CONTROL SYSTEM STUDIO
INTEGRATED OPERATING,
CONFIGURATION AND DEVELOPMENT

THC002

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

- Control System Studio Overview
- Operational Tools
  - Synoptic Display Studio (SDS)
  - Data Browser
  - Alarm Displays
- Configuration Tools
  - Database Creation Tool
  - Device Database
  - Digital Logic Editor and Simulator
  - Configuration of the Alarm Management System
- Development Editors
  - State Notation Language Editor
- Outlook
CSS is an Eclipse runtime environment with an enhanced set of core functionalities specific to control system environments
- Locale setting (e.g. to Japanese) are possible for all strings in CSS

CSS releases consist of CSS core and a set of control application plug-ins. They can be copied from the DESY ftp server.

CSS sources are free available from the DESY cvs repository under the Eclipse Public License (EPL) policy (ask us for a DESY cvs account).

Several sites create their own set of CSS products according to their desire.

CSS 1.2.0 is available since two weeks
- Based on Eclipse 3.5
- Java 1.6 (in a 1.5 compatible manner – to avoid conflicts with MAC users)
- Using the Eclipse Communication Framework (ECF) for remote management
- Bug fixes in CAJ
  - Thread safety, synchronization …
- SDS and ADL-Converter
  - Converting stripTool config files into dataBrowser config files
  - Calling dataBrowser from a related display button
The three most prominent applications:

- Synoptic Display Studio (SDS)
- Data Browser
- Alarm Displays

Data Interfaces:

- Data Access Layer (DAL)
- Archive API (AAPI)
- Java Message Service (JMS)
Operational Tools – Synoptic Display Studio (SDS)

- Based on GEF
  - The Eclipse
  - Graphical Editing Framework

- Edit Mode
  - any property can be dynamic

- Runtime Mode
  - Contribution: -> to DataBrowser
Alarm Displays

- Alarm- and Log-Tables are registered with **JMS** topics
  - ALARM (general alarm topic)
  - More topics can be configured in the Alarm Management System and are filled from the alarm filter system
  - Log topics: SYS_LOG, SNL_LOG, PUT_LOG

- Configuration of Table settings in Preferences
  - Color Coding (not only for EPICS alarms)
  - Column Labels – and their order

- Alarm Trees are defined in LDAP
  (LDAP is also used as the EPICS name server at DESY)
Control System Studio
Integrated Operating, Configuration and Development

Alarm Displays – Alarm(Log) Table(s)

- Alarm – Tree
- Log Table
- Alarm Table
- Archive Table

Contribution to Archive Table (default 24h)
Configuration Tools

- Database Creation Tool
- I/O Configurator -> Device Database
- Digital Logic Editor and Simulator (Diles)
- Configuration of the Alarm Management System
 EPICS specific database creation tool
 Starting from a hierarchical approach of so called prototypes.
   - Prototypes can consist of records and other prototypes
 Names are created in the prototype hierarchy according to naming macro substitutions (rules)
 Instances are created by resolving the final level of macro substitution
 Persistence in XML file
 Output is an EPICS db file

 Plan:
 Record names and IO_NAMES are stored in a RDB
 Graphical display of the prototype hierarchy - for documentation purpose only (for now)
Configuration Tools - Database Creation Tool

Control System Studio
Integrated Operating, Configuration and Development

Configuration Tools - Database Creation Tool

>ioname($(box)CV$(nr12)$(lfdrn))
Configuring the structure of I/O devices

- First implementation is available for Profibus I/O
  - Necessary to configure Profibus I/O on NON-Windows Systems
    - Standard Tools only run on Windows and configure Profibus Systems running in PLCs or on Windows
  - It is using the Profibus configuration files provided by the hardware vendor GSD (Geräte Stamm Datei) to configure the actual installed hardware

- Second implementation planned for Siemens S7

- Writing configuration into XML (not EPICS specific)
  - Parsed by Profibus driver on the EPICS IOC to configure the DPM memory in the Profibus controller card

- Central store for documentation (‘information on your fingertip’)
  EPICS channel -> IO_NAME -> I/O device -> Documentation
Store the internal logic program of the intelligent Profinet controller in the device database.
Configuration Tools – Device Database

EPICS address string:

@PROFI1:7/0 'T=UNSIGN8,B=4'

Hardware Channel
I/O_NAME
EPICS address string

CONFIGURATION TOOLS – DEVICE DATABASE          INTEGRATED OPERATING, CONFIGURATION AND DEVELOPMENT

CONTROL SYSTEM STUDIO

European XFEL
IO_NAME the Link between I/O Data and DCT

CSS I/O Configurator

CSS DCT

Device Database

CSS IOC

Create EPICS db file

INP: @PROFI1:7/0'T=UNSIGN8,B=4'

EPICS DB file

RDB

cvs repository

XML Config File

INP: @PROFI1:7/0'T=UNSIGN8,B=4'

Create EPICS db file

EPICS DB file

EPICS IOC

18
Work in Progress

Initial implementation by a student from the Jozef Stefan Institute
Configuration Tools – AMS Configuration

- Configuration of the Alarm Management System
  - User
  - User-Groups
  - Filter conditions
  - Filter (set of conditions)
  - Actions
    - Short Message Service (SMS)
    - Into another JMS topic
    - Mail
    - Voice mail

- Activation of this plug-in is controlled by the role based authentication/authorization scheme which is part of the core CSS functionality

TUP015: A Framework for Authentication and Authorization in Plug-in-Based Control System Software
- Special Features
  - Language Sensitive Editor (LSE)
  - Syntax check
    (no code completion – yet)
  - Colour coded keywords
  - Start compiler on save operation
    - Return compiler warnings into problem view
  - Outline View showing variables, event flags, state sets, states
  - Jump from Outline View back into editor
  - SNL Diagram Editor illustrates states and conditions
Development Editors – SNL Editor

```
#define OK 1
#define NOT_OK 0
#define SUPERERROR 0
#define CLOSING_LOOP 1
#define AUI 1
#define DII 0
#define PID_MANHOLD 1
#define PID_AUTO 0
#define STOP 0
#define START 1
#define CLOSE 0
#define OPEN 1
#define BETRIEB 3

#define EPS(x, y) \{|setexpr x, y|}
#define PV_SET(pv, val) pv = val; pvPut(pv);
#define PV_ASSIGNED(pv, type, name) PV_ASSIGNED(pv, type, name)

#define MDK1 (kompBits & K1)
#define MDK2 (kompBits & K2)
#define MDK3 (kompBits & K3)
```

```
#include <stdio.h>
#include "snl/SNL.h"
#include "snl/SNL.More.h"
#include "snl/Str strSqlMsg.h"

char text[101];

#define TEXT "\"text\"
#define VALUE "\"value\"
#define OTHER "\"other\"
#define WARNING "\"warning\"
#define ERROR "\"error\"
#define DEBUG "\"debug\"
#define INFO "\"info\"
#define TRACE "\"trace\"
#define FATAL "\"fatal\"

#define LIMIT 1.0

#define TYPE_RESOURCE 0
#define TYPE_PATH 1
#define TYPE_LOCAL 2
#define TYPE_TYPE 3
#define TYPE_DESCRIPTION 4
#define TYPE_WARNING 5
#define TYPE_ERROR 6
#define TYPE_DEBUG 7
#define TYPE_INFO 8
#define TYPE_TRACE 9
#define TYPE_FATAL 10
```
SNL Editor – Configuration / Preferences

- Preferences
  - Compiler Options (Linux)
  - EPICS Base
  - Colour coding
Outlook

- Preparation for CSS 1.2.1
  - Collecting requirements based on the experience during the current commissioning phase with the configuration tools and the synoptic displays
  - Change requests from other CSS/SDS users
  - Merge in the SDS enhancements shown during the EPICS meeting
  - Adding DAL plugs (basic-TANGO, CA-V4?)

- Collect requirements for an EPICS Integrated Configuration Environment (EPICS-ICE)
  *An initial implementation by Kenneth Evans might be a good starting point*
  

- Closely following the ‘Eclipse Way to the Web’

THP109: Eclipse RCP on the Way to the Web
Summary

- CSS core provides an excellent platform to integrate new applications.

- New CSS configuration and editing plug-ins were successfully used to improve the development cycles for the ongoing commissioning of the (former HERA) now FLASH cryogenic plant.

- Decoupling the definition of the I/O address space from the EPICS database configuration by unique IO_NAMES reduces the potential risk of address mismatches.

- A new EPICS ICE would help to organize the configuration of bigger installations.

- CSS 1.2.0 is now available requirements for 1.2.1 are currently collected
Thank you for listening

CSS 1.2.0 is available from:
http://css.desy.de