



Inspector

A Zero Code IDE For Control Systems User Interface Development

Valter Costa, Bertrand Lefort

BE Beams Department | Radio-Frequency | Operation

Developing operational User Interfaces (UI) can be challenging, especially during machine upgrade or commissioning where many changes can suddenly be required. An agile Integrated Development Environment (IDE) with enhanced refactoring capabilities can ease the development process.

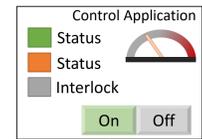
Inspector is an intuitive UI oriented IDE allowing for development of control interfaces and data processing. It features a state of the art visual interface composer fitted with an ample set of graphical components offering rich customization. It also integrates a scripting environment for soft real time data processing and UI scripting for complex interfaces.

Furthermore, Inspector supports many data sources. Alongside the short application development time, it means Inspector can be used in early stages of device engineering or it can be used on top of a full control system stack to create elaborate high level control UIs.

Inspector is now a mission critical tool at CERN providing agile features for creating and maintaining control system interfaces. It is intensively used by experts, machine operators and performs seamlessly from small test benches to complex instruments such as LHC or LINAC4.

Concept

- Control applications made in a Visual IDE with no programming
- Focusing on data presentation
 - Data correctness
 - Good visual elements
- Technology abstraction
 - Disregard of underlying UI technologies
 - No code maintenance
- Rapid application prototyping



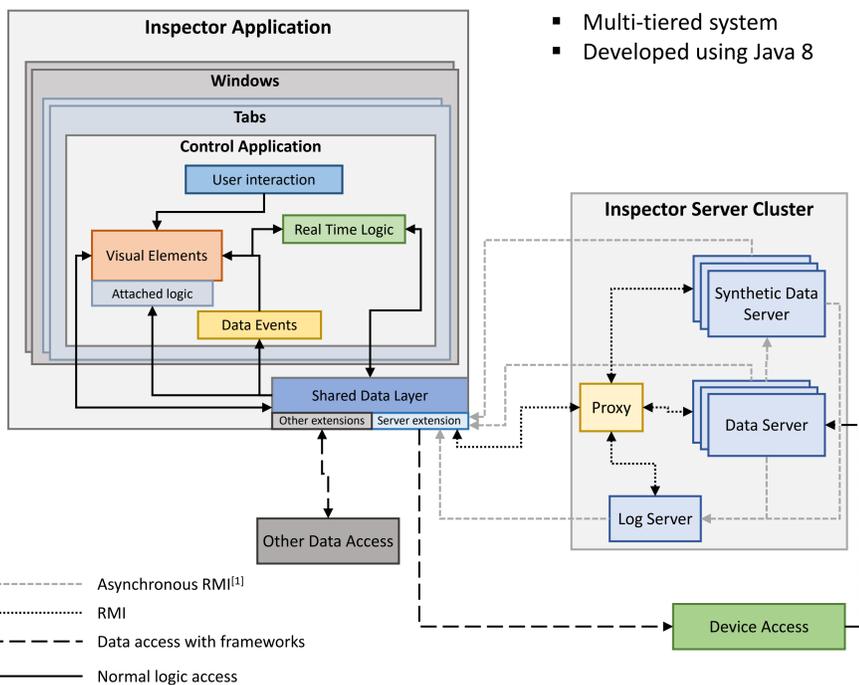
- Simple User Interactions
- What You See Is What You Get

Inspector

- Java
- Frameworks

Host system

Implementation



- Multi-tiered system
- Developed using Java 8

----- Asynchronous RMI^[1]
 RMI
 - - - - - Data access with frameworks
 _____ Normal logic access

Implementation Details

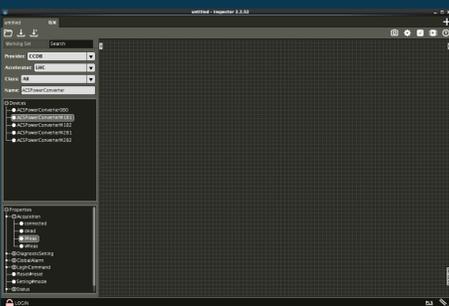
- Client**
 - Portable - Runs on any machine with Java
 - Modular architecture - Modules can be replaced with no consequences to the application core
 - Abstract implementation - Supports communication with any data source
- Servers**
 - Redundant and Scalable - Servers can be spawned dynamically and in different hosts
 - Load Balancing - Data subscriptions are delegated to the server with smallest load
 - Centralized Acquisition - For all clients subscribed to a property, only one connection is made with the data source

Features

- Rich Visual IDE
 - Vast selection of UI elements
 - GUI design tools
- Soft Real Time Data processing
 - Visual Coding
 - Scripting support
 - Data processing can be delegated to servers
- Collaborative and iterative development
- Integrated data source browser
- Customizable via plugins
- Integration in CERN control system via JAPC^[2] and access control with RBAC^[3]

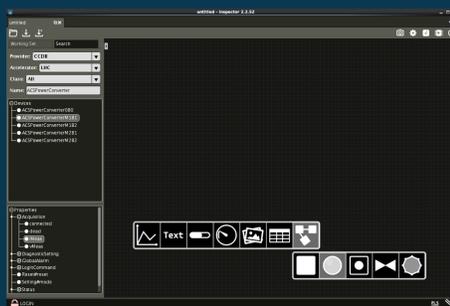
Inspector as an application development tool

1. Data Source Browsing



Data Source browsing integrated in the application.

2. Monitor creation



Inspector analyzes the data attributes and shows only adequate representation elements.

3. Customization and layout



Visual elements can be customized and organized with several layout components and features.

4. Finished application

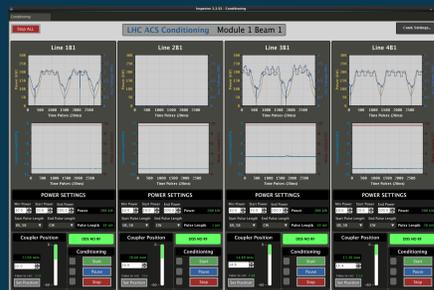


Applications can be exported and easily made available to users.

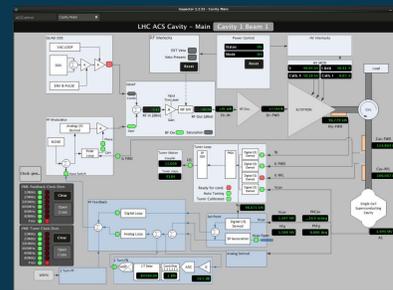
Benefits of Inspector for developers and users

- Easy to develop and maintain applications
- Applications can be created by machine experts with no programming skills
- Underlying code and libraries maintained by Inspector
- UI uniformity across systems
- Free updates and improvements to applications created in Inspector

Example of applications created with Inspector



LHC^[4] RF Conditioning application.



LHC RF Cavity control application.



LINAC4^[5] Tuner Loop control application.

References

- Java Remote Method Invocation, Oracle, <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html>.
- V. Baggiolini et al, "JAPC - the Java API for Parameter Control", ICALPCS'05, Geneva, Switzerland, October 2005.
- S. Gysin, A.D. Petrov, P. Charrue et al, "Role-Based Access Control for The Accelerator Control System At CERN", in Proc. ICALPCS'07, Knoxville, Tennessee, USA, 2007, paper TPPA04, pp.90-92.
- The Large Hadron Collider, <https://home.cern/topics/large-hadron-collider>.
- Linear accelerator 4, <https://home.cern/about/accelerators/linear-accelerator-4>.

ICALPCS 2017, Barcelona, Spain

Valter Costa • valter.costa@cern.ch

Bertrand Lefort • bertrand.lefort@cern.ch

Beams Department, CERN, Geneva, Switzerland