

TANGO-KEPLER INTEGRATION AT ELI-ALPS*

Péter Ács, Sándor Brockhauser, Lajos Jenő Fülöp, Veronika Hanyecz, Miklós Kiss, Csaba Koncz, Lajos Schrettner, ELI-ALPS, Szeged, Hungary

Abstract

ELI-ALPS will provide a wide range of attosecond pulses which will be used for performing experiments by international research groups. ELI-ALPS will use the TANGO Controls framework to build up the central control system and to integrate the autonomous subsystems regarding software monitoring and control. Beside a robust central and integrated control system, a flexible and dynamic high level environment could be beneficial. The envisioned users will come from diverse fields including chemistry, biology, physics or medicine. Most of the users will not have programming or scripting background. Meanwhile workflow system provides visual programming facilities where the logics can be drawn, which is understandable by the potential users. We have integrated TANGO into the Kepler workflow system because it gives a lot of actors for all natural scientific fields. Moreover it has the potential for running the workflows on HPC or GRID resources. We demonstrated the usability of the development with a beamline simulation. The TANGO-Kepler integration provides an easy-to-use environment for the users therefore it can facilitate e.g. the standardization of measurements protocols as well.

INTRODUCTION

ELI-ALPS is one of the three pillars of the European Extreme Light Infrastructure project. The Attosecond Light Pulse Source (ALPS) facility will provide a wide range of ultrafast light sources (such as coherent XUV and X-ray attosecond pulses) for performing material, condensed matter and surface science, chemical, biological, physical or medical experiments. Besides, the development of the technology for generating 200PW peak intensity pulses is also a main mission. As a research facility, the infrastructure will contain a large number of experimental devices and equipment which have to be managed and controlled by a robust and flexible system. We found that the TANGO Control system [1] is able to address this complexity, it has already been used at several large research infrastructures for more than a decade.

TANGO Controls is an open-source control system framework which provides a foundation to develop control systems. Architecture of a TANGO-based control system has three main components: the Device Server, the Database and the Client. The Device Server component is the base element that provides monitoring and control capabilities over a set of devices. The Database includes

information about the Device Servers and Devices. A TANGO Client may connect to Device Servers using the reference acquired from the Database to perform actions on a certain Device. Experimentalist and scientist users may create programs by using the available APIs. However, not all of the users have a programming or scripting background necessary for such a task.

Scientific workflow systems [2-3] provide a graphical interface to create scientific applications without a deep scripting and/or programming background. A workflow is basically a collection of jobs connected to each other. Each job has input(s), performs a specific function and produces output(s) that can be used by other jobs. The jobs are usually represented as boxes with input and output ports. This kind of graphical representation of unit jobs increases productivity and helps maintainability. It allows the user to save the workflows as well as to share them with others.

The motivation of the paper is to develop a prototype integration of the Kepler scientific workflow system [4] with the TANGO framework. This way a high-level graphical environment could be provided for the users to create programs controlling and monitoring research equipment. Some other institutes already follow a similar approach [5-6]. However, this work has some further motivations, e.g. to collect relevant experience about how to integrate the TANGO framework into other environments.

THE INTEGRATION

A Kepler workflow has an arbitrary number of actors and exactly one director. Every actor represents a job with its necessary input and output ports, and an additional configuration dialog may help to fine-tune the behaviour. Actors communicate by sending messages via interconnected ports. The role of the director is to supervise the execution of the workflow, e.g. it manages the number of executions, enables parallelization if possible, etc.

A fundamental component of the TANGO Control System is the TANGO Device Server, which represents a set of devices with all of their attributes, commands, events and other relevant information. The goal was to create a universal *TANGO actor* in Kepler, which can dynamically adopt any functionalities (commands, attributes, etc.) a TANGO Device may have. However, it is implemented in a way that when the TANGO actor is instantiated, then the necessary functions can be chosen and will only be available afterwards.

It is not completely obvious how to represent and execute a TANGO workflow in Kepler because the functional structure of Kepler is fundamentally different from that of TANGO. As it was mentioned before, a Kepler workflow is controlled by directors.

* The ELI-ALPS project (GOP-1.1.1-12/B-2012-000, GINOP-2.3.6-15-2015-00001) is supported by the European Union and co-financed by the European Regional Development Fund.

[†]<https://www.eli-alps.hu/?q=en>

Kepler inherited five directors from Ptolemy II: PN (Process Network), DE (Discrete Events), CT (Continuous-Time), DDF (Dynamic Dataflow) and SDF (Synchronous Dataflow). Every director is suitable for a different kind of task. For example an SDF might be an appropriate choice for a simple data transformation, however, for example it is not the right decision if one would like to create time-based simulation workflows, because it has no understanding of passing time.

Examining the directors we found that the PN Director is the best choice as a TANGO workflow execution supervisor. A process network director models a workflow as a network of processes where each actor is running on a dedicated thread and generates output data only if all of the input ports have available input tokens to process [7]. Using this model, actors can run simultaneously without blocking each other unnecessarily. Therefore a TANGO actor can continuously gather data from a TANGO device while another TANGO actor is waiting for an event to be triggered from the same TANGO device. With the flexibility of the PN Director a workflow is able to reach that level of functional variety that the TANGO system requires. Figure 1 shows the architecture of the TANGO Controls System and the place of the integration in the system (Kepler-Tango binding in the top-right part).

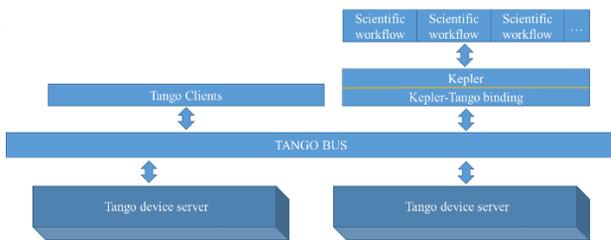


Figure 1: TANGO Controls architecture with Kepler and scientific workflows.

Main Concept

A universal actor is created that behaves as a Device Server, e.g. handles the execution of commands, reads/writes attributes while supervises the connection between the Device Server and the actor. The requirements the solution has to fulfil:

- A TANGO Actor represents a TANGO Device
- The user is able to select a TANGO actor from the list of the available ones (i.e. the TANGO Devices in the TANGO database). TANGO Devices may have a complex name which is easy to mistype, therefore bypassing this interaction makes the user's job more convenient and efficient.
- An input/output port of a TANGO actor represents an attribute or a command of the corresponding TANGO Device
- The user is able to filter which TANGO attributes and commands appear as ports of the corresponding TANGO actor. A Device Server may have tens or even hundreds of commands/attributes, therefore filtrating of them makes the user's job more convenient and efficient.

Steps of the Development

The development is separated into four phases. Each phase is ended by a test sub-phase.

Hard-coded Integration. In this phase a basic workflow component with only the necessary functions was implemented. The goal was to discover the Kepler workflow system whether it is capable of adopting unique functionalities of TANGO.

The developed component is able to handle the connection to a predefined, hard-coded device, manage the execution of a preselected command and read/write of some attributes with different types.

Database Level Integration. The TANGO Database may have thousands of entries of devices, therefore it might lead to complex device instance names that the user may not be able to memorize easily and correctly. When one wants to deploy a TANGO device component on the workflow canvas, it requires the domain name of the TANGO Device so the workflow will be able to connect to and perform actions on that. The basic solution would be that the user types in the full path of the device which is time-consuming, inconvenient and the risk of mistyped name is not ignorable. A more sophisticated solution is needed in a form of a list of available devices with which the user is able to put the actors on the workflow canvas easily.

The developed extension is able to query the available devices from the TANGO Database as well as to integrate it in a way that the user can conveniently deploy TANGO devices as TANGO actors. This feature appears in the context menu, in a combo box as well as in a separate tab.

Complete Type Conversion. One of the biggest and most significant challenges of integrating TANGO into Kepler was the complete type conversion.

TANGO has all the main and basic datatypes, e.g. bool, double, string, etc. However, TANGO defines some unique types which makes the process of conversion ambiguous, e.g. array or 2D array of scalar types, doublestringarray and so on.

In Kepler, data is passed with data tokens, and each data type has its own structural type. If one wants to use data coming from a TANGO device, one has to convert the TANGO data into an environment-dependent structural type, and then wrap it into the corresponding data token. For basic datatypes, the conversion is obvious: TANGO's SCALAR DevBoolean can be converted into a Boolean variable, and then wrapped into a Boolean token.

For IMAGE DevString (matrix of strings) the conversion is more complex. First the data has to be converted into an array of array of strings. Although Kepler has a Matrix token, it is not available for strings, therefore the matrix has to be wrapped into an arrayType(arrayType(string)) token. Although these datatypes conversions may take time to implement, they can be hidden from the user. However, there are exotic types in TANGO, which is a challenge for the developer. In case of DoubleStringArray type of TANGO, which is an array of doubles and string, the

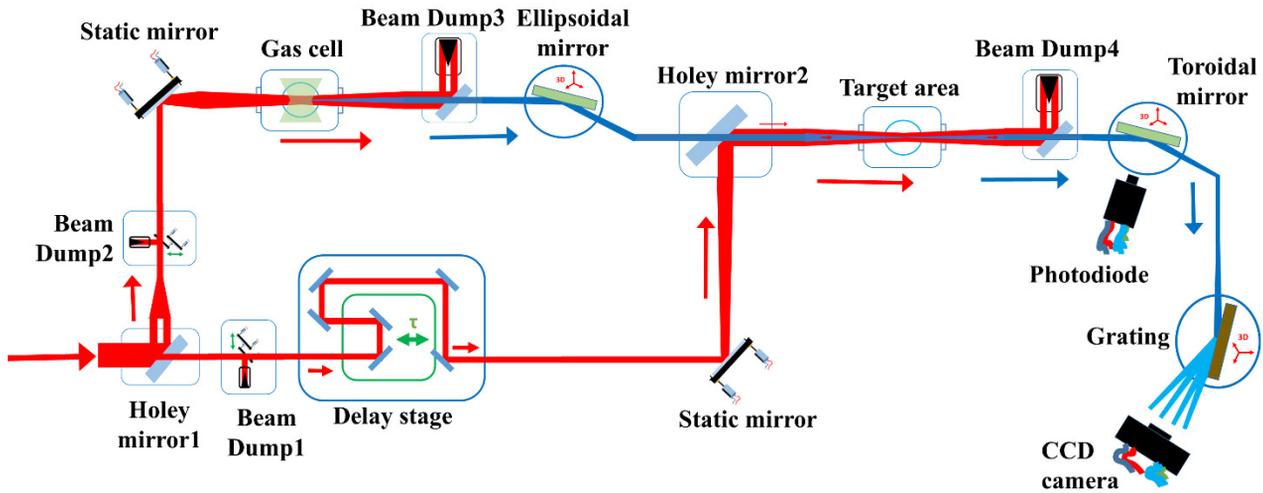


Figure 2: Beamline control system architecture.

developers have to agree on a general integration of how to instantiate it into Kepler.

Configuring ports (attributes, commands). As mentioned before, one TANGO Device may have large number of attributes and the same is true for commands. Most of the components of workflow systems display individual ports for every possible inputs and outputs. Following this concept, a TANGO actor may have hundreds of input and output ports for attributes and commands, which leads to an untraceable communication of workflow actors. Therefore limiting the number of ports for a more user-friendly appearance is a must.

The base concept is that the users should choose the ports (TANGO attributes/commands) in advance. However, the solution should guarantee the opportunity to add and remove ports later as the workflow may change.

In addition, it is restricted that an actor can have only one command, so the role of the actor stays clear and will be easier to handle.

BEAMLINE SIMULATION

For a proof of concept, a simple control system workflow is created on top of a simulated beamline to demonstrate the usability of the prototype. The architecture of the system is shown in Fig 2.

In the beamline simulator the initial InfraRed (IR) beam is split in two parts with a special mirror with a hole in the middle (Holey mirror1). After the beam splitting, both arms have a beam dump with motorized mirrors. The outer ring is focused into a gas cell to generate the XUV attosecond pulse. The generated beam is cleared out from the initial beam by a holey mirror (Beam Dump3). The next element in the beamline is a grazing incidence ellipsoidal mirror with 3D tilting option, and the light is directed to the Holeymirror2. The other arm is delayed with a delay stage and directed to the same mirror. Both beams are directed to the target area. The final beam is cleared out of the IR

Pre-Press Release 23-Oct-2015 11:00

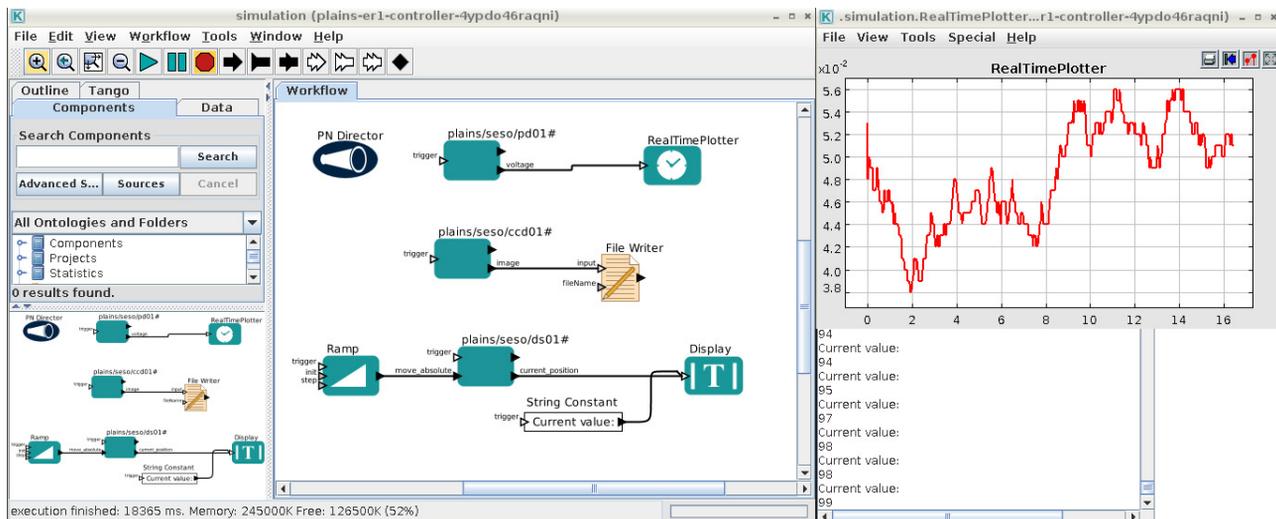


Figure 3: Kepler-TANGO integration: workflow and displayed output.

beam (Beam Dump4) and the XUV pulse is directed by a toroidal mirror to the Grating and detected by the CCD camera unit. A photodiode is placed in the toroidal mirror's chamber for the pulse intensity monitoring. The delay stage has an important role in the recombination process, so called pump-probe experiment. It is delaying the IR pulse to the XUV pulse. The elliptical mirror and the toroidal mirror are positioning the generated XUV beam. The beam dumps with sliding mirrors (1 and 2) can block the beam in both arms.

From control side the following parts are static: holey mirrors, gas cell, beam dump 3 and 4 and target area while beam dump 1 and 2, ellipsoidal mirror, spherical mirror, photodiode and CCD camera can be controlled. In this Kepler demonstration only the delay stage was moved.

A demonstration of the integration can be seen in Fig. 3. The workflow has eight actors plus one director. The plains/seso/pd01 actor represents the photodiode detector, which provides voltage values every time the director executes the actor. The output port of the voltage parameter is connected to a RealTimePlotter, which draws the voltage value on a plotter as the time goes by. The output of the plotter can be seen in the top right corner of the figure.

The second TANGO actor is the plains/seso/ccd01, which provides images from the CCD camera on its output and which will be saved in a file with the help of the File Writer actor.

The last TANGO actor is the plains/seso/ds01, which represents the delay stage component of the control system. Its move_absolute input port is connected to a Ramp actor. The Ramp actor counts from an initialized value to a maximum or infinite with defined steps. With every step, the magnitude of the delay increases with a short delay. With the help of the display actor, the current value can be followed on a dialog on the bottom right corner.

CONCLUSIONS AND PERSPECTIVES

The ELI-ALPS facility will use the TANGO Control system for managing and controlling the experimental devices and equipment. A prototype integration of TANGO with the Kepler Workflow system has been developed. This development could facilitate the creation and modification of experimental protocols. It could be a big benefit for the users and developers, too.

REFERENCES

- [1] A. Götz et al. (2003). "Tango a Corba Based Control System" in Proceedings of ICALEPCS2003, Gyeongju, Korea
- [2] V. Curcin, M. Ghanem (2008). "Scientific Workflow Systems – Can One Size Fit All?" in Proceedings of the 2008 IEEE, CIBEC'08
- [3] B. Ludäscher et al. (2009). "Scientific Process Automation and Workflow Management" in *Scientific Data Management: Challenges, Technology, and Deployment*, ISBN: 978-1-4200-6980-8, chapter 13
- [4] Kepler homepage: <https://kepler-project.org>
- [5] A. Götz et al. (2011). "Data Analysis Workbench" in Proceedings of ICALEPCS2011, WEPKS019, Grenoble, France
- [6] G. Abeillé et al. (2007). "A Graphical Sequencer for Soleil Beamline Acquisitions" in Proceedings of ICALEPCS07, RPPB20, Knoxville, Tennessee, USA
- [7] C. Brooks et al. (2008). "Heterogeneous Concurrent Modeling and Design in Java (Volume 3: Ptolemy II Domains)", EECS Department, University of California, Berkeley