



OperationalSupport

Michael Westergaard

Tuesday, July 12, 2011

OperationalSupport implements version 2.0 of the operational support infrastructure in ProM. The package supersedes but can coexist with OSSupport, implementing the old version. The protocol used in this version is documented in

M. Westergaard and F.M. Maggi

Modelling and Verification of a Protocol for Operational Support using Coloured Petri Nets

Proceedings of 32nd International Conference on Application and Theory of Petri Nets. Newcastle, England, June 2011.

which you can find at www.springerlink.com/index/U7124032T411KP86.pdf.

OperationalSupport itself is only interesting for advanced developers. It comprises the operational support service (OSS) as well as a simple operational support provider for testing purposes. This document also explains how to test providers using Declare.

External references

Video demo of new operational support service: <http://westergaard.eu/2010/10/intro-to-the-new-operational-support-service/>

Paper describing protocol of new service: www.springerlink.com/index/U7124032T411KP86.pdf

Presentation of above paper: <http://westergaard.eu/2011/06/modeling-and-verification-of-a-protocol-for-operational-support/>



Starting Operational Support Service

Starting the operational support service is easy; simply:

1. Go to the Actions perspective.
2. Select the Start Operational Support Service 2.0 plug-in (not that there may be another plug-in mentioning Operational support, but that it the old version, so avoid using that).
3. Press Start.
4. You now obtain the Operational Support 2.0 configuration screen. See below for an explanation of this screen. Once you have set the options as you desire, ensure that the Running checkbox is checked.

Note! You can only start/stop the operational support service once inside ProM. If you need to restart it, restart ProM or go to the Workspace and delete the OSS and create a new instance.

Due to how ProM serializes objects, the settings of the operational support service as well as all running sessions are lost when ProM is closed. This can only be fixed when serialization in ProM is fixed.



Configuration of Operational Support Service

The operational support service has a few configuration options. All except for the port can be changed with the service running.

The configuration options are:

Option	Meaning	Who Should Use It
Port	Which port is the server running at	If you really need the service to run on another port due to conflict or firewall issues
Running	Whether the service is running or not	Everybody to start service
Trace	Whether to provide debugging information	Developers and advanced users
Trace messages	Trace communication between service and clients	Advanced developers
Trace individual providers	Trace communication between service and individual providers	Provider developers

The running option can be used to start and subsequently stop the service. It then becomes inactive. If you want to restart the service, either restart ProM or delete the operational support service from the workspace.

We explain the Providers part of the view when discussing starting providers.

Log

You can get a view of the log by selecting Log at the left side of the screen. The log is always up-to-date, but not saved to disk. You can select text in the log and copy/paste it to another program for further processing (copy using $\text{⌘}-C$ on Mac or $\text{Ctrl}-C$ on Windows).

Starting and Stopping Providers

Here we demonstrate starting providers using the demo provider included in the





OperationalSupport package. The same procedure is used for other providers as well, though they may require more configuration. See the documentation for individual providers for details.

To start the demo provider do as follows:



1. Make sure you have started the operational support service as explained earlier.
2. Find the operational support service in the ProM workspace and click on Use resource.
3. Select the provider you want to start from the list of plug-ins (here, select Start Drunken Monkey Provider).
4. Click Start.
5. The provider is started and attached to the selected operational support service and the ProM Workspace.

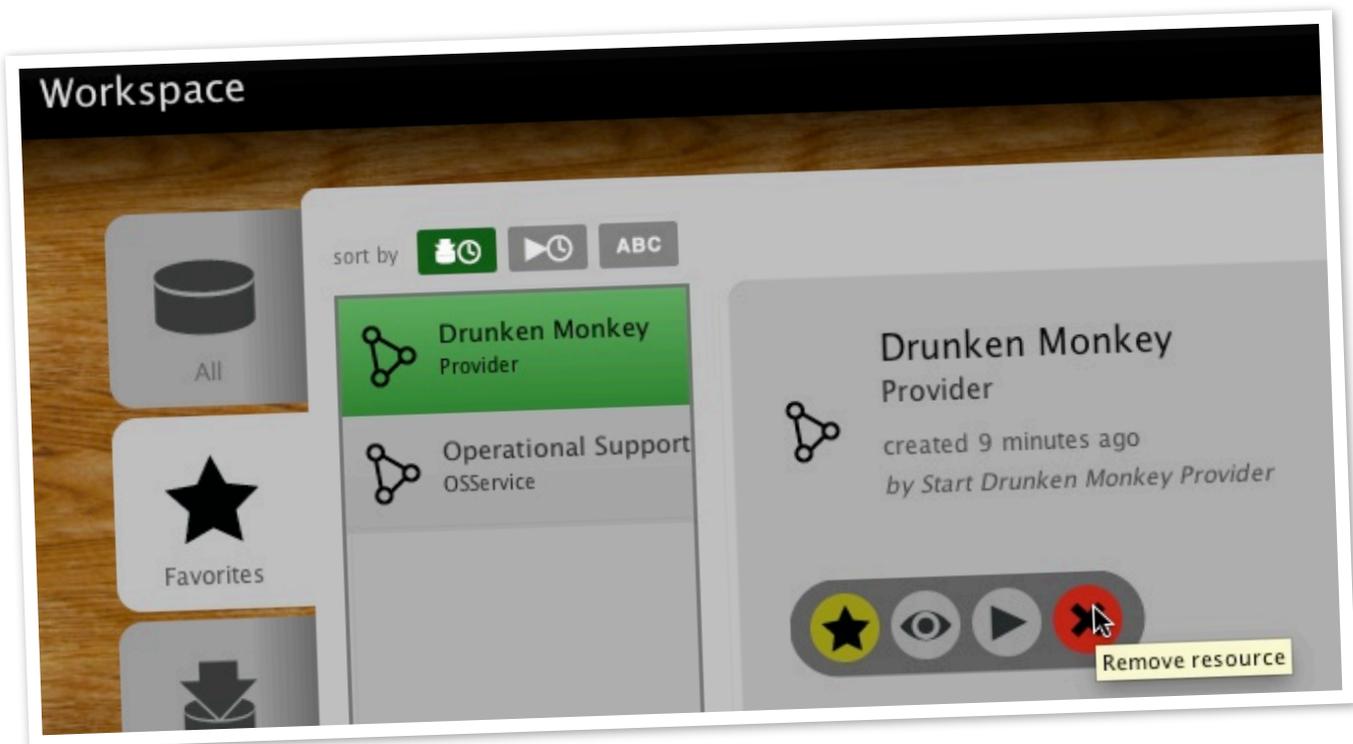


To ensure that your provider is running, you can perform these steps:



1. Select the operational support service in the workspace.
2. Click View resource (the eye thingamajig).
3. The provider is now visible in the Providers list of the Overview screen.
4. You can also see the provider registering in the Log.

The number after the @ sign is an instance-specific number making it possible to distinguish multiple instances of the same provider.



Removing Providers

Providers are removed by deleting them from the ProM workspace. Do this:

1. Select the provider you hate more than Christina Aguilera herself in the Workspace.
2. Click on Remove resource (the cross thingy).
3. The provider is gone, never to come back (until we decide to milk the cow with another sequel...).

The provider is automatically de-registered from the operational support service, all running sessions, and sessions are automatically garbage collected if they need be.

Session Overview

You can get an overview of all active sessions and their associated providers as well; do this:

1. Select the operational support service from the Workspace.
2. Click on View resource.
3. Select the Sessions tab.

Only currently active sessions are shown (i.e., not any terminated nor any garbage collected sessions). You can for each session see which providers are currently handling it.





Testing Providers Using Declare

You can test providers by interacting directly with the operational support service. This can be seen in the static method `main` in `org.processmining.operationalsupport.client.SessionHandle`, which provides a simple test. Make sure to set the query language and modeling language to something handled by the provider you wish to test.

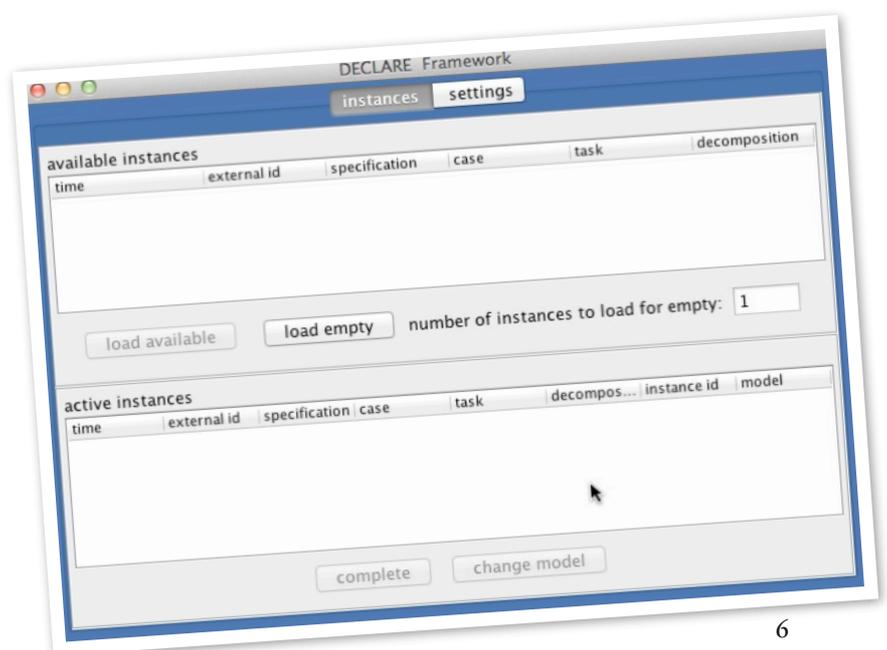
This is excellent for automatic testing, but sometimes more explorative testing is needed. This is easily done using Declare.

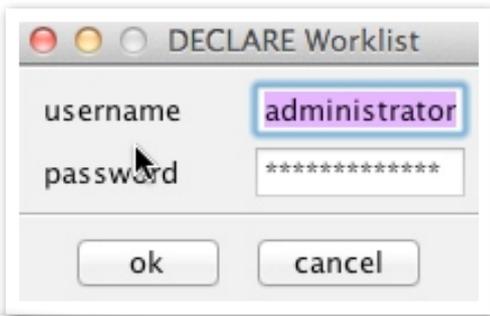
Install Declare

1. Use the latest source version of Declare from <https://svn.win.tue.nl/repos/prom/Declare/Prototype/trunk/> (and do not confuse this with the Declare ProM package).
2. Check this out as an Eclipse project.

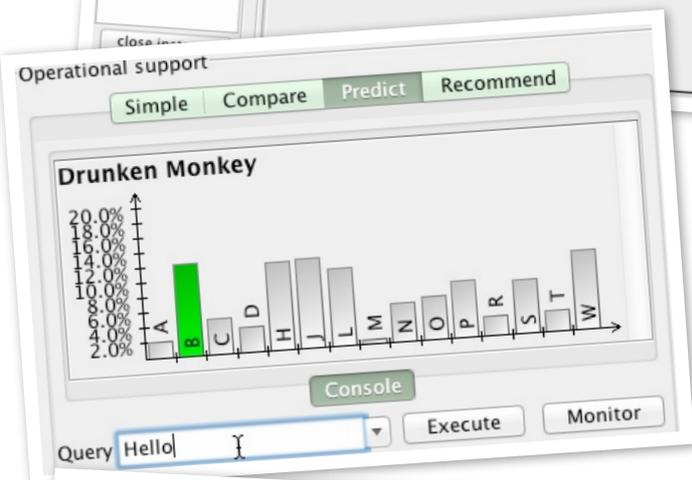
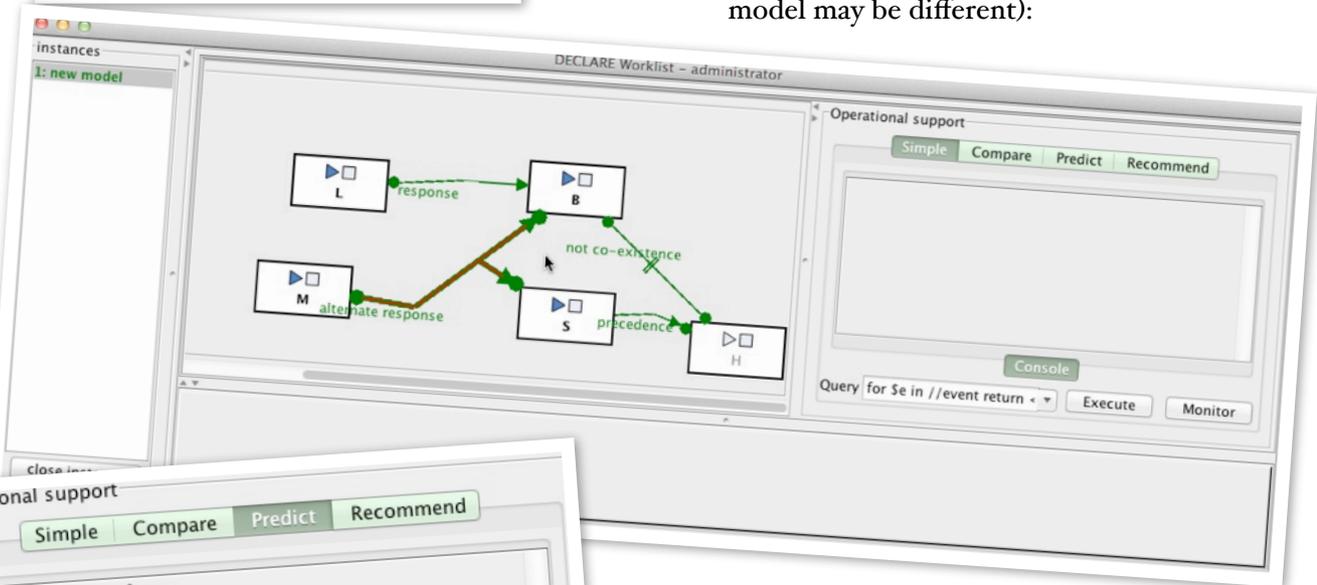
Starting Test Environment

1. Start the operational support service and one or more providers as explained earlier in this document.
2. Start the Declare Framework (use the launch configuration in the declare tool project).
3. Load one or more models using `load empty` in Declare Framework.
4. Start the Declare Worklist.





5. You may be asked to log in; the defaults are fine for the prototype.
6. You may need to resize the Worklist to reveal the operational support tab.
7. Declare should now look like this (except the model may be different):



Running Tests

Declare contains 4 tabs for executing different kinds of queries: Simple, Compare, Predict, and Recommend.

Simply enter a query in the Query Field and hit Execute to execute it. Not all providers may be able to respond to all kinds of queries.

To start a new session, simply close the instance from within the Declare Worklist of Framework and create a new one. This is, e.g., needed when adding providers.

