

MAKING IT ALL WORK FOR OPERATORS

I. Kriznar, Cosylab, Ljubljana, Slovenia
 S. Marsching, Aquenos GmbH, Baden-Baden, Germany
 E. Hertle, E. Huttel, W. Mexner, N. J. Smale, A.-S. Mueller,
 KIT, Eggenstein-Leopoldshafen, Germany

Abstract

As the control system of the ANKA synchrotron radiation source at KIT (Karlsruhe Institute of Technology) is being slowly upgraded it can become, at key stages, temporarily a mosaic of old and new panels while the operator learns to move across to the new system. With the development of general purpose tools, and careful planning of both the final and transition GUIs, we have been able to actually simplify the working environment for machine operators. In this paper we will explain concepts, guides and tools in which GUIs for operators are developed and deployed at ANKA.

INTRODUCTION

The machine control system of the synchrotron radiation source ANKA at KIT (Karlsruhe Institute of Technology) is migrating from the ACS CORBA based control system to the Ethernet TCP/IP devices with an EPICS server layer and visualisation by Control System Studio (CSS). This migration is driven by the need to replace ageing hardware. Approximately 500 physical devices, are being gradually replaced (or have their I/O hardware changed) and are integrated to the EPICS/CSS control system. [1]

We can already say that the ANKA control system is based on EPICS. All core services and tool-kits are done with EPICS/CSS. However, there are still additionally other control systems, which need to be used in parallel or integrated into EPICS for various reasons. Namely PVSS and ACS (Figure 1).

GENERAL CONVENTIONS

CSS is based on Eclipse RPC and on Java, for which there exist several good references for building functional and pleasant GUI applications [2, 3]. In addition to these extensive documents ANKA has several simple guidelines that address most frequent mistakes, which are encountered:

- Use default look and feel, default fonts and sizes as provided by CSS/Eclipse.
- Design elements must be consistent across all applications and components. Buttons, check-boxes, radio-boxes, combo-boxes have well known and expected functionality; Don't misuse them or change behaviour. Don't change their labels, functionality and position in runtime ether.
- Some colours are reserved, borders are reserved for alarm notifications. It is important to use them only for designated purposes, don't confuse users with design choices that are similar to reserved use.
- Leave just the right space between components, organize them so they are equally spaced and lined to same base-lines. There are tools in CSS visual composition toolbar that help you do just that. This point should be trivial and obvious, but for some reason it is often not followed.

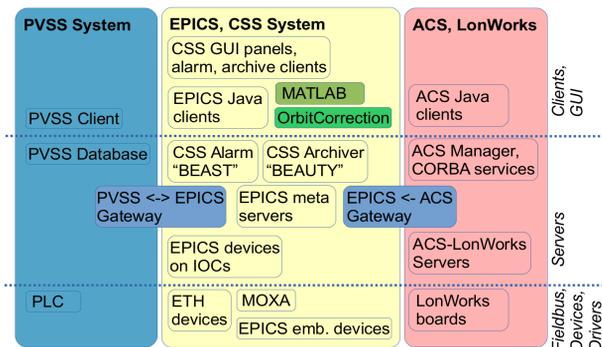


Figure 1: Patchwork or different components and technologies in ANKA control system.

The storage ring is generally operated at an energy of 2.5GeV with a typical beam current of 200 mA and a lifetime of 20 hours. Due to the finite lifetime the storage ring is emptied of electrons twice per day (8:00 and 18:00) and refilled. The refilling process involves a two stage accumulation process. The machine is then left unattended until the next injection time, which leads to two important demands on the control system, namely, robust and sensitive alarm notification in the case of reduced machine performance and a very intuitive GUI machine interface; as a single operator will only attend to the machine for approximately 10 hours per 2 months.

THE ANKA CLIENTS BUNDLE

ANKA has several distribution channels for different generations of control system clients. Distribution channels are kept in Subversion repository. Installation and updating are performed through Subversion operations on target computers. Subversion was chosen for its convenience over competing solutions. Subversion can effectively keep history of changes and it is used primary for one way distribution, so something like Git would not be fully utilized and would increase complexity for simple update operations.

Subversion distribution channels of control system clients covers: old ACS client distribution, the PVSS distribution and the main ANKA client distribution. The main ANKA client distribution, called ANKA-Clients, is kept in two branches: the main trunk is used for storing stable releases running on all operator's computers. The

“work” branch is intended for development work. Since there are not so many developers, and they work on different devices, the chances for the conflicts are very low. During a shut-down period the development branch is copied/merged to the trunk, tested, and installed on all control system computers through subversion update feature.

THE LAUNCHER

ANKA Launcher is started from within ANKA-Clients distribution bundle and is the preferred way on how to start control system applications or perform certain automated tasks. The ANKA Launcher icon can be found on all operators machines (Figure 2).

Buttons in the launcher can execute scripts on local disk or on a remote server over SSH protocol. If the local computer has XServer installed (like Exceed in Windows), then it can open an application window from a remote Linux computer.

Launcher buttons and their tasks are defined in a launcher configuration file and act as groups of buttons. The following groups are defined:

- Common Clients group: start the ACS Java panels.
- CSS Clients group: opens the CSS Main, Alarm and Archive (Data Browser) applications.
- EPICS Clients group: opens some third party provided EDM EPICS panels from a remote Linux computer, needs local X11 server to work. These are: Libera Launcher panels, Matlab ML panel and BBB Expert panel.
- VNC Servers: button opens a VNC to a server.
- VNC Oscilloscopes: opens VNC to oscilloscopes.
- ACS Server group: starts/stops the central ACS system over SSH connection.
- EPICS IOCs and Servers group: starts/stops EPICS IOCs and servers over SSH connection.

THE CSS APPLICATIONS

There are two distinct type of users of the machine control system and therefore two distinct approaches to the control system GUI applications: machine operators, they need a stable and a predictable environment which helps to go through an optimized procedure which should give a reproducible end result: a stable orbit and beam. And the experts and machine developers: they work with vague procedures and they want to have all options open and tools available, because it is not known in advance what will be needed. Tools or procedures could be developed during work and then perhaps not used again.

Even at the design stage an application developer must decide if an application is intended for operators or experts. In most cases it is even desirable to develop

different panels or sets of panels of the same application or device specifically for these two user groups.

For operation purposes the following is considered the best approach:

- ANKA CSS main application window should be used if possible in full screen mode.
- Restart of CSS instance would always bring back same main screen layout and same panels.
- Sub-panels should be opened within the main screen and not in a separate window. As further specified in the document section below.

In case of expert panels we have additional requirements: expert panels must be equipped with warnings so that a casual operator would not accidentally run them. And dangerous operations, which are not used in day-to-day operation, must be protected with additional confirmation dialogues.



Figure 2: The ANKA Launcher.

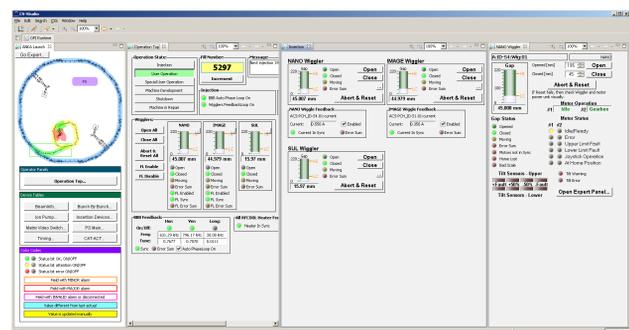


Figure 3: This is main CSS panel at ANKA and demonstrate on a case of wigglers panels the four levels of the panel hierarchy from the left to the right: i) the CSS launcher, ii) the main process/operation panel, iii) the overview device group panel and iv) the device details panel.

There are general guidelines on how to organize various panels within the main ANKA CSS application. They are demonstrated in Figure 3 and are described from left to right.

- **The CSS launcher** is considered the most important panel. It contains action buttons, which open process and overview panels in tabs to the right of the CSS launcher. The CSS launcher should be always visible. It has two modes: operator and expert.
- **Process oriented panels** contain controls for automated and semi-automated processes. Different process stages and procedure can be organized in different sub-tabs. The most distinguished process oriented panel is “top operation” panel, which gathers together all information and operations directly connected to the daily operation. By convention device widgets included (by linking container) in this panel are found in a file with the name “Top.opi”.
- **Overview oriented panels** should, by convention, open right to the process oriented panels. Here are panels which provide a group view for devices and panels for individual devices. An individual device widget is by convention found in a file called

“Overview.opi”, when included in the device group panel. An individual detailed device panel is usually opened from a device group overview panel and the a tab should appear right to the group overview panel. It allows access to the individual operations of the device and provides the most details about the device. Usually operators don't go down to this level, these panels are usually used by particular device experts. But still they should be operator-friendly, so OPI file should be named “Operator.opi” if possible.

- **Expert panels** offer more functionality than what an operator needs in daily operation and they can be safely used only by device experts. The GUI is not expected to be polished up to the same level as an operator's panels. The detailed device expert panel is opened from device panel (with button “Open Expert Panel...”). It contains additional options not presented in Operator.opi, mainly to get them away from operators. Intended for expert use only by device experts. The OPI file for device expert panel is usually named “Expert.opi”.

Generally operators are able and allowed to reorganize tabs within panels as they find more usable. But because this is potentially dangerous and could lead to unwanted results, these changes are reversible. Each time CSS panel is restarted at ANKA its workspace is cleared and replaced with a template so they always appear looking the same way.

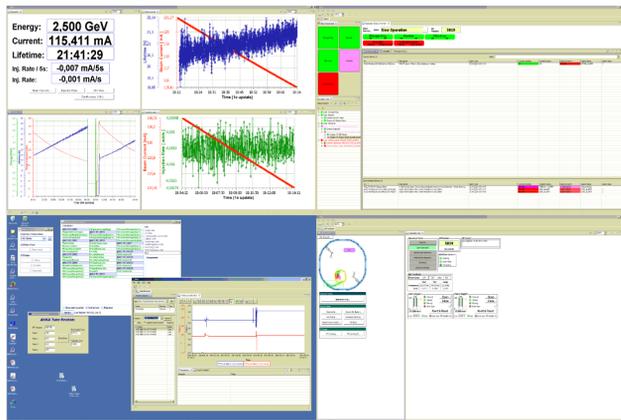


Figure 4: Layout of panels on operator's computer with four screens. From upper left corner clockwise: BeamInfo, Alarm, CSS Main and Archiving with some other control applications.

In the ANKA-Clients distribution bundle there are only three main CSS application short-cuts and several special purpose ones (Figure 4):

- ANKA CSS Main application: here all controls system CSS panels are included and launched within the main panels. Instead of separate windows they are intended to be opened as tabs and rearranged as necessary. For this reason CSS Main application window is used in full-screen mode and is usually the only window on the computer monitor.
- Alarm application: opens BEAST alarm perspective.
- Archiving application: opens the DataBrowser perspective to the Cassandra database.

- BeamInfo application: CSS panel which shows only beam parameter information and trends, interesting during injection and ramping.
- Special panels to be run on wall monitors.

Device Type Convention

Devices, which are of the same type, are represented in the top operator's panels also with the same CSS widgets (by linking container included OPI file) which provide standardized access to the common device values and operations; Even if the actual physical devices might be from different manufacturer and has a quite different set of features. For example, ANKA has several different kinds of power supplies and wigglers, yet they are all operated in the same way from the top operations panel. But of course they might have very different device detail panels, which reflect their actual capabilities.

This is possible by standardisation of a set of common PV names and types. The ANKA EPICS naming convention defines device parts of the PV and leaves the property part free. The ANKA Device Type Convention further defines which are common mandatory PVs and in what way a device must be supported in order to be included into standardized panels and automation tools. Usually these are PVs for the most common operations in a simplified format, from the point of view of operation, and device summary signals, in similar manner as earlier mentioned Status:ErrorSum.

CONCLUSION

The ANKA machine during normal user operation time is refilled twice a day and in the meantime is left unattended. Therefore, a single operator will only work with the machine for approximately 10 hours per 2 months. This puts strong constraints on the control system in that there needs to be a very intuitive GUI machine interface design. New panels and tools must be well designed and carefully introduced into operation in order not to lose trust in the control system. Operators expect to find panels on the same computer organized in the same way so they can easily go through the operation procedure after weeks of absence.

ACKNOWLEDGEMENT

We would like to thank all people who helped in the planning and upgrade of the ANKA control system, in particular Guenther Rehm, from the Diamond light source, for his continuous support. And we would like to mention Cosylab developers, who provide excellent development work and continuous support on EPICS panels and drivers according to our requirements and conventions.

REFERENCES

- [1] N.J. Smale et al., “The ANKA Control System: On a Path to the Future”, MOPPC099, ICALEPCS'13.
- [2] Design guidelines for Eclipse website: http://wiki.eclipse.org/User_Interface_Guidelines
- [3] Java development guidelines website: <http://www.oracle.com/technetwork/java/hig-136467.html>