

# CONTROL SYSTEM STUDIO

## INTEGRATED OPERATING, CONFIGURATION AND DEVELOPMENT

M. Clausen, J. Hatje, M. Moeller, H. Rickens, DESY, Hamburg, Germany

### *Abstract*

Control System Studio (CSS) is a platform for many kinds of different control system applications based on Eclipse. Many applications are now available in CSS for operating, configuration and development of control system related projects. Since CSS provides common interfaces, a plug-in structure and common data types it is possible that all applications are tightly connected and can for instance exchange data objects. The presentation will give an overview of the configuration and development tools and a short update of new features in the operating plug-ins. With the Database Creation Tool (DCT) EPICS databases can be structured hierarchically with prototypes, instances and parameters. It provides an extension to set the hardware addresses for in- and output channels by the IO Configurator that holds the device structure. Applications like SNL Editor and Debugger support programming control systems while other tools manage the device namespace or alarm system. In the future it is planned to add a control system (EPICS) IDE in CSS which shall help to configure and set up front-end-controllers for control systems.

### INTRODUCTION

Control System Studio (CSS) [1] is often identified as a next generation operator interface. The OPI aspect of CSS is only a small part of the whole story. The operational plug-ins in CSS are of course the most prominent ones and the first ones which have been developed for the control room but CSS is not limited to that. CSS basically provides the infrastructure for a whole suite of plug-ins which are related to control systems. Two important aspects of this infrastructure are the interfaces to the archived data (Archive API) and the process data (Data Access Layer) DAL. One full implementation of the EPICS channel access protocol and two basic implementations for TINE and TANGO are currently available.

In this paper we want to discuss the basic control system components of CSS: Starting from the operational tools we will continue with tools for configuration as well as an editor for the development of control system components.

### OPERATING

Three applications typically comprise the basic components for machine- and process control operations: Synoptic displays, alarm displays and trends.

### *Synoptic Display Studio (SDS)*

SDS is designed to cover all of the functionality which current display managers provide. Typical candidates are medm, dm2k or edm in the EPICS community. The data access layer (DAL) opens the regime for CSS plug-ins to all control systems supported by control system protocol plug-ins in the DAL. SDS is developed using the Eclipse Graphical Editing Framework (GEF). The WYSIWYG editor creates XML configuration files which are visualized by the runtime engine. Using GEF eases the creation of editors and widgets. The fact that only two interfaces must be implemented to integrate a new widget into SDS is the best proof of this elegant concept. Converters are available to turn existing configuration files of the display managers mentioned above into SDS XML files.

### *DataBrowser*

The data browser [2] is the best example of the functional CSS collaboration - since it has been developed at SNS in Oakridge. The coordination was carried out by email. Like the whole CSS development it is using the central cvs repository which grants access to all the CSS developers worldwide.

Like the other CSS applications the data browser is implemented as an Eclipse plug-in. Archived data are retrieved by the browser through a common archive API (AAPI). It comes with a rich functionality to display and correlate data either in strip-chart mode or from a set of archives defined in the CSS preference pages which can be changes interactively.

### *Alarm Displays*

The plug-in nature of the CSS applications provides great flexibility to the end user. While DESY has implemented an alarm system based on JMS (Java Message Service) [3] SNS has developed their own alarm approach independently from this. Both implementations can be used in conjunction with the other plug-ins without restrictions.

Alarm displays like alarm trees and alarm tables are available for CSS. Any type of messages (alarm, information, log, sys-log, SNL-log) are converted into JMS based messages and can be displayed by the same kind of graphs. A central database service is writing all of the messages into a relational database.

All of the alarm and trending services are implemented in a redundant manner running on a redundant SUN cluster. Since the 'headless' CSS application services are in any case Java based Eclipse products they can be run in any redundant configuration.

Alarm displays are very often the first places which demand further diagnostic information. Questions like “what does the trend of the time before this alarm occurred look like?” or “which kind of alarms occurred in the past for this channel?” should be answered immediately. The concept of ‘information on your fingertip’ is implemented by the Eclipse contribution mechanism in CSS. The user can call the list of available plug-ins in the contribution menu with a right mouse button click on the channel name and activate the application e.g. to list the alarms for this channel within the last 24 hours.

## CONFIGURATION

It’s been a long history in the EPICS community to configure your control system rather than program it individually. This way it is possible to hide the complexity of drivers and network protocols from the end user. The process- or machine- engineer can concentrate on the control aspect. Besides the already mentioned graphical editor in SDS several configuration tools have been integrated into CSS. Four candidates will be described here.

### *Database Creation Tool (DCT)*

The core configuration of an EPICS IOC is defined in its text based configuration file. There are various ways to create this text file. From the UNIX vi editor to the graphical – or visual VDCT (Visual Database Creation Tool). All of them create a database in the flat name space of an EPICS IOC. The CSS-DCT is starting where the control components are formed into a control device which might be part of an even bigger device based design. This way it is possible to create so called prototypes (one might call them control objects) which can consist of other control objects or just control records. These prototypes are finally instantiated into a flat EPICS database. Naming rules can be applied on the level of prototypes and instances. This eases the creation and instantiation of many control object of a similar type. The configuration of DCT is persisted into an XML file and stored in a cvs repository. This way it is possible to work on a project with several developers. Adding an instance of yet another control device is as simple as creating a new object in an object oriented language. The same applies to changes in existing control objects which are inherited in all instances of that object.

CSS-DCT also comes with a very basic graphical representation of the record structure of prototypes. This is meant to be used for documentation purpose only. Needless to mention that the access to this kind of documentation will be available yet by another contribution in CSS.

### *Device Database*

Configuring memory mapped I/O devices into a process control database can be a night mare. The Profibus field bus is – sort of – memory mapped. Changing the order of I/O cards or adding a new one in a

Profibus node will change the whole I/O mapping – at least for this node. But which process channels (or EPICS records) are currently using the I/O channels of this node? Mis-configuration is the consequence.

The device database with its device configurator is the answer to this problem. The device configurator – is of course a CSS plug-in – and defines the complete layout of one or more Profibus networks. The necessary configuration for the driver in the EPICS IOC is derived from the device database during the record instantiation process in CSS-DCT. The mapping is organized by the so called IO\_NAME. This unique name is used in CSS-DCT instead of the real input or output configuration string. In the device database on the other hand this IO\_NAME is attached to the hardware channel. This mechanism ideally decouples the EPICS process database from the I/O configuration. More different kinds of devices will be implemented in the device database in the future.

### *Digital Logic – Editor and Simulation (Diles)*

Digital logic and digital processing is the typical regime of PLCs. An EPICS Input Output Controller (IOC) is designed for process control. This kind of control often also requires some kind of digital logic processing. Therefore a digital logic record is in operation at DESY for at least a decade. The logic programs are ‘programmed’ by a very basic set of logical components realized in macros in ‘C’. This way it is possible to decouple the logic from the record. The logic can be selected by number or name out of a logical library. Input and output channels can be configured in the standard EPICS manner for each record instance.

The disadvantage of this approach is the compiled digital library which does not follow the EPICS ‘pattern’ of loading configuration data rather than programs or code to the IOC. The lack of a graphical configuration tool added enough momentum to the requirement to design a new plug-in in CSS comprising this functionality. Besides the ability to store the logical and graphical representation of a digital logic - also a simulator has been added to the CSS plug-in. This will speed up the development process of new digital logic algorithms.

While the configuration part is on the way the counter part on the IOC is still under development. This will parse the XML file from Diles and dynamically create the logic in a new digital logic record.

## ALARM MANAGEMENT SYSTEM

Operators responsible for certain subsystems are interested to see only those alarms on their alarm table which originate from those systems. Operators on call want to receive SMS messages on their GSM phone to be informed when major components need their attention.

In both cases it is necessary to define filters and actions to separate these specific alarm messages from the main stream of messages.

With the Alarm Management System (AMS) users can define complex filters to get only the messages they are interested in. To each filter one or more actions can be assigned that specify whether the message should be forwarded by Mail, SMS and voicemail or to another JMS topic. The AMS configuration plug-in is designed to configure a Filter Manager which is of course another CSS headless application.

## DEVELOPMENT

Besides all of the configuration activities for an EPICS IOC there are also programming developments going on. Typical candidates are the sequence programs in other control systems or state programs on EPICS IOCs.

### *State Notation Language (SNL) Editor*

There is currently very little support in EPICS to create SNL programs. Typically these are created by standard text editors. Syntax checks are only provided by the following pre-compiler and compiler runs on the code. Unfortunately the error messages are sometime cryptic and therefore the development process can take longer then expected.

The primary intention of the CSS-SNL editor is to speed up the development process. Another important aspect of the development is to improve the usability in that way that also experienced operators can work with the editor.

The current implementation is far from being comparable with current IDEs for Java or 'C'. But it's getting close to a language sensitive editor (LSE) which helps the programmer to avoid typing errors by issuing basic syntax checks. Colour coding of keywords and an outline view indicating variables, event flags and state sets are helping the programmer in addition.

## INTEGRATION

System integration by means of the DAL, archive integration by the AAPI and application integration in CSS by contributions have already been described. In addition there are several other levels of integration in the CSS context.

### *Messages all over*

Many different kinds of messages are sent and received through JMS which is supported in the CSS core as well as in many CSS applications:

- Alarm messages from record processing on the IOC.
- Alarm messages from scripts on other systems.
- System log messages from IOCs, workstations and CSS instances (RCP based or 'headless').
- Put log messages from IOCs.  
(changing a value on the IOC)
- Messages from SNL programs running on an IOC.

All of these are supported by the applications in CSS, the message archiver and last not least the alarm management system (AMS) which in itself is another CSS headless application.

Operational Tools

### *Code Management and Versioning*

Whenever configurations files or source code are created or modified it is important to save the current version and to coordinate the work with other developers. The rich set of plug-ins from the Eclipse community offers implementations to link projects in the user's workspace into cvs or subversion. Since CSS in its core is an Eclipse product it is possible to load these plug-ins into CSS. CSS by default is loaded with cvs support.

### *Elogbook and Bug Tracker*

Two plug-ins are building the bridge from CSS into electronic logbook and the mail system on one side and into the CodeBeamer [4] for bug tracking on the other side.

## OUTLOOK

The current version of CSS (CSS-1.2.0) comes with a rich set of plug-ins which supports the day to day operations in the control room as well as the control system designer and -engineer. More specialized plug-ins for SNL debugging and for IOC management are under development and will be available soon.

It is a continuing effort to add more plug-ins to the CSS core set of applications which will offer more 'information on your fingertip'.

### *EPICS IDE*

Managing a rich set of different kinds of IOC can be a demanding task. Different hardware, driver, operating systems and -versions combined with different EPICS versions require a configuration tool on this higher level. An EPICS IDE could fulfil these tasks.

Future will tell whether new projects with new ideas and resources will bring these developments forward.

### *On the Way to the Web*

The Eclipse RAP (Rich Ajax Platform) project will bring Eclipse RCP applications to the Web. This will have a significant impact on future CSS developments. It is discussed here:[5]

## REFERENCES

- [1] J.Hatje, M.Clausen et.al., "Control System Studio [CSS]", ICALEPCS 2007, Knoxville, 2007, MOPB03 ; <http://www.JACoW.org>.
- [2] K. Kasemir, "Control System Studio (CSS) Data Browser", ICALEPCS 2007, Knoxville, 2007, TUP009; <http://www.JACoW.org>.
- [3] M.Clausen, J. Hatje et.al., "Managing Alarms and (Log)Messages - the CSS Way", ICALEPCS 2009, Kobe, 2009, TUP017; <http://www.JACoW.org>.
- [4] CodeBeamer, <https://codebeamer.com/cb/user>.
- [5] J. Rathlev, J. Hatje et.al., "Eclipse RCP on the Way to the Web", ICALEPCS 2009, Kobe, 2009, THP109; <http://www.JACoW.org>.