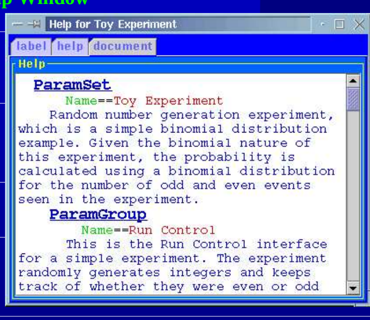
Dynamic Graphical User Interfaces using XML and JPython

Fermilab

Goal

The dynamic GUI building effort is part of the Online Data to Experiments (ODE) project at Fermilab. The main goal of dynamic GUI effort is to develop a framework which will allow greatly enhanced flexibility and customization of application GUI's for testing, data acquisition and monitoring environments. We are also developing a way for the user to dynamically redefine the underlying code to be executed when a widget is selected (a button click for example).

Help Window



Current Status

The initial phase of the project has been completed. The prototype dynamic GUI building program was written in Jpython and uses XML for defining the configuration information. All of the colors can be set in the display configuration file. The actions associaited with each displayed widget are named in the configuration data file. If an action is a simple single line of JPython code, then that code can be placed directly in the configuration file. For executing more complex instructions, the name of the method is listed in the configuration file and the method code is placed in a file of user methods and classes. This user method file can be re-sourced on the fly allowing an application GUI to dynamically change the execution code for it's actions without having to rebuild the GUI.

The current prototype only defines buttons and text entry field widgets for each parameter defined in the configuration file. Widgets can be grouped within a page or placed on separate tabbed panes in the application GUI. We have restricted ourselves to a minimal set of widgets initially since it is sufficient to demonstrate the feasibility of the concept. There is also an automated help window that displays the documentation information listed in the configuration file.

Introduction

Modern interfaces are often graphical in nature. One of the drawbacks is that they generally provide a limited user adjustable configuration. Since users have vastly different preferences, applications designers usually have to take a least common denominator approach to configuration options to keep their projects manageable. Thus, it is uncommon to find an interface that allows the user to dynamically redefine the executable commands controlled by the interface. We believe that data acquisition and monitoring environments would benefit from greatly enhanced flexibility and the modern tools like XML, Jpython and Java can provide this flexibility.

What is XML?

The eXtensible Markup Language (XML) which is like HTML (Hypertext Markup Language) with the ability to define your own tags, is well suited as a language for the configuration information. The advent of XML provides a good base on which to develop documents as well as applications. There are now XML parsers available from several sources and for several different programming languages. The availability of parsers and the numerous examples of projects using XML that can be found on the webdemonstrate that XML is rapidly becoming a recognized and accepted tool for software development.

Basic Principles

The dynamic GUI builder should use a file driven configuration approach which allows the user to easily save their preferred configuration. The builder should provide the ability to dynamically rebuild the application GUI incorperating any changes. Information related to functionality such as parameters, actions and documentation can be considered the application data, whereas information such as widget appearance is display information. Keeping to the XML philosophy, the data and display information should be stored separately. The application GUI should be able to redefine the action code (code executed when a widget is selected) dynamically. The program should be as platform independent as possible.

Future Plans

The initial phase has been quite successful and is an encouraging sign as we begin the next phase of the effort which is to design and develop a real dynamic GUI builder. Keeping data and display information separate will be a challenge. The design of the configuration infrastructure will be expanded. A system for scheduling repeated activities implemented. Additional widget types will be supported. Work on providing the infrastructure to enable integration with other components of the ODE project will begin. Some of the other components include merlin (a message logger), LabView., and an electronic logbook.

Performance will be a big challenge for the project. Java is slower than native languages like C or C++ because it is an interpreted language. JPython is also an interpreted language makes matters worse. Initially most of the code will be written in JPython, and to a lesser extend Java, to make paltform independence as easy as possible. If performance becomes a problem, some sections of the code will have to be written in a native language and incorperated using the Java to Native Interface (JNI).

The Main Control Panel

What is Jpython?

JPython is a rewrite in pure Java of the Python Object Oriented scripting language. JPython is an easy to use scripting language that is Java aware and thus allows one to use the Swing classes for rich GUI development. Like most scripting languages, Jpython is an ideal tool for rapid prototyping of applications. Being Java aware also means that re-implementing sections of code in Java if needed would be fairly easy to do. Jpython, like Java, is also highly platform independent.

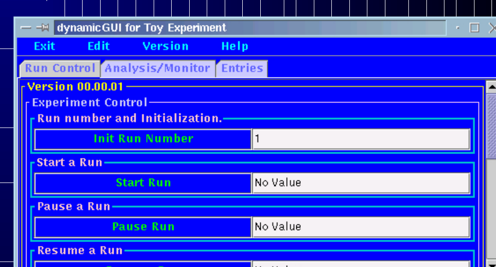
Initial Phase

The initial phase of the dynamic GUI building effort was to develop a simple prototype for evaluating the feasibility of the dynamic GUI concept. The focus of the prototype was on demonstrating that various requirements for a real system could be achieved as opposed to being an alpha version of the program.

Development Environment

We are currently using Java 1.1.6, JPython 1.1 alpha 3, and the xml4j parser 2.0.15 from IBM on Linux platforms.

Application Gui



Additional Resources

- XML: http://www.w3.org/XML/
- Jpython: http://www.jpython.org/
- ODE: http://www-ese.fnal.gov/ods/ode/
- xml4j parser: http://www.alphaWorks.ibm.com/
- merlin: http://www-b0.fnal.gov:8000/merlin/

Fermi National Accelerator Laboratory, a Department of Energy Laboratory,operated under contract by Universities Research Association.