

INTEGRATING HIGH LEVEL APPLICATIONS AND SOFTWARE COMPONENTS USING A SCADA TOOL

K. Saintin, A. Buteau, V. Hardion, M. Ounsy
SOLEIL Synchrotron, Gif-sur-Yvette, France.

Abstract

To build integrated high-level applications, SOLEIL is using an original component-oriented approach based on GlobalSCREEN, an industrial Java SCADA [1]. The simpler components are based on the ATK (Application Tango Toolkit [2]) library, a rich set of Java graphical widgets that include tools for visualizing scientific data, and encapsulating TANGO communication. More complex components are Java rich client applications that provide a high level of service and user-friendly interfaces for end users, providing access to data logging, scanning services, sequencing setup and control, and so on. Packaged as JavaBeans in GlobalSCREEN's components library, all these components are ready to use by simply dragging and dropping them into place and configuring their properties to connect them to the control system. In this way, using GlobalSCREEN as a rich client platform allows even inexperienced developers to quickly build, without writing any Java code, very complex and fully integrated applications dedicated for example to the end users of a beamline. The work organization, software architecture and overall system design will be presented, as well as the current status of deployment at SOLEIL for the accelerators and beamlines.

INTRODUCTION

Providing end users and scientists of synchrotron facilities with user-friendly high-level applications has led software teams to adopt a wide range of solutions. For a long time, these solutions consisted of developing a new standalone application for each new need, using different tools based on different technologies. These solutions, however, have proven to be lacking in interoperability, consistency and ease of maintenance. The desire for standardization led developers to look for a “magic” tool that would address users’ needs in a unified way. Different standalone applications were nevertheless developed for each need. Interoperability between applications was handled by the underlying distributed control system bus. But at the HMI level, applications were still inconsistent and strongly software developer oriented.

SOLEIL’s usage of the GLOBALSCREEN environment has proven to be effective in providing users with different applications for different needs, but with a coherent and maintainable overall system. The fundamental rule of thumb is not to use it as “**the solution**”, but to find the right tool for each service and to use GLOBALSCREEN as a configurable container for integration of software components.

INTEGRATION OF SERVICES

The critical step in designing this software architecture was to determine the services the user has to interact with through high level applications. These services were broken down as follows:

- Operating equipment and monitoring measured data by interacting with the TANGO control system.
- Logging control data for abnormal facility behavior diagnosis or late data processing.
- Sequencing complex operations processes, with features for controlling, pausing, resuming, stopping and logging the process.
- Storing physical or experimental data, with features for viewing and processing the data.

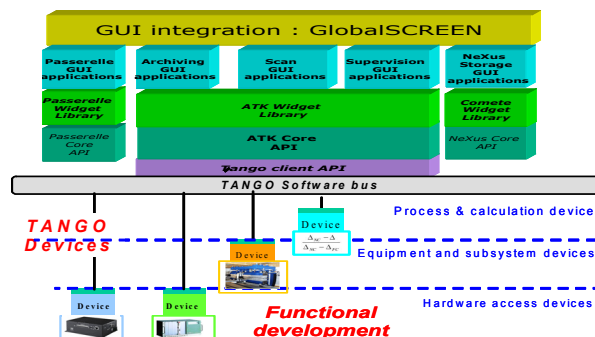


Figure 1: SOLEIL software architecture.

For each of these needs a dedicated solution was devised.

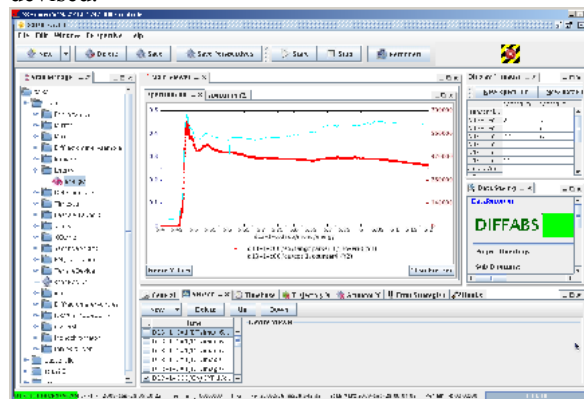


Figure 2: Salsa, the scan application.

Application Tango Toolkit

This is the first building block of the system. Unitary Java graphical widgets incorporating interaction with the control system are available for operating and monitoring equipment. In this way, operating an actuator, viewing a measurement trend or processing a CCD camera image is

just a matter of choosing the right widget, including it in an application and connecting it to the control system (see Fig. 2 for an example of a standalone application developed using elementary ATK widgets).

Archiving Service

To log control data, an external service was developed, incorporating a set of Tango devices and ATK components for configuration, extraction and visualization [3]. These components can be integrated in any other application. They were used to provide users with a standalone application (Mambo, see Fig. 3) if they need to access all archiving features.

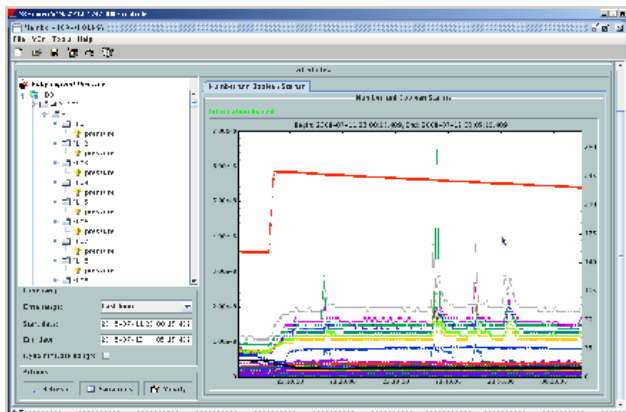


Figure 3: Logged data extraction with Mambo.

Sequencing Processes

The PASSERELLE [4] environment was chosen by SOLEIL as the service for graphically designing any process as a data workflow, using components known as “Actors” that handle interaction with the control system, flow control, calculation steps, and so on. This service comes with its own development environment. Dedicated Java graphical components are provided to configure, launch, pause, resume, stop or log sequences. These components can then be integrated in any other Java application.

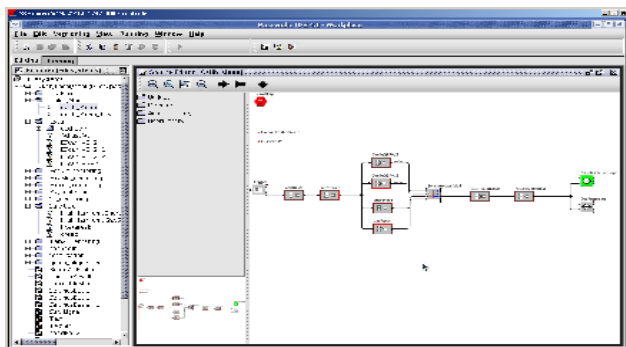


Figure 4: Development of a sequence in the Passerelle IDE.

Data Storage[5]

In the same way as for the archiving service, an external service for storing experimental data files was

developed using various Tango devices and ATK components for configuration, extraction and visualization. A set of graphical components are available for accessing these functions. Standalone applications are also available to operate them as a whole.

COHERENCE OF THE SYSTEM

We may wonder how it can be possible to use these different services and yet provide users with integrated applications in which they can access these services simultaneously and at the same time provide developers with environments where the services can functionally interoperate.

These goals are achieved at SOLEIL by asserting a number of rules governing how the services are developed:

- The functional part of each service is developed as a Java library and made available to client developers through an abstract interface.
- Elementary “ready to use” graphical components are made available to end user application developers.
- Components addressing a common concern must be gathered in a coherent development framework.
- All graphical components must comply with the Java Bean standard.

Hence:

- ATK comes with ATKCore for its functional part and ATKWidget for its graphical layer.
- The archiving service comes with a set of ArchivingApi libraries conforming to the Java JDBC standard.
- The sequencing service provides PasserelleApi to handle functional features and PasserelleWidgets for HMI development.
- NexusApi provides an interface to NeXus library [5] functions. Graphical components for browsing and processing data are built on top of the ImageJ open source library and gathered in the COMETE framework [6].
- All graphical components comply with the Java Bean standard.

This brings us to the principal added value of GLOBALSCREEN as a tool for integrating high level applications and software components. JavaBeans are reusable graphical unitary components and GLOBALSCREEN is a SCADA supporting Java Bean technology. It provides a user-friendly GUI in which components are simply dragged and dropped into place and their properties configured to connect them either to a control system, to the Passerelle sequencing engine, or to the data storage system. GLOBALSCREEN is used as the end-user development environment, because application developers do not need knowledge of Java but can use the environment directly to assemble standalone applications to fit their own ergonomic needs.

The other very important added value of the SCADA environment is that it enables our software team to fulfill various important requirements:

- Machine and beamline needs can be handled by a small team of developers.
- The team can focus on providing an assessed and maintainable set of frameworks made of shared libraries and reusable components.
- Users have sufficient autonomy to develop HMIs that fit their needs.
- The resulting HMIs are guaranteed to share a standard and homogeneous look and feel.

Last but not least, GLOBALSCREEN is compatible with the **Java Web Start** technology. Java Web Start is a framework developed by Sun Microsystems that allows users to start application software for the Java Platform directly from the Internet using a web browser. This enables SOLEIL users to access remotely the same rich client platform applications they use locally, in daily operation.

To achieve this, all the API Interface libraries needed for client access to each of the services are available with two different implementations, one local and one remote.

For the remote implementation:

- ATKCore was adapted to access the control system via an intermediate layer. This layer translates Tango requests to the TangORB API in the local case or to HTTP requests reaching a WebTangORB server for remote access. The WebTangORB server then translates these requests to TangORB API calls to the underlying control system.
- The PasserelleAPI Interface is implemented as HTTP requests addressing a remote server complying with the W3C Webservice standard for software systems designed to support interoperable machine-to-machine interaction over a network.
- GLOBALSCREEN itself comes with an internal web server, for which the requests are handled transparently for the user by the deployed environment.

The remote services are deployable in a J2EE application server (Jboss at SOLEIL).

APPLICATION EXAMPLES

Beamline Alignment

The following GLOBALSCREEN view shows a beamline synoptic. Each optical equipment (*primary slits, monochromator, etc.*) has a PasserelleWidget for its associated alignment sequence. Once the widget is selected, the sequence configuration panel shows up at the bottom left. The operation of the sequence is monitored with the ATK spectrum viewer at the bottom right (here showing beam intensity evolution measured on a sensor).



Figure 5: A GLOBALSCREEN view for alignment.

Beamline Data Acquisition

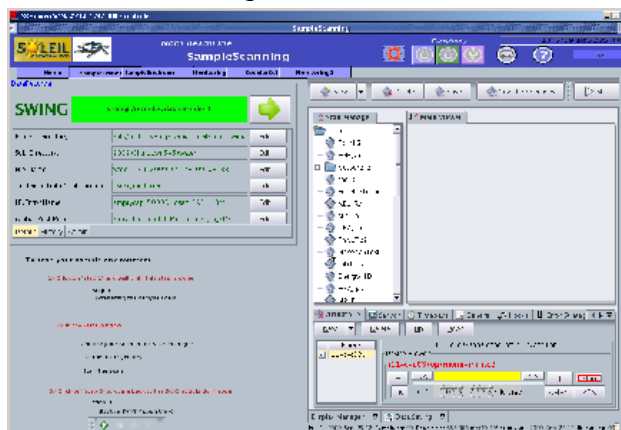


Figure 6: A GLOBALSCREEN view including the Data Storage configuration bean, a sequence launcher bean and a Salsa panel.

REFERENCES

- [1] V. Hardion, M. Ounsy, K. Saintin "How to Use a SCADA for High-Level Application Development on a Large-Scale Basis in a Scientific Environment", ICALEPS 2007.
- [2] F. Poncet, J.L. Pons, "Tango Application Toolkit", ICALEPS'05, Geneva, Oct 2005.
- [3] J. Guyot, M. Ounsy, S. Pierre-Joseph Zephir "Status of the TANGO Archiving System", ICALEPS 2007.
- [4] A. Buteau, M. Ounsy, G. Abeille "A Graphical Sequencer for SOLEIL Beamline Acquisitions", ICALEPS'07, Knoxville, Tennessee - USA, Oct 2007.
- [5] A. Buteau, "Experimental Data Storage management in the Nexus format", ICALEPCS 2009.
- [6] G. Viguier, K. Saintin
<https://comete.svn.sourceforge.net/svnroot/comete>.