Rapidly Installable High Performance Control System Facilitates BESSY II Commissioning*

R. Bakker, J. Bergl, T. Birke, B. Kuner, B. Kuske, R. Lange, B. Martin, I. Müller, R. Müller,
G. Pfeiffer, J. Rahn, H. Rüdiger, BESSY, Berlin, Germany

Abstract

The BESSY II control system takes advantage of the mature stage of the EPICS toolkit and its contributed generic applications. Development activities have been focussed on three aspects. (1) Dominant role is given to device control IO, based on distributed local intelligence of embedded controllers and CAN fieldbus networks. (2) Cooperative development of adaptable physics applications is supported by the granularity of the programming environment. Toolkits with well designed interfaces handle GUI, modelling etc. (3) ORACLE RDBMS (*Relational DataBase Management System*) and automatic generation procedures guarantee consistency of configurations for real time data bases, programs etc. The resulting control system combines convincing installation speed, performance and functionality with simplicity, reliability and transparency offering help on all levels of the BESSY II commissioning process.

1 OVERVIEW

Control systems of third generation light sources strongly reflect the high demands on specified beam quality as well as on non-interrupted service. As an early design decision at BESSY II [1], emphasis has been given to ultimate precision and stability of all beam guiding components. Residual perturbations (of e.g. orbit stability) are then expected to be small enough to be controllable by relatively simple feedback systems that might be subject of cost effective upgrade activities.

In a bottom-up approach signal conditioning for the power supplies has been specified first and relatively independent from the control system architecture. Back-bone of the power supply control are plug-in units that consist of high precision, moderate speed IO cards and piggy-back modules with i386EX CPU and CAN (*Controller Area Network*) controller [3]. A 'standard model' control system architecture connects to the remaining devices and concentrates data from the CAN segments. Core software for communication between HP workstation consoles and VME single board computers is based on the EPICS toolkit.

For the commissioning of the storage ring 2 dual headed C class workstations, 2 B class auxiliary consoles with a total of 4 knob boxes and a Windows-NT X-terminal serve as control room console equipment. It properly balances display area requirements, simultaneous (eventually concurrent) device interaction possibilities and access to off-line data analysis. Presently a central RAID system with file and database server functionality is installed to allow for adequate load distribution.

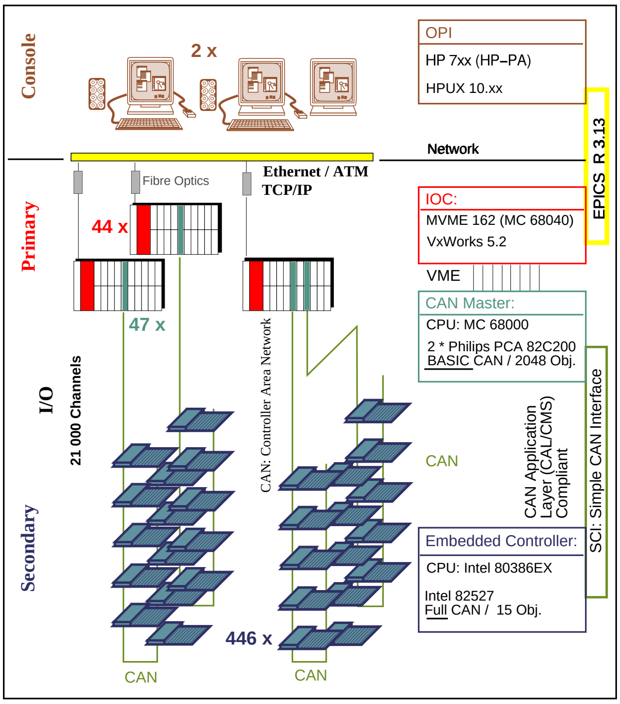

Figure 1: Control System Architecture

On field level a total of about 21 000 control points are handled by 44 moderately loaded VME crates (*Input Output Controllers*, IOCs), 47 CAN segments with 446 nodes (Fig. 1) and 3 GP-IB busses.

2 DEVICE ACCESS

Wherever possible, preference is given to the secondary CAN based I/O layer. The direct primary I/O on the VME level is only used for specific systems.

2.1 CAN Nodes

The standard power supply connection using an embedded controller on a plug-in card has a variety of advantages[3, 4]. Due to the generic software design and the simple cabling of the shielded twisted pair CAN cable, installation of this part of the control system was very rapid. Control system based equipment tests were enabled literally the same day the device has been set up and powered. For convenience

---

*Funded by the Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie and by the Land Berlin

several mobile consoles on trolleys with radio frequency based network connection are made available.

Performance and reliablity of the control system is commonly appreciated. Command execution via CAN and embedded controller is typically completed within 4 msec. Only about 30% of the total response time is due to the additional CAN field bus layer. Largest fraction of this overhead time is the embedded controller processing time, rather then the CAN transmission speed (200 μsec write cycles, 20 kB/s data transfer rate). Local system inspection via serial lines as well as remote health checks via LAN/IOC provide flexible means to track-down error-conditions or hardware failures. Set-points and status informations are persistent on the embedded controller as long as the controlled device is operational. By reading the data stored by the embedded controller at start up, the system returns to the previous operational state after an IOC reboot (‘stateful recovery’). VME crates are configured to maintain status OFF after a power fail. With abandonment of uninterruptable power supplies for the IOC systems this seems to be a sufficient hardware protection.

For specific devices without appropriate connector, that require more digital I/O (e.g. Kicker/Septa) and for the vacuum system, the embedded controller module is mounted on a carrier board that controls I/O cards in an ISA bus crate. Similarly status read out of the personnel safety system, a standalone programmable logic array ISA bus installation, is provided by an embedded controller.

Each of the 5 RF power amplifiers is controlled by a hybrid system. It consists of a digital I/O board for command transmission and the local Siemens S5 PLC control unit equipped with a commercial CAN I/O adapter for exchange of command and status information.

Beam intensity is monitored by a parametric current transformer. Data are aquired by a stand alone PC running LabViews under Windows NT. A PC CAN card is used by the LabViews program to provide primary current data as well as derived data like life time.

2.2 VME based I/O

For specific requirements like a video multiplexer board, a programmable delay unit and a counter system, solutions are available that meet the specifications and are integratable into EPICS on the VME level with moderate effort.

The aquisition system for the beam position data has very specific demands on data volume and processing speed. The resulting set up for the injection system consists of a VME crate with a fast digitizer board. For the storage ring 16 VME crates control operation modes and gather closed orbit data while a 17th crate serves as master controller. This 17th crate is also packed with fast digitizer boards collecting the first turn data[5]. Hardware set up and data preprocessing is completely handled by the diagnostic group.

GP-IB connection between EPICS and various measurement devices is provided by a specific device support layer establishing network communication to commercial Ethernet–GP-IB converter boxes.

3 APPLICATIONS

3.1 Embedded Controller

Beside the standard I/O control software on this level, only the download protocol has been implemented as a prerequisite for the envisaged ID compensation scheme[2].

3.2 IOC (VME)

IOC applications, providing device protection, conditioning procedures, PID control algorithms as well as data selection and presentation mechanisms, turned out to be very helpful. Typically it has been feasible to react promptly on new requirements reflecting operating experiences. Device collections with changeable configurations form additive pairs, synchronized power supply families or closed orbit bumps. These pseudo devices are useful accelerator control handles that are better adapted to physics parameter space.

3.3 Console (EPICS)

Most standard requirements of accelerator operations may be covered by configuring the generic applications available as EPICS extensions. For commissioning the display manager *medm* is a tool of outstanding importance. Device control panels are constantly developed from the early device access development phase on and ready to be used as soon as the piece of equipment is installed. In combination with a few comprehensive overview screens a synoptic evolves that is ready to be used during the installation phase and rapidly adaptable to changing requirements in functionality, presentation and navigation capabilities.

Accelerator experiments and data analysis is largely based on the SDDS toolkit[6]. The typical approach to use scripts for tests and give tcl/Tk user interfaces to the more established procedures, has proven to be very efficient.

Other essential commissioning tools have been developed from available, but insufficiently generic tcl/Tk scripts (e.g. save/restore/compare) or they are custom wrappers around specific EPICS tools (e.g. cycle magnets).

3.4 Console (Non-EPICS)

Accelerator physics applications have been planned to be composed from object oriented toolkits that cover certain functionalities and are independently developed and maintained [7].

The resulting beam shaping programs available for the commissioning are ‘relatively’ generic. Each program is capable to handle the different accelerator sections with the same functional blocks wherever functionality matches.

The orbit control program for example determines the accelerator target system (e.g. the storage ring) from a command line option (Fig. 2). It requests the appropriate graphical user interface from the graphical server[7] and detects the application variables associated with user control and monitor handles. A model object is created using the *goemon*[8] toolkit. The model is used to create the associated device access object: ORACLE queries collect the informations needed to find the power supplies connected to the magnets of the model and to set up the synchronization between control system set points and model parameters.

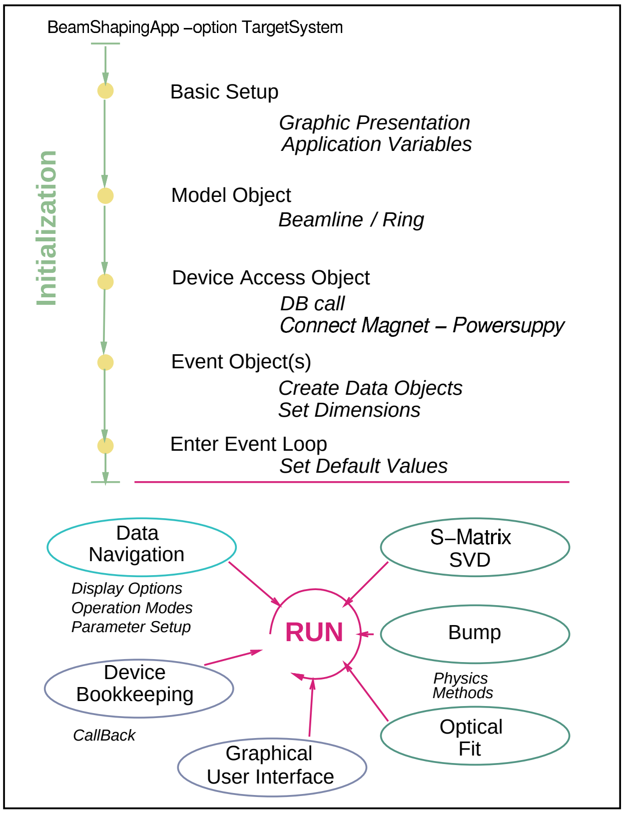

Figure 2: Sketch of ‘Generic’ Beam Shaping Programs

Event objects that monitor control system data relevant for the program functionality are established. The orbit control program e.g. requires status and set point information for the corrector power supplies. A BPM monitor object handles a dynamic history buffer of orbit data and provides statistical informations and data navigation methods.

Within the running program the physics methods dynamically adapt to the specific system (Fig. 2). For device bookkeeping and screen updates generally callbacks from the event objects are applied.

As residual non-generic program elements the data navigation features, like display options, meaningful operation modes etc., remain and reflect the specific system differences.

4 RELATIONAL DATABASE

Generic high level software tools, a flexible and layered distributed I/O system and online modelling programs require a huge amount of configuration data having a large variety of relationships. An ORACLE RDBMS has been set up very early as a single and unique repository of reference data. Configuration data have been consequently stored in the DB and the data needed in various contexts and formats have been generated by extraction procedures[9].

4.1 RTDB Generation

Templates of the EPICS real-time DB (RTDB) are designed with a graphical editor. RTDB instantiation is generated by scripts from the ORACLE DB. Equipment description data have been loaded from genuine sources and are maintained by the equipment specialists. Changes are automatically propagated into the download area and may be monitored on a WEB page.

4.2 Console Level Configuration

Description files turning generic tools into meaningful applications known as *display manager*, *alarm handler* etc. are automatically generated wherever appropriate. Physics applications retrieve the mapping between control system and model data with DB calls or from an equivalently generated file caching the corresponding informations. Despite all inherent complexity the result is a consistent, transparent and maintainable system of operation programs.

5 SUMMARY

Achievements of the EPICS collaboration have been utilized in combination with strongly focussed developments adding value to the core system. The resulting system provides adequate commissioning tools that are strong in reliability, timely availability, consistency and performance.

6 REFERENCES

[1] D. Krämer et. al., This Conference, KRE0301

[2] J. Bergl, B. Kuner, R. Lange, I. Müller, R. Müller, G. Pfeiffer, J. Rahn, H. Rüdiger, Proceedings of the 1995 ICALEPCS, Chicago, 1995, 1017

[3] J. Bergl, B. Kuner, R. Lange, I. Müller, R. Müller, G. Pfeiffer, J. Rahn, H. Rüdiger, *Embedded Controllers, Field Bus and a Modular IO Concept: Central elements of BESSY II Controls*, Proceedings of PAC 97, Vancouver, 1997

[4] J. Bergl, B. Kuner, R. Lange, I. Müller, G. Pfeiffer, J. Rahn, H. Rüdiger, *CAN: a Smart I/O System for Accelerator Controls – Chances and Perspectives* Proceedings of the 1997 ICALEPCS, Beijing, 1997

[5] R. Bakker et. al., This Conference, BAK0556

[6] M. Borland, L. Emery, N. Sereno, Proceedings of the 1995 ICALEPCS, Chicago, 1995, 382

[7] T. Birke, R. Lange, R. Müller, Proceedings of the 1995 ICALEPCS, Chicago, 1995, 648

[8] H. Nishimura, Nucl. Instr. Meth., A 352 (1994), 379

[9] R. Bakker, T. Birke, B. Kuske, B. Martin, R. Müller, *Significance of a Comprehensive Relational Database System for Modern Accelerator Controls*, Proceedings of the 1997 ICALEPCS, Beijing, 1997