with network diagram animation. The tasks, queues, and networks are shaded to indicate when they are being executed.

Random Number Seed

Determines which set of random numbers are generated to calculate distributions of task times, trait attribute values, and anthropometry variables by considering the mean, standard deviation, and frequency distribution.

Enable VACP Calculation

Select this checkbox to update operator VACP data during model execution.

Enable W/Index Calculation

Select this checkbox to update operator W/Index data during model execution.

4.3.1.2 Mission Tab parameters

Mission Name

Contains a descriptive name of the mission (optional).

Description

Contains a detailed description of the mission (optional).

Enable Time Limit

Controls the duration of the execution. To set a time limit, select the Enable Time Limit check box and enter a time value in the adjacent field

Scenario Events

Controls which scenario events are selected for execution. Events are defined in the project tree.

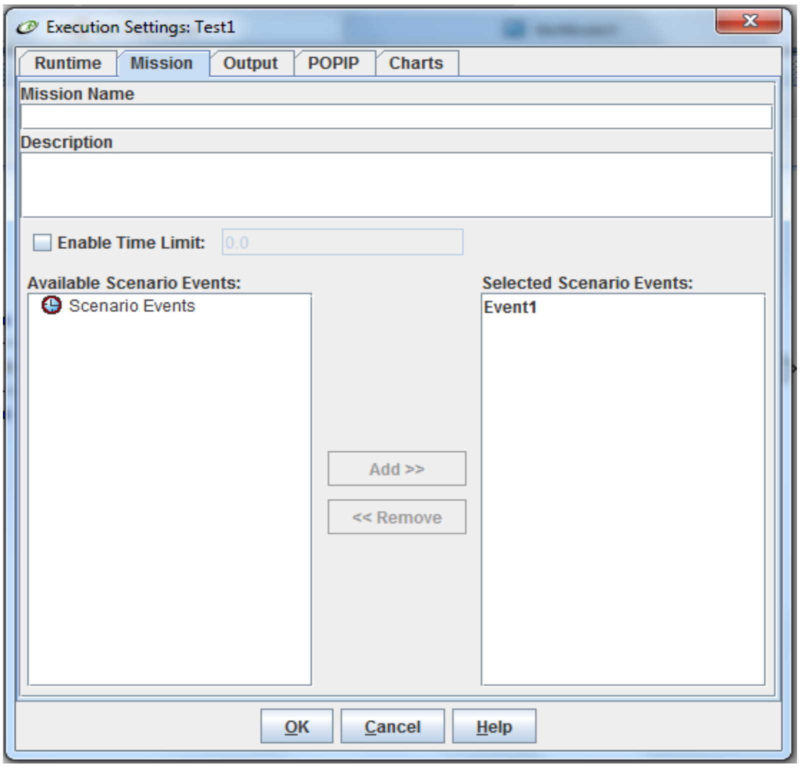


Figure 6: Execution Setting – Mission Tab

4.3.1.3 Output Tab Parameters

The Output tab contains the following parameters:

System Output Directory

Contains the location where the simulation results files will be stored.

To change the Output Directory: Click Browse. (The Choose A Directory dialogue displays). Browse to and select the desired output directory. Click Open (Output will be saved to the specified directory)

Output Execution Trace (*.trc)

Controls whether a file is produced that contain the beginning time and ending time for each task executed during the model run. The trace file shows the steps taken by each entity as it travels through the network diagram, and can be a useful tool for debugging the model. Trace files have the extension *.trc*.

Output Execution Viewer File (*.evf)

Controls whether an *.evf* file is generated. These files can be viewed in the Execution Viewer. The files created using this option are a graphical representation of the model simulation.

Output Critical Path Results (*.cpr) - Controls whether critical path data are collected for the simulation. The critical path is the route through the network that takes the longest to execute. The critical path is useful in determining the path to optimise for the model. Critical path data are stored in an ASCII file with the extension *.cpr*.

Output External Interface Log (*.log) - Controls whether a log file is collected showing the events transmitted to and received from another application. This is used when you are executing IPME in conjunction with other simulations.

The *.log* files are stored in the output directory.

Output Snapshots (*.ssr) - Determines whether snapshots of variables are collected during model execution. Snapshots are defined in the project tree. Snapshot files have the extension *.ssr*.

Output VACP File (*.vacp) - Select this checkbox to collect the VACP data. This file has the extension *.vacp*.

Output W/Index File (*.windex) -Select this checkbox to collect the W/Index data. This file has the extension *.windex*.

Output Operator Time Pressure (*.itp) - Select this checkbox to collect the operator time pressure data. This file has the extension *.itp*.

Output Run Times File (*.ttm) - Select this checkbox to collect the run times data. This file has the extension *.ttm.* This file collects the final clock value for each run.

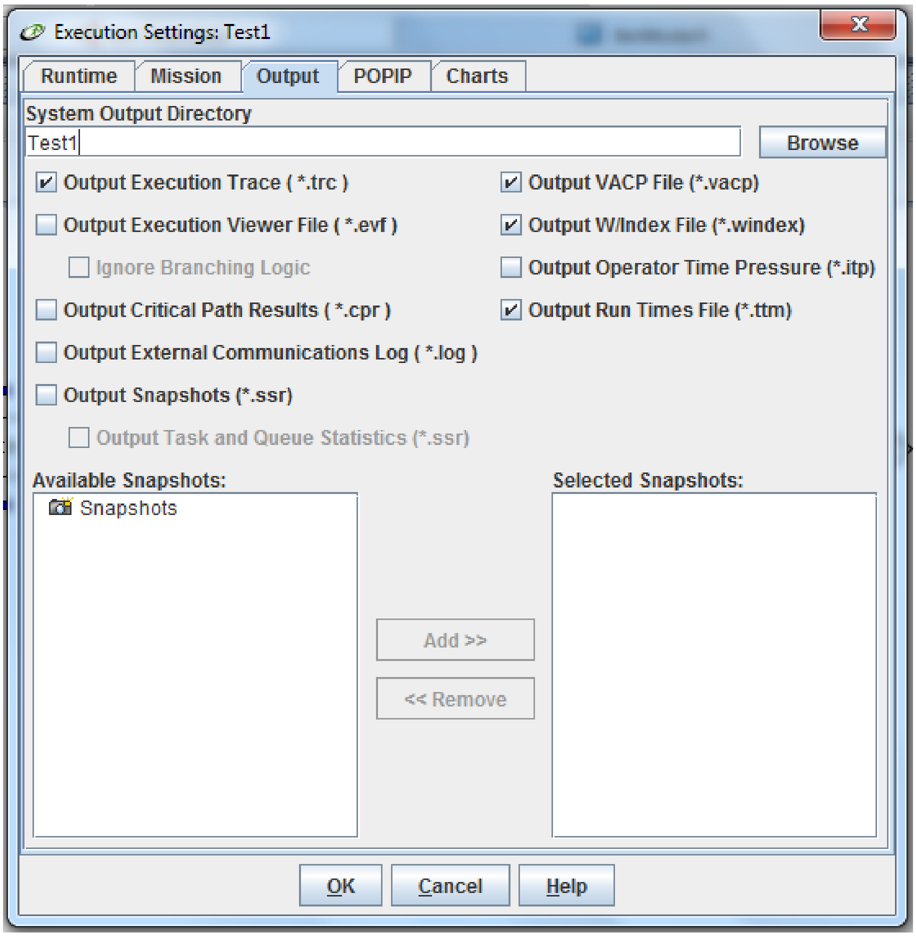


Figure 7: Execution Setting – Output Tab

4.3.1.4 POPIP Tab Parameters

The POPIP tab contains three subtabs that are used to control execution when POPIP mode is selected: Main, Compatible Tasks, and Visual Areas.

4.3.1.4.1 POPIP - Main Subtab

The Main subtab contains these parameters:

Enable Structural Interference

Controls whether Structural Interference is applied during model execution. Enabling this option will cause tasks to get delayed, interrupted or shed based on structural interference in the selected channel(s). Select the appropriate categories from the following list: Visual, Auditory, Psychomotor**.**

Visual

Controls whether structural interference due to the visual domain is considered when tasks with a visual component are scheduled. This option is only available when Enable Structural Interference is selected.

Auditory

Controls whether structural interference due to the auditory domain is considered when tasks with an auditory component are scheduled. This option is only available when Enable Structural Interference is selected.

Psychomotor

Controls whether structural interference due to the psychomotor domain is considered when tasks with a psychomotor component are scheduled. This option is only available when Enable Structural Interference is selected.

Enable POP Workload Calculation

Controls whether the POP Workload Calculation will be used to schedule tasks during model execution.

POP mode - Controls the POP mode used for the simulation. When you click this button, a menu displays with a selection of POP (Prediction of Operator Performance) modes. The POP model is based on a set of rules that define work load demand and the consequences on performance depending on the factors that are inherent to a task.

POP modes include the following:

Post Processing - POP data processed after model execution. The Execution Settings dialogue box is labelled POP Postprocessing Mode.

a) *TDM Display* - task demand modifiers (TDM) are calculated. The Execution Settings dialogue box is labelled POP Runtime Mode (display only). The TDM values are not applied to tasks during model execution.

b) *TDM Effects* - task demand modifiers (TDM) are calculated, displayed, and automatically applied to tasks during model execution. The Execution settings dialogue box is labelled POP Runtime Mode (time/error effects).

c) *TDM Scheduler* - task demand modifiers (TDM) are calculated, displayed, and automatically applied to tasks during model execution. Tasks are also rescheduled (delayed) accordingly. The Execution Settings dialogue box is labelled POP Runtime Mode (overload rescheduling).

The POP mode is only available when Enable POP Workload Calculation has been selected.

TDM Calculation

Controls the type of TDM calculation that is used. When you click this button, a menu displays with a selection of TDM (Task Demand Modifiers) Calculations. Select one of the following calculations to use in the model simulation: Standard, Weighted, Maximum, and Weighted Maximum. The TDM Calculation is only available when Enable POP Workload Calculation has been selected. During execution, Task Demand Multiplier values are color-coded to indicate the severity of the TDM effects.

Error Multiplier

Controls the error multiplier that is used. In determining the effect of the workload on error rate, an error constant is used. The constant is determined experimentally.

States Limit

Controls the limit used on the number of states. To prevent the processing time from exceeding a reasonable limit, the number of states can be limited. This processing bound reduces the accuracy of the workload calculations.

mean time pressure value

is recorded at the end of the sliding window.

The POPIP scheduler runs on unitless time. If you create a model with the time units based on minutes, the time pressure variable would also be based on minutes.

Width

Contains the value for the total time to be used in time pressure calculations. A time window value equal to four times the mean of the longest task duration is a recommended starting point.

Resolution

This value is used to scale the width of the time window. For example, if events are hours apart, you would want to avoid having a 60 second window.

Task Resume Penalty

This field contains a multiplier for the mean task execution time to be applied when a task is interrupted and scheduled to be restarted.

Enable Task Priority Time Pressure - When this option is selected, IPME will prioritise tasks being considered for execution according to each tasks' priority time pressure value, with tasks with the highest priority time pressure value will be selected first (highest priority).

Reset to Defaults - Click this button to reset all POPIP parameters on the Main subtab to their default values.

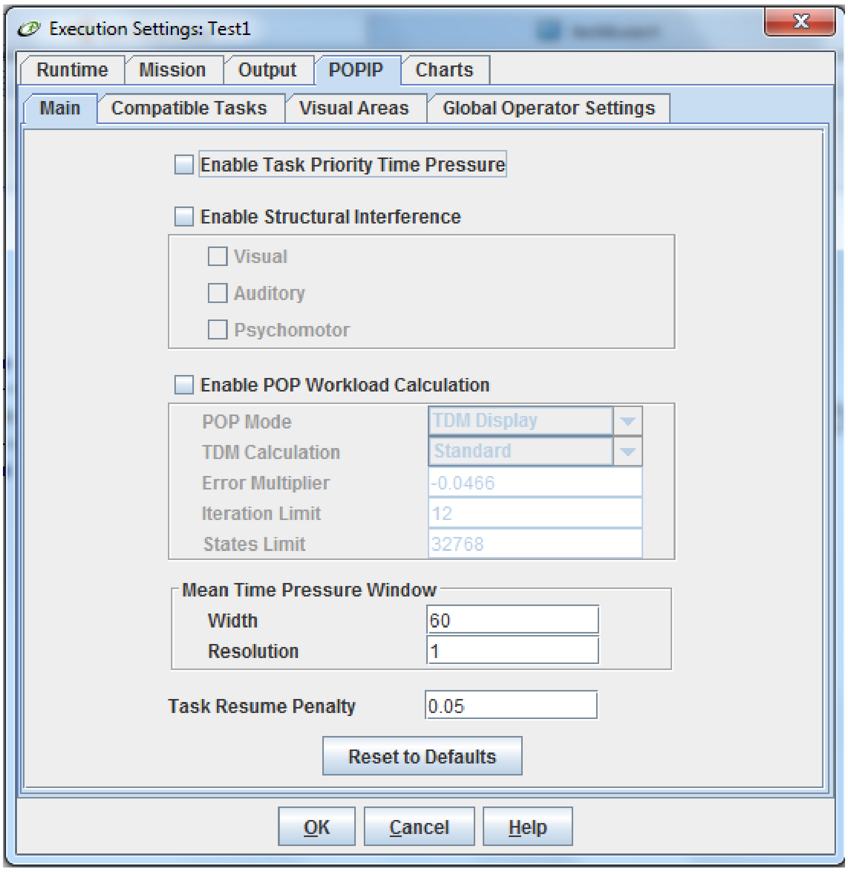


Figure 8: Execution Setting – POPIP Tab,Main subtab

4.3.1.4.2 POPIP - Compatible Tasks Subtab

Compatible task pairs are pairs of tasks that may be compatible for dual tasking in certain combinations, such as using multiple digits on the same hand. Compatible tasks are not involved in serial processing and their default interference is set to zero. The Compatible Tasks subtab is used to select the compatible pairs of tasks.

Task A/Task B

Each list contains the tasks defined for the model. Select a task in the Task A list, and then select the compatible task in the Task B list. Click the **Add** button. The compatible task pair displays in the table at the bottom of the dialogue box.

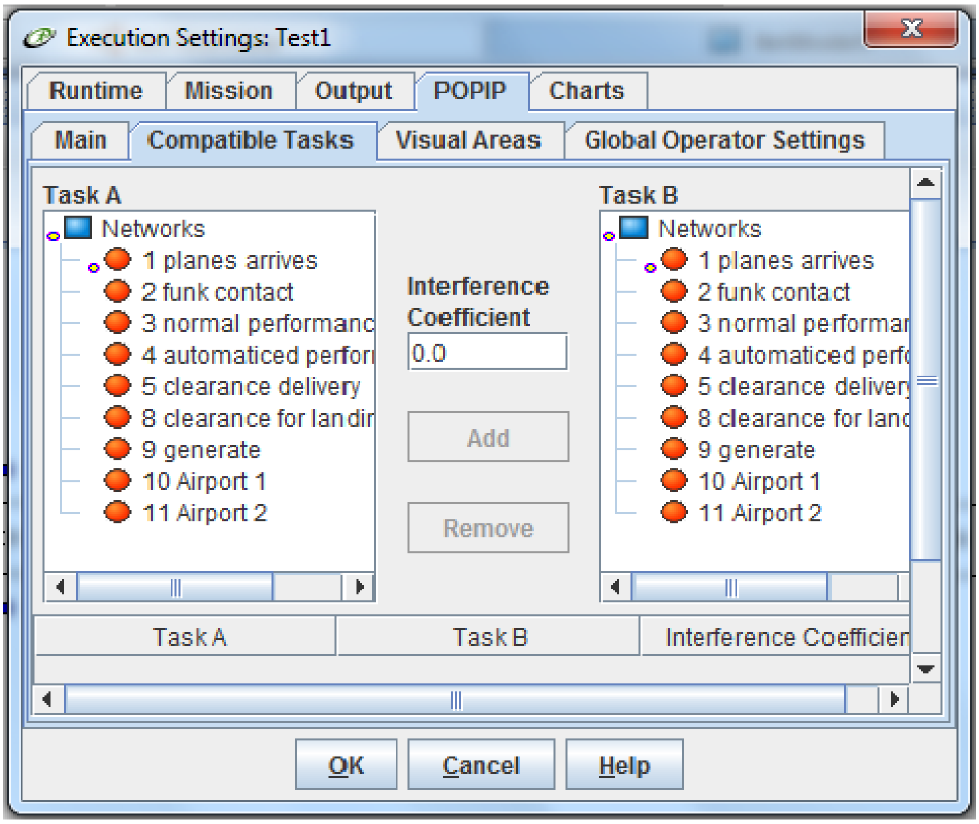


Figure 9: Execution Setting – POPIP Tab, Compatible Tasks Subtab

Interference Coefficient

After selecting a pair of tasks, you can enter an interference coefficient for the task pair. Although this value defaults to zero, the POPIP algorithm allows for a nonzero interference coefficient.

Contains a listing of the compatible task pairs defined for the model. To delete a compatible task pair, select the row in the table that contains the task pair and click the **Remove** button.

4.3.1.4.3 POPIP - Visual Areas Subtab

The visual areas are the areas in a work environment where the operators will be looking, such as a monitor, dial, or gauge. Each visual task is assigned to an area in the visual scene, such as Area1, Area2, and so on. The Visual Area table contains the X, Y, and Z positions for the areas. The X, Y, and Z positions are in metres and are used to locate the visual area in a 3-D environment. The locations are used to calculate angles to use in the assessment of visual interference coefficients and probabilities of detection for externally initiated stimuli.

The Visual Area tab contains the following parameters:

Area Name

Contains the areas or targets in the visual scene. Type a descriptive name for the area, such as Speedometer or Radar Display.

X, Y, Z

Contains the position of the object in the work environment. Type the X, Y, and Z values (units are meters).

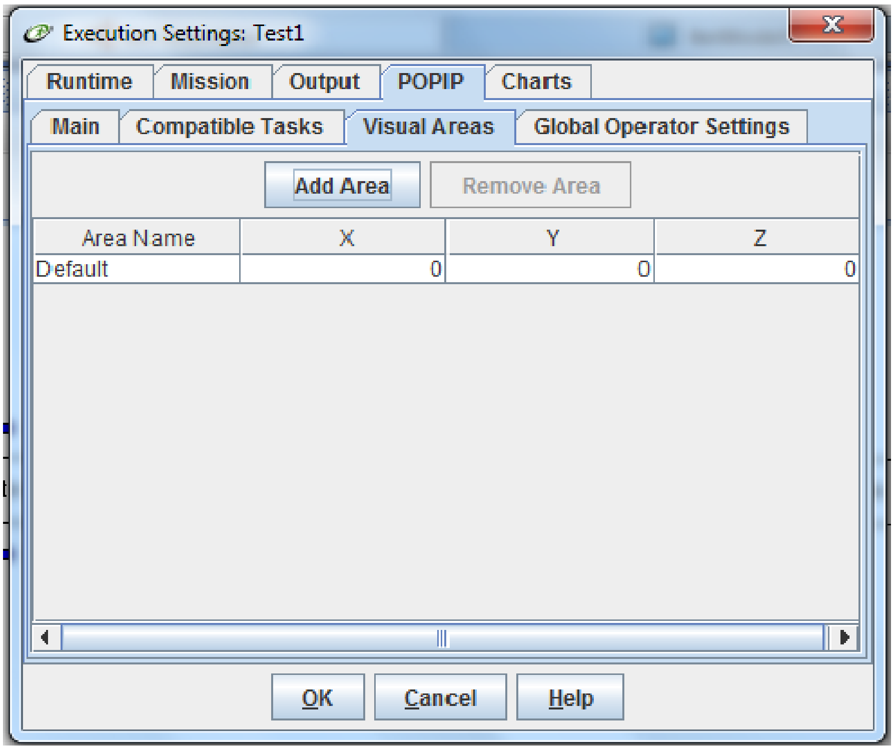


Figure 10: Execution Setting – POPIP Tab, Visual Areas Subtab

To add a visual area: Click the Add Area button. Enter the area parameters.

To remove a visual area: Select the visual area to remove in the table. Click the Remove Area button. The visual area is removed.

4.3.1.4.4 POPIP - Global Operator Settings

Settings on the Global Operator Settings tab can be used to override values set for individual operators.

Enable Global Operator Settings

Select this check box to make the Global Operator settings on this tab override any individual operator-level settings entered in the crew model. When this check box is enabled, values entered at the operator level are preserved, but are not applied during execution.

Critical Interference

If the sums of the individual coefficients for tasks in the queue plus the coefficient for the task trying to be scheduled are below the value entered in this field, then the task can be added to the active queue.

Critical Time Pressure

When the value in this field is reached for the operator, multitasking is stopped and processing is done serially. If the instantaneous task time pressure for any of the tasks that the operator is currently performing exceeds the critical time pressure value, then tasks can no longer be interrupted.

Critical % Complete

When the value in this field is reached, the task will not be stopped even if another task of higher priority is trying to be scheduled for execution. This feature of the scheduler represents task momentum.

Short Term Memory

This field contains the number of tasks that can be held in the queue for short-term memory. Short-term memory is the lowest form of cognition. It represents the number of tasks that can be held in human memory for immediate execution or attention. Short-term memory is generally limited to between three and five items.

Forgetting TPM

Refers to the forgetting of a task. The probability for a successful outcome for a task usually decreases if a task is interrupted or delayed, or dropped from the queue (forgotten) of tasks considered for processing by the operator if it is not serviced within a certain time period.

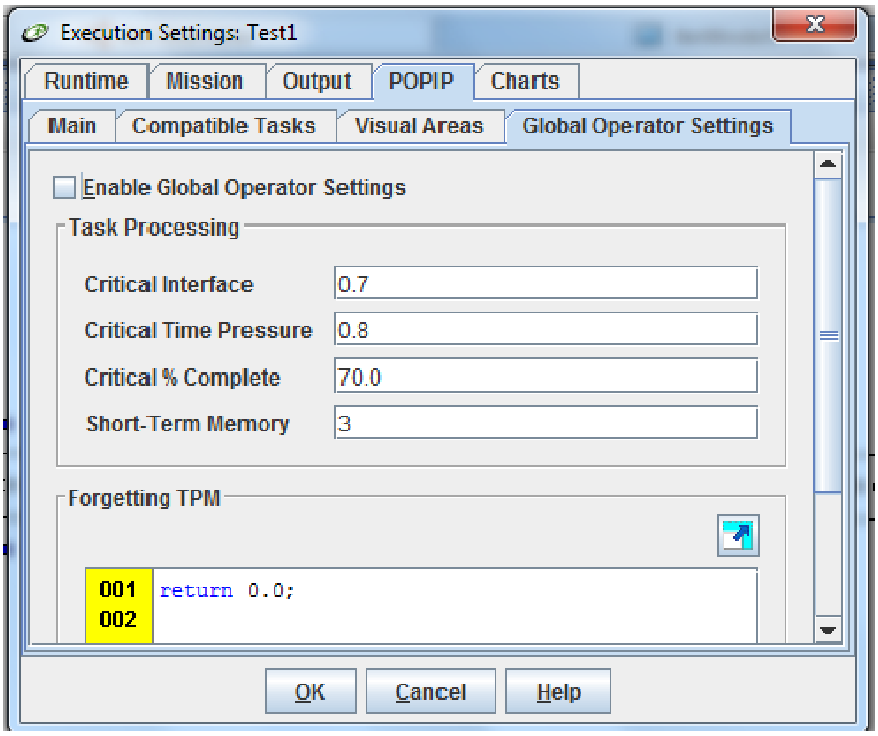


Figure 11: Execution Setting – POPIP Tab, Global Operator Settings Subtab

Dropping a task from the operator's queue can simulate forgetting a task. The probability that an item is forgotten will likely increase with time.

4.3.1.5 Charts Tab contains the following parameters:

The Charts tab contains settings used to control charts and graphs displayed during simulation execution.

The Charts tab contains these parameters:

Enable Run-time Charts - Select this checkbox to make run-time charts available for viewing during simulation execution. Then choose any of the following run-time charts to display.

VACP by Operator - Select this checkbox to display the VACP values for each operator as the simulation executes. The chart may be viewed as a line graph or a histogram.

Workload Index by Operator - Select this checkbox to display the Workload Index values for each operator as the simulation executes.

The W/Index graph is best viewed as a histogram. For each operator, four pieces of information display: WithinChannel, Interference, WorkloadIndex, and Concurrent Tasks. **The Concurrent Tasks value** is the number of active tasks assigned to the operator at that simulation clock time. **The WithinChannel value** is the calculated workload contribution value for each channel for all active tasks. **The Interference value** that is graphed takes into account any interference between channels for each pair of active tasks. **The WorkloadIndex value** in the runtime graph is simply the sum of the WithinChannel and Interference values

POPIP

Task Demand Modifiers (TDMS) by Operator -Select this checkbox to display the task demand modifiers for each operator as the simulation executes.

Time Pressure by Operator - Select this checkbox to display the operator time pressure data as the simulation executes..

POP Workload by Operator - Select this checkbox to display the POP workload data for each operator as the simulation executes.

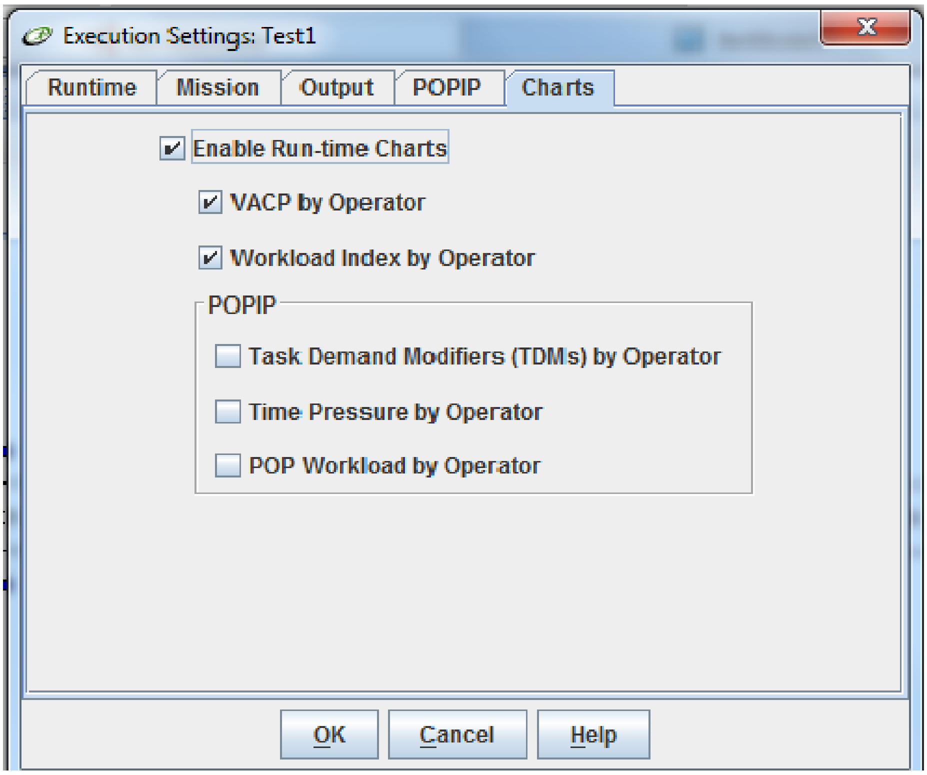


Figure 12: Execution Setting – Chart Tab

4.4 NetModel menu

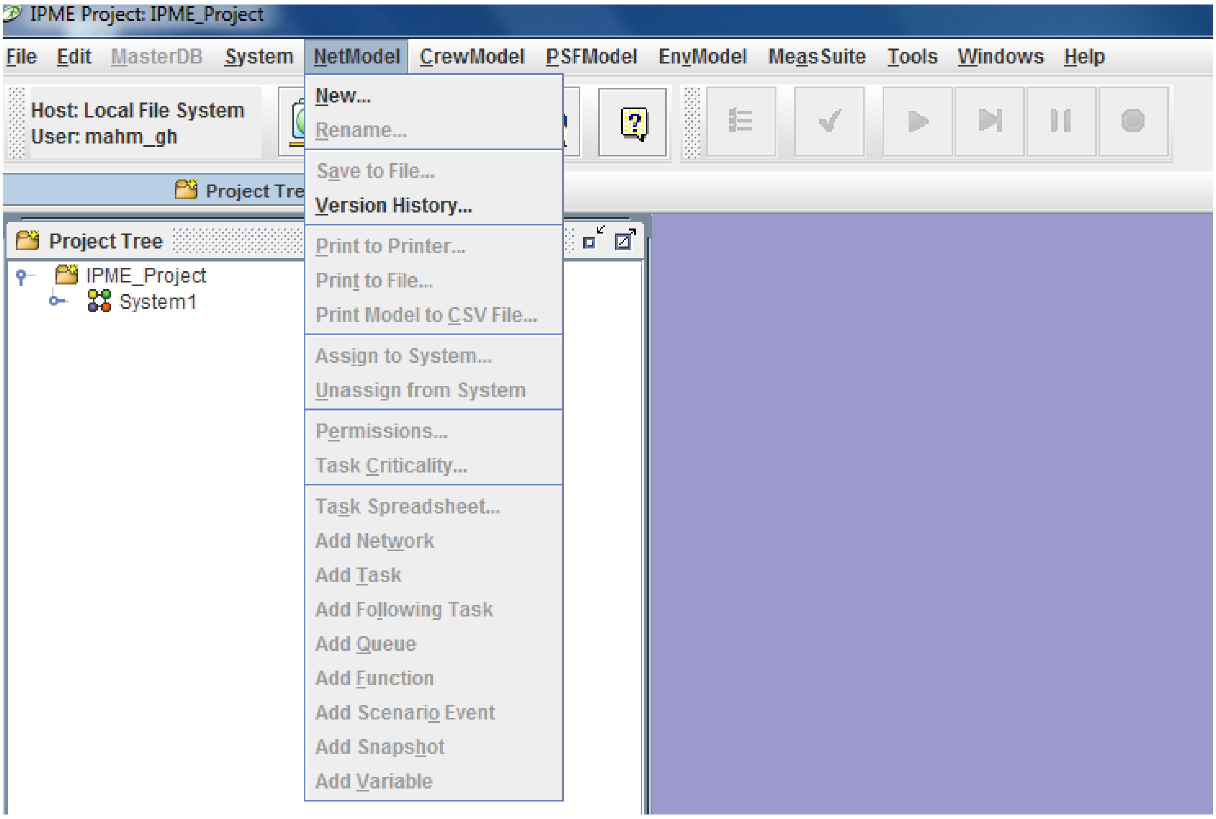


Figure 13: NetModel menu

New	You add a new network model by creating a new model of that component type.
Rename	You can rename a system if you have the appropriate permissions.
Save to File	You can save a network model to a file. The file is saved in XML format.
Version History	Contains a list of all versions for the selected item. Created: Contains the date (dd/mm/yyyy) and time (24 hour clock) on which the selected version of the item was created. Created By: Contains the name of the user that created the selected version of the item. Note: Enter any additional information about this version (optional). Description: Contains a brief description of the selected version.
Print to Printer	You can print a network model to a printer. Before you print the item, use the Print Options dialogue box to select printing options.
Print Model to CSV file	Basic parts of the Task network model can be printed to a CSV file. The Task ID, Task Name, Beginning Effects, Ending Effects, Release Condition, Time Distribution, Mean Time, Standard Deviation, Operator Assignment, and Following Tasks are included.
Assign to System	You can construct a system by assigning network models to the system. You cannot assign a component model unless you have the appropriate access privilege to the model.
Unassign	Network models can be unassigned from a system. Unassigned network models are not deleted and reside in the project.

from System	
Permission	As the owner of the model or a user with full permissions, you can change the access level for a model on a global or individual basis. Individual access assignments override global access. Access levels include no access, read-only, or full read and write permissions.
Task Criticality	The criticality of a task refers to the mission category of the task. Task criticality definitions are used in the Task Criticality tab in the Task dialogue box to determine whether a task is critical. A task is critical if the Base Criticality Rating is greater than the user-defined threshold. Values range from 0 to 10 inclusive. Overlapping ranges are not allowed. Not all descriptions must be used, but the full range of integers [0-10] must map to some text description. For example, you could define [0-5] to be *Moderate,* and [6-10] to be *Critical,* but could not specify [0-2] as *Moderate,* and [6-10] as *Critical* without mappings for the remaining range [3-5]. To omit a range from the mapping, the minimum and maximum range values must be equal, and some other range must map to this value.
Task Spreadsheet	The task spreadsheet viewer allows you to view and edit network and task information for all tasks and networks within a project from the same dialogue. This view is useful for copying and pasting the same information between tasks or networks. It also allows for comparison of task information. Each row contains information for a task or network contained in the project, with one column dedicated for each data field. Changes made in the spreadsheet viewer are reflected in the model. Each object's id displays at the far left of the row. Objects are indented according to their hierarchy in the project tree. Icons also display to indicate whether the row belongs to a task or a network.
Add Network	When an activity in your model is very complex, you can break it down further into more specific sub-activities and represent the main activity as a network, while representing the sub-activities as subnetworks or individual tasks.
Add Task	A task is the basic building block of a model and contains timing information, execution constraints, effects of the task, POPIP, and workload data. IPME automatically assigns a default name of *UnNamed* and a default ID to the task.
Add Following Task	You can indicate the tasks that follow a specific task. This order determines the passage of an entity through the network diagram. Any time you have two or more tasks following a task, IPME automatically adds a decision node. The Following Tasks dialogue box displays. All available tasks in the network model display in the Possible Following Tasks box, and any tasks assigned as following tasks display in the Following Tasks box.
Add Queue	The queue is a waiting area in front of a task where entities accumulate while they are waiting to execute the task. Queues adopt the release condition of the associated task. If you have a queue on a repeating task, there would be a "loop-back" and a new entity would enter the queue again. For a queue to accumulate entities, each entity must be defined as a separate *tag* number. If a tag enters the queue and the tag number is already present, the two will be merged as a consequence of the release condition.
Add Function	Functions are used to group a complicated expression or group of expressions in a model. In all instances where the function name is encountered in the model, the expressions are executed. The user-defined functions for the network display towards the bottom of the list of network elements. The function list displays functions that are defined for the current model. You can add, edit, or delete user-defined functions.
Add Scenario Event	Scenario Events are time- or condition-based events that you can use to change the state of the simulation. Events can be used to start sections of the network or assign values to variables, independent of when an entity begins or ends a task or enters or

	departs a queue. The expressions assigning the values can be scheduled to occur at a specific clock time or when a specific condition is met in the model. You can use events to start, stop, suspend, or resume specific tasks or groups of tasks The scenario events to execute are specified in the Execution Settings dialogue box.
Add Snapshot	Snapshots are used for custom data collection during simulation. A snapshot records the values of particular variables at specified points during model execution. These data can then be used to generate statistics and plots describing model execution. Defined snapshots can be enabled in the Execution Settings dialogue box.
Add Variable	Variables include task network, user-defined, and physical environment (mission, crew, physical, and threat) variables. You can use variables in any expression field.

Table 5: NetModel menu

4.5 CrewModel menu

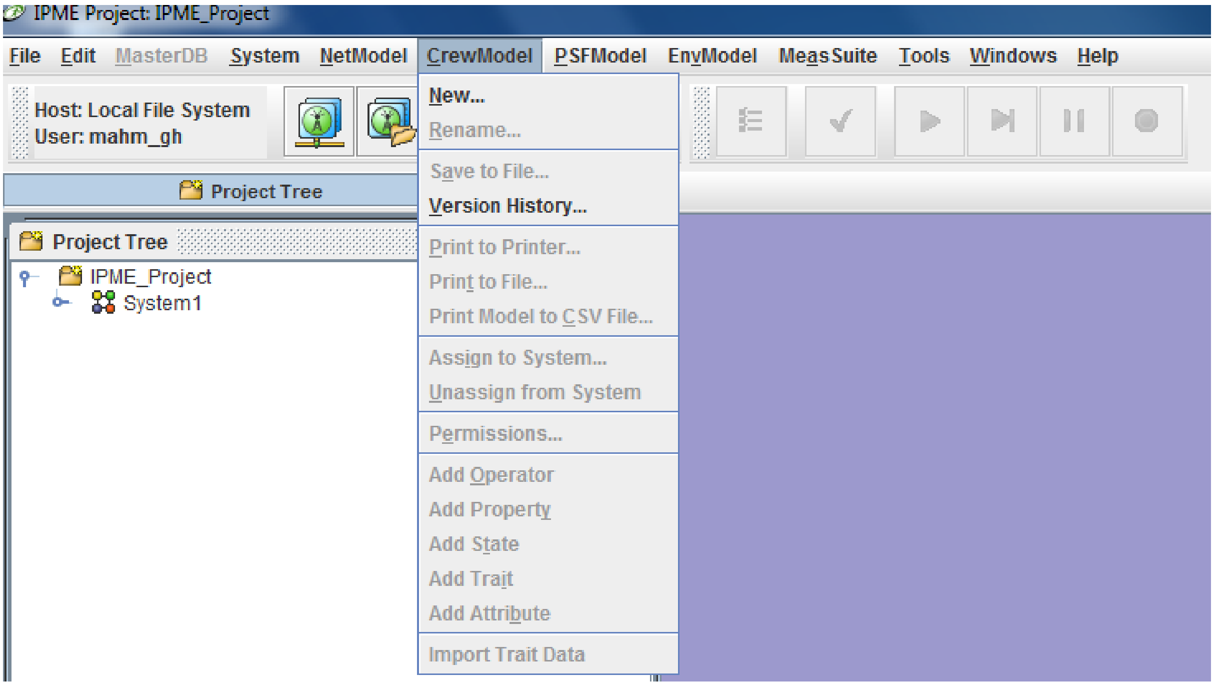


Figure 14: CrewModel menu

New	You add a new crew model by creating a new model of that component type.
Rename	You can rename crew model if you have the appropriate permissions.
Save to File	You can save a crew model to a file. The file is saved in XML format.
Version History	Contains a list of all versions for the selected item. Created: Contains the date (dd/mm/yyyy) and time (24 hour clock) on which the selected version of the item was created. Created By: Contains the name of the user that created the selected version of the item. Note: Enter any additional information about this version (optional). Description: Contains a brief description of the selected version.

Print to Printer	You can print a crew model to a printer. Before you print the item, use the Print Options dialogue box to select printing options.
Print to File	You can print a crew model to a file. The file is saved in HTML format.
Print Model to CSV File	Basic parts of the crew model can be printed to a CSV file. "Operator" ,"Type","DefinedBy","Name","Attribute","Sampling","Unit","Minimum","Maximum","Mean Expression" and "Standard Deviation" are included.
Assign to System	You can construct a system by assigning crew models to the system. You cannot assign a component model unless you have the appropriate access privilege to the model.
Unassign from System	Crew models can be unassigned from a system. Unassigned crew models are not deleted and reside in the project.
Permissions	As the owner of the model or a user with full permissions, you can change the access level for a model on a global or individual basis. Individual access assignments override global access. Access levels include no access, read-only, or full read and write permissions.
Add Operator	Operators are individual simulation objects that contain default human physical and psychological characteristics. An operator is defined by a set of characteristics that consist of properties, traits, states and anthropometry. Each operator characteristic can consist of one or more attributes. An attribute is the name of a value or expression that indicates the status of a characteristic.
Add Property	Operator properties consist of the default physical characteristics for each operator: *Eyes*, *LeftFoot*, *RightFoot*, *LeftHand*, and *RightHand*. These default properties are fixed for IPME, but you can add other properties.
Add state	States are operator characteristics that can be formulated from the combination of operator properties and traits in conjunction with environment and task network variables. States include factors such as *mental alertness, clothing, encumbrance, hunger, thirst, morale, or fear.*
Add trait	Traits describe non-physical operator personality characteristics such as *agility, susceptibility to motion sickness, fitness, or cognitive ability*. Traits are generally viewed as those characteristics of an operator that *will not change* as a function of the simulation.
Add Attribute	An operator attribute is a single value that indicates the status of the operator. The default attribute for each characteristic is named *Value*. To add an attribute, select the characteristic in the project tree to which you want to add an attribute.
Import Trait Data	This feature can be used to import trait attribute values from an *.xml* file. The xml file must adhere to the format required by IPME.

Table 6: CrewModel menu

4.6 PSFModel menu

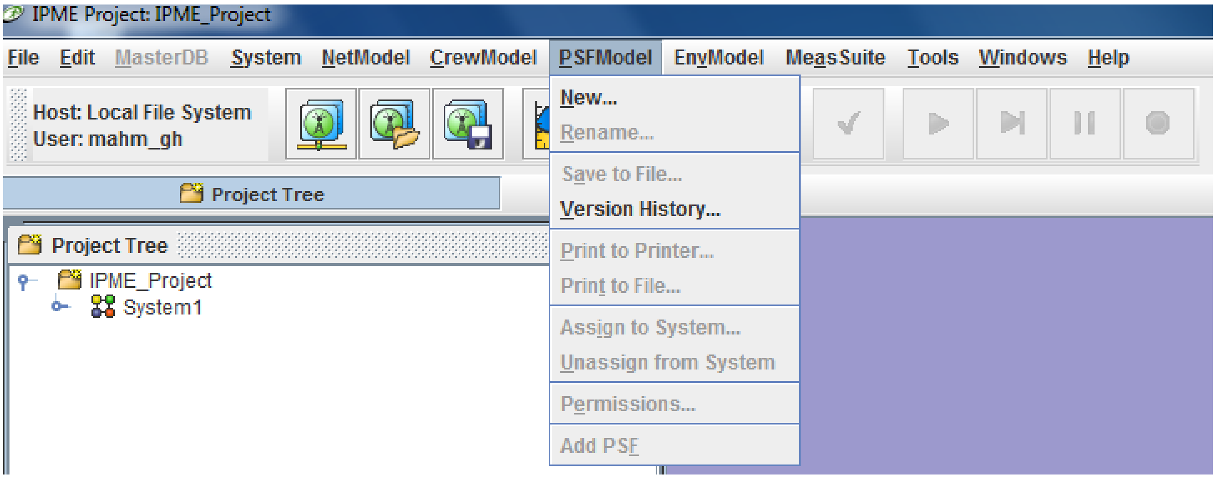


Figure 15: PSFModel menu

New	You add a new PSF model by creating a new model of that component type.
Rename	You can rename a PSF model if you have the appropriate permissions.
Save to File	You can save a PSF model to a file. The file is saved in XML format.
Version History	Contains a list of all versions for the selected item. Created: Contains the date (dd/mm/yyyy) and time (24 hour clock) on which the selected version of the item was created. Created By: Contains the name of the user that created the selected version of the item. Note: Enter any additional information about this version (optional). Description: Contains a brief description of the selected version.
Print to Printer	You can print a PSF model to a printer. Before you print the item, use the Print Options dialogue box to select printing options.
Print to File	You can print a PSF model to a file. The file is saved in HTML format.
Assign to System	You can construct a system by assigning PSF models to the system. You cannot assign a component model unless you have the appropriate access privilege to the model.
Unassigned from System	PSF models can be unassigned from a system. Unassigned PSF models are not deleted and reside in the project.
Permissions	As the owner of the model or a user with full permissions, you can change the access level for a model on a global or individual basis. Individual access assignments override global access. Access levels include no access, read-only, or full read and write permissions.
Add PSF	Performance Shaping Functions can be added by the user or from the Master Database.

Table 7: PSF Model menu

4.7 Environment Model menu

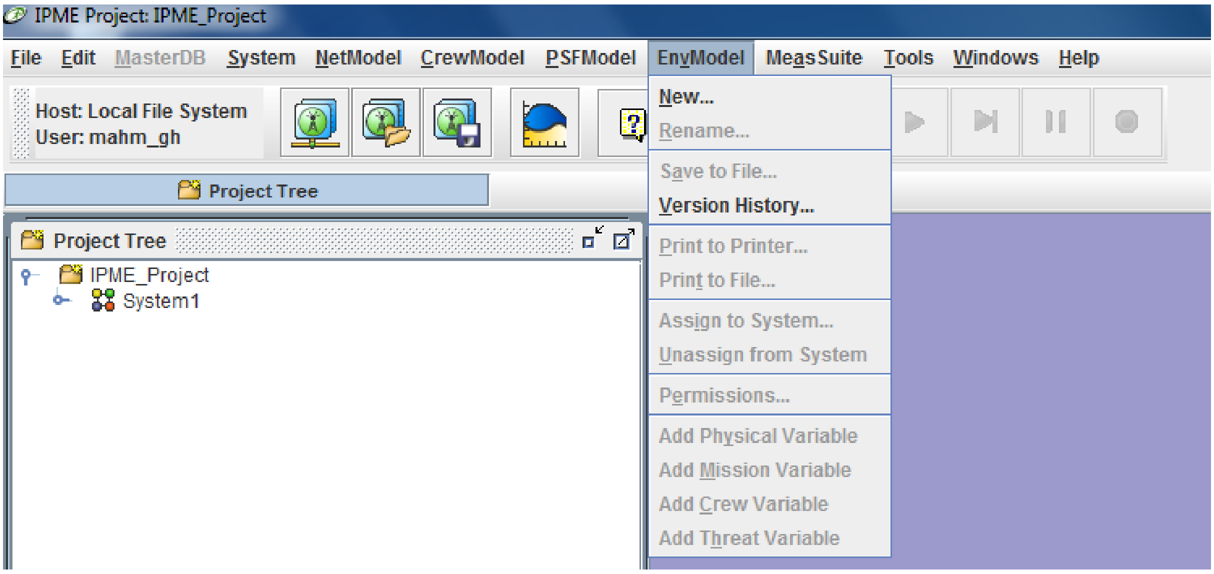


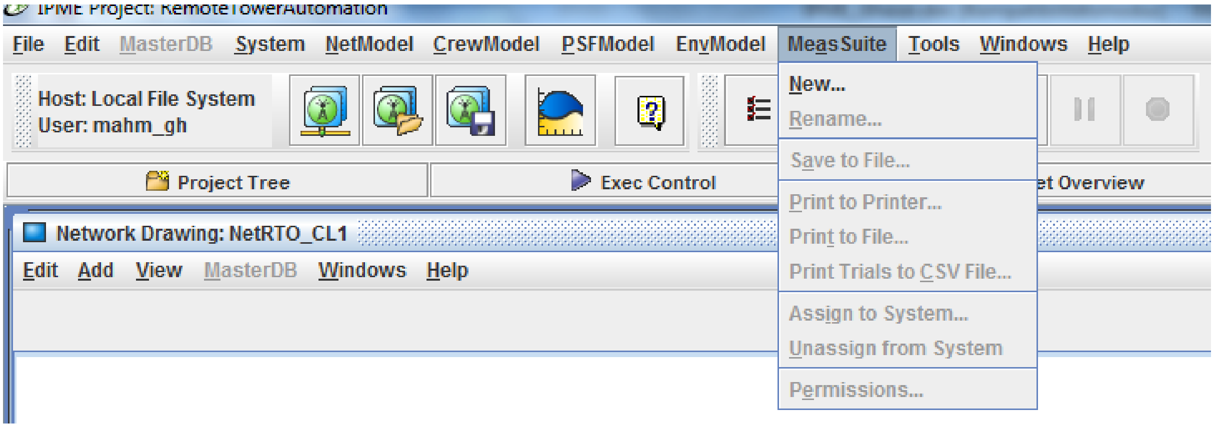
Figure 16: EnvModel menu

New	You add a new environment model by creating a new model of that component type.
Rename	You can rename an environment if you have the appropriate permissions.
Save to File	You can save an environment model to a file. The file is saved in XML format.
Version History	Contains a list of all versions for the selected item. Created: Contains the date (dd/mm/yyyy) and time (24 hour clock) on which the selected version of the item was created. Created By: Contains the name of the user that created the selected version of the item. Note: Enter any additional information about this version (optional). Description: Contains a brief description of the selected version.
Print to Printer	You can print an environment model to a printer. Before you print the item, use the Print Options dialogue box to select printing options.
Print to File	You can print an environment model to a file. The file is saved in HTML format.
Assign to System	You can construct a system by assigning environment models to the system. You cannot assign a component model unless you have the appropriate access privilege to the model.
Unassign from System	Environment models can be unassigned from a system. Unassigned environment models are not deleted and reside in the project.
Permissions	As the owner of the model or a user with full permissions, you can change the access level for a model on a global or individual basis. Individual access assignments override global access. Access levels include no access, read-only, or full read and write permissions.
Add physical Variable	You can use variables in any expression field. Ambient_Noise, Contamination_ Level, Contamination_Type, Digability, Dry_Bulb_Temperature, Footing, Humidity, Illumination, Pressure, Sea_State, Temperature, Terrain, Terrain_Direction, Terrain_Slope, Thermal_Radiation, Time_Of_Day, Turbulence, Weather, Wind_Direction, Wind_Speed, Wind_Strength

Add Mission Variable	Adequacy_of_Procedures, Communications_Density, Intelligence, Platform_Reliability, Surveillance_Reliability, Time_Stress, Weapons_Reliability.
Add Crew Variable	Clarity_of_Role, Cooperation, Leadership_Style, Supervision, Team_Experience, Team_Morale, Team_Training.
Add Threat Variable	Target_Bearing, Target_Elevation, Target_Location, Target_Obscuration, Target_Range, Threat_Severity, Target_Signature, Target_Speed, Target_Value.

Table 8: EnvModel menu

4.8 Measurement suite menu



New	You add a new measurement suite model by creating a new model of that component type.
Rename	You can rename a measurement suite if you have the appropriate permissions.
Save to File	You can save a measurement suite a file. The file is saved in XML format.
Print to Printer	You can print a measurement suite to a printer. Before you print the item, use the Print Options dialogue box to select printing options.
Print to File	You can print measurement suite to a file. The file is saved in HTML format.
Print Trials to CSV File	
Assign to System	You can construct a system by assigning measurement suite to the system. You cannot assign a component model unless you have the appropriate access privilege to the model.
Unassign from System	measurement suite can be unassigned from a system. Unassigned measurement suite are not deleted and reside in the project.
Permissions	As the owner of the model or a user with full permissions, you can change the access level for a model on a global or individual basis. Individual access assignments override global access. Access levels include no access, read-only, or full read and write permissions.

Table 9: Tools menu

4.9 Tools menu

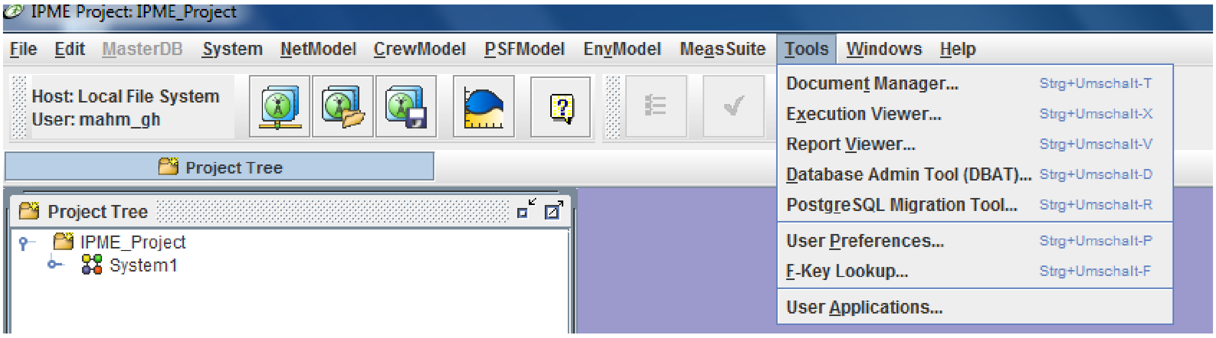


Figure 17: Tools menu

Document Manager	Before you can add any documents to the Document Manager, you must define at least one editor. After adding an editor, you can add document folders and then add the documents. The topmost folder is titled Documents, and the nodes immediately below that are the user-defined folders, which are groupings of documents. Documents associated with each grouping are listed below the folder name. From the Document Manager window, you can add, rename, and delete folders and documents, associate a document with its editor, and open a document for viewing.
Execution Viewer	To collect an Execution Viewer file, select the Output Execution Viewer File check box in the Main tab of the Execution Settings dialogue box in IPME. An output file is generated. Execute the system more than once so that dynamic operator assignments are reported more accurately in the Execution Viewer data file. The Execution Viewer displays critical tasks in colour and options allow the user to only view tasks executed during a specific time range. If a workload scheduling algorithm is applied, the scheduling effects (i.e., when a task is shed, delayed, or interrupted) can also be viewed. All tasks are considered to be at the same top network level because the Execution Viewer does not accommodate subnetworks. The Execution Viewer tool has the following limitations: · Execution with Measurement Suite enabled is not supported. · Time is represented using the IPME time unit, not minutes and seconds.
Report Viewer	When you view a file as a report, you can specify parameters that define the report view. The file contents will be shown in a Print Preview dialogue according to the specified report view parameters. You can define many report views for the same file. Available Values (Snapshots Only) : Contains a listing of all the available values in the file. Selected Values (Snapshots Only) : Contains a listing of all the variables for which you would like to display the values. Each time a snapshot is taken, IPME stores the current value of each selected variable. Those stored results are displayed for selected variables. **Crews -** Enter the range of crews for which results will be generated. In the left text box, enter the first crew number for which you want to view results. In the right text box, enter the last crew number for which you want to view results. **Trials -** Enter the range of trials for which results will be generated. In the left text box, enter the first trial number for which you want to view results. In the right text box, enter the last trial number for which you want to view results.

	Runs - Enter the range of runs for which results will be generated. In the left text box, enter the first run number for which you want to view results. In the right text box, enter the last run number for which you want to view results.
Database Admin Tool (DBAT)	The Database Administration Tool (DBAT) is a Java-based application created to assist users in managing IPME and Master databases. Users can view a list of and delete databases, add and remove users, and view, assign, and revoke access to any of the IPME or Master databases. The MySQL root password is used to log into the Database Administration Tool.
Postgre SQL Migration Tool	Migrate projects from MySQL to PostgreSQL
User Preferences	
E-Key Lookup	· F1 opens Help for the currently displayed screen. Pressing F1 is the same as clicking the Help button in any open dialogue box. · F2 checks the model for valid syntax. · F3 displays a variable list of all available system variables. · F4 displays a file browser. You can select a file from the file browser to be inserted into any text field within IPME. The text field must be active for the file browser to display. · F5 causes the currently selected system to execute. · F6 will step through execution of the currently selected system. · F7 causes the currently executing system to pause execution. · F8 causes the currently executing system to abort execution. · F9 expands/contracts an expression field. · F10 moves to the previous expression field. · F11 moves to the next expression field.
User Application	You can create menu shortcuts to run other applications, such as frequently used statistical packages, from inside IPME. You can specify the parameters to launch the external application. The list of user applications is specific to each user for a specific machine. Each user may define a separate list of user applications.

Table 10: Tools menu

4.10 Windows menu

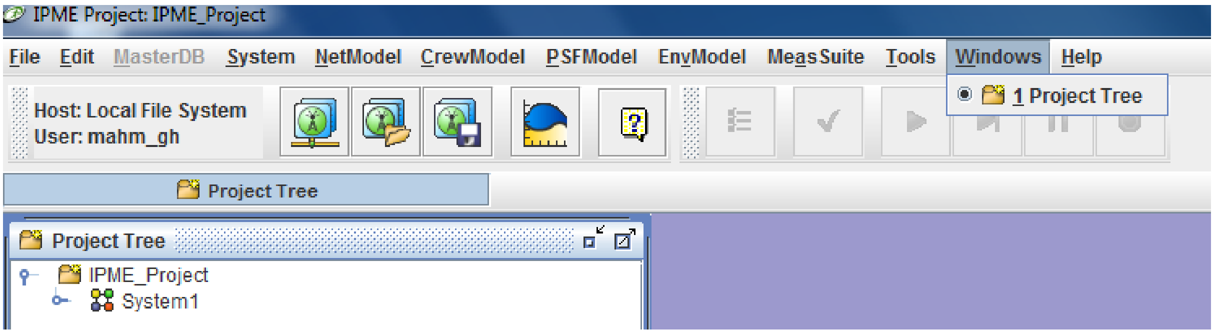


Figure 18: Windows Menu

1 Project Tree	displays all current open window

Table 11: Windows Menu

4.11 Help menu

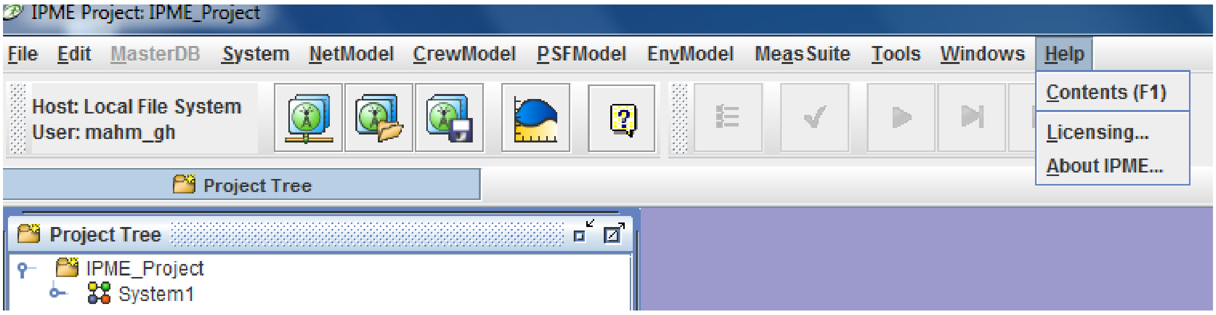


Figure 19: Help menu

Contents (F1)	Open IPME's Assistant and get a search box to type in. Word displays possible matches for you to read about. See an index of all topics available in IPME's Help documentation.
Licensing	This dialogue displays the location of the licence file, the licence type,company name to which the licence is registered, registered contact person,registered machine name, registered users, the date after which the licenceis valid, and the expiration date.
About IPME	This dialogue displays contact information for Alion Science and Technology, technical support contact information, the IPME application version, and the IPME and Master Database versions.

Table 12: Help menu

5 IPME System model

The IPME system model consists of four component models, the measurement suite, and an external model. The component models are the task network model, environment model, crew model, and performance shaping function model; the external model is the optional intranet or internet connections to other IPME client models or other software applications. The measurement suite can be used for data collection. The integration of these models creates a unique environment for human system modelling. Once models are created, they can easily be assigned or unassigned from a system model.

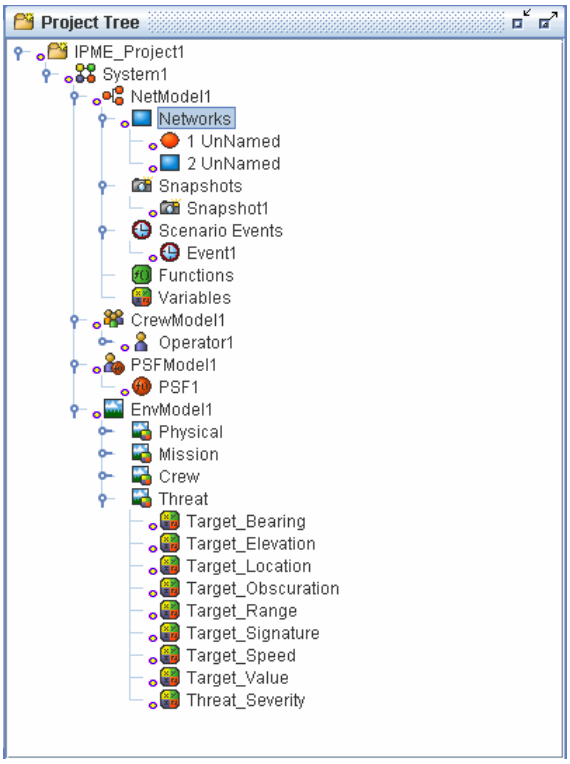


Figure 20: Project Tree

5.1 Task Network Model

The task network model consists of the following interrelated parts: network simulation modeling, and a collection of micro model functions. Micro models use empirical human performance data to calculate human performance times that affect the network simulation model.

Network Simulation

The network simulation is a hierarchical modelling tool
with an interface that is flexible enough to model any process that can be represented in a flow diagram as a network of tasks. The degree of complexity in the models is up to you. You can build a simple, functional model of the process that a human operator must go through to perform some activity by just drawing a network diagram and filling in some task timing information. You can also build more complex models that include dynamically changing variables, probabilistic and tactical branching logic, conditional task execution, human performance task time information calculated by the micro model functions using the work space design, and extensive data collection. You specify all of these by making menu choices or providing expressions to execute under specific circumstances.

Micro model functions

Micro model functions are calculation algorithms that calculate various types of human performance times for general classes of activities. Most of the micro model functions calculate times for very detailed activities such as eye fixations or button pushing. The link to the task network diagram is through the micro model functions. The value that is calculated from the micro model can be used in any IPME expression field.

5.1.1 Network Diagram

This chapter describes the task network diagram. IPME incorporates a network diagram for viewing graphical representations of the task network model. Any changes that are made to the task network in the network diagram are reflected in the project tree and any changes made in the project tree are reflected in the task network diagram.

The network diagram contains all the task network model objects, including networks, tasks, queues, and decisions. The network diagram helps to clearly identify starting tasks and the task sequence. The network diagram also offers a tool palette, an overview window, and cut-paste, copy-paste, and undo-redo capabilities. You can easily navigate multiple windows using the Windows

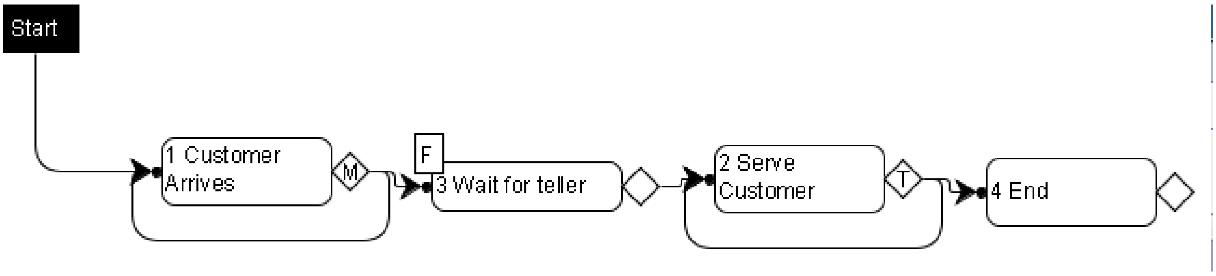


Figure 21: Task Network Model

5.1.2 Snapshots

Snapshots are used for custom data collection during simulation. A snapshot records the values of particular variables at specified points during model execution. These data can then be used to generate statistics and plots describing model execution. Defined snapshots can be enabled in the Execution Settings dialogue box.

Snapshot Parameters

Name

Contains a name for the results document where the data is to be stored. The default name is *Snapshot#*, where # is the next available integer. You can rename the snapshot to reflect its contents. Snapshot names have the same restrictions as variable names.

Description

Contains a description of the snapshot; this field is used for documentation only.

Triggered By

Selections available to trigger the snapshot include the following:

a) End of Run

Triggers the snapshot at the end of the run.

b) Task

Triggers the snapshot at the beginning or ending of the task. You specify the following: **Selected Task -** Name and ID number of the task that will trigger the snapshot. **Task Begins -** Snapshot is triggered at beginning of task. **Task Ends -** Snapshot is triggered at ending of task.

c) Queue

Triggers the snapshot as the queue is entered or departed. You specify the following: **Selected Queue -** Name and ID of the queue that will trigger the snapshot. · **Entering Queue -** Snapshot is triggered at beginning of queue. **Departing Queue -** Snapshot is triggered at ending of queue.

d) Clock

Triggers the snapshot at a specific time. Events can happen once (Discrete) or they can repeat over specified intervals (Repeating).

Discrete snapshots are triggered when the clock reaches the time specified for the snapshot. You enter the following:

Start Time - The time the snapshot will occur.

Repeating snapshots execute at a specific clock time and repeat at regular intervals until the end time specified or the end of the run. You enter the following:

Start Time - The time the snapshot will occur. **Repeat Interval -** The amount of time between snapshots.

Repeat Until - The point in the simulation when the repeating snapshot will cease to repeat. Choose one of the following: **Time -** The ending time for the snapshot. Enter the ending time in the Stop Time field. **End -** The snapshot repeats until the simulation is finished.

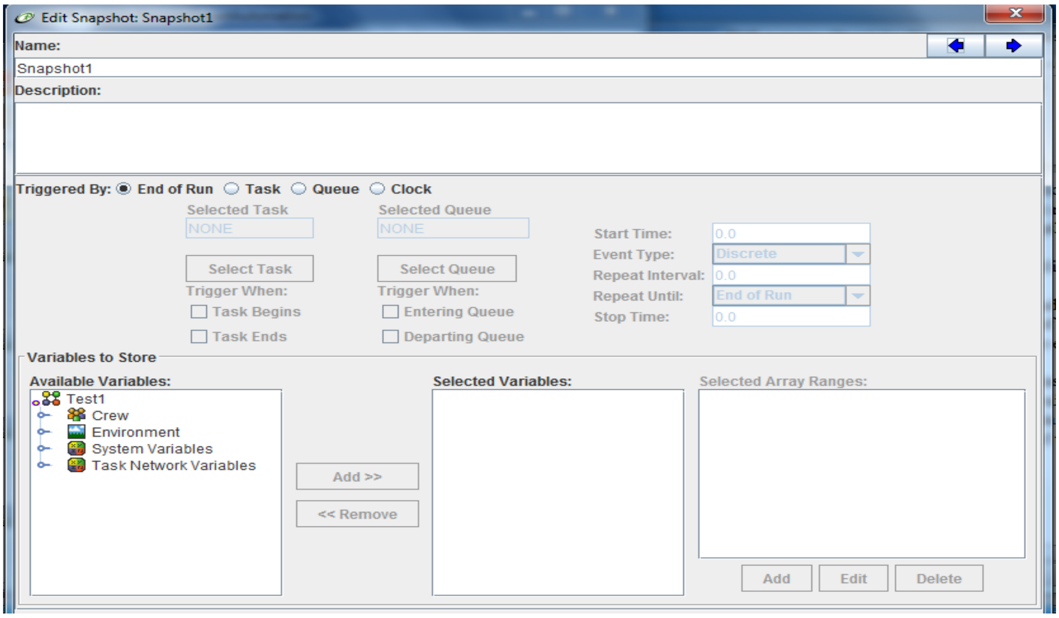


Figure 22: Snapshot

5.1.3 Scenario event

Scenario Events are time- or condition-based events that you can use to change the state of the simulation. Events can be used to start sections of the network or assign values to variables, independent of when an entity begins or ends a task or enters or departs a queue. The expressions assigning the values can be scheduled to occur at a specific clock time or when a specific condition is met in the model. You can use events to start, stop, suspend, or resume specific tasks or groups of tasks The scenario events to execute are specified in the Execution Settings dialogue box.

Scenario Event Parameters

Name - Type the name for the scenario event in this field. Scenario events have a default name of *Event#* where # is incremented each time you define a new event.

Description - Contains notes and comments about the event.

Triggered By -When you click the Triggered By button, a drop-down menu displays a list of conditions to trigger the event. Drop-down menu include the following:

a) **Time -** triggers the event at a specific time. Events can happen once (Discrete) or they can repeat over specified intervals (Repeating).

aa) **Discrete** events are triggered when the clock reaches the time specified for the event. You enter the following: **Start Time -** The time the event will occur.

ab) **Repeating** events execute at a specific clock time and repeat at regular intervals until the end time specified or the end of the run. You enter the following: **Start Time -** The time the event will occur. **Repeat Interval -** The amount of time between events. **Repeat Until -** The point in the simulation when the repeating event will cease to repeat. Choose one of the following: **Time -** The event will end a specific clock time. Supply the ending time for the event in the **Stop Time** field. **End -** The event repeats until the simulation is finished.

b) **Event -** Triggers the event when a specific condition is met during the simulation. You enter the following:

Event Condition - The conditional expression that triggers the event.
The conditional expression is a logical expression that tests the value of a variable against a value of another variable.
Expression - The expression that executes when the condition is met.

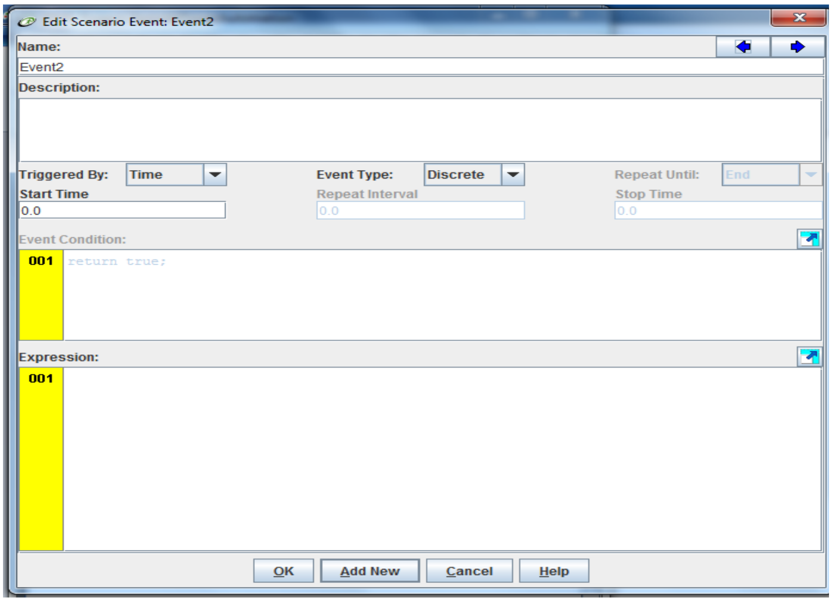


Figure 23: Scenario Event

5.1.4 Function

5.1.4.1 System function

Functions are used to group a complicated expression or group of expressions in a model. In all instances where the function name is encountered in the model, the expressions are executed.
The user-defined functions for the network display towards the bottom of the list of network elements. The function list displays functions that are defined for the current model. You can add, edit, or delete user-defined functions.
The Edit Function dialogue box contains the following parameters:
Return Type - Select the return type for the function. Options include int (whole numbers), double (numbers that are not whole numbers), string (variables that consist of alphanumeric characters), boolean (true/false) and task_id (values in the format of a task or network ID number, such as 2.1.9).
Name - Contains the name of the function. All user-defined functions have a default name of *Function#* where # is incremented each time you define a new function. You can use this default name, but it is recommended that you name the function something that reflects its use. It is also recommended that you capitalise function names to distinguish them from other expressions in your model. Function names have the same restrictions as variable names.
Description - Contains notes and comments about the function. This is an optional field.
Arguments - Arguments can be passed to a user-defined function. Arguments appear in parentheses next to the function name in the project tree. For example, *(int status)* means that an integer variable named *status* will be passed to the function. The arguments passed to functions can be integers, doubles, strings, booleans, or task_id.

Expressions - Contains the expressions defining the function. Be sure to separate expressions with a semicolon.

A user-defined function is useful when you have a complicated expression or group of expressions used in your model. In all instances where the function name is encountered in the model, the expressions are executed.

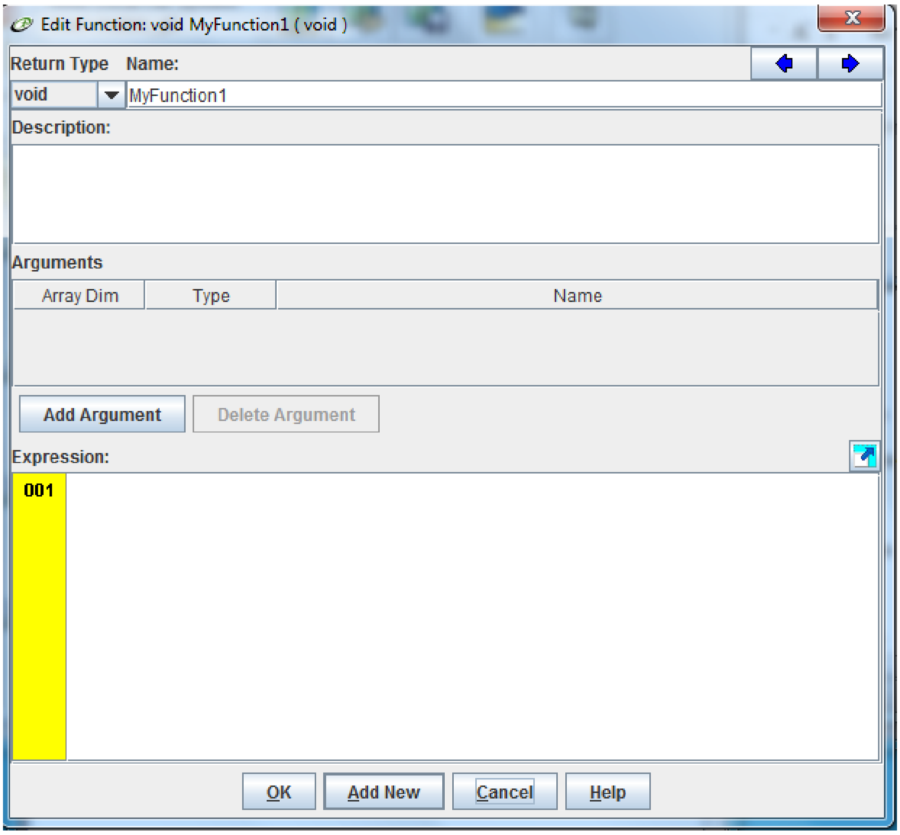


Figure 24: Function

5.1.4.2 System micro models Function

Micro model functions are included with IPME and show some of the various types of human activity models that exist. These activities range from the very detailed movements such as eye movement and eye fixation to larger movements such as walking. For some of the micro model functions, the time values have already been calculated. For others, the time value is calculated based on the values passed to the function.

IPME provides the following classes of Micro Model functions:

· Cognitive Micro Model Functions: Cognitive micro model functions address time required for thought processes, such as simple reaction times, response to an on or off condition, reaction time requiring a physical match, and times for the perceptual, motor, and the cognitive subsystem.

· Motor Micro Model Functions: Motor micro models address time required for small motor movements, such as moving a cursor, moving a hand to a target, pushing buttons, turning rotary dials, and typing rates.

· Perceptual Micro Model Functions: Perceptual micro model functions address times required to visually locate a target such as eye fixation, movement, and head movement times. Listening, reading, and searching rates and visual acuity are also included.

5.1.5 Variables

5.1.5.1 System variables

clock or Clock - Records elapsed time (in simulation time units) since the beginning of model execution. *clock* is a variable of double type. You can use *clock* in any expression in your model, but use care in changing its value.

duration - Records the time each entity spends in the current task or queue. *duration* is a variable of double type. You can use *duration* in any expression in the ending effects for a task, but use care in changing its value.

group - Records the identity of a collection of tasks, similar to the *tag* variable. *group* is a variable of integer type. The group number must be an integer. Groups can be used to stop, suspend, and resume a group of tasks.

meantime - Represents the mean time for the current task. m*eantime* is a variable of double type. This variable is only valid after mean time for the task has been determined (otherwise the value is zero).

run - Records the current run number in cases where there are multiple runs. You can use *run* in any expression in your model. *run* is a variable of integer type

tag = tag + 1 - to increment the tag numbers. Once an entity has a *tag* value, the value stays with that entity through the remainder of model execution or until you change it. When you use *tag* in an expression, IPME interprets it as the number of the entity that is currently being processed.

Split tags - Whenever you have a multiple decision that routes entities to more than one following task, IPME splits the entity into multiple entities that have the same *tag* value as the original entity. If you are using tag values as unique identifiers, you need to be sure either to renumber the "extra" entities, remove them from the model, or merge the split entities later.

Rejoining tags - IPME automatically rejoins any entities that have the same tag number if they both enter an intersection at the same time.

5.1.5.2 Variable

Variable Parameters:

Name - Contains the name for the variable.

Description - This is an optional field that can contain any descriptive notes about the variable.

Type:

Boolean - flag variables. The boolean variable is often used to determine whether a condition is true or false. Booleans can be initialised as *false* or *true*.

Integer - whole number. Variables that count discrete objects should be defined as integers.

Double - numbers that are not whole numbers. IPME displays double variables with six digits to the right of the decimal point. Variables that store clock values or measurements of other continuous quantities should be defined as double variables.

Task_id - values in the format of a task or network ID number, such as 2.1.9. Task_ids can be used to control task execution.

Array - an ordered set of values or elements identified by a single name. Each array element is referred to by the array's name and the element's position in the array.

For each array, you must also assign the following:

· **Array Dimensions -** Specifies the number of dimensions in an array. Click the button adjacent to the appropriate array dimension.

·

Highest Index - Highest Index specifies the highest index number in the array. Be sure to define the index range large enough to handle all the possible elements you may have for the array. There is currently no limit on the size of an array.

Initial Value

An initial value of zero is assumed unless otherwise specified. The initial value of a variable is reset at the beginning of each run. If the variable is an array, the initial value applies to all elements.

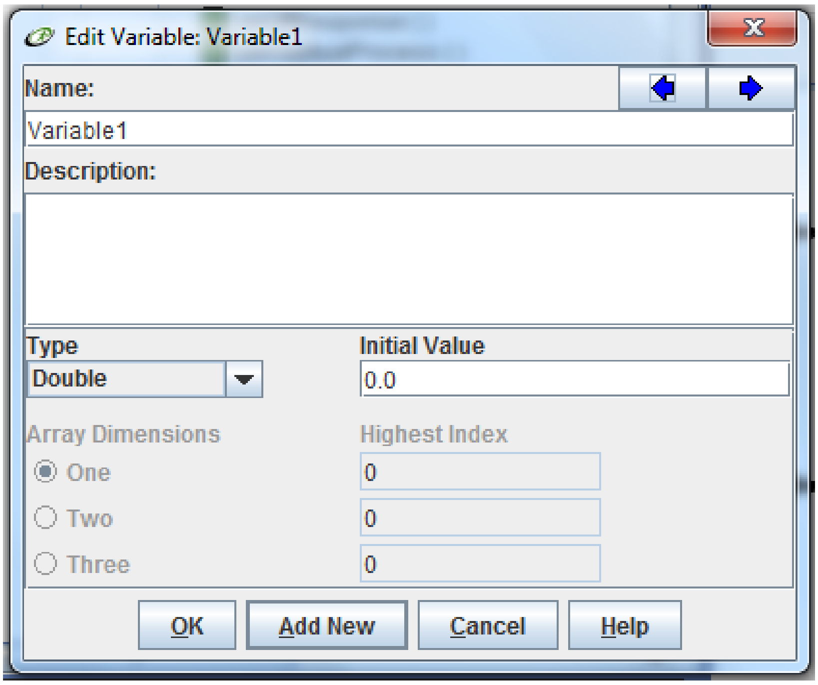


Figure 25: Variable Dialoge

5.2 Crew Models

A crew model consists of a group of operators that are available for use in taskassignments. An operator is defined by the characteristics properties, traits,states, and anthropometry. Operator variables refer to factors affecting the operator, such as clothing, hunger, thirst, or physical fatigue. Operator variablesare assigned in the crew model and can depend on environment expressions. Operator assignments can easily be made at the task level in the Task Information dialogue box. Operators can also be assigned a work load for each task. The work load level is dependent on the network task the operator is executing. Additionally, IPME assigns the percentage demand for the abilities required to execute a task by the operator based on the performance shaping functions.

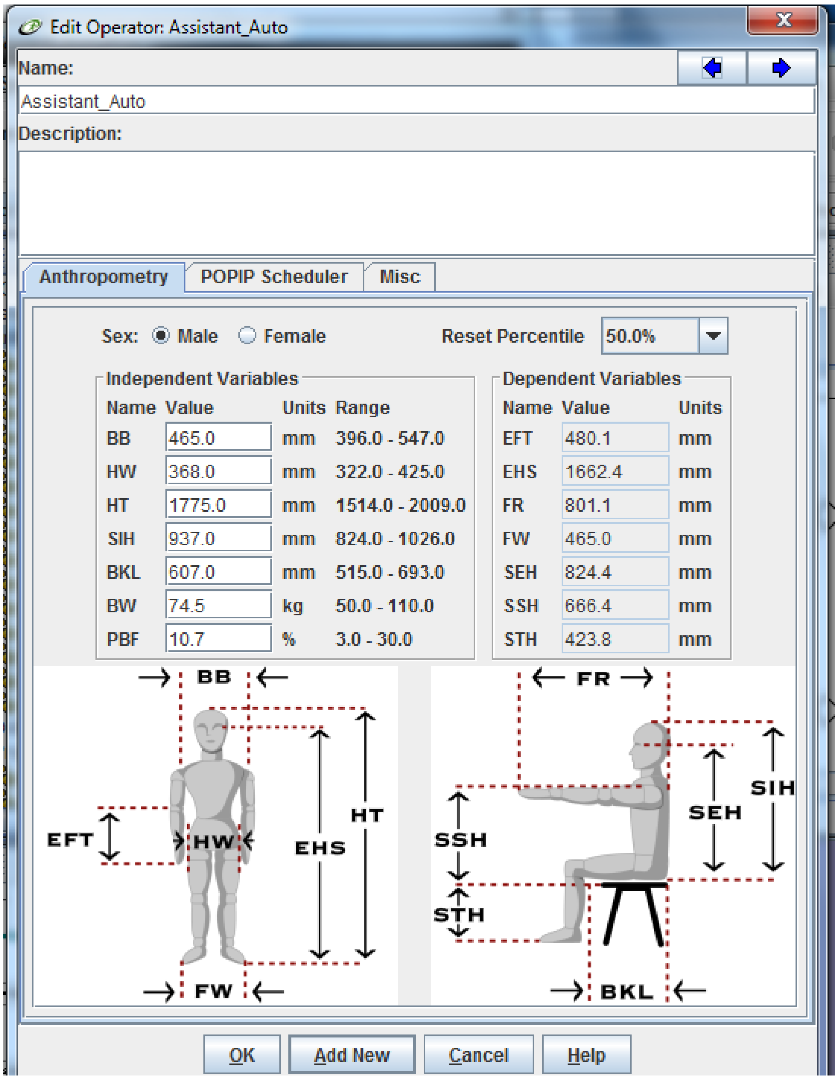


Figure 26: Operator, Anthropometry

• **Properties**

consist of the default physical characteristics for each operator: Eyes, LeftFoot, RightFoot, LeftHand, and RightHand. These default properties are fixed for IPME, but you can add other properties. An attribute is a single value that indicates the status of the operator.

• **Traits**

describe non-physical operator personality characteristics such as agility, susceptibility to motion sickness, fitness, or cognitive ability. Predefined traits are available by linking to an operator in the master database list and cannot be deleted, but you can add and delete userdefined traits. Traits are generally viewed as those characteristics of a human that *will not change* as a function of the simulation.

• **States**

are operator characteristics that can be formulated from the combination of operator properties and traits in conjunction with environment and task network variables. States include factors such as mental alertness, clothing, encumbrance, hunger, thirst, morale, or fear. Predefined states are available by linking to an operator in the master database and cannot be deleted, but you can add and delete user-defined states. States are generally viewed as those characteristics of a human that *can* change as a function of the simulation.

• **Anthropometry**

can be used to determine whether an operator violates reach or fit restrictions when the reach or fit is modeled through expressions.

5.3 Environment Models

The environment model consists of a set of environment variables. Environment variables are based on external factors that make up the scenario and consist of the following:

· **Physical variables**

include factors of the physical environment such as temperature, humidity, wind, and terrain.

· **Mission variables**

include communications, intelligence, and weapons reliability.

· **Crew variables** relate to how well the group performs as a team, and includes factors such as supervision, leadership style, morale, and supervision.

· **Threat variables** refer to the degree of threat from an enemy, and the position of the target.

Operator characteristics can depend on environment factors such as clothing, hunger, thirst, or physical fatigue. Operator characteristics are assigned in the crew model.
PSF expressions can depend on or use environment variables to build stressor effects on modelled human performance.

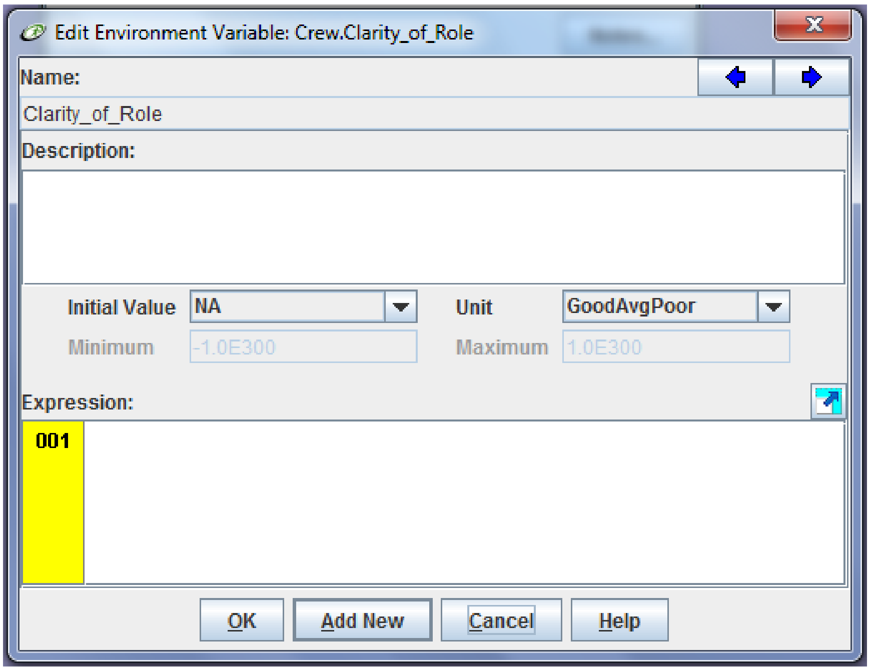


Figure 27: Environment Variable

5.4 Performance Shaping Function Models

The relationship between the human-environmental factors is measured in IPME with performance shaping functions. These are a group of conditions that may influence an operator while he or she is performing a task or activity, and can degrade the task time and task accuracy. The time to perform and/or the error rate may be altered as a result of the performance shaping function. Performance shaping functions are qualified with a taxonomy that classifies tasks according to basic human skills. Degradation functions are then created for each performance shaping function.

The taxonomy for skills used by IPME is as follows:

• **Attention** – *Vigilance*.

The ability to actively attend to a complex stimulus for extended periods of time to detect specified changes that indicate the occurrence of some phenomenon that is critical to task performance. Attention is required for a task such as driving a vehicle. In this case, attention would be a combination of cognitive and sensory resources, but also has the added characterisation that the person must be focused on the task.

• **Perception** – *Visual*, *Auditory*.

The ability to detect and categorise specific stimulus patterns. Visual perception would be used for tasks where the operator would seesomething, like detecting a target or reading a book. Auditory perception would be used for tasks where the operator was required to hear something, such as a warning bell.

• **Cognition** – *Spatial, Verbal/Numeric*.

The ability to apply concepts and rules to information from the environment and memory to determine a course of action. This includes communicating the course of action to others. Spatial represents cognition on spatially displayed materials, such as maps. Verbal/Numeric represents cognition on words or numbers, including mathematical calculations.

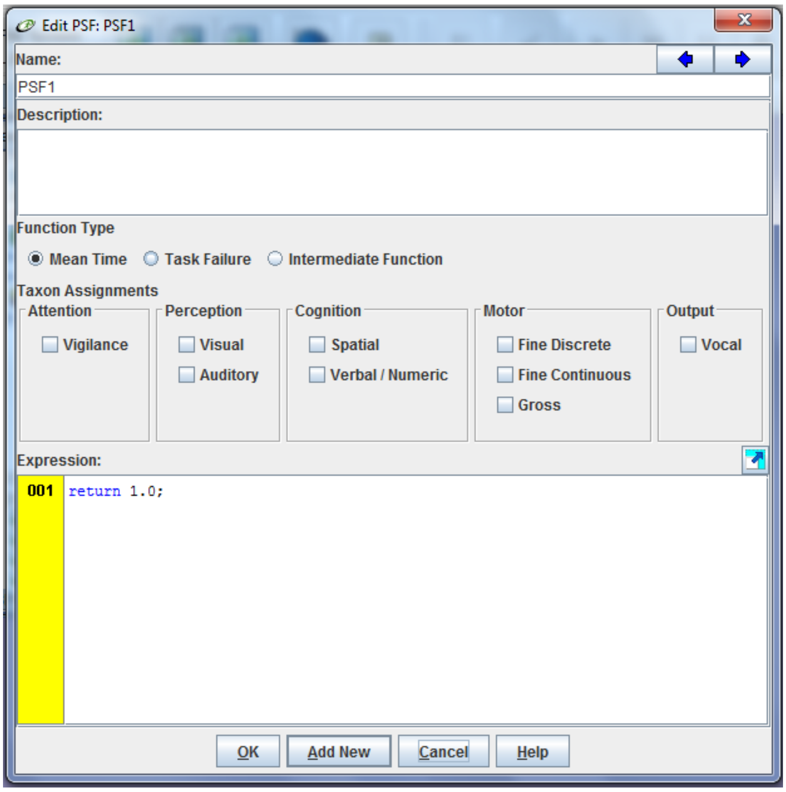


Figure 28: Performance Shaping Function Models

• **Motor** – *Fine Discrete*, *Fine Continuous*, *Gross.*
Physical action using parts of the body, such as arms or legs. Fine Discrete motor skills are motions such as setting a dial. Fine Continuous skills are motions such as tracking with a mouse. Gross motor skills are movements such as lifting or moving heavy materials, closing a door, or other motions not demanding a significant amount of attention.

• **Output** – *Vocal.*
The ability to send a vocal message. The vocal skill would be used for a person to speak.

PSF equations are normalised around the value 1.0, with 1.0 representing no performance effect (either enhancement or degradation). The following table lists whether the resulting task performance effects would be an enhancement or a degradation.

Category	**Evaliuates Less Than 1.0**	**Evaluates to 1.0**	**Evaluates to Greater Than 1.0**
Mean Time	Decrement (longer task mean time)	no change	enhancement (shorter task mean time)
Task Failure Probability	Decrement (higher probability of failure)	no change	enhancement (lower probability of failure)

Table 13**:** PSF-equations

Example:

The formula for the mean time PSFs is the following:

((Mean time * Taxon_1 %) / (PSF_1 * PSF_2*. . .*PSF_n)) + ((Mean time * Taxon_2%)/ (PSF_1 * PSF_2 * . . .PSF_n) + . . .

	Example 1	Example 2	Example 3
Mean Time	10	10	10
St. Dev.	0	0	0
Taxon & %	vigilance 100%	vigilance 50%, none 50%	vigilance 50%, auditory 50%
PSF Expression	return .8;	return .8;	return .8;
PSF Taxon	vigilance only	vigilance only	vigilance, visual

Table 14**: PSF** Result: (mean time * taxon %) / (PSF Expression)

Example 1: 10 / .8 = 12.5

Example 2: ((10*.5) / .8) + ((10*.5)/1) = 6.25 + 5 = 11.25

Example 3: ((10*.5) / .8) + ((10*.5)/.8) = 6.25 + 6.25 = 12.5

5.5 Measurement Suite

The optional measurement suite can be used to encapsulate the results produced by varying a specific set of variables in a disciplined manner. The results can be grouped into blocks of data by using the blocking factors. Results can be triggered by the end of run, a specific clock time, or by a task or queue.

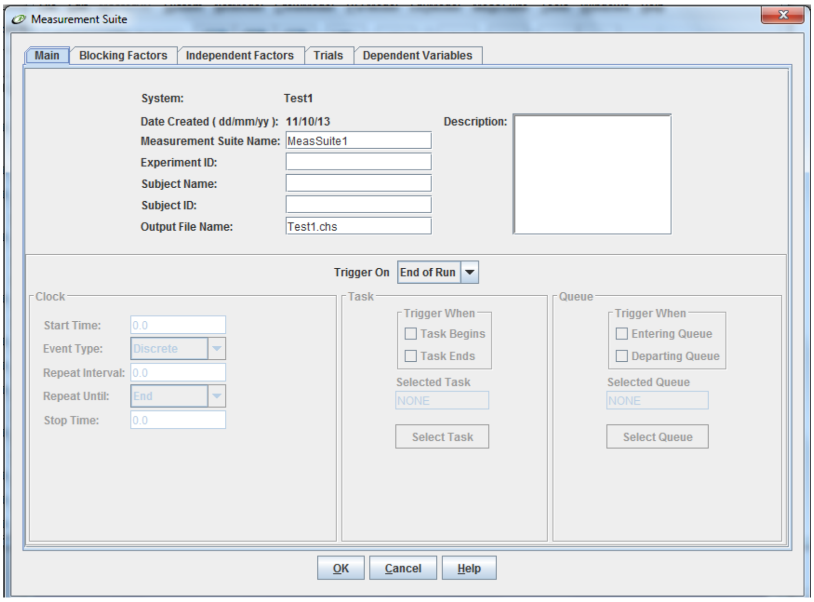


Figure 29: Measurement Suite

5.6 External Models

A system description can also include external models. These models can be custom-developed applications or other IPME simulators that execute across a network. State information is transferred between simulations through common variables.

6 Task

A task is the basic building block of a model and contains timing information, execution constraints, effects of the task, POPIP, and workload data.

6.1 Main Tab

6.1.1 Timing and Effects Subtab

Time Distribution

- This field has a drop-down menu with a list of distributions used for calculating execution times. Available distributions include the following: normal, gamma, exponential, rectangular, and lognormal. Select a distribution to define the appropriate timing for the task.

Type

- Displays the type of task: discrete, repeating, or continuous. The task type can also be selected on the POPIP tab using the Scheduling subtab

Mean Time

- The mean time field is a value that represents the average number of time units (this can be seconds, minutes, hours, and so on) to execute a task.

Standard Deviation

- Contains the number or expression for the standard deviation (in time units) to use in task time calculations. The actual execution time for a discrete task is based on the time parameters in this window

Release Condition

- This field contains an expression that determines whether the task can execute. A task can only execute when the logical evaluation of the release condition is true. You supply the release condition when you define the task. The default value is *true*.

Beginning Effects

- This field contains an expression(s) indicating what happens when the task begins execution. Use this field to change the value of variables that reflect the state of the system when a task begins executing.

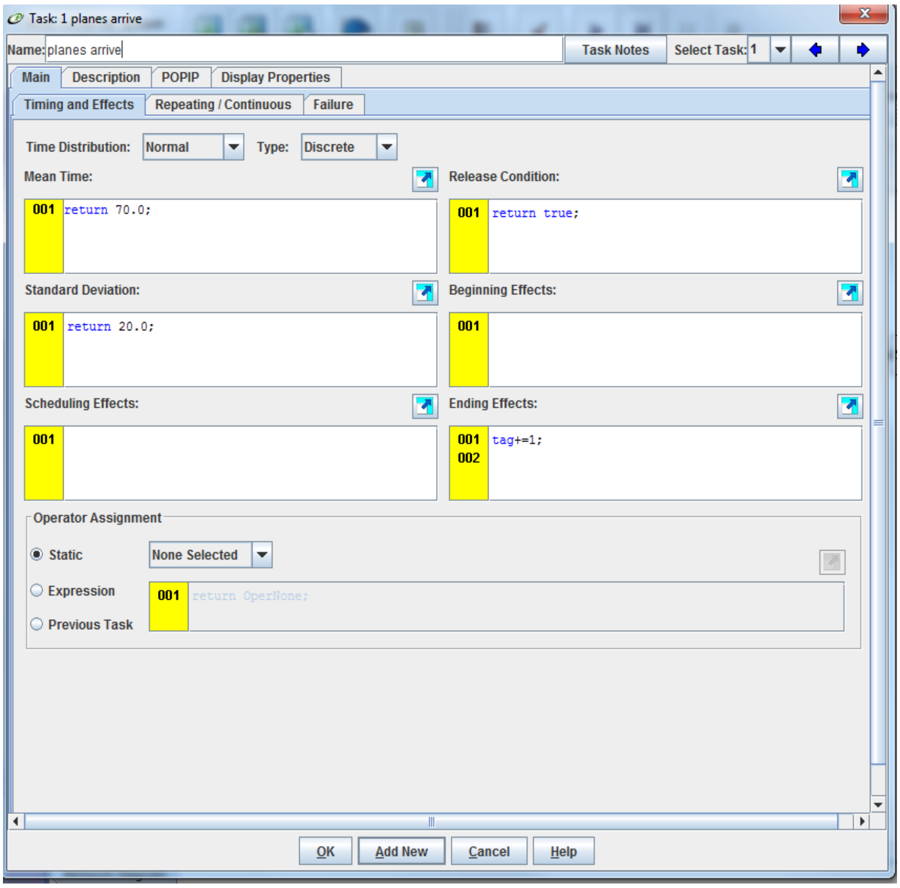


Figure 30: Task - main Tab, Timing and Effects Subtab

Ending Effects

- This field contains an expression(s) indicating the state of the simulation once a task has executed.

Operator Assignment

- Three types of assignment are available:

Static

- the same operator always performs the task. Select the operator from the pull-down list.

Expression

- the operator specified in the expression performs the task. Select the **Expression** radio button and then type the operator assignment in the text box. The assignment must be in the form of one of the following:

6.1.2 Repeating/Continuous Subtab

Repeating

- Repeating tasks are similar to continuous tasks except that the active period is defined by the *active time*. The total task cycle time is: (active time & active time standard deviation) + ((cycle time-active time) & cycle time standard deviation).

When this type of task is selected, fields pertinent only to other task types are disabled.

Cycle Time

- The cycle time is the inactive portion for a repeating task. The cycle time is used for repeating tasks only.

Cycle Time Standard Deviation

- This field contains the standard deviation for the cycle time. It is applied to (*cycle time-active time*). The cycle time standard deviation is used for repeating tasks only

Active Time

- Active time is the time actually spent performing the task (attending time). Active time is used for continuous and repeating tasks.

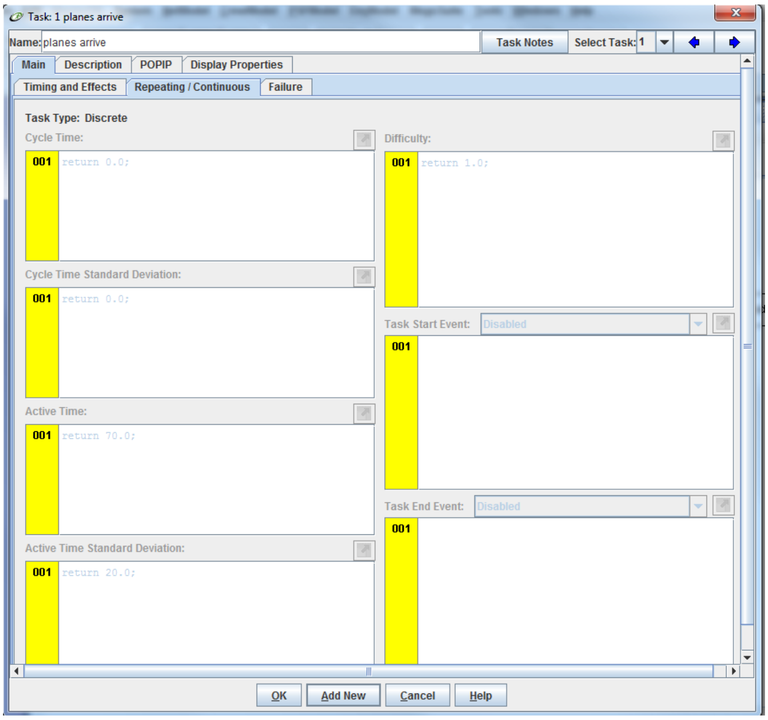


Figure 31: Task - main Tab, Repeating/Continuous Subtab

Active Time Standard Deviation

- This is the standard deviation for the active time. The active time standard deviation is used for continuous and repeating tasks.

Difficulty

- This is a subjective rating of the task difficulty that is applied to processing time. Permissible values are between 0 and 1 inclusive; the default is 1. **Task Start Event** - This field contains the time or condition for a continuous or repeating task to begin execution. Expressions or numeric values are allowed. Click this button to specify whether the task starts at a specific time or when a specific condition is met. The Task Start Event is used for continuous and repeating tasks.

Task End Event

- This field contains the time or condition for a continuous or repeating task to stop execution. Expressions or numeric values are allowed. Click the button adjacent to the label Task End Event to specify whether the task ends at a specific time or when a specific condition is met. The Task End Event is used for continuous and repeating tasks.

6.1.3 Failure Subtab

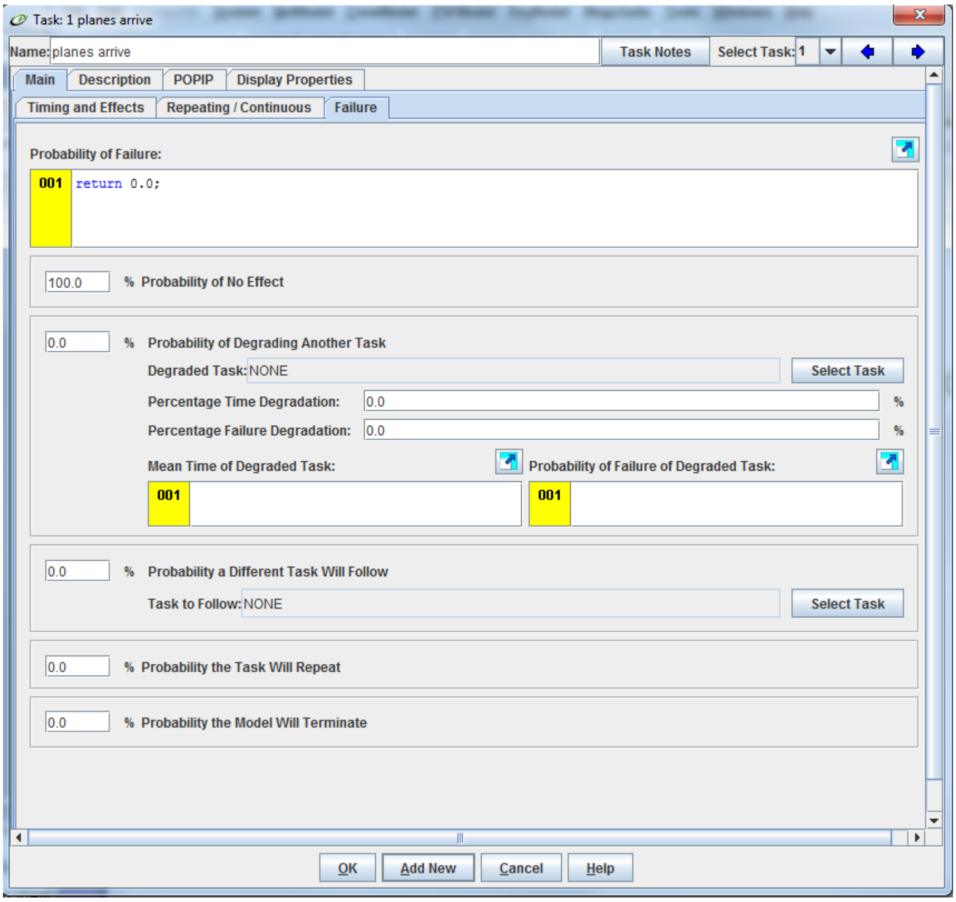


Figure 32: Task - main Tab, failure Subtab

Failure subtab contains an estimate of the likelihood that the task will be performed unsuccessfully, based on a scale of 0 to 1.0, where the value 0 indicates no failure and the value 1 represents 100 percent failure. If the task is very easy to perform, this field contains a very low number, such as .01. If the task is difficult to perform, the estimate will be higher, such as .3.

The probability of failure can be an expression. If the expression evaluates to a number greater than or equal to 1, the task fails. If the expression evaluates to a value less than zero, a runtime error is produced and simulation is continued as if the expression evaluated to 0. Expressions that evaluate to a value between 0 and 1 are treated as the percentage chance of failure for the task. The result of this field is accessible by using the *Entity.FailProb* variable.

Consequences of Failure

For a probability of failure greater than zero, five consequences can occur if the task is not successfully completed. Assign the probability that each consequence will occur when the task fails. The consequence probabilities must sum to 100 percent.

6.2 Description Tab

6.2.1 VACP and W/Index Subtab

Within IPME Modus workload is calculated by means of the Demand Algorithm (VACP und Workload Index (W/Index)). VACP measures operator workload within four separate channels (Visual-, Auditive-, Cognitive-und Psychomotor) . VACP and Workload Index (W/Index) data measure the resource demand imposed upon an operator. VACP measures within resource demand. W/Index provides an overall demand measure and includes both a sum of within resource demand and the sum of inter-resource demand (Multiple Resource Theory). These are included in the lower part of the Task Assignment (and Workload) dialogue box.Select the VACP values from the drop-down lists for each channel and enter the values for the W/Index channels. You can manually change any of the VACP weights.

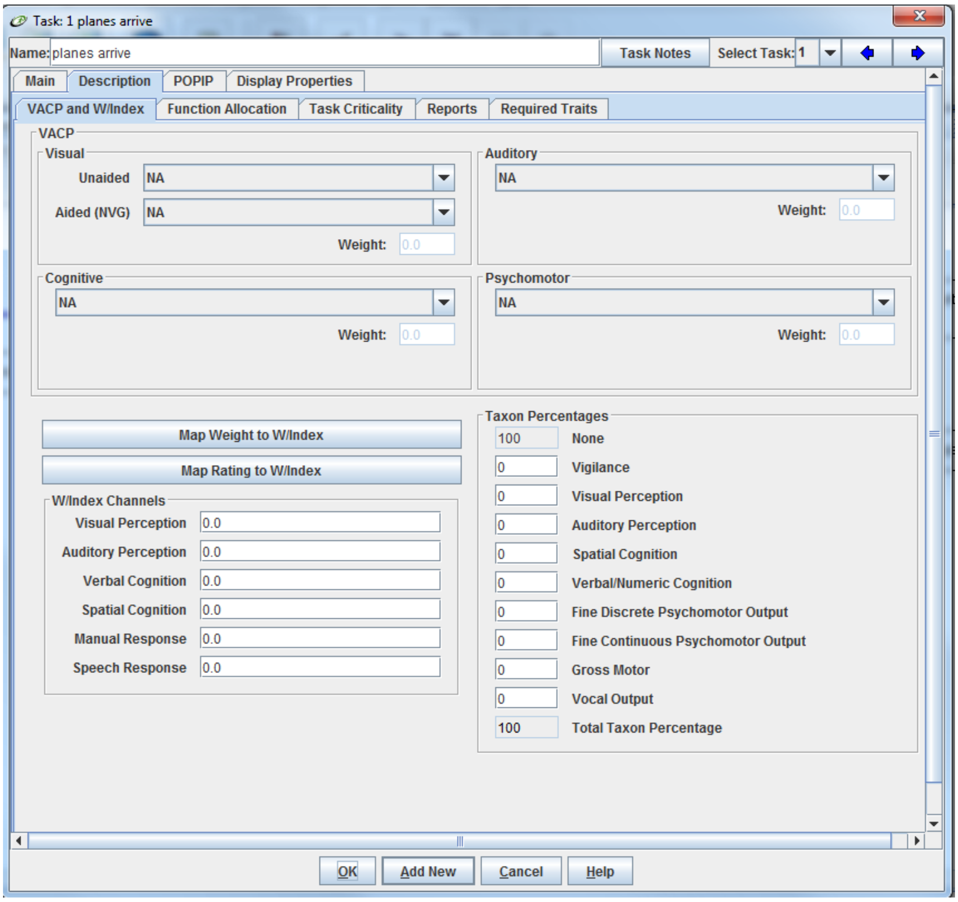


Figure 33: Task – description Tab, VACP and W/Index

The channels used for this model include visual perception, auditory perception, verbal cognition, spatial cognition, manual response, and speech response. Ratings are based on ordinal categories (integers from 1 to 7) as shown in the following table.

Ordinal Rating(ordnungszahl Bewertung)	**Interval Rating(Zeitraum bewertung)**	**Descriptors**	**Maps to**
Visual - Unaided (Naked Eye)			
1	1.0	Register/Detect (Detect Occurrence of Image)	Visual Perception
2	3.7	Discriminate (Detect Visual Difference)	Visual Perception
3	4.0	Inspect/Check (Discrete Inspection/Static Condition)	Visual Perception
4	5.0	Locate/Align (Selective Orientation)	Visual Perception
5	5.4	Track/Follow (Maintain Orientation)	Visual Perception
6	5.9	Read (Symbol)	Visual Perception
7	7.0	Scan/Search/Monitor (Continuous/Serial Inspection, Multiple Conditions)	Visual Perception
Visual Aided (NVG - Night Vision Goggles)			
1	1.0	Register/Detect (Detect Occurrence of Image)	Visual Perception
2	4.8	Discriminate (Detect Visual Difference)	Visual Perception
3	5.0	Inspect/Check (Discrete Inspection/Static Condition	Visual Perception
4	5.6	Locate/Align (Selective Orientation)	Visual Perception
5	6.4	Track/Follow (Maintain Orientation)	Visual Perception
6	6.7	Read (Symbol)	Visual Perception
7	7.0	Scan/Search/Monitor (Continuous/Serial Inspection, Multiple Conditions)	Visual Perception
Auditory			
1	1.0	Detect/Register Sound (Detect Occurrence of Sound) Anticipated Sound)	Auditory Perception
2	2.0	Orient to Sound (General Orientation/Attention)	Auditory Perception
3	4.2	Orient to Sound (Selective Orientation/Attention)	Auditory Perception
4	4.3	Verify Auditory Feedback (Detect Occurrence	Auditory

		of	Perception
5	4.9	Interpret Semantic Content (Speech)	Auditory Perception
6	6.6	Discriminate Sound Characteristics (Detect Auditory Differences)	Auditory Perception
7	7.0	Interpret Sound Patterns (Pulse Rates, and so on.)	Auditory Perception
Cognitive			
1	1.0	1 Automatic (Simple Association) 2	Spatial Cognition
2	1.2	Alternative Selection	Verbal Cognition
3	3.7	Sign/Signal Recognition	Spatial Cognition
4	4.6	Evaluation/Judgment (Consider single Aspect)	Verbal Cognition
5	5.3	Encoding/Decoding, Recall	Verbal Cognition
6	6.8	Evaluation/Judgment (Consider Several Aspects)	Verbal Cognition
7	7.0	Estimation, Calculation, Conversion	Verbal Cognition
Psychomotor			
1	1.0	Speech	Speech Response
2	2.2	Discrete Actuation (Button, Toggle, Trigger)	Manual Response
3	2.6	Continuous Adjustive (Flight Control, Sensor Control)	Manual Response
4	4.6	Manipulative	Manual Response
5	5.8	Discrete Adjustive (Rotary, Vertical Thumbwheel, Lever Position)	Manual Response
6	6.5	Symbolic Production (Writing)	Manual Response
7	7.0	Serial Discrete Manipulation (Keyboard Entries)	Manual Response

Table 15: VACP channels

The default Workload Index Conflict Matrix is shown here.

	Visual Perception	**Auditory Perception**	**Verbal Cognition**	**Spatial Cognition**	**Manual Response**	**Speech Response**
Visual Perception	1.0	0.3	0.0	0.2	0.2	0.0
Auditory Perception	0.3	1.0	0.2	0.0	0.0	0.2
Verbal Cognition	0.0	0.2	1.0	0.5	0.0	0.2

Spatial Cognition	0.2	0.0	0.5	1.0	0.2	0.0
Manual Response	0.2	0.0	0.0	0.2	1.0	0.3
Speech Response	0.0	0.2	0.2	0.0	0.3	1.0

Table 16: Conflict Matrix

Example:

In the following example, two tasks are assigned to the same operator. These tasks execute simultaneously. These tasks have the following Workload Index values.

Example 1	**Visual Perception**	**Auditory Perception**	**Verbal Cognition**	**Spatial Cognition**	**Manual Response**	**Speech Response**
Task 1	5	0	0	0	0	0
Task 2	4	0	0	0	0	0
Example 2	**Visual Perception**	**Auditory Perception**	**Verbal Cognition**	**Spatial Cognition**	**Manual Response**	**Speech Response**
Task 1	5	0	0	0	0	0
Task 2	0	4	0	0	0	0

Following are the Workload Index results.

Results	**Within Channel**	**Interference**	**Workload**
Example 1	9	9	18
Example 2	9	2.7	11.7

Table 17: Workload Index results

6.2.2 Function Allocation Subtab

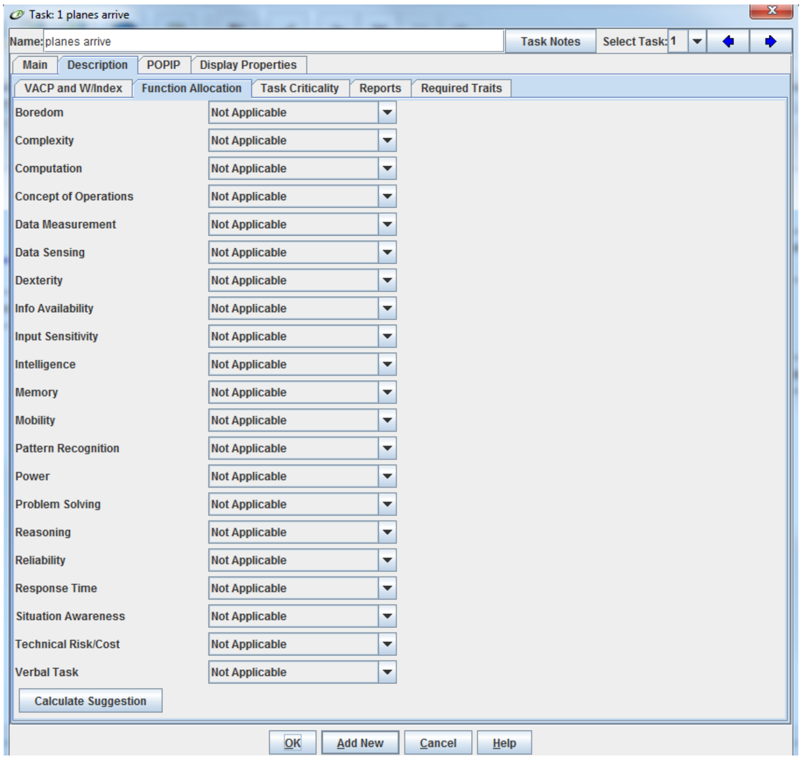


Figure 34: Task - description Tab, Function Allocation Subtab

6.2.3 Task criticality

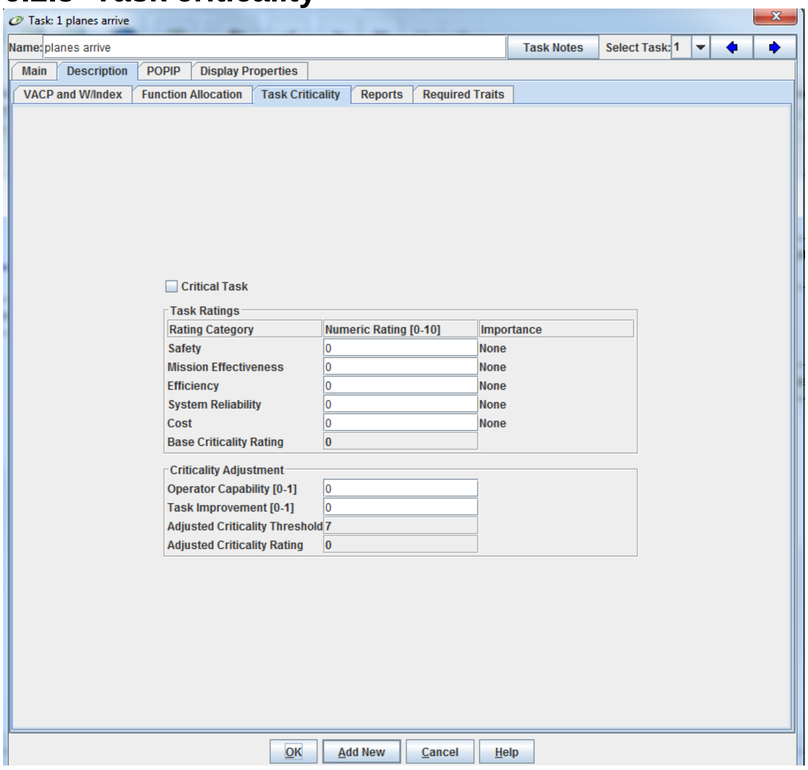


Figure 35: Task - description Tab, Task Criticality Subtab

6.2.4 Report

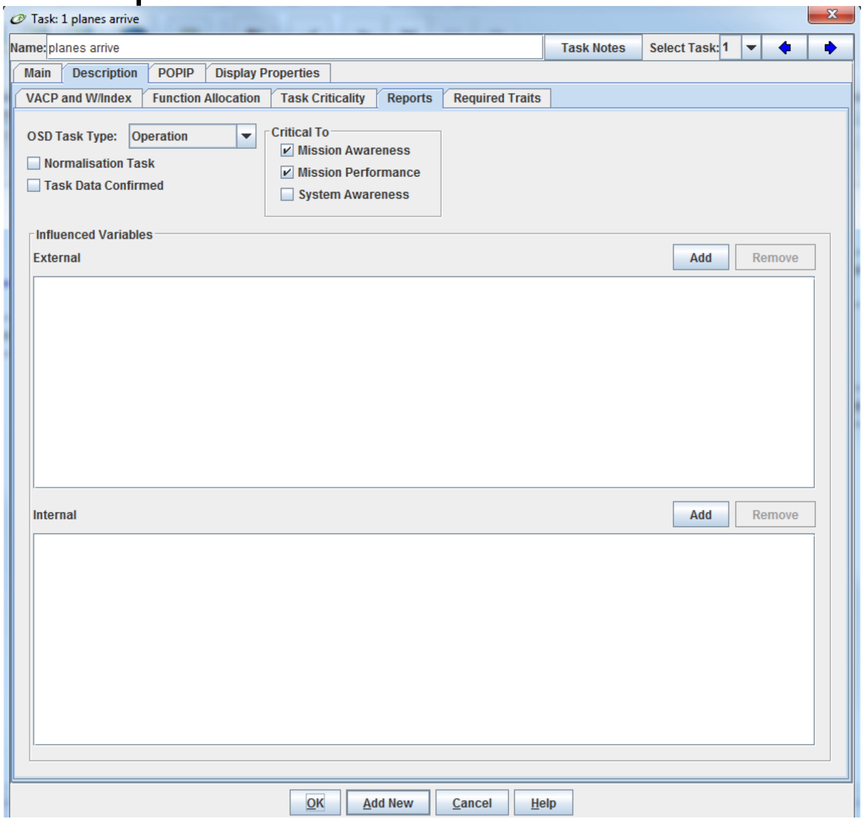


Figure 36: Task – Description Tab, Reports Subtab

6.2.5 Required Traits Subtab

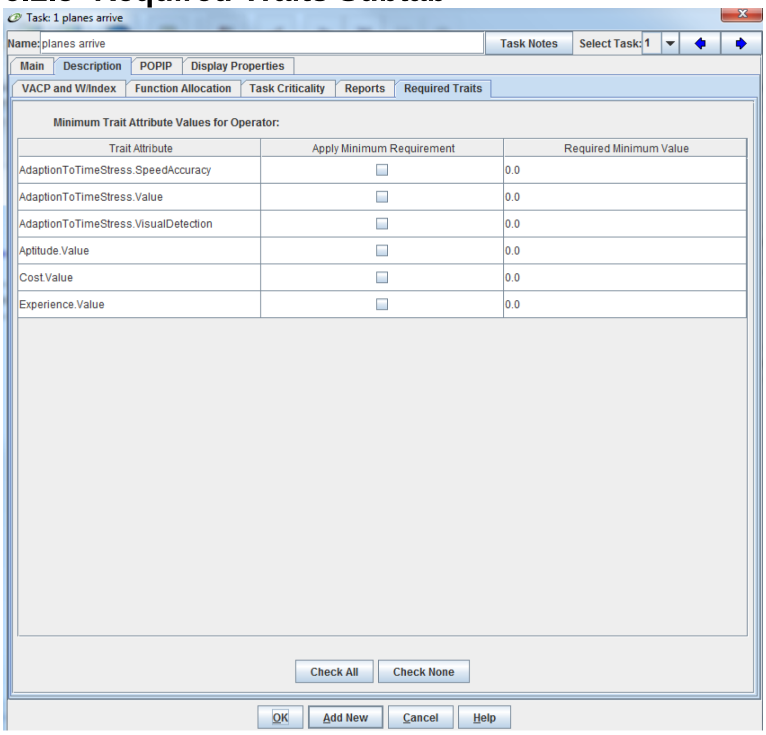


Figure 37: Task - description Tab, Required Traits Subtab

6.3 POPIP Tab

6.3.1 Scheduling subtab

This subtab contains the scheduling priorities for the task. Task type can be selected in this tab from the Category drop-down list. If the task is discrete, the fields in the Repeating/Continuous tab of the dialogue box cannot be edited. Timing parameters for discrete tasks are set in the Timing and Effects tab. If the task is continuous or repeating, different fields for the timing information in the Repeating/Continuous tab are available.

Task Pacing

Internally paced

tasks are those tasks accomplished by an operator that would be considered self-scheduled. If the operator was performing internally-paced tasks and reached the POPIP workload overload threshold, the expected real-life behaviour is to reschedule or postpone the task to prevent workload overload. Therefore, internally-paced tasks do not cause the POPIP workload calculation to exceed the workload overload threshold of 100.

Externally paced

tasks are those tasks performed by an operator that are scheduled by an external source. They are scheduled based on an external need, independent of operator availability or the operator's status in performing other tasks. For example, a time-sensitive event such as receiving a communication and detecting and reacting to a radar plot is externally-paced. Externally paced tasks can cause the POPIP workload calculation to exceed the workload overload threshold.

Scheduling and Priority

Fields in the Scheduling and Priority section of the dialogue set the priority category for the task, the initial task time pressure, and whether the tasks can be interrupted, resumed, or shed.

Category

specifies the priority for a task according to the latest processing time for the class. Categories are described in the following table. The k value in the table is the multiplier used to calculate initial task time pressure.

Interruptible

a task that can be stopped during execution. An interrupted task can be resumed or restarted later. A task that is not resumable will be restarted if possible. An uninterruptible task must run to completion once it is started.

Resumable

a task that can be continued from the point of interruption. A task that is not resumable will be restarted if possible.

Discretionary

a task that can be abandoned without penalty if prospective memory exceeds capacity.

Shed if Late

A task that is forcibly shed if the current time is greater than the latest execution time that the task can be scheduled.

Task Priority Time Pressure

Enter the expression for the category multiplier used to calculate the Task Priority Time Pressure.

Enable Operator Forgetting TPM

Select this checkbox to apply the operator's forgetting TPM expression.

Consequences of Shedding

The consequences of shedding indicates what happens when a task is shed. The following options are available:

Continue Branch

The task is shed and the model execution continues with the next connected task in the network. This is the default selection.

Terminate Branch

The branch terminates after the task is shed. If you select Terminate Branch with a repeating/continuous task earlier in the branch, then the tag's progress is stopped through the branch.

Start Other Task

Another task is started in place of the task following the task being shed. Click the Select Task button to display a listing of available tasks in the network and select one of the tasks

· **At Shed Time**

The selected task is started at the time the shed event occurs for this task.

· **At Ending Effect Time**

The selected task is started at the scheduled ending effect time for this task. The end time of a shed task is calculated strictly on the task timing information (Mean Time, Standard Deviation, and Time Distribution). It does not include adjustments to the task time due to PSF effects or TDM values.

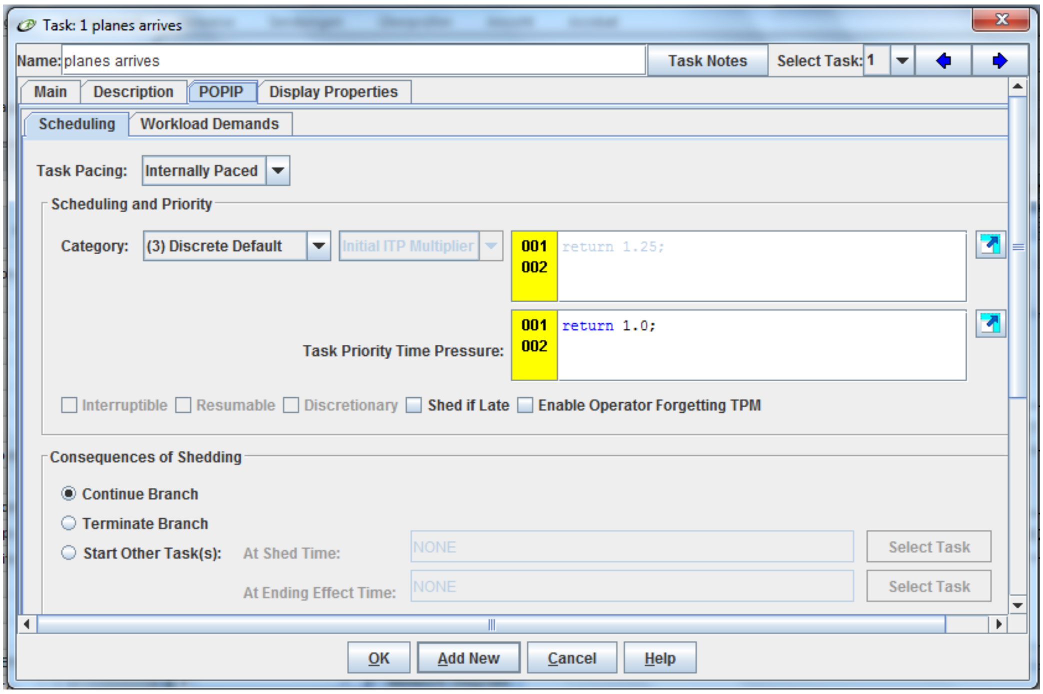


Figure 38: Task - POPIP Tab, Scheduling Subtab

6.3.2 Workload Demands subtab

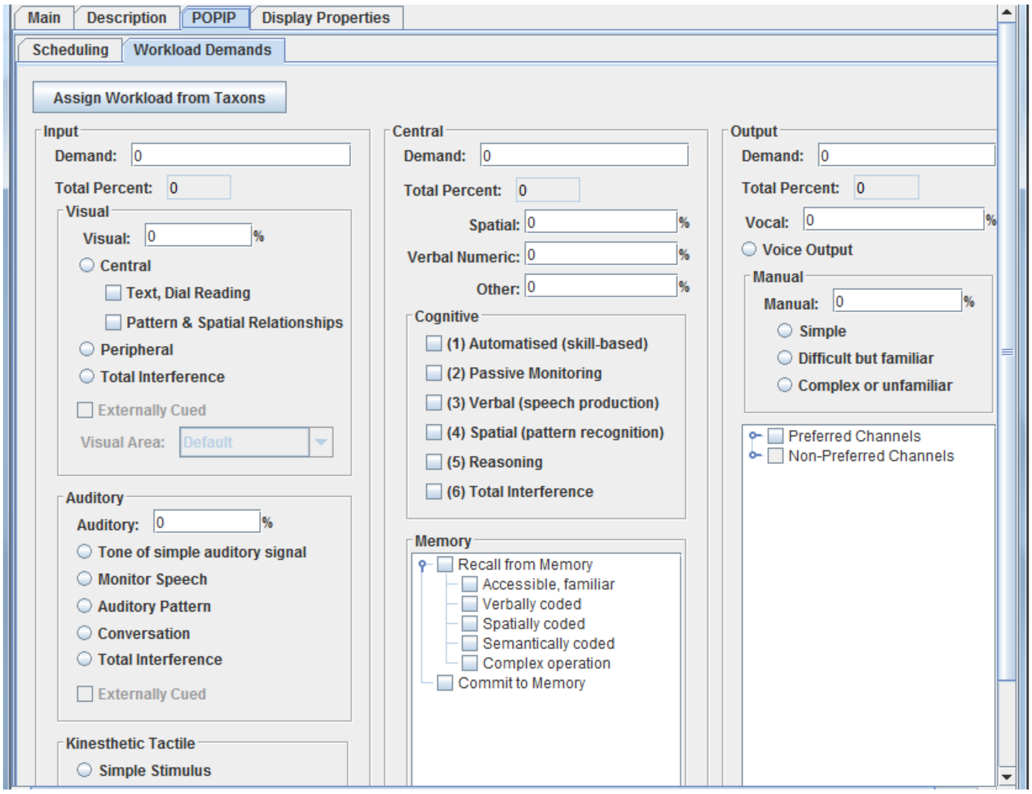


Figure 39: Task - POPIP Tab, Workload Demands Subtab

6.4 Display Properties

Selected Object -Select the object for which you want to change the display properties. Options include: Task, Queue, and Decision.

Background Colour : The selected background colour displays as the fill colour for the object. Click the Change button to select a different background colour. The currently selected colour is displayed on the Change button.

Foreground Colour - The selected foreground colour displays as the outline colour for the object. Click the Change button to select a different foreground colour. The currently selected colour is displayed on the Change button.

Size - Select the font-size to display from the drop-down list.

Font - Select the desired font from the drop-down list.

Style - Select the desired font style from the drop-down list.

Colour - This option controls the colour of the font. Click the Change button to select a different font colour. The currently selected colour is displayed on the Change button.

Preview - The preview box shows what your selections will look like.

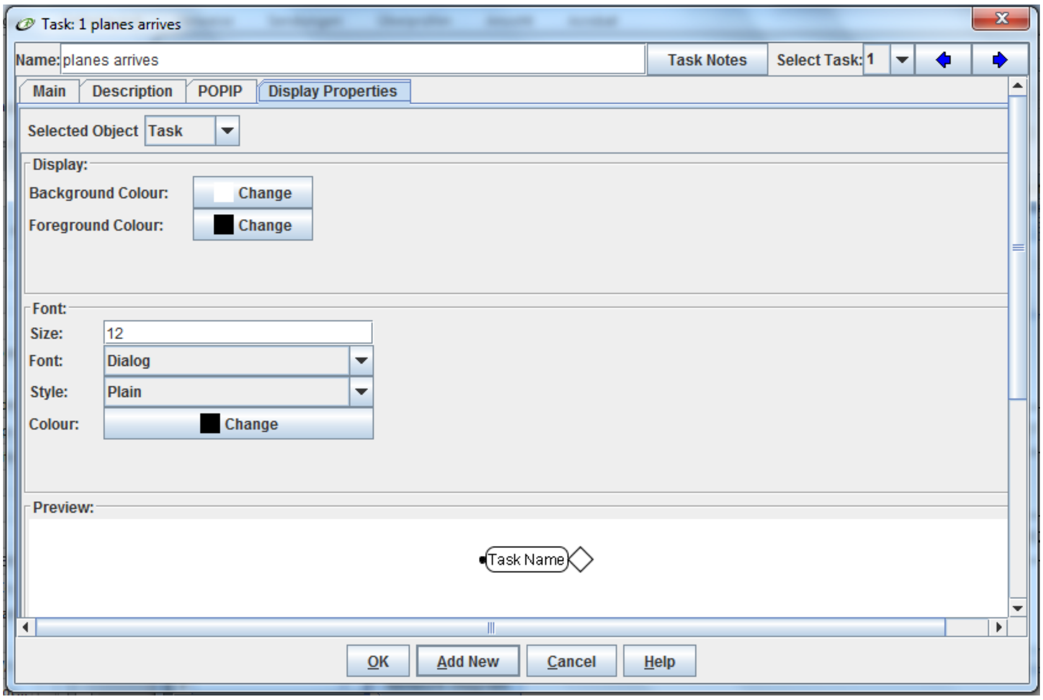


Figure 40: Task – Display Properties Tab

7 Queue

The queue is a waiting area in front of a task where entities accumulate while they are waiting to execute the task. Queues adopt the release condition of the associated task. If you have a queue on a repeating task, there would be a "loop-back" and a new entity would enter the queue again.

For a queue to accumulate entities, each entity must be defined as a separate *tag* number. If a tag enters the queue and the tag number is already present, the two will be merged as a consequence of the release condition.

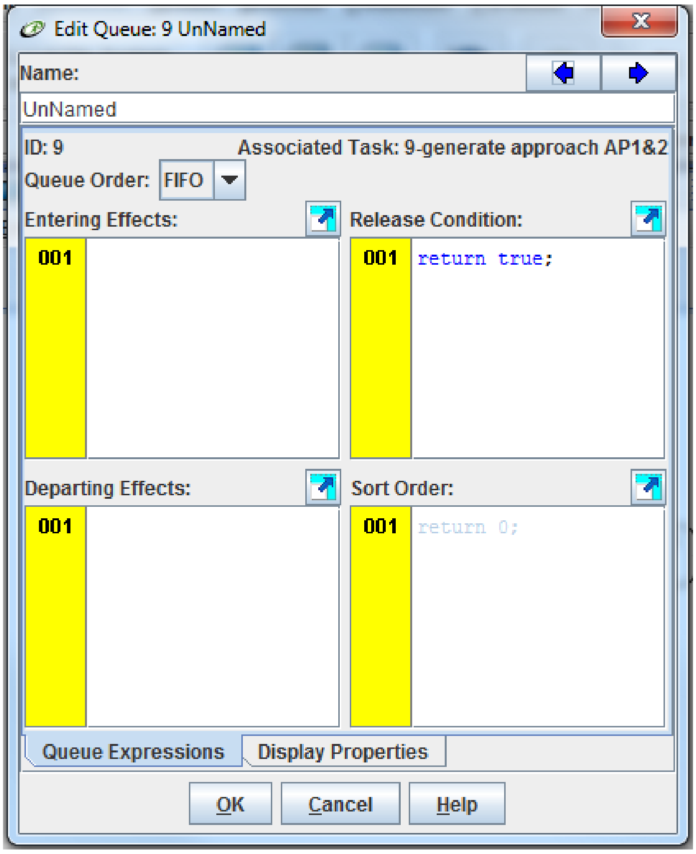


Figure 41: Task – Queue Dialoge

Queue Parameters:

The Queue Parameters are contained in the Edit Queue dialogue.

Name - Contains the name of the queue.

ID - Contains the queue number. The queue is automatically assigned the reference number containing the associated task.

Associated Task - Contains the task_id and name of the task associated with the queue.

Queue Order - Specifies the order for sorting the entities. When you click the **Queue Order** button, a drop-down menu displays with the following options:

FIFO (First in First Out) - The entity that has been in the queue the longest is the next to be selected for execution. This is the default order and the one used most often. FIFO entities are time-based, and are thus independent of tag number.

LIFO (Last In First Out) - The most recent entity to enter the queue is the next to be selected for execution. This option represents a stacking order, so that the last one placed on top is the next one to be taken out. LIFO entities are time-based, and are thus independent of tag number.

Sort - This selection sets the next entity from the queue based on a priority value that you assign. When you select this option, you must define the priority expression identifying which entities have priority over other entities. These types of entities can be dependent on the tag number or any other variable.

Entering Effects - Contains the expression indicating what happens when an entity enters a queue. You can use this parameter to change the value of variables when an entity enters a queue.

Departing Effects - Contains the expression indicating what happens when an entity leaves a queue. You can use this parameter to change the value of variables that reflect the state of the system when an entity leaves the queue.

Release Condition - This field contains an expression that determines whether an entity can be released from the queue to the associated task. An entity can only be released when the release condition value is true. The default value is *true*, so that a queued entity can be released from the queue as soon as it is scheduled. The release conditions for a queue are the same as for the task to which the queue is attached. The release conditions can be edited in the Queue Information or Task Information dialogue box.

Sort Order - Contains the expression used in sorting the entities in a queue; this field is only used when the Queue Order is sorted. When it is time to consider an entity for release from the queue, the value of the priority expression is calculated for each entity in the queue. The release condition for the entity with the highest priority value is evaluated, and if the release condition is satisfied, the entity with the highest priority is then released into the task. For the priority expression to be meaningful, it must have different values for different entities, usually by involving the *tag* variable. When you use a sorted release order, you must enter priority expressions that define the priority of entities released from the queue.

8 Decision

In the task network diagram, a decision node is associated with each task in IPME. In the project tree, a decision item is added to a task whenever that task has more than one possible path emerging from it. Decision nodes exist in the network drawing and decision items exist in the tree.

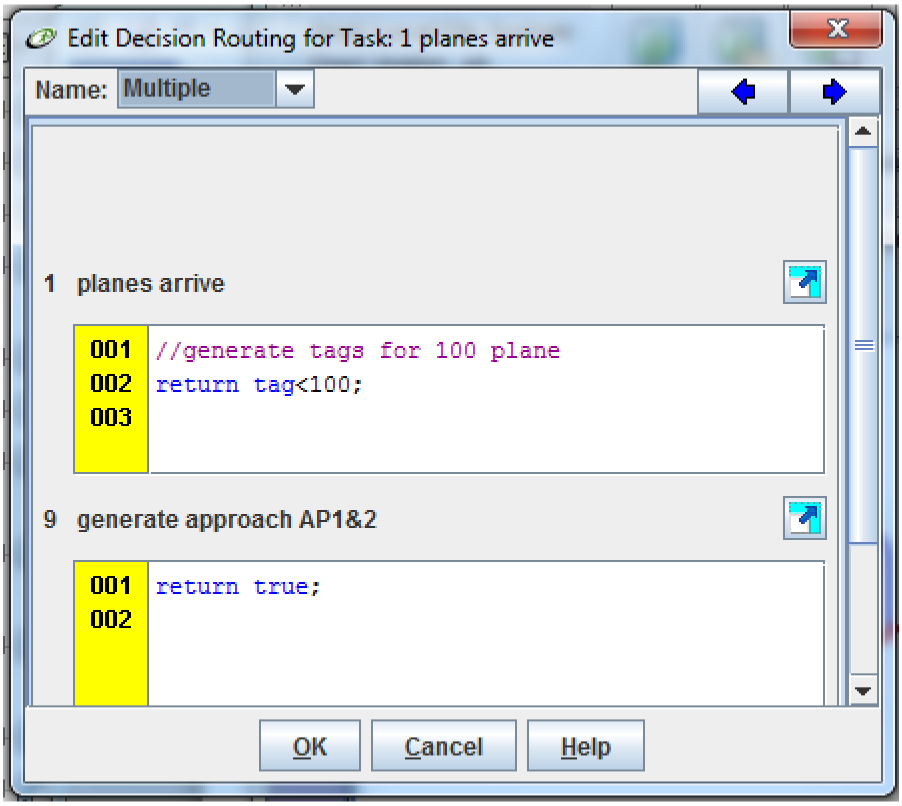


Figure 42: Task – Decision Dialoge

Decision Parameters:

Decision Type - Contains the type of decision. The three types of decisions are tactical, multiple, and probabilistic. Multiple and tactical decision types are rule-based decisions, while the probabilistic decision type is mathematically based. The rules for tactical and multiple decisions are defined through logical or mathematical expressions using the IPME simulation language. A drop-down list contains the following decision types:

Multiple - All of the following paths with *true* routing conditions begin execution simultaneously following execution of the current task. When this happens, the entity exiting the current task splits into multiple entities, one for each following task. These entities all have the same tag value unless you change the tag value in the following tasks. In some models, you may want to change the tag values to avoid having the same tag value executing in the model in more than one place. You can define an expression later in the network model, such as in a release condition, to rejoin the multiple entities so that it can complete the model as one entity. In the case where all evaluated path expressions return false, none of the paths is followed.

Probabilistic - Only one of the following tasks executes next; you supply the probability of each path executing next in its routing condition. The probability that a particular task follows is equal to its probability value in the routing condition field divided by the sum of the probability values of all possible following tasks. For example, if the probability value is 25 percent for Task A, you could enter return .25, return 1, or return 25 in the routing condition for Task A and enter return .75, return 3, or return 75, respectively, for the routing condition for Task B. In the case where all evaluated path expressions are false (or equal to zero), a run-time error will occur.

Tactical - The tactical decision type is used for rule-based decisions where one, and only one, path will be followed. The path with the tactical expression that evaluates to the highest value executes next; you supply the tactical expression as the routing condition. In the case where all evaluated path expressions are equal, the path listed first in the Decision Routing dialogue box will be followed.

9 Operator

You can use a wide variety of operators to perform operations on numeric values. These operators are summarised in the following table.

Operator precedence controls the order in which operations are performed. Operations with the same precedence are performed in order according to their associativity (usually left to right). Operator precedence can be overridden with the explicit use of parentheses.

Associativity of an operator specifies the order that operations with the same precedence are performed. In the following table, a value of L specifies left-to-right associativity, and a value of R specifies right-to-left associativity.

Precedence	Operator	Operand Types	Assoc.	Operation Performed
1	( ) [ ]	arithmetic	L	Grouping or index
2	Math.Pow	arithmetic	R	Exponentiation
	++	arithmetic	R	Increment
	--	arithmetic	R	Decrement
	+, -	arithmetic	R	Addition, Unary Plus, String Concatenation Subtraction and Unary Minus
	!	boolean	R	Logical Negation
	(type)	any	R	Grouping
3	*, /, %	arithmetic	L	Multiplication, Division, Modulus or Remainder
4	+, -	arithmetic	L	Addition, Unary Plus, String Concatenation Subtraction and Unary Minus

	+	string	L	Addition, Unary Plus, String Concatenation
5	<, <=	arithmetic	L	Less Than, Less Than or Equal to
	>, >=	arithmetic	L	Greater Than, Greater Than or Equal to
6	==	primitive	L	Equal to (have identical values)
	!=	primitive	L	Inequality (have different values)
7	&&	boolean	L	Logical AND
	||	boolean	L	Logical OR
8	=	variable, any	R	Assignment
	*=, /=	variable, any	R	Multiplication Assignment, Division Assignment
	%=	arithmetic	R	Modulus Assignment
	+=, -=	arithmetic string (for +=)	R	Addition Assignment, Subtraction Assignment

Table 18: Operator

10 Example

Every 70 seconds with standard deviation 20 seconds one plane arrives on 2 airport with a single runway. The air traffic controller will be asked by pilots for landing clearance. The traffic controller is checking either automatically or manually.

The simulation is defined for 100 aircraft. A/C and the tags will leave the model and the model will halt automatically when all tags have been used.

10.1 Definition of the variables

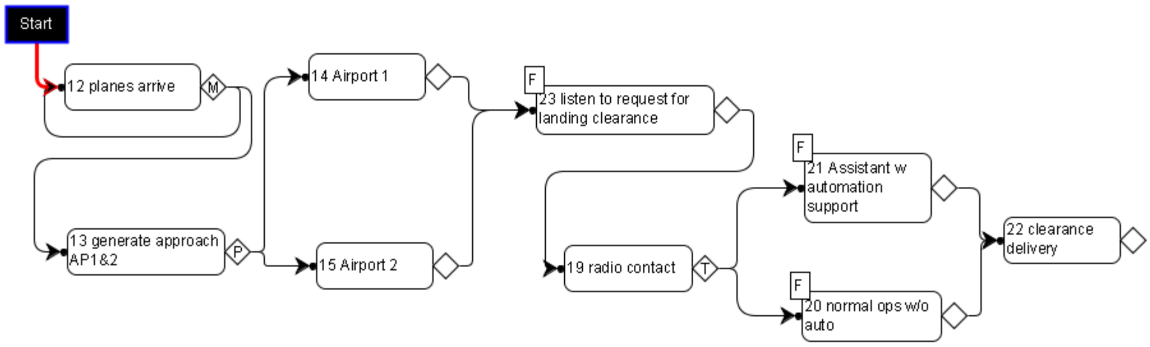
variable name	Type	Initial value	description
ControllerSpeaking	boolean	False	
NormalOpsBusy	boolean	False	
Auto busy	boolean	False	
normalOps	boolean	False	
requestBusy	boolean	False	
Avwait	double	0.0	
Maxwait	double	0.0	
Waitotal	double	0.0	
planeCleared[1000]	Array of integer	0	Highest index:1000 One dimension
runningTime[1000]	Array of double	0	Highest index:1000 One dimension
waited [1000]	Array of double	0	Highest index:1000 One dimension
q1counter	integer	0	
q2counter	integer	0	
t1counter	integer	0	
t2 counter	integer	0	

Table 19: Definition of variables for aircraft landing simulation

10.2 Definition of the Operators

I have two operators defined (Assistant _Auto and Controller). The Assistant _Auto is checking automatically and the Controller is checking manually. In the main task screen, you need to select an operator in the operator assignment box. I would use the static drop down box to choose one.

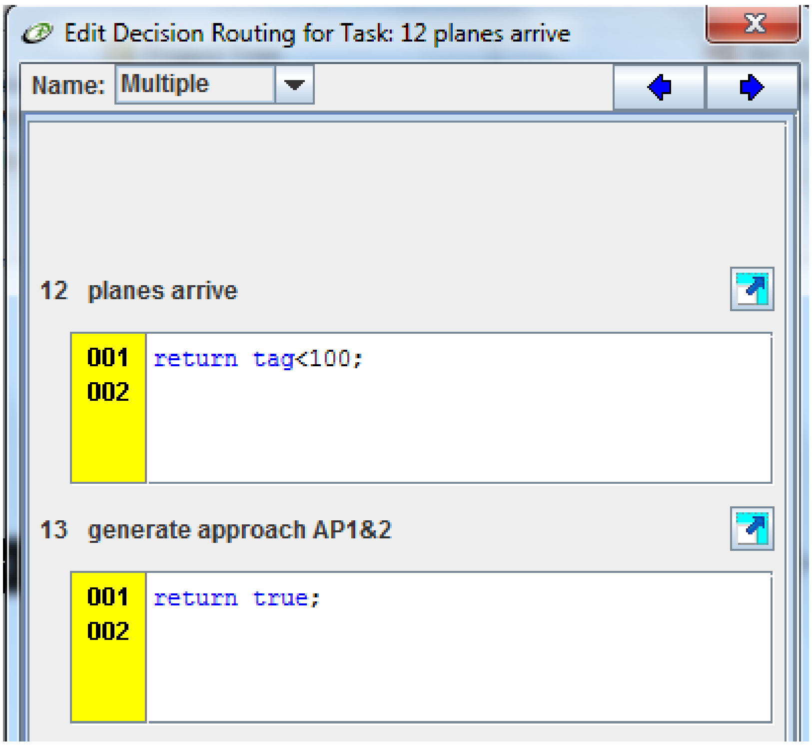
10.3 Network Diagram



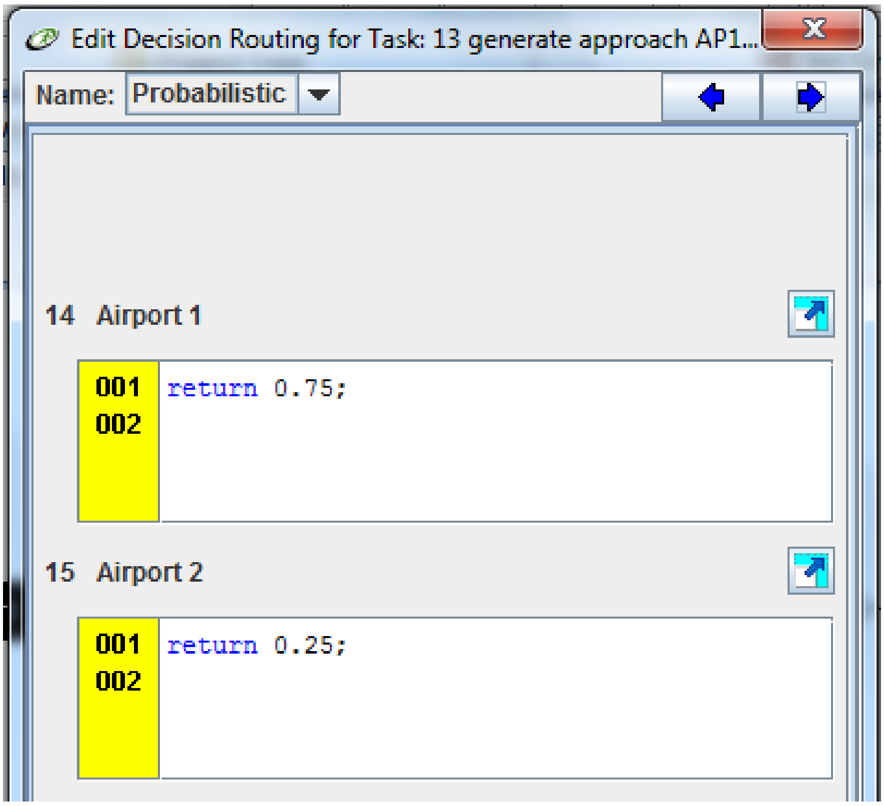
10.4 Decision Type

A) the decision should be multiple because the spinner task would then be based on number of airplane. You would continue to generate new planes (tags) until 100 airplane (tag<100) had passed.

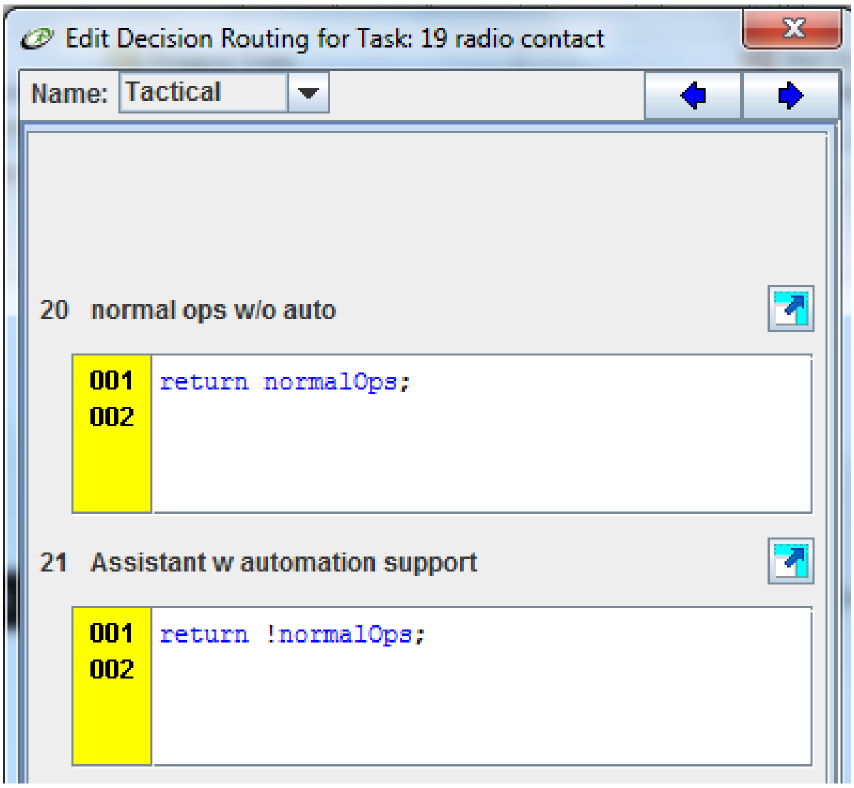
The ending effect of task planes arrive should increment the tag number to create unique tags, also tag+=1



B) the decision should be probabilistic because Only one of the following tasks executes next; you supply the probability of each path executing next in its routing condition. The probability that a particular task follows is equal to its probability value in the routing condition field divided by the sum of the probability values of all possible following tasks .75% of the airplanes land at airport 1 and 25% at airport 2

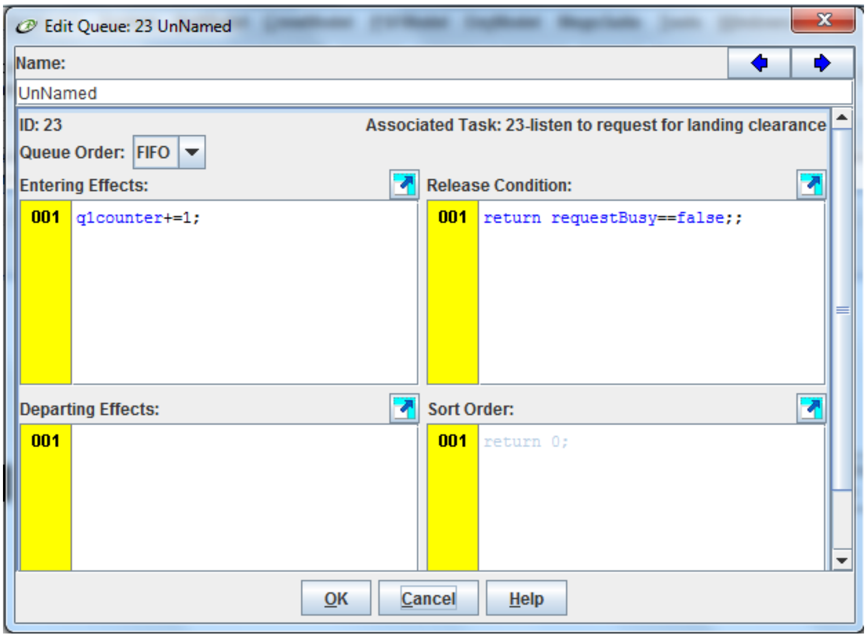


c) the decision should be tactical because the operator is only doing one OR the other, not both. I added a boolean variable called normalOps to the variable list. I changed the decision logic so that if normalOps is false, the tags all go to the normal task. If it is true, they go to the automatl task. This is one example of how this could be done to represent the case where the operator handles an entire group of planes in the same manner. Run the model, collect results, change the value of the variable and rerun. Then compare the two data sets. You can change the initial value in the variable list or set it in a scenario event at time zero.

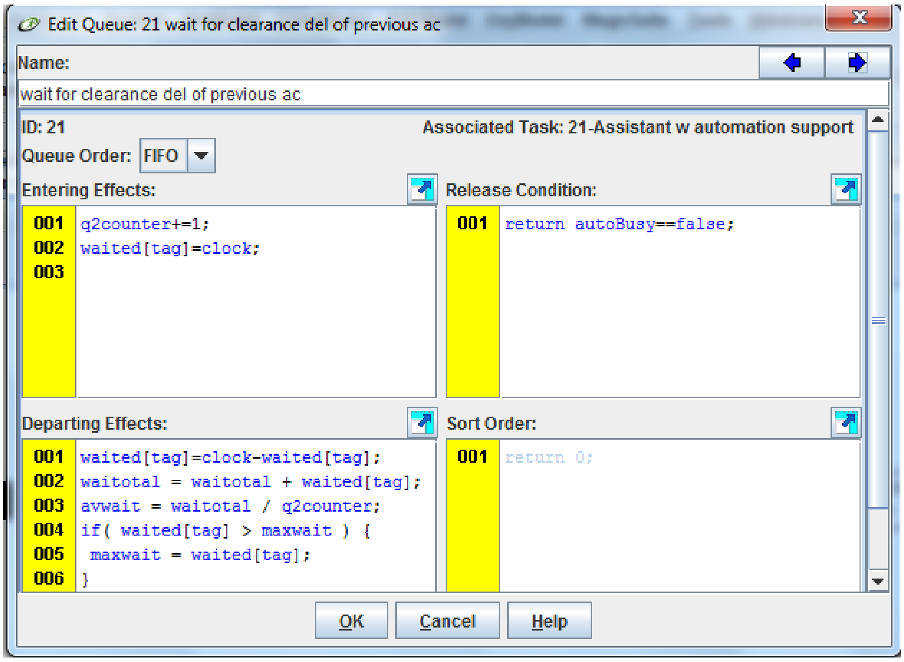


10.5 Queue

For Task 23: “listen to request for landing clearance”:



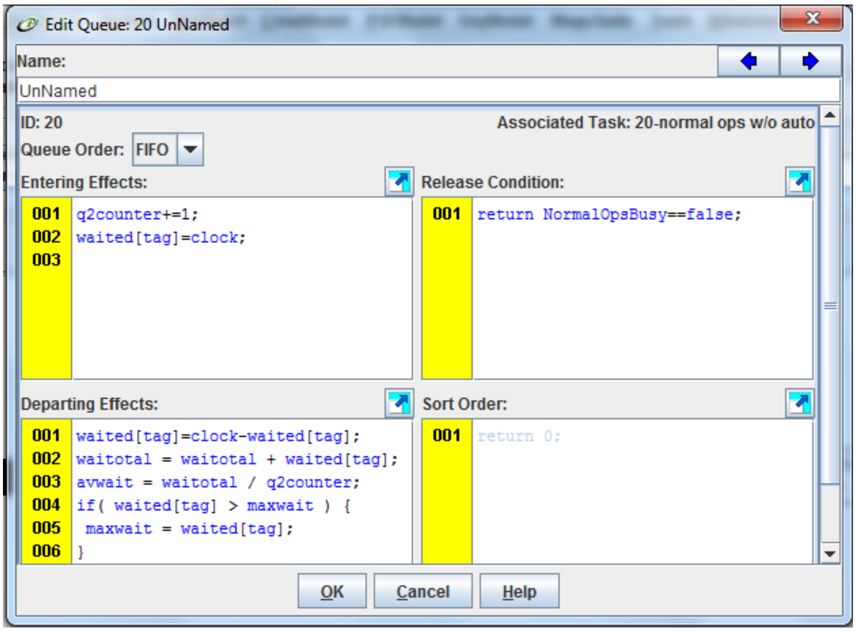
For Task21: “Controller Assistant with automation support”:



you can define simulation time with using variables to record how long running time is. For example, if the time to complete tasks (task 23/Listen to request for landing clearance) until (task 22/Clearance delivery), you would record the start time in the Beginning Effect of task 23 (or whatever your starting task is in the sequence) (runningTime[tag]=clock;) and subtract that value from the current clock time in the ending effect of(task 22/Clearance delivery) (or whatever the ending task is in the sequence).

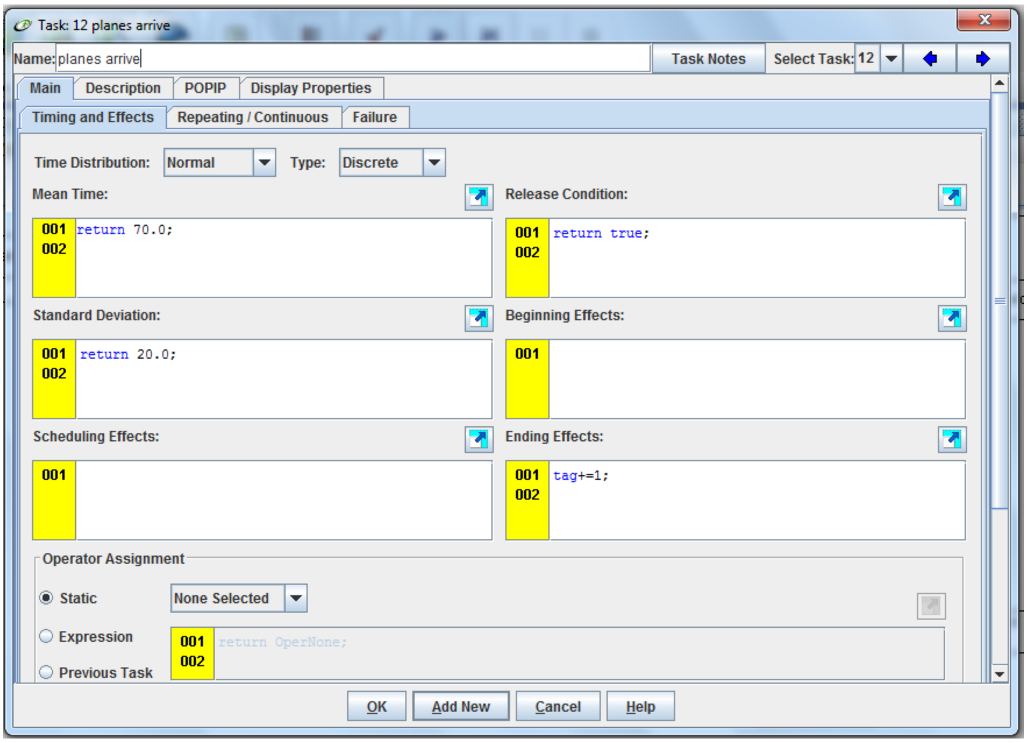
- You would need to add a counter in at least one of the tasks and check the counter variable value in the decision to determine where to send the next plane.
- I added a debug statement so you can see how the time varies with the standard deviation.

For Task20: “ normal ops w/o auto”:

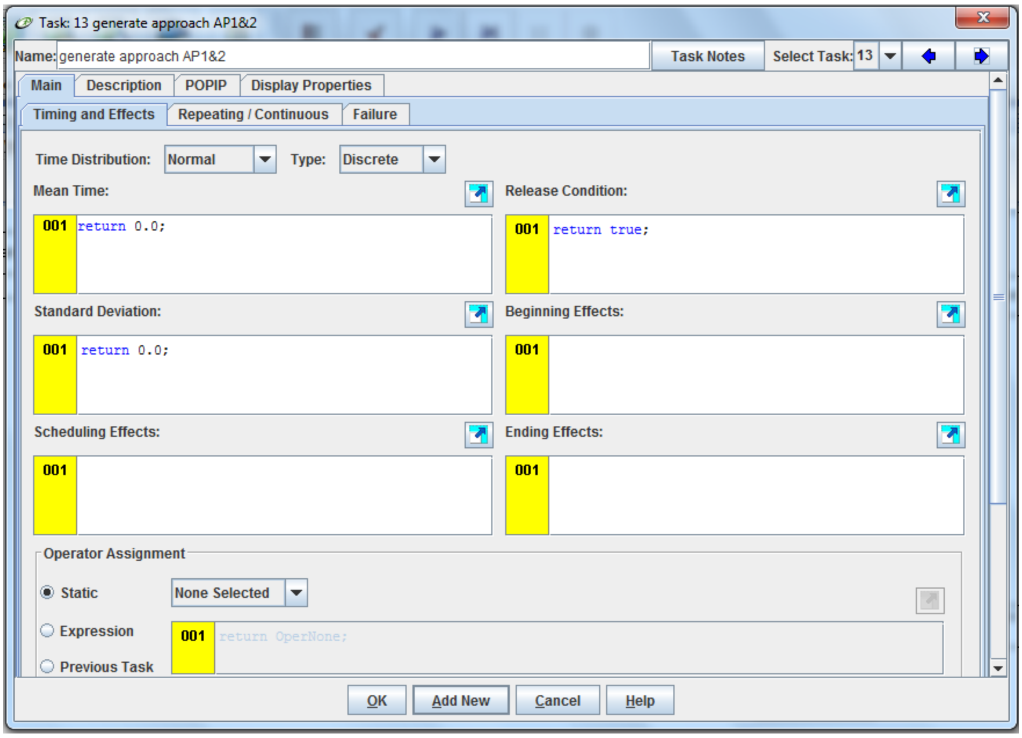


10.6 Tasks

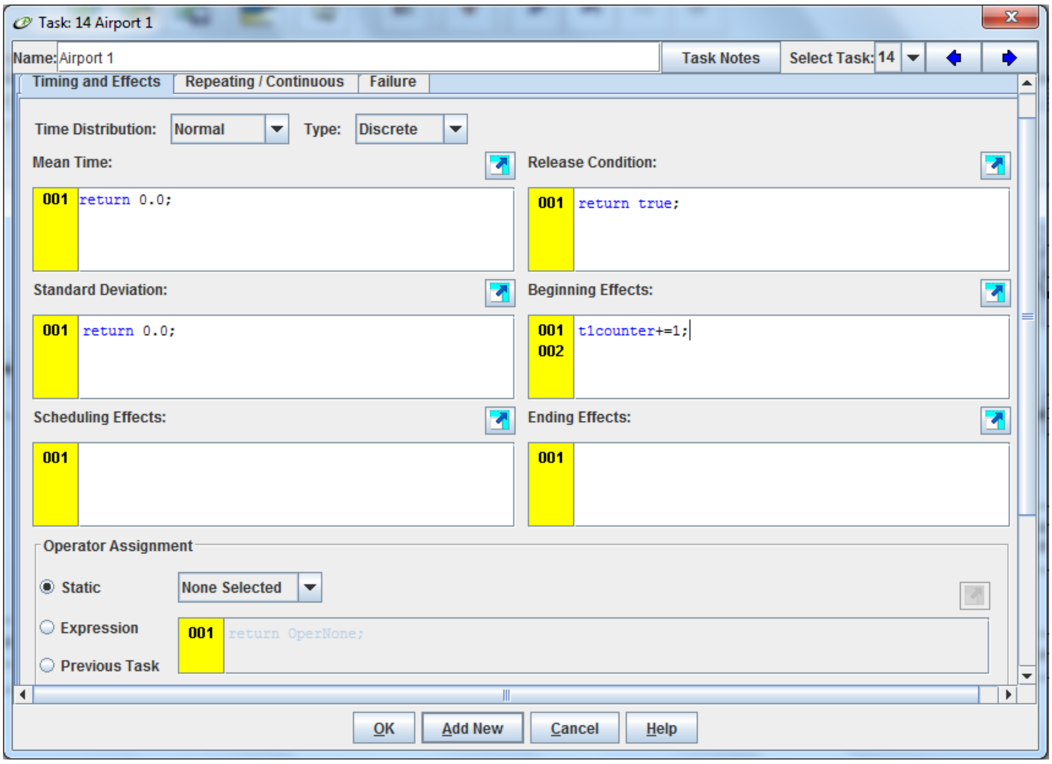
Planes arrive:



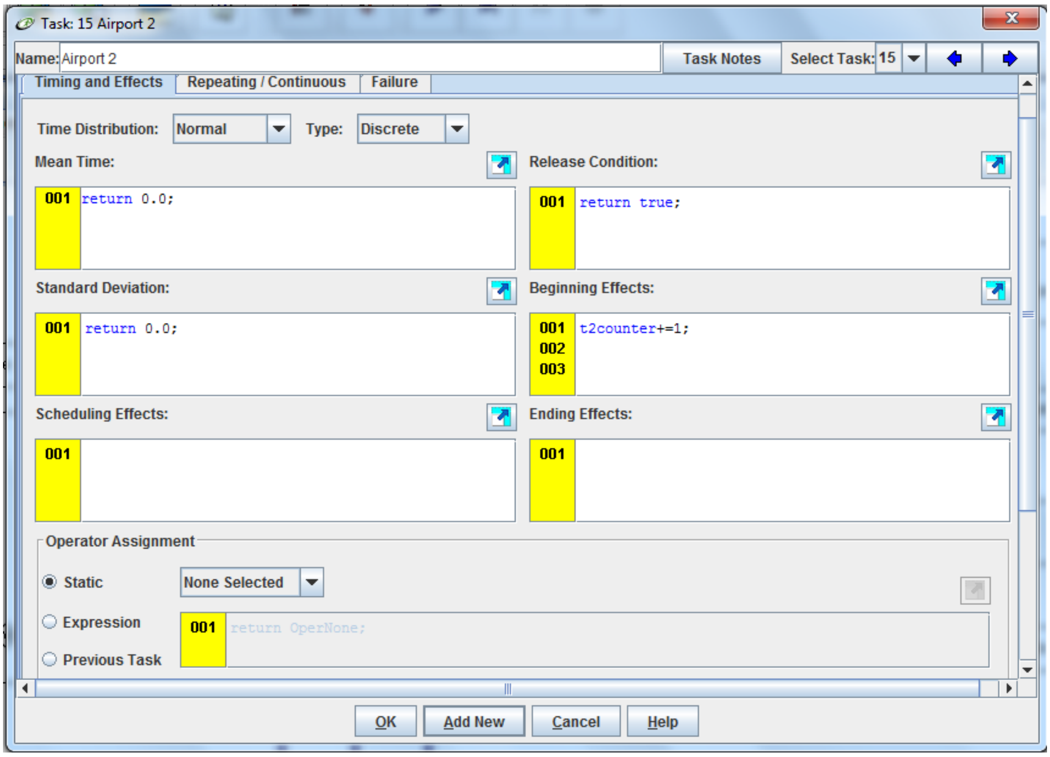
Generate approach AP 1&2:



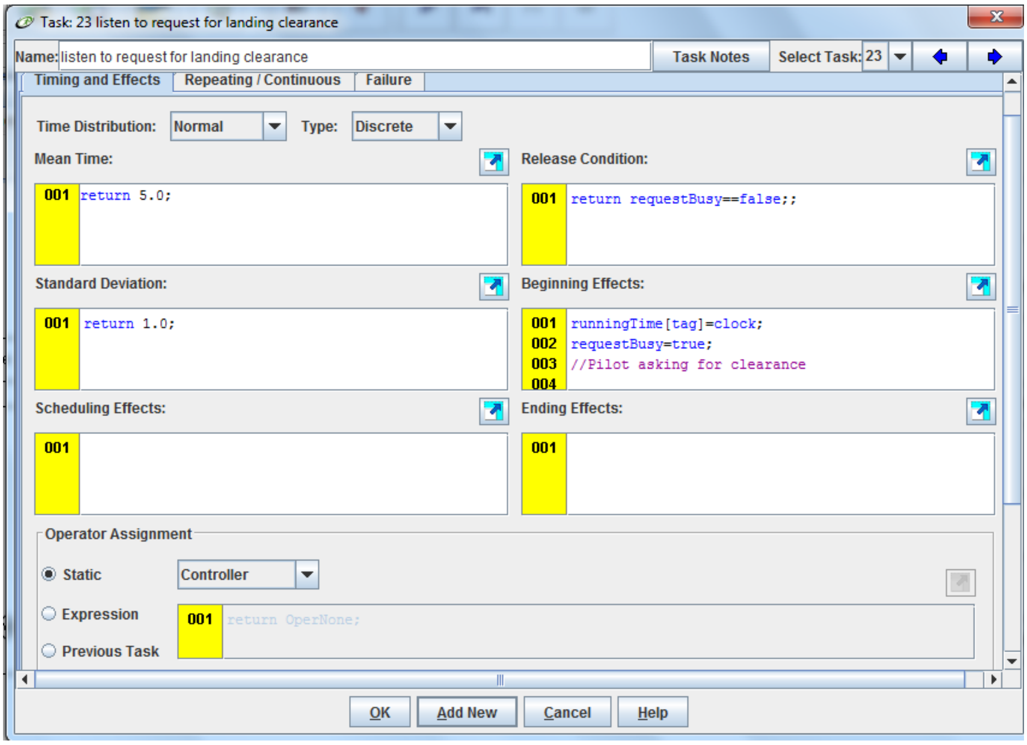
Airport1:



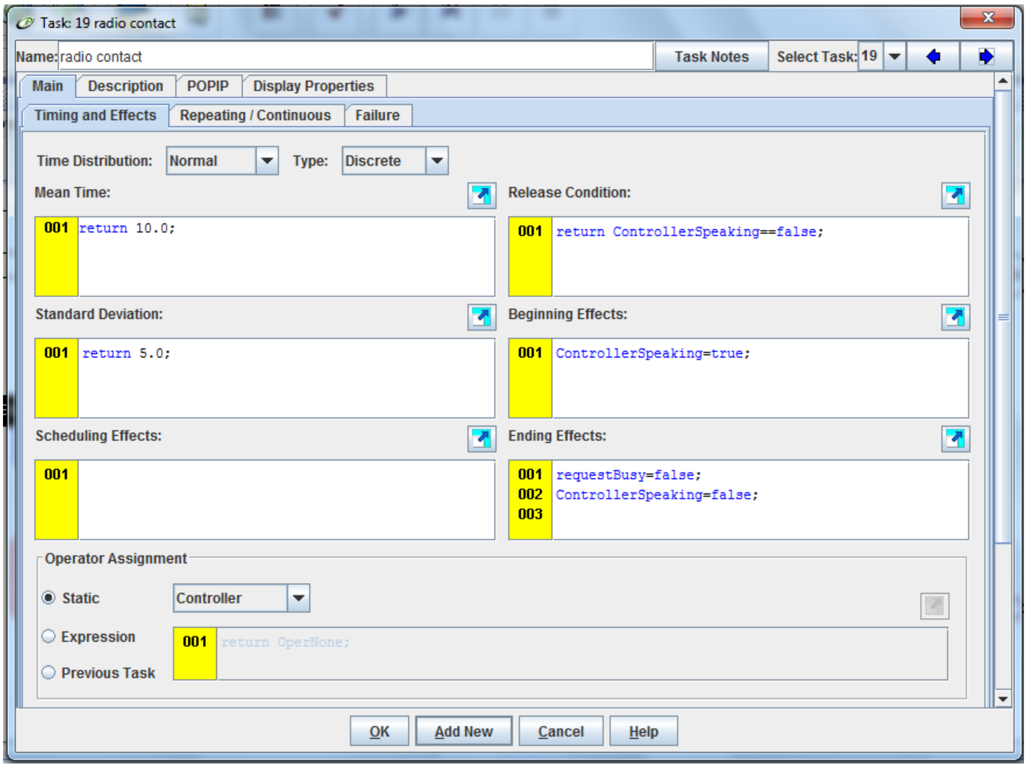
Airport 2:



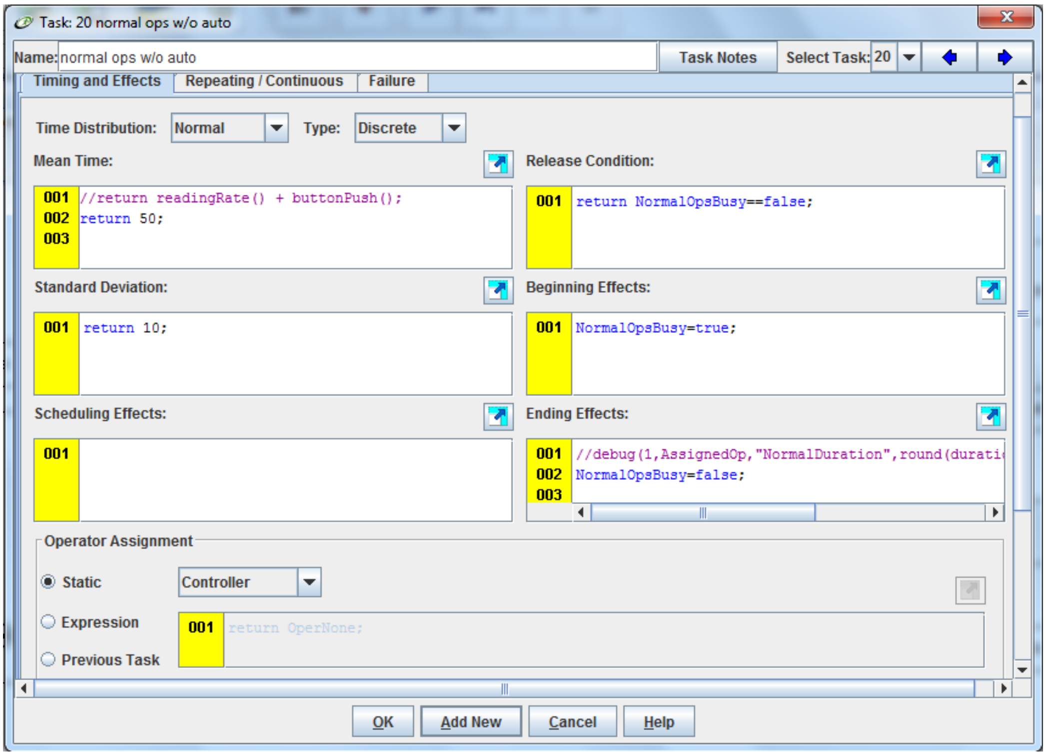
Listen to request for landing clearance:



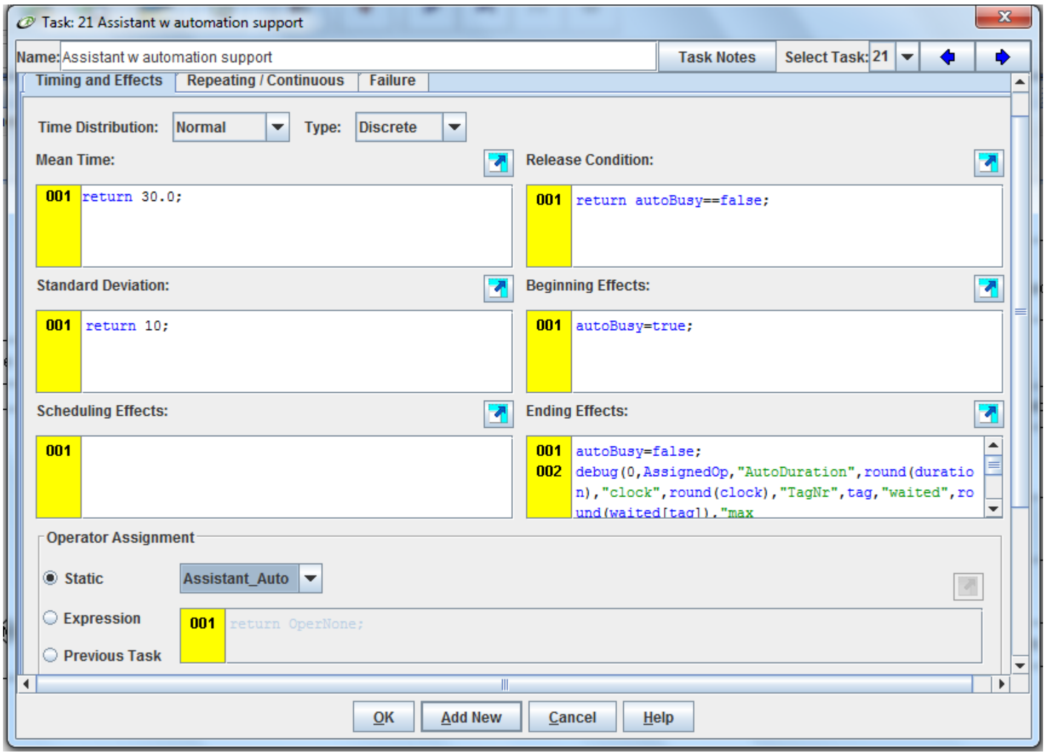
Radio contact:



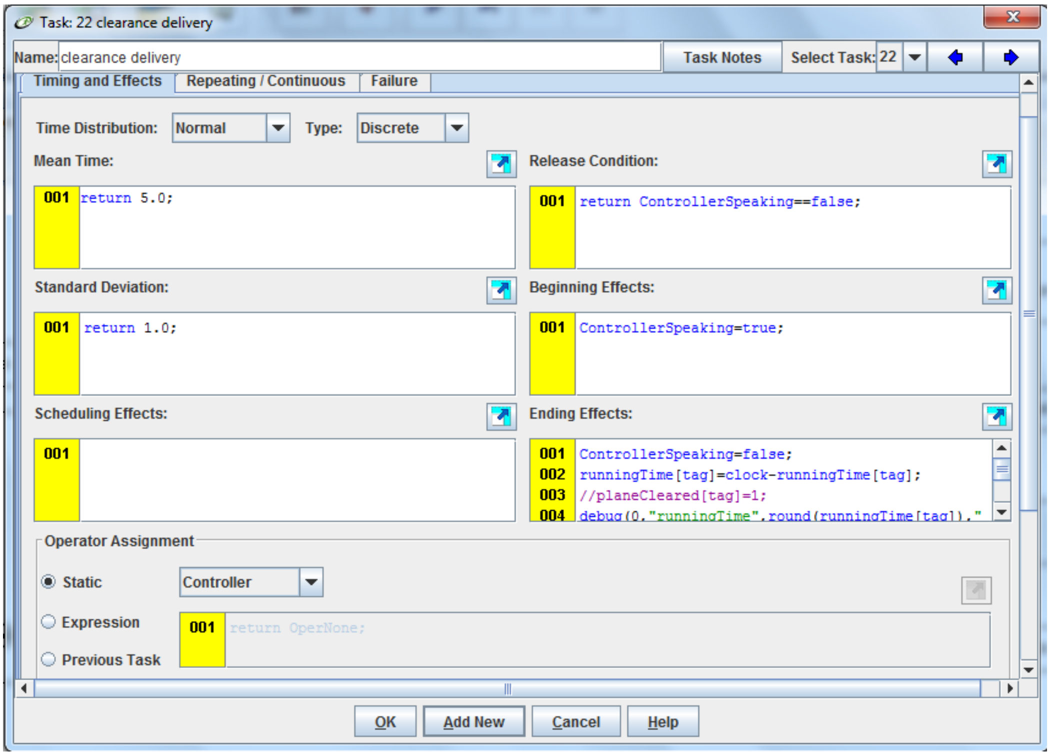
Normal operations w/o automation support:



Assistant w automatic support:



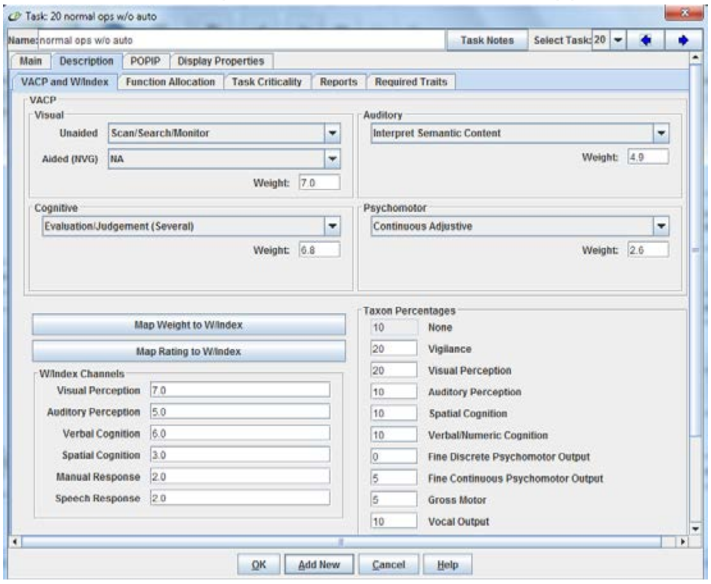
Clearance delivery:

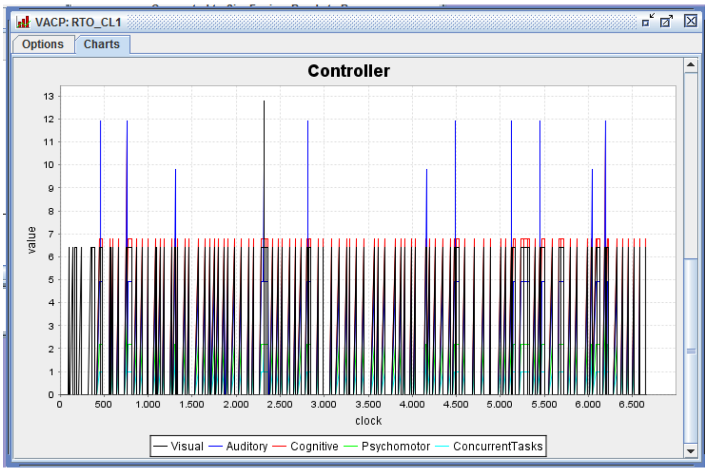


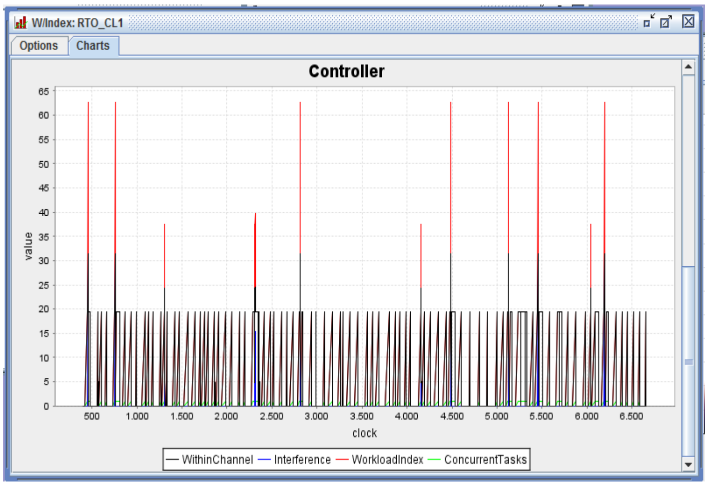
10.7 VACP and Workload chart

for Controller:

Task 20: Normal operations w/o automation support

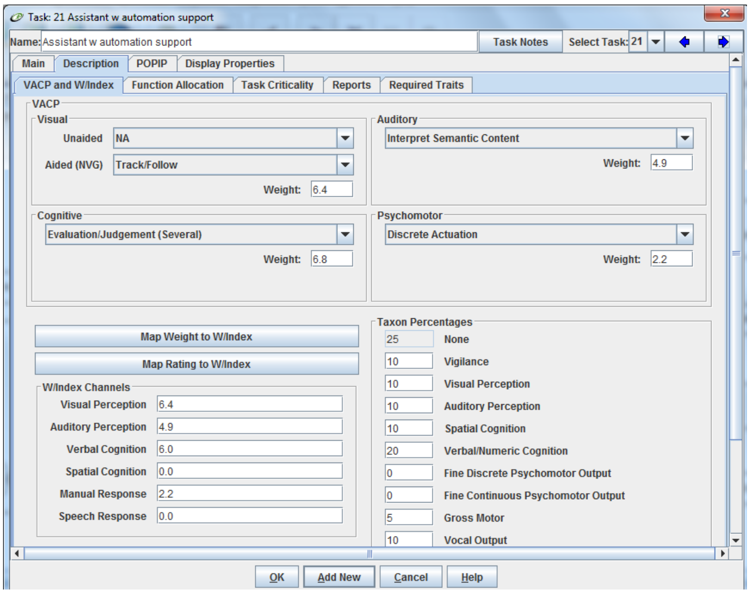


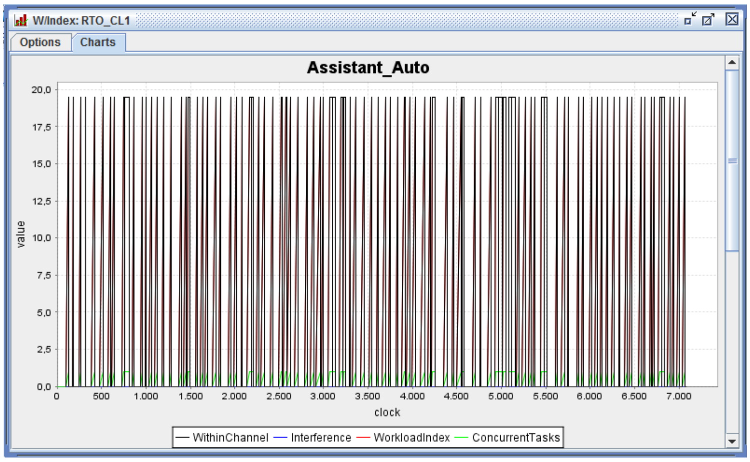


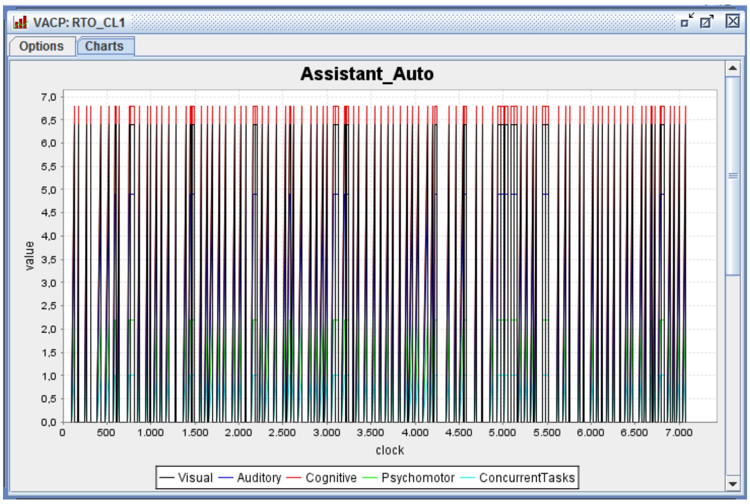

WIndex: RTO_CL1
Options
Charts
Controller
value
clock
WithinChannel
Interference
WorkloadIndex
ConcurrentTasks

For Assistant:

Task 21: Controller Assistant with automatic support







- You only need to enter VACP values if you intend to collect VACP workload values. If you are using VACP for your definition of workload, then this is necessary to enter in all operator tasks. For VACP to work, the tasks with VACP values must be assigned to an operator. The aided selection in the VACP is for an operator using night vision goggles. Operators are probably not wearing night vision goggles, so I would make a selection from the unaided list. You can only use either unaided or night vision goggles, not both. Also, in VACP is it common to assign every task a cognitive component, so I would add that. It is also possible that the normal task has some psychomotor load if the operator has to adjust the binoculars.
- Taxons are only used if performance shaping functions are going to be used to impact task times or task failure.

11 Appendix

11.1 Abbrevations

IPME	integrated performance modeling environment
IP-TP	information processing – time pressure
NVG	night vision goggles
PCT	perceptual control theory
POPIP	Prediction of Operator Performance
PSF	performance shaping function
TP	time pressure
TPM	Task performance modifier
TDM	task demand modifiers
VACP	visual, auditive, cognitive, psychomotor
W/Index	workload index

11.2 References

[1] Fowles-Winkler, A. M. (2003). Modelling with the Integrated Performance Modelling Environment (IPME). Proc. 15th European Simulation Symposium. SCS European Council, ISBN 3-936150-29-X.

[2] Hendy, K.C., Jianquiao, L., Milgram, P. (1997): Combining Time and Intensity Effects in Assessing Operator Information-Processing Load. Human Factors vol 39 (1). pp 30-47

[3] Powers, W. (1973). Behavior: The Control of Perception. ISBN 0-7045-0092-2

[4] Aldrich, T.B., Szabo, S.M., Craddock, W.P. (1986) A computer analysis of LHX automation options and their effect on predicted crew workload. Tech. Report ASI479-063-85 Fort Rucker, AL: Anacapa Sciences, Inc..

[5] Bierbaum, C.R., Szabo, S.M., Aldrich, T.B. (1989): Task analysis of the UH-60 Mission and Decision Rules for Developing a UH-60 Workload Prediction Model. Tech. Res. Prod.89-08, US Army Res. Inst. Beh. And Social Science, Army Proj. no.2Q263007A793, Anacapa Sci. Inc.

[6] Aldrich, T.B., Szabo, S.M., Bierbaum, C.R.: The development and application of models to predict operator workload during system design. In: Applications of human performance models to system design. Defense Res. Series, vol.2. Plenum Press, New York, London 1989

[7] Wickens, C.D., Hollands, J.G. (2000). Engineering Psychology and Human Performance. Prentice-Hall / N.J.

[8] Hendy, K.C., Farrell, P.S.E. (1997). Implementing a Model of Human Information Processing in a Task Network Simulation Environment. Defence & Civil Inst. Of Environmental Medicine (DCIEM) Report no. 98-R-71

11.3 List of Figures

Figure 1: IPME-main window..12

Figure 2: File menu....13
Figure 3: Edit menu....15
Figure 4: System menu....16
Figure 5: Execution Setting – Runtime Tab....19
Figure 6: Execution Setting – Mission Tab....20
Figure 7: Execution Setting – Output Tab....22
Figure 8: Execution Setting – POPIP Tab,Main subtab....24
Figure 9: Execution Setting – POPIP Tab, Compatible Tasks Subtab....25
Figure 10: Execution Setting – POPIP Tab, Visual Areas Subtab....26
Figure 11: Execution Setting – POPIP Tab, Global Operator Settings Subtab....27
Figure 12: Execution Setting – Chart Tab....28
Figure 13: NetModel menu....29
Figure 14: CrewModel menu....31
Figure 15: PSFModel menu....33
Figure 16: EnvModel menu....34
Figure 17: Tools menu....36
Figure 18: Windows Menu....37
Figure 19: Help menu....38
Figure 20: Project Tree....39
Figure 21: Task Network Model....40
Figure 22: Snapshot....41
Figure 23: Scenario Event....43
Figure 24: Function....44
Figure 25: Variable Dialoge....46
Figure 26: Operator, Anthropometry....47
Figure 27: Environment Variable....48
Figure 28: Performance Shaping Function Models....49
Figure 29: Measurement Suite....51
Figure 30: Task - main Tab, Timing and Effects Subtab....52
Figure 31: Task - main Tab, Repeating/Continuous Subtab....53
Figure 32: Task - main Tab, failure Subtab....54
Figure 33: Task – description Tab, VACP and W/Index....55
Figure 34: Task - description Tab, Function Allocation Subtab....59
Figure 35: Task - description Tab, Task Criticality Subtab....59
Figure 36: Task – Description Tab, Reports Subtab....60
Figure 37: Task - description Tab, Required Traits Subtab....60
Figure 38: Task - POPIP Tab, Scheduling Subtab....63
Figure 39: Task - POPIP Tab, Workload Demands Subtab....63
Figure 40: Task – Display Properties Tab....64
Figure 41: Task – Queue Dialoge....65
Figure 42: Task – Decision Dialoge....67

11.4 List of Tables

Table 1: Components of main window....13
Table 2: File menu....15
Table 3: Edit menu....16
Table 4: Edit menu....18
Table 5: NetModel menu....31
Table 6: CrewModel menu....32
Table 7: PSF Model menu....33
Table 8: EnvModel menu....35

Table 9: Tools menu 35
Table 10: Tools menu 37
Table 11: Windows Menu 37
Table 12: Help menu 38
Table 13**:** PSF-equations 50
Table 14**: PSF** Result: (mean time * taxon %) / (PSF Expression) 50
Table 15: VACP channels 57
Table 16: Conflict Matrix 58
Table 17: Workload Index results 58
Table 18: Operator 69
Table 19: Definition of variables for aircraft landing simulation 70