

XG Demonstration Software – User’s Guide

ICS-16763-TR-08-02
SRI Project No. 16763
Contract No. FA8750-05-C-0230

Prepared by
Rukman Senanayake (rukman@csl.sri.com)
Grit Denker (grit.denker@sri.com)
SRI International
Computer Science Laboratory
333 Ravenswood Ave
Menlo Park, CA 94025-3493

Prepared for
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

Acknowledgment and Disclaimer

This material is based upon work supported by the Defense Advanced Research Projects Agency and the United States Air Force under Contract Number FA8750-05-C-0230.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency or the United States Air Force.

Approved for Public Release, Distribution Unlimited

1. System Requirements

Operating System

This version of the XG policy reasoner developed by SRI International must be executed on a Linux 64 bit platform. Therefore, the provided demonstration software currently functions only on this platform. Support for Apple's Mac OS X is on the way. Windows platforms such as Windows XP or Windows Vista will not be supported because the underlying reasoning engine used for this XG Policy Reasoner is Maude (<http://maude.cs.uiuc.edu/download/>) and Maude runs on many of the Unix variants, but not on Windows platforms..

Java Runtime Environment

The XG software requires Java runtime environment version 5 or later. In a Linux console, type 'java -version' to find out which version of Java is available to you. Version 1.5 or later is required to run the XG Graphical User Interface (GUI).

The XGDemo folder contains two subfolders: XGGraphicalInterface and XGReasoner. The XGGraphicalInterface folder contains the GUI-related executables and dependencies. The XGReasoner folder contains the policy reasoner executables and dependencies. All the external libraries and modules used by the reasoner and the GUI are provided. For the GUI these external Jar files are located in the XGGraphicalInterface/libs folder. For the policy reasoner the required libraries are located in the XGReasoner/lib folder.

2. Launching the XG Demo

Two shell scripts are provided for executing the GUI and the XG Policy Reasoner (XGPR). The XGPR must be launched prior to executing the GUI.

1) Within a console window, change your current directory to <SomeFolder>/XGDemo/XGReasoner. <SomeFolder> is the location where you extracted the XGDemo-Linux-64bit.tar.gz archive.

2) Launch the XGPR using the command './run_pr.sh'.

You should see an output similar to the following:

```
-----  
PR: Starting up on localhost:7777  
Creating maude reasoner  
Loading reasoner specification...  
Loaded in 0.25 seconds  
Entering the server loop...  
-----
```

Depending on the Linux Kernel headers in your system, you may also see a warning message similar to

```
-----  
./pr: Symbol `__ZTIN11xercesc_2_713XMLEntityDeclE' has different size in shared object, consider  
re-linking  
-----
```

You can safely disregard this.

3) Within a separate console, change your current directory to <SomeFolder>/XGDemo/XGGraphicalInterface. <SomeFolder> is the location where you extracted the XGDemo-Linux-64bit.tar.gz archive.

4) Launch the XG GUI using the command './run_gui.sh'. You should see the XGLaunchBar application on the top left of your screen. This allows you to launch any number of XG graphical components (e.g., Policy Editors, Region Editors, Powermask Editors).

To use the information in the following sections, you must have successfully installed and launched the XGPR as well as the XG Demonstration GUI.

Note: The XGPR is a TCP/IP server and therefore requires you to actively terminate it when you no longer wish to interact with it. When you have finished using the XG software (after you have closed the GUI), in the terminal window used to launch the XGPR you must use the key combination 'Ctrl + C' to terminate the application. If the GUI is terminated you **MUST** terminate and restart the XGPR to avoid unpredictable reasoning results due to corrupted XGPR state tracking.

3. XG GUI Components

We implemented a GUI for a stand-alone reasoner demonstration of our Maude-based policy reasoner. The GUI features the following modular components: XG Launch bar, Policy Manager, Policy Editor, Region Editor, Powermask Editor, and Strategy Reasoner.

The Policy Manager allows adding and removing policies from the Policy Reasoner. In the Policy Editor, policies are described using the following parameters: name of policy, role (role required of the requesting radio), network ID (identity required of the requesting radio), region (region where the policy applies), emissions (maximum allowed emission under this policy described by a powermask), sensed power (maximum allowed sensed energy defined by a powermask), min and max time (date and time when the policy applies). The Region Editor allows specifying circular and rectangular regions, and the Powermask Editor defines powermasks that correspond to step functions. The Strategy Reasoner (SR) allows defining goals and features of a radio. Goals can be to maximize bandwidth or to maximize power. Goals guide the SSR in picking alternatives among the returned sets of solutions. Features of the radio can be used to pick among the opportunities returned by the Policy Reasoner and discard those that require capabilities not present in the radio. Capabilities are location detector, time detector, network membership, location evidence, time evidence, and network member. In the following we illustrate each of these components in more detail.

XG Launch Bar



Figure 1: XG Launch Bar - Main XG demonstration application

The initial screen with which you are presented in the XG GUI is the XG Launch Bar (Figure 1). This is the main application window. The editors that you require (such as Policy Editors and Powermask Editors) are launched from this application. The XG Launch Bar allows you to launch five different types of sub-applications:

1. Policy Manager
2. Policy Editor
3. Region Editor
4. Powermask Editor
5. Strategy Reasoner

You can launch multiple instances of the same editor or application. For example, it is common practice to have several Policy Editor panels open to compare policies, or to have several Strategy Reasoner panels open to compare interactions with the XGPR. A caveat regarding this is to not launch more than one Policy Manager. The Policy Manager is used exclusively for adding and removing

policies to and from the XGPR and is responsible for maintaining the state information of the policies loaded onto the XGPR. Having multiple instances of the Policy Manager running at the same time will lead to unpredictable behavior.

The first step of a typical XG software demonstration is to load required policies to the XGPR, which requires using the Policy Manager.

Policy Manager

Launching the Policy Manager brings up a window similar to Figure 2. This window has the following components.

Add Button: This button is used to load policies to the XGPR. Pressing this button displays a list of policies that have been defined using the Policy Editor. This list contains policies that have not yet been loaded into the XGPR. A particular policy cannot be loaded twice into the XGPR. The drop-down list is shown in Figure 3. If a policy loads successfully, the Policy Manager displays a message at the bottom of the panel in green as shown in Figure 4. If the policy fails to load, a failure message appears at the bottom of the screen in red.

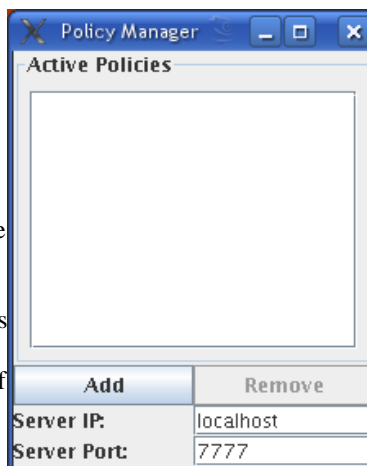


Figure 2: Policy Manager at startup

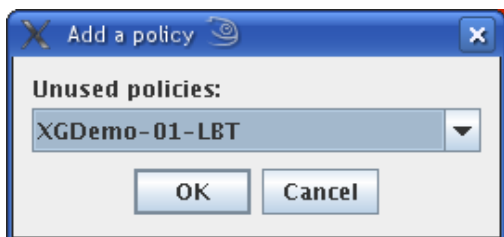


Figure 3: Drop-down list used to add policies to the XGPR

Remove Button: This button is used to unload policies from the XGPR. A policy that is successfully loaded into the XGPR appears on the 'Active Policies' list in the Policy Manager. To unload a policy, you must select it from this 'Active Policies' list.

Server IP: The system that is hosting the XGPR TCP/IP server must be provided to the Policy Manager, by using either a fully qualified domain name of a system such as 'cougar.csl.sri.com', where the system name is 'cougar' and the domain is csl.sri.com, or by using 'localhost' if the XGPR is

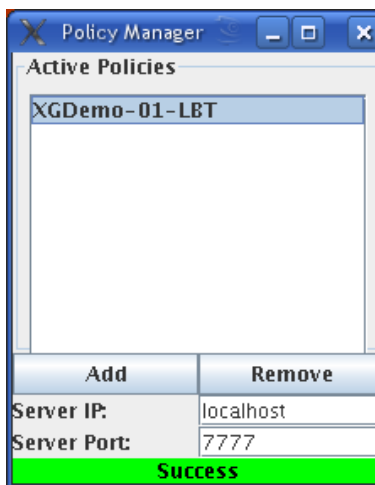


Figure 4: Policy Manager after successfully loading a policy

in the same system as the XG GUI (which is the default case, if you downloaded and ran the provided archive). You can also specify the IP address directly, for example, '192.168.1.33', if this is the IP of the system hosting the XGPR.

Server Port: This is the default port used the by the XGPR for communication and should NOT be changed.

Policy Editor

Policy Editor panels are used to edit or compare policies. When launched this provides a panel similar to Figure 5.

The 'Policy Name' field is used to specify the current policy's unique name. Every policy must have a unique name.

The 'Role' field is used to specify whether this policy has a 'Master' or 'Slave' attribute. The values for the 'Role' field must be defined in the XG-SRI Ontology.

The 'Network ID' field is used to specify the network identification string of the policy. This can be any string value.

The 'Region' drop-down list allows you to select a geographic region within which this policy is applicable. These regions are defined using the Region Editor.

The 'Emissions' drop-down list allows you to select from a predefined set of powermasks that is the applicable emission mask for this policy.

The 'Sensed Power' drop-down list allows you to select from a predefined set of powermasks that is the applicable sensed spectrum mask for this policy.

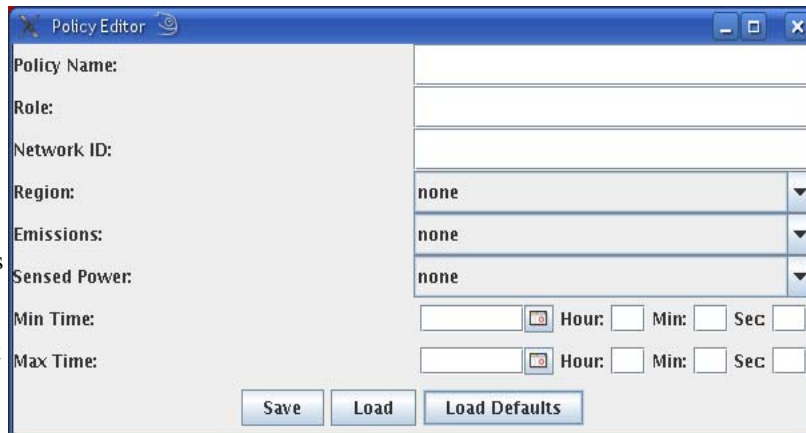


Figure 5: Policy Editor at startup

Formatted: Font: Lucida Grande, 14 pt, Bold

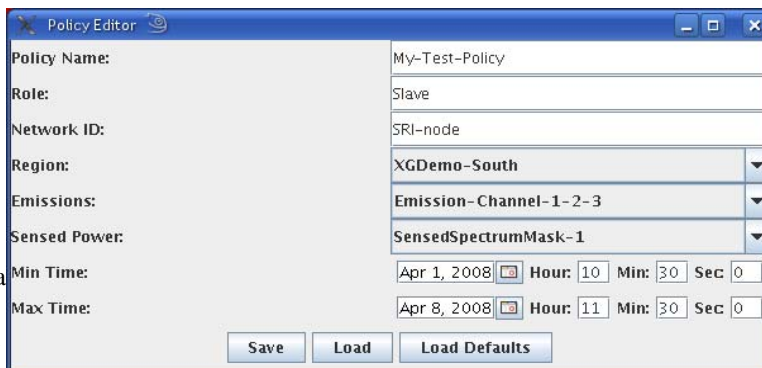


Figure 6: A fully specified policy example

The 'Min Time' and 'Max Time' calendar fields allow you to specify the date and time range to which the policy is applicable.

The 'Save' button is used to store the current policy. A saved policy can be loaded to the XGPR using the 'Add' button in the Policy Manager'.

The 'Load' button allows you to load a predefined policy. If you press this button, you will be presented with a drop-down list consisting of predefined policies.

The 'Load Defaults' button can be used to load some default values to a few of the fields in the Policy Editor.

A fully specified policy example is shown in Figure 6.

Region Editor

The Region Editor is used to define regions that can be then used in the Policy Editors. Two types of regions can be created: circular and rectangular. A rectangular region is defined by specifying the top left and bottom right coordinates of the region in Latitude/Longitude values. A circular region is specified by giving the Latitude/Longitude values for the center and the radius of the region in meters. The 'Regions' list on the right side of the Region Editor (see Figure 7) allows you to select a region from the currently defined set. The 'Add Rectangle Region' and 'Add Circular Region' buttons are used to add new regions.

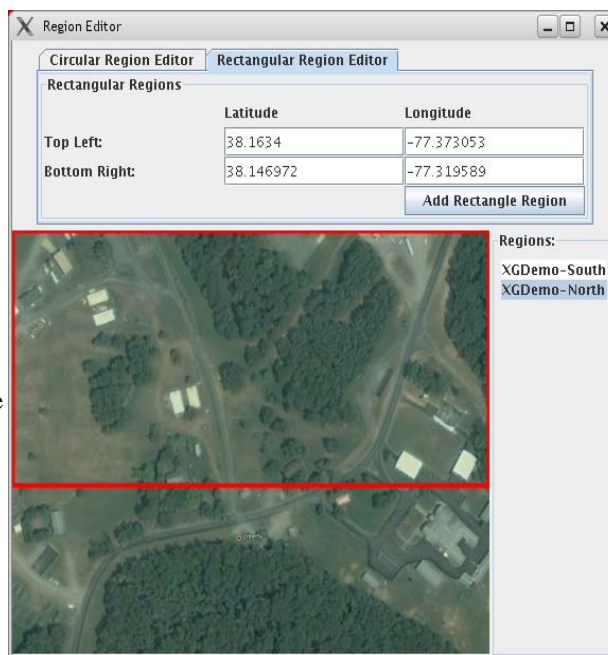


Figure 7. Region Editor – rectangle regions being accessed

Powermask Editor

The Powermask Editor is used to create emission masks and sensed spectrum masks. The initial view of the Powermask Editor is shown in Figure 8.

The Powermask Editor has three components: the table on the left edge that is used to specify Frequency and Power values, the central part that is used to show the current powermask, and the bottom section that is used to specify mandatory parameters as well as load, save, and update the display.

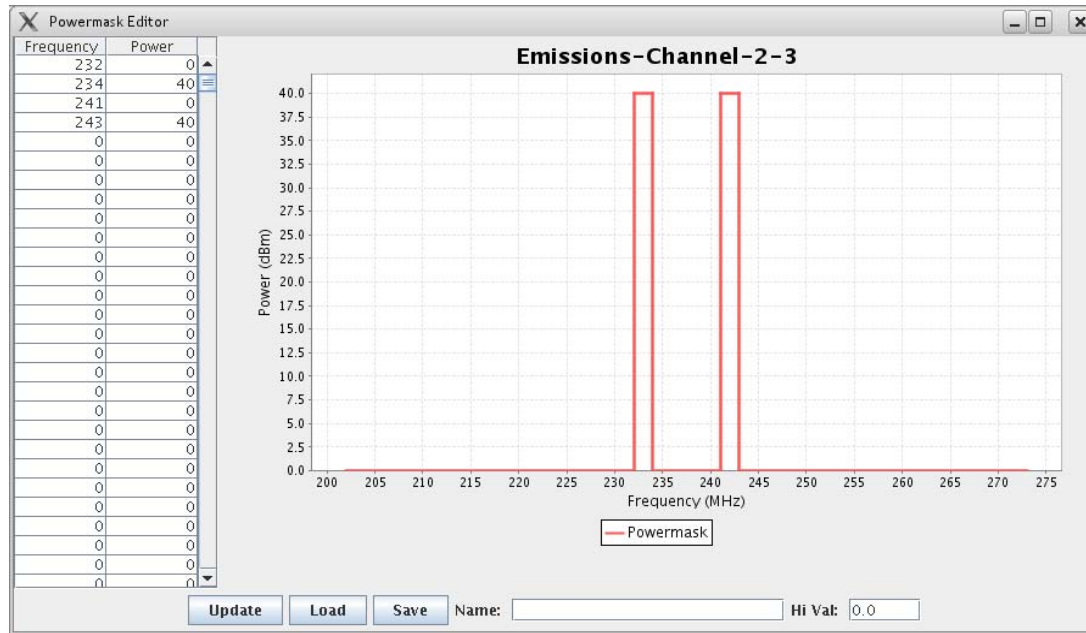


Figure 8: Powermask Editor with Emission-Channel-2-3 powermask loaded

For ease of explanation, let us load one of the default powermasks that ships with the XG software. Use the 'Load' button to load the 'Emission-Channel-2-3' powermask from the drop-down list. You should see a panel similar to Figure 8.

The powermask is specified as pairs of Frequency and Power values. Each pair of such values is interpreted as the range up to which the given Power value is true. Therefore, the two frequency/power value sets (232,0) and (234,40) in conjunction would be interpreted as any frequency less than 232 has a power value at most of 0 and any frequency between 232 and 234 has a power value of less than 40. The 'Hi Val' field is used to specify the final power value that is applicable to any frequency greater than the highest frequency specified in the table.

As with the other editors, the name provided for an emission mask or a sensed spectrum mask through this editor must be unique as well.

Strategy Reasoner

The Strategy Reasoner (SSR) panel is the main interaction method for sending requests to the XGPR and displaying the corresponding replies. The SSR consists of two tabular windows and two buttons (Manual Request, Launch SSR). When the SSR is first launched, you will see the Goals/Features tabular window, which is depicted in Figure 9.

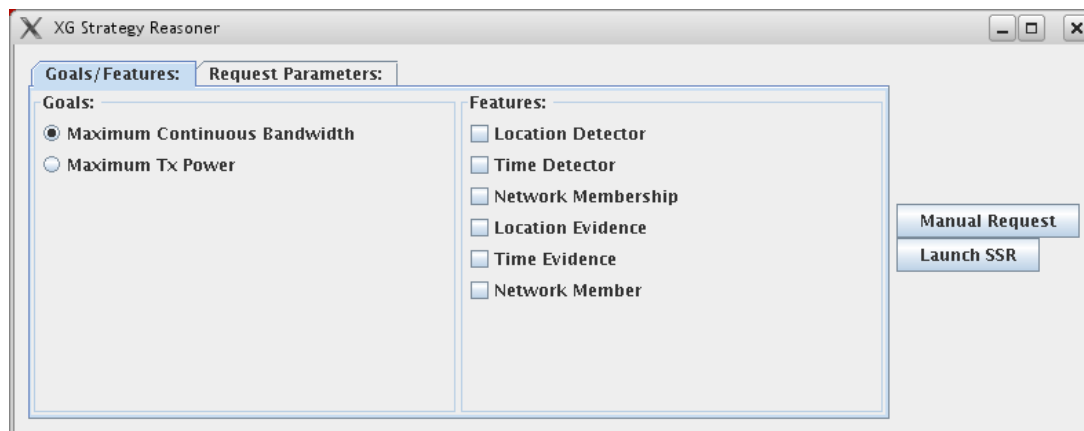


Figure 9: Strategy Reasoner - Goals/Features tabular window

As the name implies there are two parts to the Goals/Features tabular window: the panel that allows you to select the Goal of an interaction (either Maximum Continuous Bandwidth or Maximum Transmission Power) or to specify the features supported by the radio. The goal specified in this panel is used when the SSR is launched using the 'Launch SSR' button to determine what kind of request must be constructed automatically to fit the selected goal. The features of a radio are required if the policy needs a radio to have certain capabilities prior to transmission. For example, a policy that specifies a range in time for transmission requires that a radio have a 'Time Detector'. Another example would be that a policy that allows transmission within a certain geographic region requires that the radio provide 'Location Evidence'.

The 'Request Parameters' tabular window allows you to manually specify the parameters for a request. Any values specified in this tabular pane are used in constructing the request that will be sent to the XGPR (using either 'Manual Request' or 'Launch SSR'). The values that can be specified in the Request Parameters panel are analogous to the fields specified in the Policy Editor, except for Location, which is specified as a Latitude/Longitude pair instead of a region. The default view of the Request Parameters panel is shown in Figure 10.

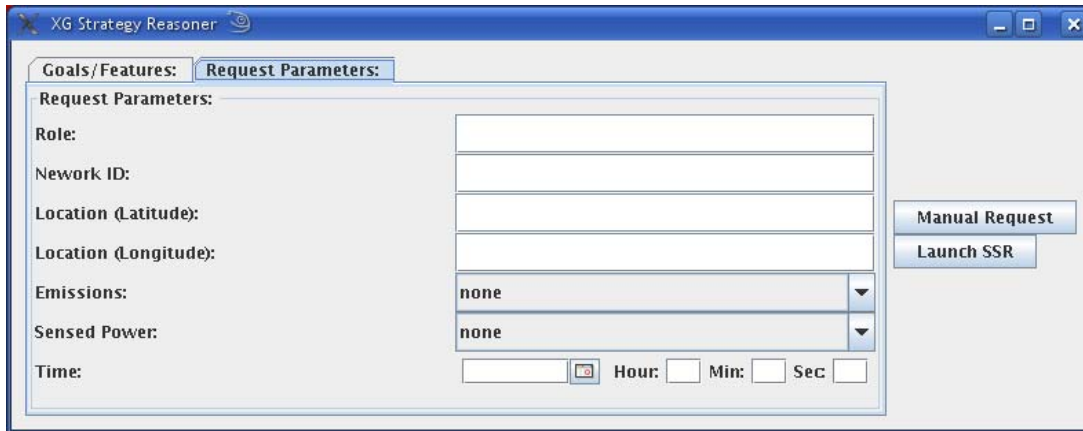


Figure 10: Strategy Reasoner - Request Parameters tabular window

For illustrative purposes, let us assume that the policy currently loaded into the XGPR is 'XGDemo-01-LBT' (which is one of the policies that ships with this software). This is a simple policy that allows transmission if the emissions of the radio within the emission mask are specified by powermask 'Emission-Channel-1-2-3'. The loading of the policy is given in the snapshot of the Policy Manager (Figure 11), and the relevant powermask is shown in Figure 12.

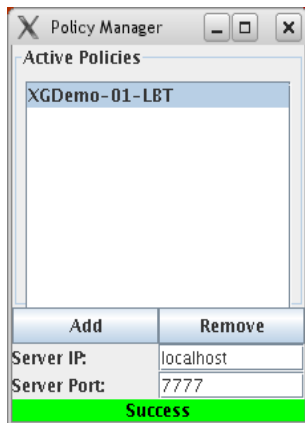


Figure 11: Policy XGDemo-01-LBT loaded into the XGPR

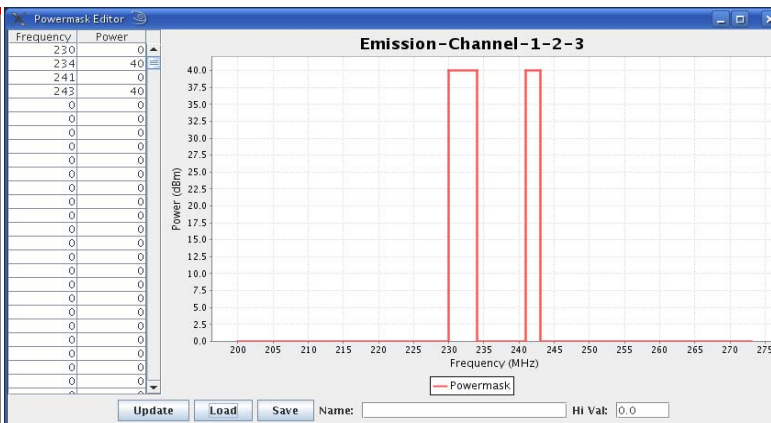


Figure 12: Powermask Emission-Channel-1-2-3

If you now send an empty request to the XGPR by leaving all the fields in the 'Request Parameters' tab empty and you use the 'Manual Request' button, the XGPR replies with three opportunities (Figure 13). An XG reply to a manual request has two main components.

In the first interaction, the graphical indicator shows whether transmission was allowed, denied, or approved conditionally. In this case the amber swatch next to the 'Conditional' label indicates that the transmission was approved within the constraints specified in each of the opportunity tabs (in this case there are three). The 'XG Reply' tab that is under the Opportunity tabs displays the complete response as an XML document object model. Note that Opportunities 2 and 3 are conjunctions; therefore, all three constraints in each of these two opportunities must be satisfied for transmission to be allowed.

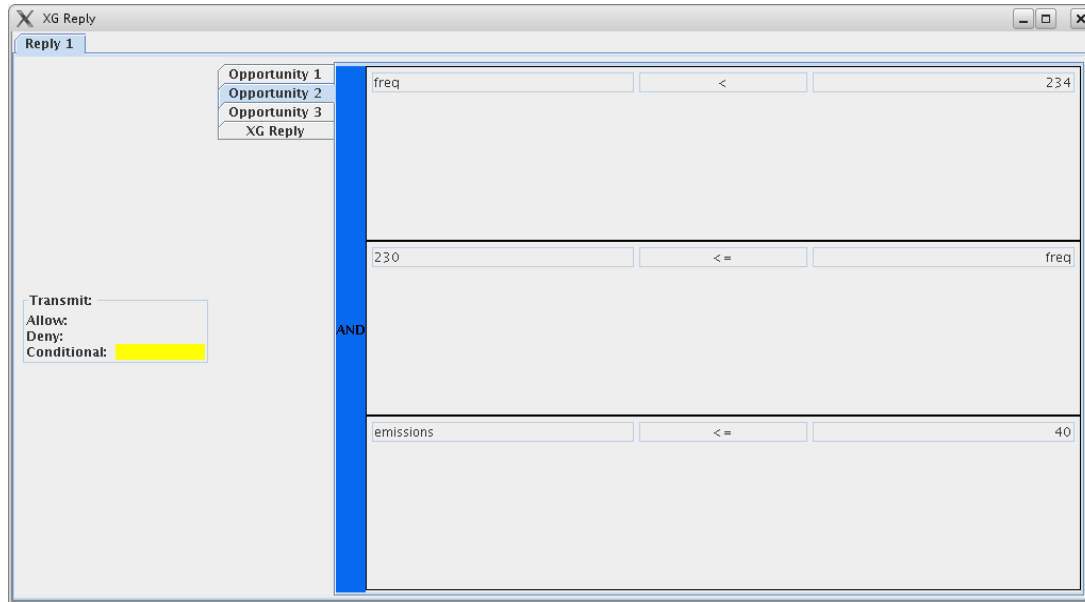


Figure 13: Reply from XGPR with Opportunity 2 selected

Let us assume now that you specified the emission mask for the radio as 'Emission-Channel-1'. The resulting reply from the XGPR is given in Figure 14, and transmission has been allowed since this channel is within the emission mask specified in the currently loaded policy.

If you select the 'Launch SSR' option, the system performs several tasks. First, it checks to see if any of the request parameters are specified in the 'Request Parameters' tab. If so, these values are added to the request. Then it checks to see if the radio supports any of the features listed in the 'Goals/Features' tab. If so, these are added to the request as well. Then the Strategy Reasoner sends a request to the XGPR and analyzes the reply. If the reply is to allow or deny transmission, this is displayed and no further action is taken.

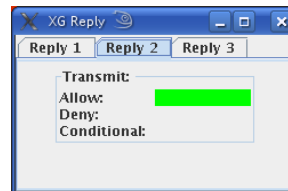


Figure 14: Reply from XGPR - Transmission allowed

If the reply is for a conditional transmission, the SSR analyzes the returned constraints and attempts to construct the best reply it can in line with the goal selected in the 'Goals' panel of the SR. Then this request is sent to the XGPR, and the reply is displayed in the same reply panel as the previous interaction. Therefore, if the 'Launch SSR' option is used and a conditional

transmission is allowed in the first reply there will always be two interactions between the SSR and the XGPR.

The second interaction may result in a reply that allows, denies, or conditionally allows transmission. In this case, the first interaction results in a conditional transmission with three opportunities (Figure 15), and the second interaction results in a reply that allows transmission (Figure 16) since the SSR had sufficient information to generate a fully specified request.

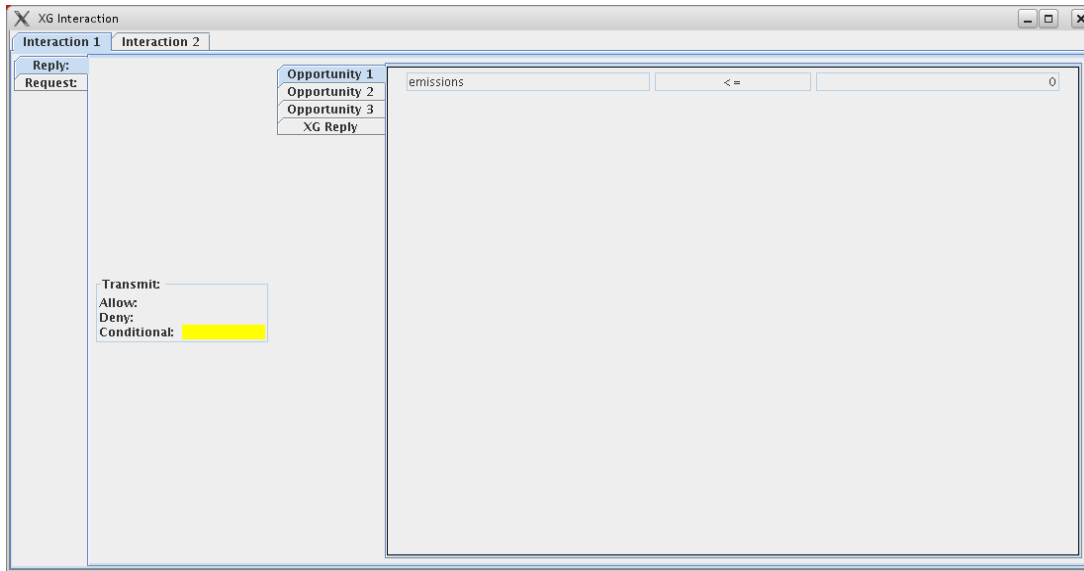


Figure 15: XGPR Reply – first interaction results in three opportunities

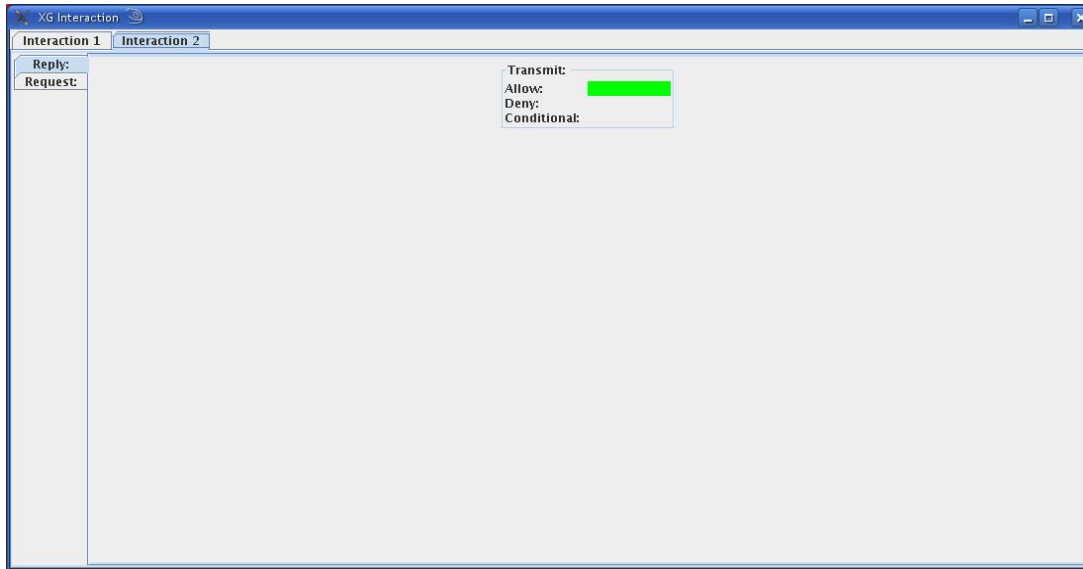


Figure 16: XGPR Reply – second interaction results in transmission being allowed

The 'XG Reply' tab under the opportunities tabs of an interaction can be used to view the XML representation returned by the XGPR (Figure 17). This panel is a Java Document Object Model of the XML string received by the SSR from the XGPR and contains the complete proof state of the interaction between the SSR and XGPR.

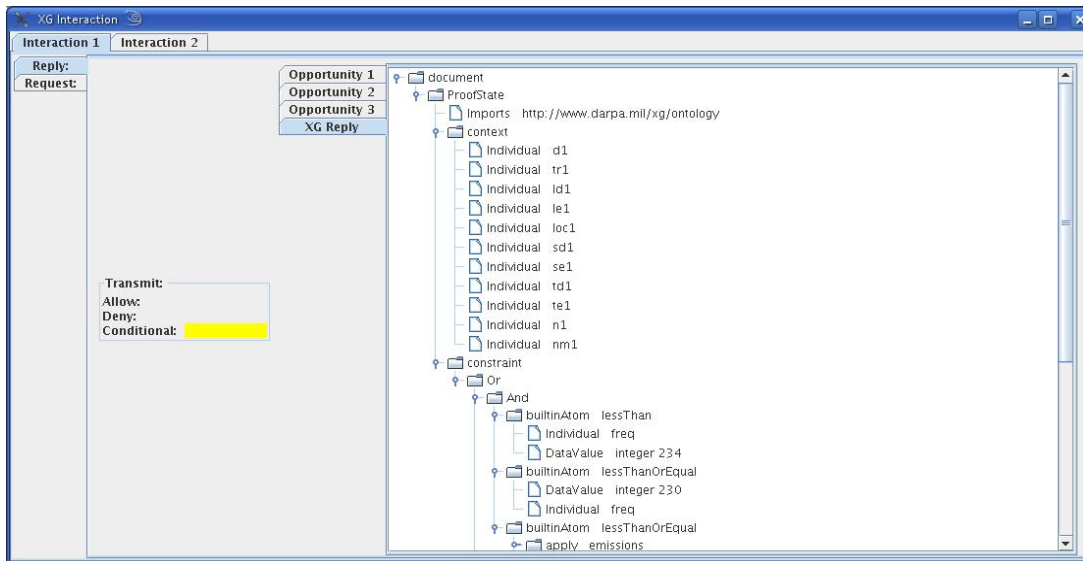


Figure 17: XML Document Object Model of the proofstate