



Contribution ID: 5

Type: **not specified**

Hard Rock CAFE

Wednesday 5 October 2011 10:00 (15 minutes)

Summary

A number of channel access (CA) interfaces to various high-level programming languages already exist within the EPICS extensions domain. Many of these are extensions to declarative and 4th generation languages each of which incorporates its own dedicated C/C++ class to establish channel access connectivity. Such CA classes are often embedded with function calls provided by the language's C/C++ API, and are thus, by definition, tied to the programming language at hand. Another approach, however, is to enforce a logical boundary between the channel access components and the specifics of a given application's C/C++ extension framework. Such a methodology then provides the basis for the modular use of a single C++ CA class across a number of C/C++ based programming languages. CAFE (Channel Access interFacE) is a new C++ library, built upon the standard channel access client library, that provides a modern, multi-faceted interface to EPICS. CAFE makes extensive use of templates and containers with multiple STL-compatible access methods that improve efficiency, flexibility, and performance. Stability and robustness, as 'Hard as Rock', are accomplished by ensuring that connectivity to EPICS records remain in a well defined state in every eventuality. CAFE presents the user with a number of options for instigating synchronous and asynchronous operations. In addition to providing basic read/write methods, a further layer of abstraction provides a convenient interface to groups of objects or collections of logical data sets, which can be easily instantiated through an XML-based configuration mechanism. CAFE aims to provide a comprehensive, yet extendable, C++ CA class that can be well tested in-house through its adoption in bindings to various C/C++ based high-level languages, and which, if proven effective, will ultimately serve to reduce the load on maintenance associated with CA client-based software. CAFE's suitability for use within different applications is demonstrated with examples from high performance Qt GUI control widgets, to event processing agents that propagate data through OMG's Data Distribution Service (DDS), and to script-like frameworks such as MATLAB.

Author: Dr CHRIN, Jan (Paul Scherrer Institut)

Presenter: Dr CHRIN, Jan (Paul Scherrer Institut)

Track Classification: Presentations