

Status of the CS framework and its successor CS++

H. Brand, D. Neidherr
GSI, Darmstadt, Germany

The CS framework is a LabVIEW based framework for developing event driven, multi-threaded control systems using an artificial object oriented approach. CS was upgraded to version 3.4 based on LabVIEW 2014. Upgrade experience is summarized.

CS++ will become the successor of CS and is based on NI's Actor Framework which is using the native object oriented programming approach consequently following the data-flow paradigm.

Motivation

The CS framework is in use since a couple of years at about 15 experiments at 10 institutes. A detailed description of CS is given in [1]. This text aims at describing the enhancements by the most recent version 3.4 and upgrade experience to LabVIEW 2014.

CS framework is using an artificial approach to object oriented programming in LabVIEW. Many features rely on convention and cannot be enforced since the support of the development environment is missing. Therefore we are working on its successor CS++, based on NI's Actor Framework [2], to be used for FAIR experiment control systems. The Actor Framework provides some simple base classes for active objects (threads) and well defined communication mechanism. The AF implements a much simpler design than the HGF Class Library [3], so it is much easier for newcomers to learn. In addition we can profit from NI support and community developments, so that for example the migration of the core components within the CS++ is already done by NI.

Status of CS framework

Last year the CS framework had its 10 year anniversary. Since 2004 the core of the framework has been migrated to the most recent LabVIEW versions every three to four years. For the developers of experiment specific CS classes this normally means upgrading to the new LabVIEW version. They just have to copy the new core system and recompile their executables. The CS upgrade to 3.4 need more work.

For an unknown reason NI changed the way references are compared with each other. This leads not only to important changes in the core system of the CS but also to changes in the source code of each class. For this reason we wrote a tool using *VI-Scripting* methods to modify parts of the class specific source code automatically.

NI found a security issue which they closed in newer LabVIEW versions. For the CS this leads to several small changes for example in the way how an executable has to be built.

HITRAP @ GSI as well as ISOLTRAP @ CERN and LEBIT @ MSU migrated their systems to CS 3.3 (LV

2012). The PHELIX experiment @ GSI has already migrated to CS 3.4 (LV 2014). The step from CS 3.3 to 3.4 is marginal.

Status CS++

CS++ follows the recommendations of the Actor Framework. It provides base classes of common interest for objects to be used as entities, *CS++Base.lvclass*, and derived actor classes *CS++BaseActor.lvclass*.

- *CS++MessageLogger* client for DSC and Syslog
- *PVConnection* and *PVMonitor* for Shared Variables or DIM communication.
- *CS++DeviceActor* and *CS++DeviceGUI* e.g. *CS++DCPwr*, *CS++DMM*, *CS++Fgen*, *CS++Osci*, etc. Concrete implementation classes wrapping the IVI-Driver are already available.
- *CS++Factory* provides the methods to create initialized objects of such classes at runtime reading initialization data from configuration file.
- *CS++Reference* can be used to maintain object references respecting data-flow.
- More passive data classes are available providing new datatypes such as PV- or Alarm&Event data.
- *CS++DSCAlarmViewer* and *CS++DSCtrendViewer* connecting to the historical DSC database.

Status & outlook

The CS framework 3.4 is stable and in good shape. It will be maintained in future since many experiments rely on it and it cannot be easily replaced.

The CS++ class libraries are still under development, but already available under the terms of the European Union Public Licence (EUPL) on Github [4]. Some simple demo applications to illustrate the usage and extension of the base classes are available. CS++ will become the base for the serial test-stand sequencer for the SIS-100 dipole magnets. CS++ applications can easily connect to CS Framework applications using DIM [5].

References

- [1] D. Beck et al., Nucl. Instr. Meth. A 527 (2004) 567-579, <http://wiki.gsi.de/cgi-bin/view/CSframework>,
- [2] <https://decibel.ni.com/content/groups/actor-framework-2011?view=overview>
- [3] F.Berck "Prototyp für ein mobiles Agentensystem in NI LabVIEW", Hochschule Darmstadt University of Applied Science, FB EIT, 2010
- [4] <https://github.com/HB-GSI/CSPP>
- [5] <http://www.cern.ch/dim>

