BRINGING ACCELERATOR MODELS TO CONTROL SYSTEM STUDIO*

N. Malitsky#, K. Shroff, BNL, Upton, NY 11973, U.S.A
C. Xiaomeng, Stony Brook University, Stony Brook, NY 11794, U.S.A.

Abstract

The paper is the next logical step in the evolution of the new EPICS-based high-level accelerator application environment. This project presents the connection of its middle layers servers with the new Eclipse-based operational toolkit, Control System Studio. The approach is illustrated by the implementation of the Model Independent Analysis application involving three key servers: Machine, Online Model, and Virtual Accelerator.

INTRODUCTION

Commission and operation of the modern accelerator facilities is a complex and composite procedure dealing with variety of control applications. In general, all programs can be divided into two categories: engineering tools and high-level applications. One of the major classification criteria is the level of associated algorithms. For example, the typical engineering tool, Operator Interface (OPI), provides direct access to process variables via a generic set of visual components. On the other hand, the high-level application for correcting orbits might integrate several alternative algorithms. This conceptual difference affects many technical-related aspects, such as the software architecture, development approach, and others. Usually, engineering tools are implemented by a control group using a common consolidated framework and one of the primary compiled programming languages. The high-level applications are developed, in most cases, by accelerator physicists preferring the scripting environment that facilitates rapid prototyping of new scenarios. As a result, high-level application software is often represented by a mixture of heterogeneous tools based on a variety of technologies and programming languages.

While resolving flexibility issues, the heterogeneous environment however diminishes other software quality characteristics, such as reliability, maintainability, etc. Furthermore, it significantly restricts shareability of successful approaches among different accelerator projects. In reality, there is a big overlap between the two categories of applications and their implementation can be reconsidered from an alternate classification perspective: the level of maturity of their techniques. This point of view immediately leads to the development of a common homogeneous environment that is able to sequentially accommodate the proven high-level applications as the corresponding remote service and/or client plug-ins.

In the context of the EPICS control system [1], the development of this approach is facilitated by two complementary projects: the new PvData-based communication protocol [2] and the Control System Studio [3] built on the Eclipse Rich Client Platform. In a previous paper [4], we explored the new communication protocol for developing the Model Independent Analysis (MIA [5]) client application using two middle layer servers, Online Design Model and Virtual Accelerator. The selection of the MIA application as the first candidate was determined by two major factors. First, MIA has become one of the mature tools in commissioning and operation of several accelerator facilities. Second, it represents a complex test bed for the different technical decisions introduced and implemented in the new high-level application framework. This paper concludes the approach by implementing the MIA application as a CSS plug-in.

ARCHITECTURE

A typical high-level accelerator application environment is designed after a three-tier architecture illustrated in Figure 1. In this approach, front end computers controlling physical devices form the bottom tier. Middle layer servers, such as Virtual Accelerator or Online Model, maintain common data structures and algorithms which are shared and used by an open collection of top tier thick and thin client applications.

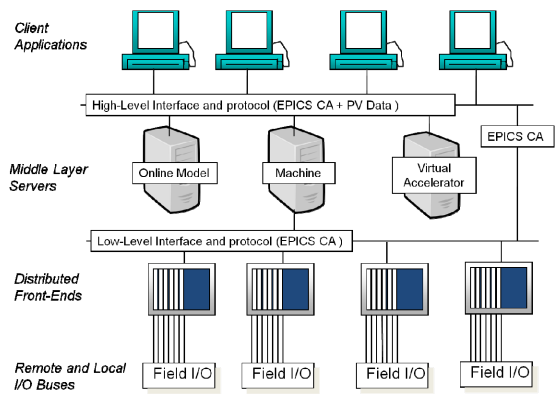

Figure 1: Three-tier high level application environment.

Despite the common environment, requirements of each layer are different. As a result, the modern accelerator facilities are using at least two different technologies: low-level control infrastructures like RHIC ADOs and high-level middleware, like JLAB CDEV, OMG CORBA, and Java JMS. Recently, The EPICS-DDS project [6] addressed this problem by providing a consolidated approach based on the Experimental Physics and Industrial Control System (EPICS) and the OMG Data Distribution Service specification.

*Work supported by DOE contract DE-AC02-98CH10886
#malitsky@bnl.gov

EPICS is an open source framework and a rich collection of tools are developed collaboratively and used worldwide for building distributed real-time control systems in large-scale scientific projects: accelerators, detector systems, telescopes and others. Its base infrastructure consists of two layers: distributed Input/Output Controllers (IOC) and Operator Interface applications. IOC provides uniform interfaces to heterogeneous physical devices which can be accessed and monitored by clients via the Channel Access (CA) communication protocol.

One of the basic ideas of the EPICS-DDS project is reuse of the EPICS waveform record for accommodating the serialized representation of structured data. The approach immediately creates a basis for developing the EPICS-based middle layer servers. According to the EPICS-DDS uniform scenario, these servers provide the states (the most current values) of the associated data structures shared by other servers and high-level client-subscribers. The majority of accelerator use cases require three data types: collection of the accelerator element parameters, design of linear and non-linear optics functions, and turn-by-turn data, measured or calculated in beam position monitors.

The Machine server represents a central component of this facility. It maintains a state of magnet strengths used by all other participants. Other middle layer servers and high-level clients subscribe to the Machine server for synchronizing their own containers. Particularly, the Online Model and Virtual Accelerator servers recalculate and update their own states of the design optics and turn-by-turn beam data respectively. By following the Service-Oriented Architecture (SOA) principles, these online services can be configured with the different off-line accelerator programs.

Control System Studio (CSS) provides the integrated client environment based on the Open Service Gateway Initiative (OSGI) framework and Eclipse toolkit. Its development started at 2005 and now CSS combines a collection of the important engineering applications [7], such as Synoptic Display, Data Browser, The Best Ever Alarm System Toolkit (BEAST), and others.

MIA PLUGIN

Model Independent Analysis (MIA [5]) is a spatial-temporal mode analysis technique used for processing turn-by-turn data measured at a large number of beam position monitors. The approach is based on Singular Value Decomposition (SVD) which decomposes the spatial-temporal variations of beam motion into a few orthogonal modes. As a result, MIA is able to provide the different types of important information, such as optical properties of accelerators and the performance of a BPM system, in an efficient and comprehensive way. Moreover, the level of mismatch between the design optics and MIA-based measurements can serve as quantitative criteria and guidance for a machine commissioning. This feature was especially appealing for developing and evaluating this application in the context of the new consolidating environment using both the Online Design Model and Virtual Accelerator, a simulation engine for generating turn-by-turn data.

The application was tested with the NSLS-II booster lattice. According to the design, the turn-by-turn data will be measured in 24 BPMs. In the current application, this data is generated by the Virtual Accelerator. The original version was implemented as a standalone Java application. Its integration with Control System Studio required a few steps and mainly dealt with packaging the Java jar files into the corresponding Eclipse plug-ins. Figure 2 shows the MIA display in CSS. It compares the design and measured Twiss functions in the ideal case without measurement and magnet errors. As expected, they agree perfectly. The application demonstrates a consolidated high-level application environment built from two integrated platforms, Control System Studio and service-oriented middle layer, connected by the new PvData-based communication protocol.

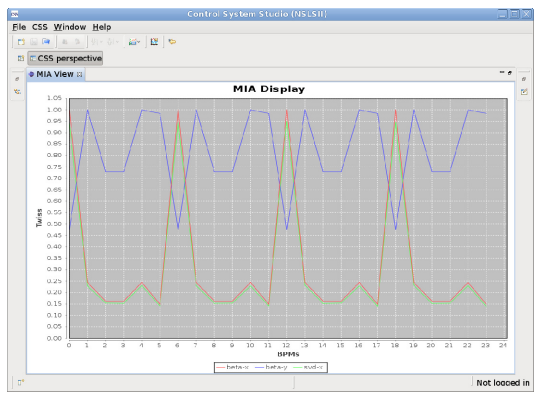

Figure 2: MIA display in the Control System Studio

REFERENCES

[1] L. Dalesio *et al.*, "Experimental Physics and Industrial Control System Architecture," ICALEPCS'93, Berlin, Germany, 1993

[2] M. R. Kraimer *et al.*, "Evolution of the EPICS Channel Access Protocol," ICALEPCS'09, Kobe, Japan, 2009

[3] J. Hatje *et al.* "Control System Studio (CSS)", ICALEPCS'07, Knoxville, USA, 2007

[4] N. Malitsky *et al.* "Application of Model Independent Analysis with EPICS-DDS", IPAC'10, Kyoto, Japan, 2010

[5] J. Irwin *et al.*, Phys. Rev. Lett. 82 (1999) 1684

[6] N. Malitsky *et al.*, "EPICS-DDS: Rationale, Status and Applications", IASTED'10, Novosibirsk, Russia, 2010

[7] K. Kasemir, "Control System Studio Applications," ICALEPCS'07, Knoxville, USA, 2007