Enhancing the *CS* Framework for Distributed Control Systems

D. Beck, H. Brand, and S. Götte

GSI, Darmstadt, Germany

The *CS* framework is a LabVIEW based framework for developing event driven, multi-threaded control systems using an object oriented approach. In December 2006, version 3.00 of *CS* has been released. The new version is a major change addressing a couple of issues that are of prime importance for large scale distributed control systems.

Introduction

The *CS* framework is in use since a few years at about ten experiments at four institutes. Instead of supporting a maximum number of process variables, the main focus of *CS* is to provide a basis for control systems requiring a high flexibility and performance. A detailed description of *CS* is given in [1]. This text aims at describing the enhancements by the most recent version 3.00.

DIM as Communication Layer

Earlier versions of *CS* allowed for distributed control systems but required the knowledge of the node name for each process variable. Moreover, a general built-in observer mechanism where many observers may register to receive updates of one published process variable was missing.

The entire communication layer has been changed to DIM (Distributed Information Management) [2], which is a light weight protocol for inter-process communication based on TCP/IP. It follows the concept of named services. Clients may receive information from a service (observer pattern) or may send a command to a server (command pattern).

Switching to DIM as communication layer was possible after implementing a LabVIEW interface for DIM. This drastically increased the performance in the communication layer of *CS* and solved a few other problems not mentioned here. Using a *CS* system with one million DIM services was tested successfully.

Process Management

Starting with version 3.00, *CS* is complemented by a simple process management system. Typically, a couple of processes must run on each node to provide the functionality of the control system. However, each node may require a different set of processes. The process management system takes care of starting, restarting or stopping processes in distributed environments. A server publishes a list of processes for each node in the distributed system together with command line parameters and options for restarting a process in case it has been stopped or crashed. On each node a client subscribes to that list and takes care to handle the required processes. The process management system is available for MS-Windows and Linux and supports a mix of both operating systems at the same time.

Access System

A distributed control system with more than one console requires a mechanism to lock subsystems for individual operators or tasks. Such a reservation mechanism is not intended for preventing malicious attacks but for preventing accidental changes of a parameter. With *CS* 3.00, a reservation mechanism has been implemented that becomes effective when using the command pattern.

Application

The new version of *CS* represents a major change in the *CS* framework. The usage of common design patterns within *CS* has become much simpler in many cases. At the same time it was tried to provide as much backward compatibility as possible. However, experiment specific code based on the *CS* framework needs to be migrated to the new version. Thus, switching to the new version requires some work by the experiments.

At GSI, the new version is already in use at two facilities. First, the PHELIX facility [3] has the most complex control system (12 PCs) based on *CS* and represents a good test case for the new *CS* version. PHELIX started to successfully use the new version already in an early stage of its development. Second, a small test-bed of eight PCs has been set up and is permanently in operation. Its main purpose is to test scalability, stability and performance of a distributed *CS* system when being operated for many weeks. Other experiments like SHIPTRAP are about to switch to the new version in the near future. The upcoming facilities HIPTRAP and MATS at FAIR have already decided to use *CS*.

Summary

The *CS* framework has done a major step forward by adding missing features that are required for larger, distributed control system. The main change is the usage of DIM as communication layer. The software has been released in December 2007 and is already in use.

References

[1] D. Beck et al., Nucl. Instr. Meth. **A** 527 (2004) 567-579, http://wiki.gsi.de/cgi-bin/view/CSframework.

[2] C. Gaspar, M. Dönszelmann, Ph. Charpentier, International Conference on Computing in High Energy and Nuclear Physics (Padova, Italy, 1-11 February 2000), http://www.cern.ch/dim.

[3] E. W. Gaul et al., GSI Scientific Report 2002 (2003) 101-103.