LabServer Architecture

Draft - 7 Sep 2009 - LJP

The architecture consists of two parts:

1. **LabServer Core Libraries** – The majority of LabServer code is common to all types of experiments. This is a result of the iLab ServiceBroker-to-LabServer API which is common to all Batch LabServers. All experiment input parameters (specification) are specified in an XML format and are sent to the LabServer for processing. The LabServer parses the specification for validity and then places the experiment specification on a queue ready for an experiment engine to run the experiment according to that specification. When the experiment has completed, the results are saved in an XML format and are available for retrieval.

2. **LabServer Specific Implementation** – Each type of experiment has it own specific set of input parameters and output results. The experiment engine needs to know the format of the specification so that it can extract the parameters needed to run the experiment. These parameters include the "setup" or particular configuration of the experiment that will be run, which in turn determines the equipment driver that will be invoked to run the experiment.

Source code on SourceForge

The source code for some Batch LabServers is available on SourceForge in the iLabsProject at: "http://sourceforge.net/projects/ilabproject/". To get to the code tree, go to: "http://ilabproject.svn.sourceforge.net/viewvc/ilabproject/" and browse to "/trunk/LabServer".

- **LabLibraries** – This common LabServer code consists of the base classes for the Experiment Manager, Experiment Engine and Experiment Setup Drivers. It also includes classes for persistent storage handling of the experiment queue, results and statistics.

  Also included here is the code which is common to all HTML LabClients. The LabClient provides a way of creating an experiment specification and submitting it to the LabServer for execution. The LabClient also retrieves the results and displays them in a form that can be viewed by the user. The LabClient also allows the results to be saved to a CSV file where they can then be processed or graphed with third-party software.

  For those experiments that use real hardware, the code common to all LabEquipment servers is also provided. This basically only contains the code for powering up equipment and then powering it down again after a timeout. This functionality will not be used with all LabEquipment servers and can be disabled by a setting in the application's configuration file (web.config).

  For development, a Dummy ServiceBroker is provided. This allows the development of the LabServer/LabClient combination without interference from an iLab ServiceBroker. It has exactly the same web service interface as an iLab ServiceBroker but implements pass-through methods which allow the LabClient to communicate with the LabServer. The Dummy ServiceBroker generates experiment numbers or IDs when experiments are submitted.

- **ACMachine** – A 3-phase 415V AC motor is run under several conditions to determine its characteristics. The input parameters include only the "setup". The output results include the phase-to-phase voltage and current, and the power factor.

  Power control equipment is used to drive the AC motor. Equipment commands are received through a network connection using the ModBus protocol over TCP/IP.

- **DCMachine** – A 3-phase 415V AC motor drives or is driven by a DC motor/generator. The DC motor/generator is run under several conditions to determine its characteristics. The input parameters, depending on the "setup", include armature voltage, field current, motor speed and load torque. The output results include the armature voltage, motor speed and field current.

  Power control equipment is used to drive the AC and DC motors. Equipment commands are received through a network connection using the ModBus protocol over TCP/IP.

- **Radioactivity** – The emissions of a radioactive source are measured by a Geiger tube and Radiation Counter over periods of time and at varying distances. The effects on the emissions by placing various materials between the source and tube can also be determined. The input parameters, depending on the "setup", include the type of radioactive source, the type of absorbing material, the distance between the source and Geiger tube and the time period of measurement. The output results include the emissions count at each distance.

  A National Instruments FlexMotion controller card is installed in the computer to drive the motors on the equipment. There is also a microcontoller board connected to the serial port of the computer to capture the emissions count for specific time intervals.

- **Time-Of-Day** – The time/date is obtained from either the local computer's clock or from a network time server. The input parameters include the "setup" and the format in which to display the time and date as an ASCII string.

  There is no hardware associated with this experiment except a network connection. This makes it ideal for beginners to get aquainted with the LabServer architecture.

- **Template** – For the beginner, implementing a LabServer for a new experiment is not a trivial task. The template provides a means for getting that new experiment up and running sooner by concentrating on only those parts of the LabServer architecture that are specific to that new experiment.