HOW THE COMETE FRAMEWORK ENABLES THE DEVELOPMENT OF GUI APPLICATIONS CONNECTED TO MULTIPLE DATA SOURCES

R. Girardot, G. Viguier, K. Saintin, M. Ounsy, A. Buteau
SOLEIL Synchrotron, Gif-sur-Yvette, France

Abstract

Today at SOLEIL, our end users require that GUI applications display data coming from various sources: live data from the Tango control system, archived data stored in the Tango archiving databases and scientific measurement data stored in NeXus/HDF5 files. Moreover they would like to use the same collection of widgets for the different data sources to be accessed.

On the other side, for GUI application developers, the complexity of data source handling had to be hidden. The COMETE framework has been developed to fulfil these allowing GUI developers to build high quality, modular and reusable scientific oriented GUI applications, with consistent look and feel for end users.

COMETE offers some key features to software developers:

- A data connection mechanism to link the widget to the data source
- Smart refreshing service
- Easy-to-use and succinct API
- Components can be implemented in AWT, SWT and SWING flavours.

This paper will present the work organization, the software architecture and design of the whole system. We'll also introduce the COMETE eco-system and the available applications for data visualisation.

CONTEXT

The SOLEIL ICA team is in charge of the control systems and data visualisation and reduction GUI for accelerators and 30 beamlines.

During the first years of SOLEIL construction, ICA team was focused on developing GUI application and TANGO [1] devices for the control systems.

Then the focus was set on providing data storage and management applications for technical and scientific data.

- For the technical data the Tango Archiving service [2] was developed with very demanding requirements on the GUI for data extraction and visualisation. Today a volume of about 10 TB of data coming from more than 30 000 Tango attributes are stored in Oracle databases.
- For scientific data, it was early decided to use the NeXus data format [3] to record measurements and metadata on all our beamlines. SOLEIL beamlines produce daily thousands of experimental NeXus files with sizes ranging from a few MB up to 100 GB.

Moreover, users needed to use uniformed GUI to view their data, whatever the origin.

Control system and supervision GUI applications were first developed using the ATK [4] toolkit which is intimately linked to TANGO.

On the other hand, for technical and scientific data visualisation and reduction, there was no toolkit available to quickly develop this kind of software not TANGO based.

It is in this context that SOLEIL launched the COMETE project [5] to propose a multi data source toolkit to help software engineers to develop applications independently of the data source.

THE COMETE SOLUTION

Architecture

COMETE is a framework composed of three parts:

1. A set of graphical components (widgets) that are completely dissociated from a data source or even a data type.
2. A data source compatible with the graphical component, each corresponding to a data type.
3. Between the widgets and the data source, a mediation layer in charge of adapting and transmitting data.

DataConnectionManagement

The DataConnectionManagement module is a layer that allows connection between two abstract entities, called "**Target**" and "**Data Sources**".

This module implements a Mediator pattern (Figure 1), as well as various other patterns such as Strategy, Observer etc. Mediator was chosen instead of MVC pattern because our two entities had to be completely independent from each other, to allow easily adding new widgets and sources.

The sources are produced by factories, which include some data refreshment mechanism.

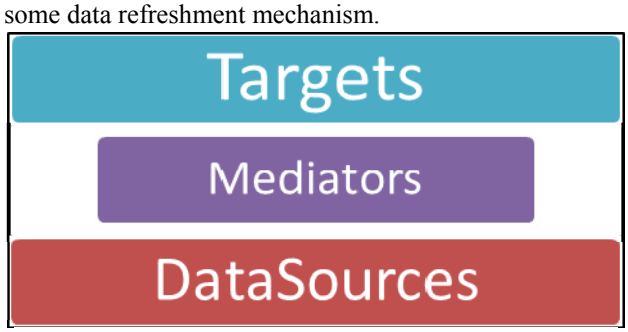

Figure 1: DataConnectionManagement pattern.

ISBN 978-3-95450-146-5

Widgets

COMETE Widgets are "**Targets**" specialisation. They intend to be components which comply with the interfaces described in CometeDefinition that makes them connectable to any data source.

Comete Widgets are available in three implementations (Swing, SWT & AWT), see Figure 2, and therefore integrate well into any existing JAVA software. Anyone can use this set of widgets because they are based on these three major toolkits.

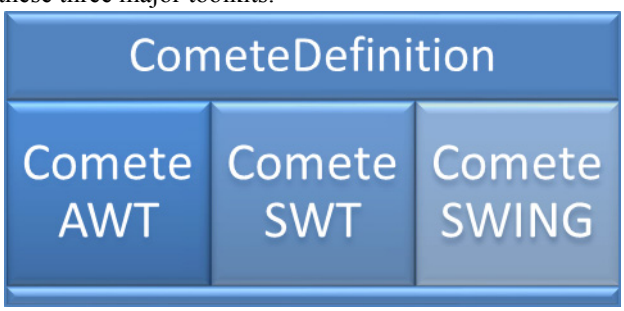

Figure 2: Comete Widget implementations.

Of course the COMETE framework allows adding easily new widgets. The current library of widgets can display:

- Scalar data (textfield, spinner, wheelswitch, slider, etc.)
- Spectrum data (chart viewer)
- Images (image viewer, tables)

Our CometeSwing image viewer (Figure 3) is based on ImageJ [6] which is a popular image processing application. ImageJ is already used by many scientists, so it is very easy for them to use our viewer and they can run their old macros through our Java applications. Since our viewer encapsulates ImageJ, it provides the same image processing features.

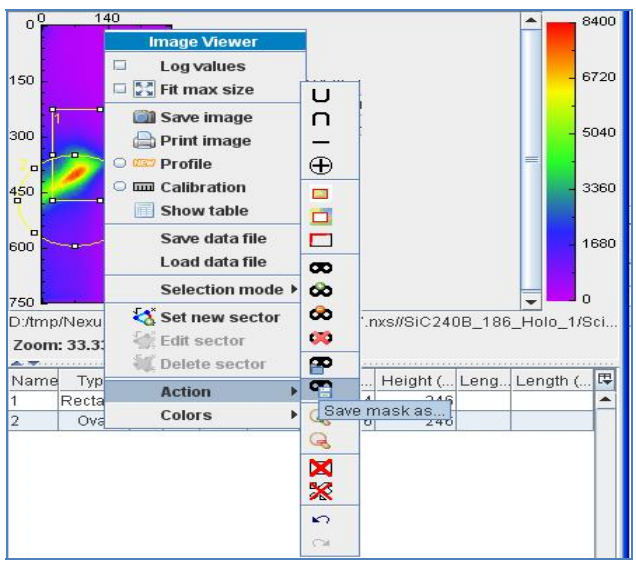

Figure 3: CometeSwing Image Viewer features.

DataSources

When someone wants to add a new data source to Comete, the only thing to do is to implement a class that extends AbstractDataSource. This guarantees that the COMETE mediator will be able to send data to the widget and vice-versa.

Moreover, to manage data refreshment you might need to implement your own IDataSourceFactory class that will instantiate your data sources.

Today the following data sources are available within the COMETE framework:

- A TANGO implementation to access Control System data
- A NeXus implementation to access scientific data from data reduction applications.
- A first SQL implementation to access technical data produced by the Tango Archiving system

CometeBox

The CometeBox module aims to simplify the use of COMETE for developers who intend to use the IDataSourceFactory.

When you want to connect a target to some source produced by an IDataSourceFactory, you have to do following steps:

- Instantiate your target
- Instantiate your IDataSourceFactory
- Instantiate a key
- Ask your IDataSourceFactory to produce your source from this key
- Instantiate your mediator
- Ask your mediator to connect your source to your target

And of course, **this allows accessing only 1 data**.
CometeBox simplifies this access, and **offers the possibility to automatically connect your target to some meta-data** around your source (*for example, a state or quality concerning your source*).
To connect your target to a source and all its meta-data, you have to do following steps:

- Instantiate your target
- Instantiate your CometeBox
- Instantiate a key
- Ask your CometeBox to connect your target to your key

This mechanism is illustrated in the following Use Cases and by the code examples in Figure 5.

ISBN 978-3-95450-146-5

Use Cases

A typical use case is to connect to the same target (*for example: a chart widget*) a Tango or a NeXus data source.

For example, anyone can superpose a spectrum coming from a NeXus file with another one from the TANGO control system (see Figure 4).

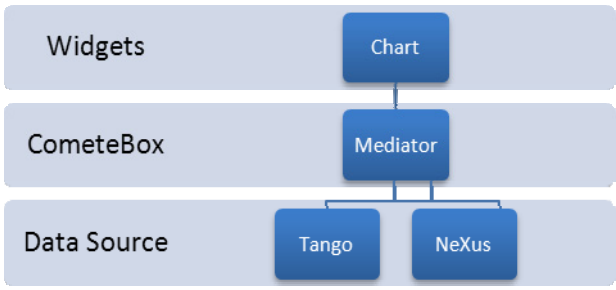

Figure 4: A widget connected to two sources.

This use case can be translated by the following code (Figure 5):

```
NeXus
•NexusKey key = new NexusKey();
•NexusKeyTool.registerAttribute(key,
 "myfile", dataset);
•chartBox.connectWidget(chartViewer, key);

Tango
•TangoKey key = new TangoKey();
•TangoKeyTool.registerAttribute(key,
 "myTangoDevice","myTangoAttribute");
•chartBox.connectWidget(chartViewer, key);
```

Figure 5: Connect a widget to two data source in 6 lines of code.

Application Examples

Here are two examples of graphical applications based on COMETE and using the same component list with different data sources (Figure 6 & Figure 7).

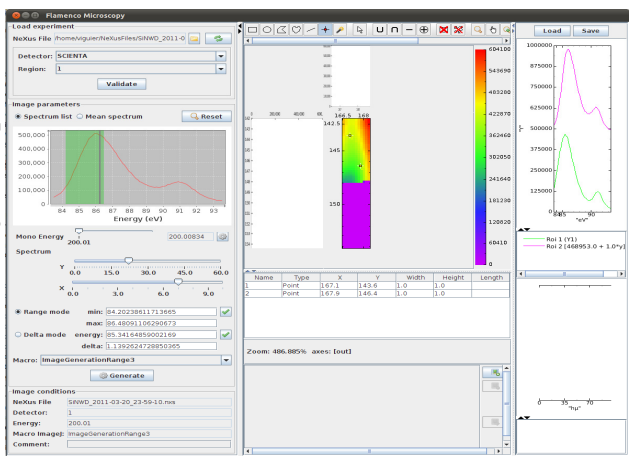

Figure 6: Data reduction application dedicated to microscopy.

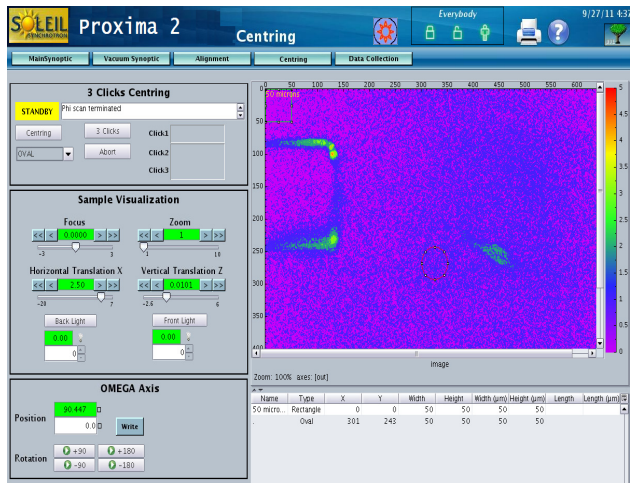

Figure 7: Data visualisation application on PX2 beamline.

CONCLUSION

COMETE is the result of about 10 years of experience on developing data visualisation and reduction applications at SOLEIL. It is now a mature and powerful framework widely and daily used by our developers.

SOLEIL is open to collaborations on the project:

- With potential **contributors** having advanced Java skills to participate to framework extension (new widgets, new data sources, data services, etc…)
- With anyone who would like to **use** existing framework to develop their own application, making feedback on their use and needs.

The easiest way to get in touch with the COMETE team is to contact them at:

comete@synchrotron-soleil.fr

REFERENCES

[1] Tango Control System
http://www.tango-controls.org

[2] J. Guyot, M. Ounsy, S. Pierre-Joseph Zephir "Status of the TANGO Archiving System", ICALEPS 2007

[3] NeXus Data format
http://www.nexusformat.org

[4] F. Poncet, J.L. Pons, "Tango Application Toolkit", ICALEPS'05, Geneva, Oct 2005.

[5] Comete Repository
http://sourceforge.net/apps/trac/comete/

[6] ImageJ: Image processing and analysis in Java
http://rsb.info.nih.gov/ij/

ISBN 978-3-95450-146-5