ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA038

EXTENDING TANGO CONTROL SYSTEM WITH KEPLER WORKFLOW, PRESENTED ON AN X-RAY CRYSTALLOGRAPHIC APPLICATION

S. Brockhauser[1], K. Csankó†, V. Bugris, Zs. Filákovics,
Biological Research Center, Szeged, Hungary
P. Ács, V. Hanyecz, ELI-ALPS, Szeged, Hungary
[1]also at University of Szeged, 6720 Szeged, Hungary, and European XFEL, Schenefeld, Germany

Abstract

Nowadays there is a growing need for user friendly workflow editors in all fields of scientific research. A special interest group is present at big physics research facilities where instrumentation is mostly controlled by a robust and reliable low level control software solution. Different types of specific experiments using predetermined automated protocols and on-line data processing with real-time feedback require a more flexible and abstract high level control system[1]. Beside flexibility and dynamism, easy usability is also required for researchers collaborating from several different fields. Tentatively, to test the ease and flexible usability, the Kepler workflow-engine was integrated with TANGO[2]. It enables researchers to automate and document experiment protocols without any programming skill. The X-ray crystallography laboratory at the Biological Research Center of Hungarian Academy of Science (BRC) has implemented an example crystallographic workflow to test the integrated system. This development was performed in cooperation with ELI-ALPS.

INTRODUCTION

Control systems are vital elements of the highly complex experiments performed on large scale experimental physics facilities like synchrotrons, neutron sources or laser-facilities. During an experiment, hundreds and thousands of widely displaced devices need to be controlled simultaneously which can't be carried out only with one control device like a single computer, but it can be handled by several computers in a network. Distributed control systems uses network nodes to connect and interface several individual systems that are responsible only for a segment of the entire experimental setup. Significant development of high-level distributed control systems happened in recent decades following the evolution of available IT solutions. The development of increasingly demanding experimental methods also contributes greatly to this and have challenged the control, acquisition and processing solutions by exponentially increasing amount of data produced (by faster and bigger detectors). This created a claim of automation [3, 4]. On the basis of these developments various control systems were born in the past, such as TANGO [5], EPICS [6], or TINE [7], just to mention a few. Despite these systems are mostly under the editorship of experiment facilities, these developments have begun to flourish also in smaller laboratories in consequence of the achieved wide-range of supported instrumentation, easy configurability, and user-friendliness. One of the most popular systems is TANGO used in many research facilities like ESRF, ALBA, Soleil, DESY, ELI-ALPS [2] and others.

TANGO is a CORBA [8] based, object oriented distributed control system being developed in C++, java and Python by a collaboration headed by ESRF in France. The fundamental unit of this control system is the "device", which is a remote object (usually a device server directly connected to the controlled hardware) registered in a database. Each device has commands, attributes and states to control its instrument and also properties for configuration. The TANGO framework facilitates the implementation of control systems at each level (see Fig. 1). Both low and high-level control environment can be developed by TANGO. Although it is flexible and semi-automated, it still requires some programming skills. It has excellent ability to direct a wide variety of devices and also offers a graphical interface; but it is not offering a dynamic interface for combining complex experiments into workflows. To overcome this problem an easy to use tool needed to be integrated with TANGO which provides a common graphical interface for instrumentation and data processing services to support the requirements of the experiments and to provide a platform for the scientists where they can easily alter the experiments with minimum programming skills and without the need to learn all details of the instrumentation.

Recently, a new initiative has appeared which enables the integration of TANGO to various process management tools [1, 9, 10, 11, 12]. This integration has several advantages because the benefits of both TANGO and the management software can be utilized. On the TANGO side which is a well-supported open source control system, there are numerous former works [11] implementing reliable low level systems and abstracting the control of various hardware facilities. With an integration keeping the modularity of TANGO, the supported hardware list can be fully maintained and experiment protocols can be adopted for different hardware sets without the need of real changes.

A generic architecture of a TANGO-based control system is presented on Fig. 1. The low level system hides the specific and more detailed environment of the facility (like type of the applied devices) while the high level system provides a standard, easy to understand tool for designing and performing experiments.

† csanko.krisztian@brc.hu.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA038

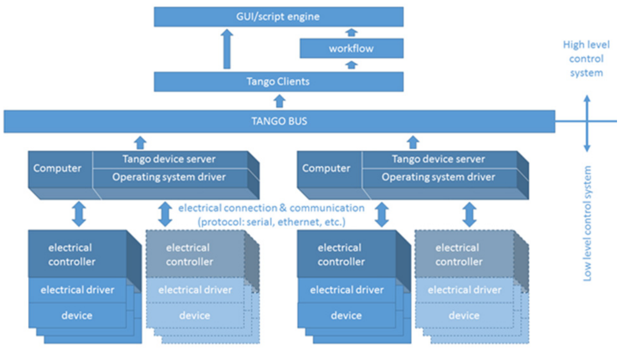

Figure 1: The connection between different parts of the control system.

During the selection process for the workflow engine to be integrated with TANGO, the following conditions played an important role: user friendliness, the main development language, scalability (the number of actors handled by the workflow engine), level of documentation and technical support. Finally, the Kepler workflow solution has been selected which is written in java, so it could be natively integrated with TANGO clients. One of the main reasons for choosing Kepler [11] as a workflow management software was that it has a lot of elaborated modules to support many fields of sciences as well as it is supporting grid computing and web services [13, 14]. These placed it above others (for example the also excellent Passerelle [12]).

WORKFLOW SYSTEMS

Workflow is an abstract description of a real-world process containing different steps and dependencies among them [15]. A workflow system is a flexible tool used for automation and dynamic design offering algorithms to run the processes. Traditionally, there are two types of workflow systems: scientific and business. The scientific workflow systems are rather dataflow-oriented while the business types are much more control flow-oriented [13]. The process steps are represented by so called "actors" or "composite actors" (sometimes called nested workflow or sub-workflow), which could correctly communicate with each other via different protocols. Actors can be grouped to "composite actor" that appear in the workflow just as one item therefore the workflow logic can be better represented. However, it should be mentioned, that not all of workflow systems support such "composite actors" which can easily lead to unmanageable and cumbersome systems. Composite actors enable an actor to act as a workflow on its own and be embedded in other applications as a subworkflow. The task of a workflow actor can be almost anything; e.g. an application can be a device actor controlling a stepper motor and another one can be a feedback device that calculates how many steps that motor should move. Thanks to the modularity, an actor can be replaced with another (with similar functionality) without redesigning the whole system and they may run independently so can take advantage of CPU multithreading if run on the same computer.

At first, it may be seen that workflows are nothing more than visualization of the long existed scripting methods or Unix pipes, which is partially right, but workflows offer more than those as their use is much easier than writing a script or a pipe [16]. Using workflow systems generally do not require scripting knowledge, because actors can be simply dragged and dropped and connected by wires to each other. Workflow systems also allow easy integration of online data analysis with experiment control and so implement feedback into the measurement process. Such automation can lead to increased experiment efficiency, and improvement of results.

Several scientific workflow systems exist due to the intensive developments. Thus, we have done a comparison to investigate the ability of the different workflow tools to find the most suitable for our aim. The primary aspects of this that it must be able to be integrated with TANGO while offering easy management and user experience. TANGO offers an API which can be used when integrating to other workflow systems which is the simplest when the development language is the same (e.g. Java). The investigated workflow systems are the widely used Taverna [17], Kepler [18], Passerelle [19], Orange [20], Akka [21], Pegasus [22], Airavata [23] and Triana [24].

Passerelle integration is already available and has successfully applied in Soleil synchrotron for data acquisition and control processes allowing also web-based, centralized, remote execution [10, 11]. Although it offers a solution out of the box, we did not choose Passerelle because of their other properties, like the limited number of available TANGO actors, the requirement of a tight Java integration for each individual actor; and the lack of useful third party extensions, like Matlab [25] or Labview [26] actors. Considering the last two criteria, Kepler seemed the most suitable workflow system offering the most flexible environment with a possibility of HPC and/or GRID integration for fast online data analysis. Also, Kepler has the most detailed and usable documentation and tutorials.

Kepler is a Java based, powerful and flexible open source software package with the underlying Ptolemy II engine that can handle multiple processes parallel. Several detailed descriptions are available, like in the work [27].

Using the graphical user interface of Kepler, users can simply drag and drop the required analytical components and data sources and then connect them with each other to create a scientific workflow, an executable representation of the steps required to generate results. By implementing actors providing access to experiment control services, the workflow can become the documentation of both the experiment and the generated results.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA038

IMPLEMENTATION

Integration of Kepler and TANGO

In the following, a short overview is presented to describe the main features and functionalities of the integration. Several possibilities exist in Kepler to connect to TANGO devices and their instrumentation services. From workflow-specific actors to generic actors, the resilience is increasing but the challenge of the task is growing simultaneously. The simplest solution is to embed a Tango client which connects to a specific device and reads/sets its hardcoded attributes. In such a case, the actor will work well, but this method is unfavorable, because an actor needs to be implemented for every device and use case. Another handicap is that tens of attributes may belong to a Tango device so an actor would inherit several input and output ports which makes the use of GUI uncomfortable. A bit better solution is when the actor can connect to any type of device using a configurable list of input and output ports. This provides some elasticity but the users have they hand tied in the sense that they can change only a predetermined number of attributes or commands and does not eliminate the errors resulting from manual operations (e.g. typo during copying the name of a device). The best solution is to create a generic TANGO actor which can connect to any attributes or commands.

In order to avoid the manual errors and facilitate its handling, a new tab is created in Kepler GUI which contains the whole TANGO database sorted in usual tree structure (Fig. 2). When instantiating an actor a pop-up window is provided to select the device, command and specify I/O ports. For commands and attributes, respective actors are generated in Kepler.

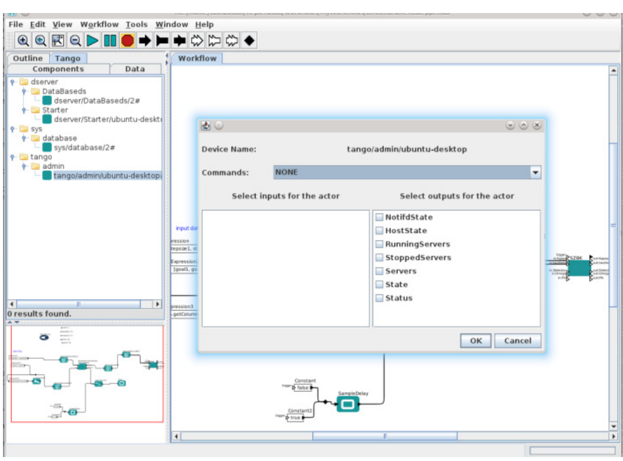

Figure 2: The TANGO database tab integrated into the GUI of Kepler [2].

Cooperation between TANGO and Kepler is hindered due to the fact that they are using different data types. Hence, a compatible conversion must be implemented. The list of the applied conversions is shown in Table 1.

Table 1: Data Type Conversions Between TANGO and Kepler

TANGO data types	Kepler data types
Boolean Scalar ->	Boolean Token
Boolean Spectrum and Image	BooleanMatrixToken
Byte, Short and Int Scalars	IntToken
Byte, Short, Int Spectrum and Image	IntMatrixToken (there are no byte or short matrices)
Long Scalar	LongToken
Long Spectrum and Image	LongMatrixToken
Double Scalar	DoubleToken
Double Spectrum and Image	DoubleMatrixToken
Float Scalar	FloatToken
Float Spectrum and Image	DoubleMatrixToken (no Float matrix exists)
String Scalar	StringToken
String Spectrum	arrayType(string) – Kepler arrays are like Java lists.
String Image	arrayType(arrayType(string)) : array of array of strings
DevState Scalar	StringToken
DevState Spectrum and Image	arrayType(string), arrayType(arrayType(string))
DevEncoded	unsupported

The connected data are stored in static variables thereby they can be reached by the actor durig its construction. The pop-up window (Fig. 2) appears during the drop operation if LSID identifier contains the 'tango' keyword. The structure of the identifier is shown on Table 2.

Table 2: The Syntax of LSID to Distinguish Actors

urn:<lsid>:<authority>:<namespace>:<object>:<revision>#<anchor>	
authority	Generally a URL or any string
namespace	The type of the entity (actor or director) or any string
object	A number which must be individual
revision	The version number of actor
anchor	Optional note

CRYSTALLOGRAPHIC APPLICATION

In this section, a new x-ray crystallographic measurement workflow is presented as a proof of concept showing the cooperation between TANGO and Kepler.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA038

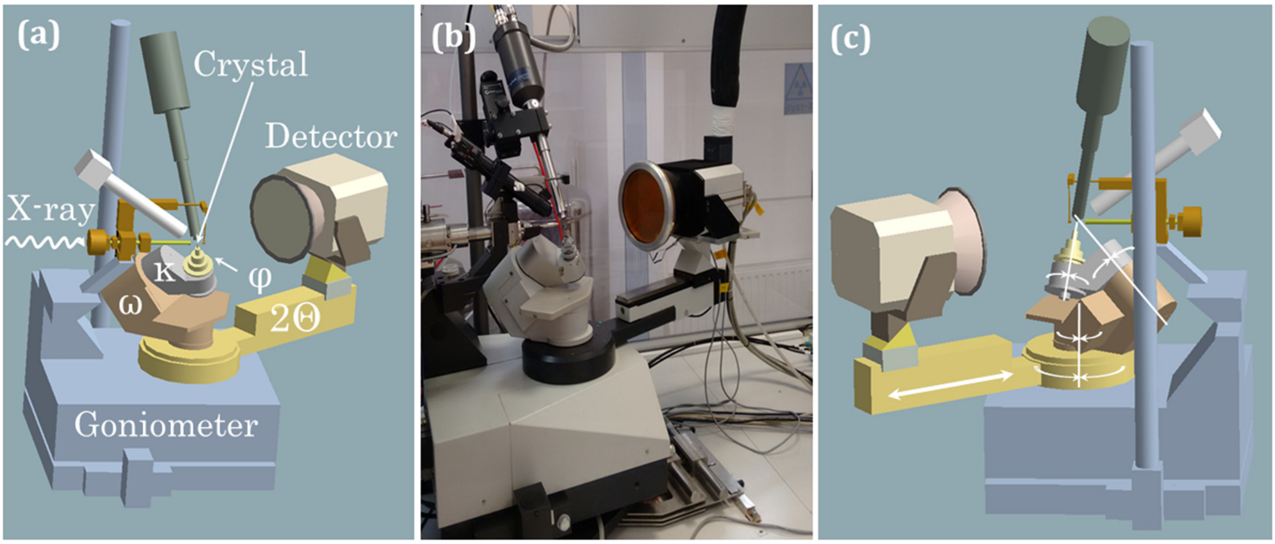

Figure 3: Schematics of an X-ray diffractometer; (a) The Bruker diffractometer as installed at BRC; (b) The 5 stepper motors controlled by TANGO.

Background

Interaction between monochromatic X-ray radiation and the electrons of a crystalline material results the diffraction pattern. The Fourier transform of the resulted diffraction pattern gives the electron density of the molecule which can be used to calculate the real 3D structure. Due to this fact, crystallography is one of the most powerful techniques for detailed molecular structural analysis.

In order to record the required information during the measurement, it is necessary to rotate the crystal and collect multiple diffraction images from different crystal orientation. Crystal rotation is performed by goniometer axes. In our model system the goniometer consists of five stepper motors (Fig. 3.c), so the crystal can be rotated around the following axes: κ, ω and φ respectively, while the detector can be rotated around the 2Θ axis, and the sample to detector distance can be adjusted by the translation stage 'distance' (Fig. 3.a).

Since the X-ray radiation is continuously damaging the internal structure of the crystal during the measurement [28], it is necessary to optimize the rotation trajectory in order to make the process as quick as possible. Several data collection strategy applications exist, and most of them calculate and apply possible crystal symmetry to decrease the rotation ranges needed (Mosflm [29], EDNA/DNA [30]). All of these require a quick scanning step to determine the initial orientation of the crystal [31].

The experiment plan is shown on Fig. 4 as a workflow containing instrument control and data analysis steps. In the first step the collision map is generated by a simulator in order to determine the spatial restrictions to be applied according to the current settings. The second step is mounting the sample on the goniometer, and centering it in the intersection of the X-ray beam and the crystal rotation axis. After moving the detector to its initial position, two images are taken in order to calculate the measurement strategy.

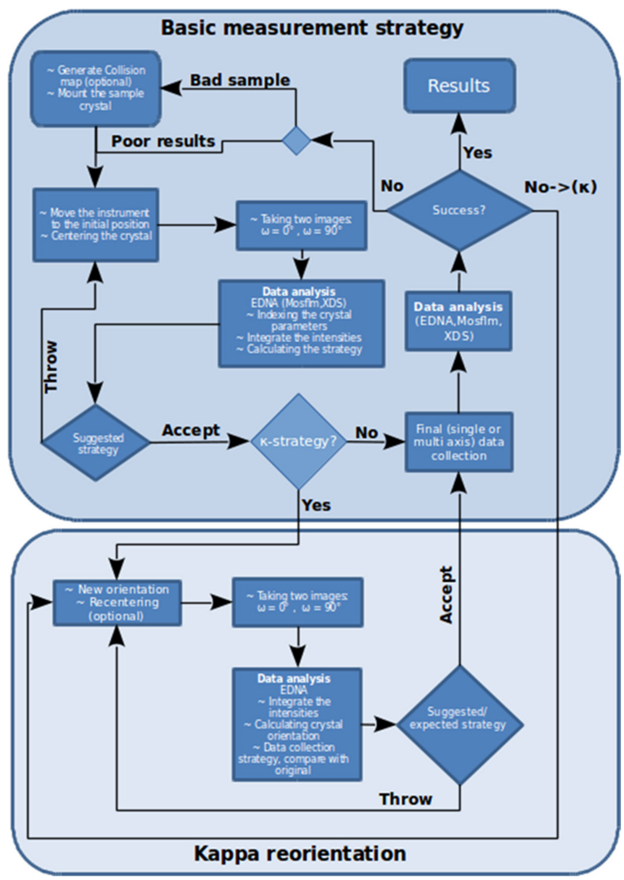

Figure 4: Flowchart of the basic X-ray data collecting strategy and the Kappa data collecting strategy.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA038

EDNA can evaluate the sample quality and suggest an appropriate data collection strategy. This can be a simple scan around a single axis or a so-called kappa reorientation strategy when data collection scan(s) are suggested at different crystal orientations. After the final data collection step(s), further analysis decides whether the strategy or the quality of the crystal were appropriate or new data needs to be collected (maybe trying a different sample).

Implementation and Testing

During an X-ray crystallography diffraction measurement, the TANGO control device server can be responsible for the movements of a goniometer, the detector, and the data acquisition, while external applications must be fed by the data collected and the respective experiment settings to evaluate the data and calculate further strategy that can be then applied to the goniometer. During the test of the new environment a virtual beamline simulator (included in the STAC [32] software package) was used to optimize and debug the test system. This simulator is controlled by TANGO as well and it holds an exact model of a κ-goniometer which is frequently used in x-ray crystallography and it has the same parameters as the one installed in BRC crystallography laboratory (Fig. 3.b), which is a Bruker Nonius – X8 Proteum X-ray diffractometer system. The simulator can also generate a collision map for the instrument and list of the positions of the motors where the parts would collide with each other.

With this approach, the complete workflow of an experiment can be developed and tested without the need of the real hardware even the movement restrictions can be considered. Either simulated data can be used, or previously collected data can be fed into the test setup. The later we have used for testing our integrated environment (Fig. 5). Based on an artificial collision map, we could avoid collisions successfully. While low level instrument control may also present collision avoidance, they tend to be more rigid, and do not necessarily follow the changes of the experiment setup, eg. if researcher insert a new nozzle to control the humidity of sample during the experiment. Using the simulator, the safe control can be maintained by regenerating the collision map with the new object added to the scene.

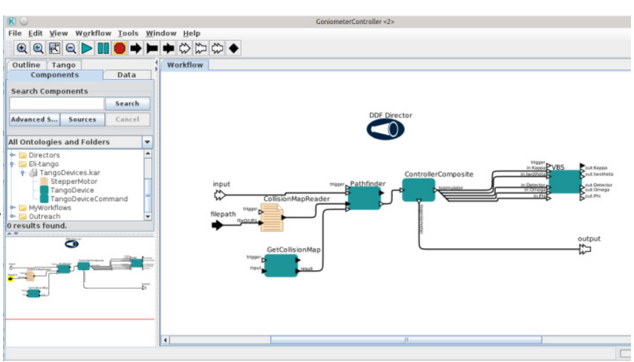

Figure 5: Subworkflow of the goniometer controller (for the simulator).

Figure. 5 shows the subworkflow which is responsible for the collision avoidance during goniometer movements. The input port receives the final position of the goniometer. ‘VBS’ actor is the virtual beamline simulator visualizing the movements in its 3D scene as motions are requested via its TANGO interface. The actor ‘pathfinder’ generates the collision-free trajectory between the actual and the requested positions. The ControllerComposite actor organizes the motor movements in due course and sends the requests to both the real and virtual instrument. The collision map can be retrieved on the fly by the TangoDeviceCommand (in here: GetCollisionMap) actor or can be fetched from file. The output port connects to the device. (The VBS actor can be moved out of this subworkflow and fed directly by the output port).

CONCLUSIONS

As presented, Kepler now can handle TANGO device servers and TANGO actors are dynamically available from Kepler, similarly to Passarelle. The integration of TANGO and Kepler establish a new dynamic environment which is sufficiently easy to handle for users but gives the opportunity the control of devices directly from workflow. Thereby users are able to build up their own workflows and to design experiment protocols. In essence, the whole experiment design and development is shifted from the low-level control software programming environment to the intuitive high-level workflow design environment provided by the Kepler Gui.

Several concepts can be followed in future developments, e.g. when TANGO is only responsible for the communication to the hardware and Kepler arranges the rest. The other use case is when data analysis steps are encapsulated by Tango device servers and Kepler is only used for providing an easy to use interface to the users for connecting Tango services. The described example works on a simulator which generates a collision map before the use of real instrumentation. We suggest offering such simulators for the users as it provides them with a great help during the preparation of experiments. Simulators show the spatial restrictions of the instrument, so prevents occurring unanticipated events in the course of progress. In addition, the remote users can see their experiments in full 3D without the need of entering the instrument hutch.

Users provided by the integrated workflow tool can design specific experiment protocols before their arrival to the large scale facilities which ease the optimization process and the integration with high-level control systems [5, 6, 7, 33] and shorten the necessary machine time, providing significant increase in productivity.

REFERENCES

[1] S. Brockhauser *et al.*, “The use of workflows in the design implementation of complex experiments in macromolecular crystallography”, *Acta Cryst.,* D68, pp. 975-984, 2012.

[2] P. Acs *et al.*, “Tango-Kepler Integration at ELI-ALPS”, in *Proc. 15th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'15)*, Melbourne, Australia, Oct. 2015, pp. 212-215, doi:10.18429/JACoW-ICALEPCS2015-MOPGF050

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA038

[3] Y. Zhao, I. Raicu, I. Foster, "Scientific Workflow Systems for 21st Century, New Bottle or New Wine?", in *Proc. IEEE Congress on Services P.I.*, Honolulu, HI, USA, 2008, pp. 467-471, doi: 10.1109/SERVICES-1.2008.79

[4] A. Beteva *et al.*, "High-throughput sample handling and data collection at synchrotrons: embedding the ESRF into the high-throughput gene-to-structure pipeline", *ActaCryst.*, vol. D62, pp. 1162-1169, 2006.

[5] http://www.tango-controls.org/

[6] http://www.aps.anl.gov/epics/

[7] http://adweb.desy.de/mcs/tine/

[8] http://www.omg.org/spec/CORBA/

[9] D. du Boulay *et al.*, "Remote Instrument Control with CIMA Web Services and Web 2.0 Technology", *IjoE.*, vol. 4, no. 1, 2008.

[10] A. Buteau, M. O. Ounsy, and G. Abeille, "A Graphical Sequencer for SOLEIL Beamline Acquisitions", in *Proc. 11th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'07)*, Oak Ridge, TN, USA, Oct. 2007, paper RPPB20, pp. 647-649.

[11] E. De Ley, D. Jacobs, and M. Ounsy, "Web-based Execution of Graphical Workflows: a Modular Platform for Multifunctional Scientific Process Automation", in *Proc. 13th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'11)*, Grenoble, France, Oct. 2011, paper TUAAULT04, pp. 540-543.

[12] A. Gotz, M. W. Gerring, O. Svensson, and S. Brockhauser, "Data Analysis Workbench", in *Proc. 13th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'11)*, Grenoble, France, Oct. 2011, paper WEPKS019, pp. 823-826.

[13] B. Ludascher *et al.*, "Scientific workflow management and the Kepler system", *Concurrency Computat, Pract. Exper*, 18:10391065,
doi:10.1002/cpe.994, 2006.

[14] J. Wang and I. Altintas, "Early Cloud Experiences with the Kepler Scientific Workflow System", in *Procedia Computer Science*, vol. 9, pp.1630-1634, 2009.

[15] V. Curcin and M. Ghanem, "Scientific workflow systems - can one size fit all?", in *Proc. IEEE CIBEC'08*, Cairo, 2008.

[16] B. Ludascher *et al.*, "Scientific Process Automation and Workflow Management", in *Scientific Data Management: Challenges; Technology, and Deployment.*, Chapman & Hall, CRC, 2009.

[17] http://taverna.sf.net/

[18] https://kepler-project.org/

[19] https://code.google.com/archive/a/eclipselabs.org/p/passerelle

[20] https://orange.biolab.si/workflows/

[21] http://akka.io/

[22] http://pegasus.isi.edu/

[23] http://airavata.apache.org/architecture/workflow.html

[24] I. Taylor, M. Shields, I. Wang and A. Harrison, "The Triana Workflow Environment: Architecture and Applications", in *Workflows for e-Science*, London: Springer, 2007.

[25] *MATLAB and Statistics Toolbox Release 2012b*, The MathWorks, Inc., Natick, Massachusetts, United States.

[26] http://www.ni.com/labview/

[27] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludscher and S. Mock, "Kepler: An Extensible System for Design and Execution of Scientific Workflows", in *Proc. 16th International Conference on Scientific and Statistical Database Management*, Santorini Island, Greece, 2004, doi:10.1109/SSDM.2004.1311241

[28] O. Carugo, K. Djinovi'c Carugo, "When X-rays modify the protein structure: radiation damage at work", *Trends in Biochemical Sciences*, vol. 30 No.4., 2005., doi:0.1016/j.tibs.2005.02.00911

[29] T.G.G. Battye, L. Kontogiannis, O. Johnson, H.R. Powell and A.G.W. Leslie, "iMOSFLM: A New Graphical Interface for Diffraction-Image Processing with MOSFLM", *Acta Cryst.*, vol. D67, pp. 271-281, 2011.

[30] M.-F. Incardona, G. P Bourenkov, K. Levik, R. A. Pieritz, A. N.Popov and O. Svensson, "EDNA: a framework for plugin-based applications applied to X-ray experiment online data analysis", *J. Synch. Rad.*, vol. 16, pp. 872–879, 2009.

[31] A. G. Leslie, "Integration of macromolecular diffraction data", *Acta Cryst.* vol. D55, pp. 1696-1702, 1999.

[32] S. Brockhauser, R. B. G Ravelli and A. A. McCarthy, "The use of a mini-[kappa] goniometer head in macromolecular crystallography diffraction experiments", *Acta Cryst.*, vol. D69, pp. 1241-1251, 2013.

[33] S. Hauf *et al.*, " The Karabo distributed control system", *J. Synch. Rad*, vol. 26, pp. 1448–1461, 2019.