

---

This space is reserved for the Procedia header, do not use it

---

# Introducing Triquetrum, A Possible Future for Kepler and Ptolemy II\*

Christopher Brooks<sup>1</sup> and Jay Jay Billings<sup>2</sup>

<sup>1</sup> Ptolemy Project  
7729 Isabel Dr. Cotati, CA, U.S.A  
cxbrooks@gmail.com

<sup>2</sup> Oak Ridge National Laboratory,  
PO BOX 2008 MS6173  
Oak Ridge, TN 37831  
billingsjj@ornl.gov

---

## Abstract

Triquetrum is an open platform for managing and executing scientific workflows that is under development as an Eclipse project. Both Triquetrum and Kepler use Ptolemy II as their execution engine. Triquetrum presents opportunities and risks for the Kepler community. The opportunities include a possibly larger community for interaction and a path for Kepler to move from Kepler's one-off ant-based build environment towards a more common Open Services Gateway initiative (OSGi)-based environment and a way to maintain a stable Ptolemy II core. The risks include the fact that Triquetrum is a fork of Ptolemy II that would result in package name changes and other possible changes. In addition, Triquetrum is licensed under the Eclipse Public License v1.0, which includes a patent clause that could conflict with the University of California patent clause. This paper describes these opportunities and risks.

*Keywords:* Triquetrum, Kepler, Ptolemy II, Eclipse, Eclipse Public License

---

## 1 Introduction

Triquetrum is a scientific workflow environment that started as an Eclipse project in 2015. Like Kepler [8], Triquetrum is a Java-based, open platform to create, manage and execute scientific workflows using the Ptolemy II [3] execution engine. We propose that the Kepler

---

\*Notice of Copyright: This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

community could benefit from Triquetrum in several ways, including leveraging some of its newer capabilities like its build system, taking advantage of the large community - the Eclipse Science Working Group - in which it is developed, and as an alternative way to maintain the Ptolemy II core. These are strongly related to each other since a large community drives the development of other technologies that are used by Triquetrum and produces more developers to contribute to and maintain its Ptolemy II core. At the same time, Triquetrum helps that large community by augmenting or replacing parts of other codes in the community. Just like Triquetrum and the rest of the Eclipse Community augment and benefit each other, we believe that Kepler and Triquetrum could have a relationship that is even more synergistic given their common heritage.

## 2 What is Triquetrum?

Triquetrum has three lines of work: an Rich Client Platform (RCP) editor, task-based processing, APIs and tools for remote services and actors that supply task-based workflow concepts for scientific data sets. Each line of work is discussed below.

As of January 2016, the Triquetrum source code for the RCP editor is available via GitHub and a nightly build has been set up. However, the vast majority of functionality is missing before Triquetrum can be useful: Actors need to be added, hierarchy needs to be supported etc.

Second, Triquetrum will include APIs and an OSGi (Open Services Gateway initiative) service implementation for Task-based processing, which defines what needs to be done, not how it should be done. The initial implementation of Task-based processing will include storage strategies for traces and provenance information. This work will be done in collaboration with the Eclipse DAWNSci [7], and Eclipse Integrated Computational Environment (ICE) [9], (described below). As part of this work, actors and services will also be developed.

Thirdly, Triquetrum will include a set of supporting APIs and tools for remote services, external packages, data sources, resource managers etc. Here, we intend to collaborate with the Eclipse Parallel Tools Platform (PTP) and Eclipse Communication Framework (ECF) projects.

## 3 The History of Eclipse/Ptolemy Efforts

Open Services Gateway initiative (OSGi) is a specification that supports the reuse of software components. Eclipse Rich Client Platform (RCP), uses OSGi and other technologies to provide a plug-in based Integrate Development Environment (IDE). Over many years, there have been various efforts to move Ptolemy II towards an OSGi-based solution and to use Eclipse RCP to display Ptolemy models. Below is a summary of prior work in this area.

The Ptolemy group has used Eclipse as a development environment since 2003, but other than a Eclipse/Ptolemy plug-in developed by Thomas Feng in 2005 [4], has not directly developed code that invokes the Eclipse APIs.

In 2007, Recito Designer was released. [2] Recito Designer was a Ptolemy II RCP program targeted at OpenOffice applications.

Passerelle, another derived work on Ptolemy II, was developed in the early 2000's by iSencia. Their original UI was an adapted subset of Vergil while a major focus was on higher-level actor APIs and workflow patterns. Since 2005 Synchrotron Soleil started using Passerelle as a subsystem in their software architecture for supporting beamline experiments. [1] In the 2010-2012 era, a Passerelle RCP application was created that was used at the Diamond Light Source



Figure 1: Claudius Ptolemaeus shown holding a triquetrum (pronounced tri-QUET-rum). (source: Wikipedia)

Synchrotron [6] as part of the Data Analysis Workbench (Dawb). Dawb became Data Analysis WorkbeNch (DAWN) in January 2012 and in October, 2014, became part of the DAWNSci Eclipse Project.

In January 2016, DAWN was described as a workbench for scientific data analysis that included support for 1D, 2D and 3D visualization, Python development, execution and debugging and workflows for scientific data. DAWN is focused on the synchrotron community, but is also involved in the neutron scattering, photon science and “any scientific communities with the above or similar needs”.

The work of Reinhard von Hanxleden’s group at the University of Kiel used Ptolemy II as an execution engine for layout work [5] that also used Eclipse facilities. They also developed a Lightweight Diagrams (KLighD)-based KIELER Ptolemy Browser [10] that uses Eclipse.

In 2012-2013, Hallvard Traetteburg updated the Ptolemy II trunk to including building smaller jar files for his work on Cal, the textual language for modeling actor-based systems using Xtext and Xbase.[11]

The creation of Triquetrum as an Eclipse RCP project was prompted by the 2015 closure of Google Code, which meant that Passerelle needed to move to a new repository. Erwin de Ley led the effort to get the Triquetrum project set up. The process was mentored by Jay Jay Billings and the effort was accepted as a project in the Eclipse Science Working Group.

The name Triquetrum was chosen because in the canonical image of Claudius Ptolemaeus used by the Ptolemy Project (Figure 1), Ptolemaeus is holding a triquetrum, which is a three-sided astronomical instrument that could measure astronomical angles more accurately than the astrolabe.

## 4 Opportunities

The opportunities presented by Triquetrum for Kepler are listed below.

### 4.1 Communities

The Eclipse community has a number of active workflow communities including The Eclipse Science Working Group that “works to solve the problems of making science programs interoperable and interchangeable by defining standards, common principles, and software development collaboration.” This group has a mailing list that focuses on how different Eclipse science

projects can interact. Two products of particular interest are DAWNSci and ICE, which will provide use cases and requirements for Triquetrum.

ICE is a full Eclipse distribution that supports scientific computing and publishes Eclipse tools to support development of scientific software. The ICE community is fairly active, with a developer mailing list with 21 members and about 50 posts per month and regularly presents at Eclipse conferences.<sup>1</sup> ICE includes a small workflow engine of its own that is specifically designed for modeling and simulation and the platform provides a large collection of extension points and OSGi declarative services. This makes it possible to easily extend the platform to, for example, create a Triquetrum workflow that analyzes data and to publish that workflow as a native ICE Item.

Triquetrum intends to collaborate with DAWNSci by using DAWNSci's core APIs and implementations to access external data sets, which would be used by extension modules of Triquetrum.

## 4.2 Build and Module Systems

Kepler uses a one of a kind, Ant-based build and module system. At the time the build system was developed, OSGi was considered and not chosen. The build system was primarily developed by one developer who is no longer with the project. The build system is maintainable, but moving towards an OSGi-based system would provide many useful features:

1. By exploiting the Java ClassLoader, allow packages to depend on different versions of sub-packages
2. OSGi is a much more common system, which would mean that Kepler would be more likely to be reused, and extended, which would increase impact
3. OSGi allows other top-level layers, such as GUIs to be used.
4. OSGi is maintained by others. Does Kepler want to be in the software configuration space or the in the workflow space?

## 4.3 Maintenance of the Ptolemy II Core

Ptolemy II consists of about 600K lines of code developed since 1996 via over 71K repository changes. Kepler uses a subset of that code for its execution engine and the basis of its UI. Future staffing changes at the Ptolemy project suggest that it would be useful to explore a more community-based continuous integration environment for Ptolemy II. In particular, moving the Ptolemy II nightly build to a new set of servers maintained by others would help future proof Ptolemy II and Kepler. In addition, creating smaller modules that provide just the set of functionality needed by Kepler would help simplify the maintenance of Kepler.

## 5 Risks

The risks presented by Triquetrum for Kepler are listed below.

---

<sup>1</sup>The kepler-dev and kepler-users lists had 50 posts in 2015. 'ptolemy-hackers' had 312 subscribers and 29 posts in 2015. Of course, much of the Kepler and Ptolemy development occurs in non-public mailing lists.

## 5.1 Forking

Unfortunately, the Eclipse organization requires that when the Ptolemy II Java classes are moved to Eclipse, the Java packages must be renamed. In addition, the version control history would be lost. A silver lining to losing the version control history is that the Ptolemy classes would be hosted on GitHub, which is the favored version control web site du jour.

Changing the package names is the greater of the two issues, but if the changes were done properly, then it could be possible to update the Kepler sources by using a script.<sup>2</sup> The Model Markup Language (MoML) filter facility could be updated to allow opening of old models by users and optionally saving them with the new package names. At a minimum, Kepler could be updated to use the Ptolemy II core found in Eclipse while Kepler retained the current Swing-based Vergil UI.

## 5.2 License

Kepler and Ptolemy II are released under the BSD license. As an Eclipse project, Triquetrum falls under the Eclipse Public License - v1.0 (EPL). This biggest issue is that the EPL includes a patent clause that conflicts with the patent agreement that many Kepler developers may have signed, (c.f. §1 and §2.b.).<sup>3</sup> An issue being raised by the University of California (UC) Office of the president is that UC employees do not have the authority to license patents. In addition, there is concern that if a UC employee unknowingly contributes software that has been already patented elsewhere, then there is a conflict. Short of convincing UC to not pursue software patents, one way around this is to have UC employees commit changes as Eclipse individual contributors. Another alternative would be to appeal to the Eclipse Foundation Board for an exception to release Triquetrum, Kepler, Ptolemy II, etc. under a BSD license, which is allowed where warranted. The Eclipse Advanced Visualization Project, (EAVP), was recently approved to use both the EPL and BSD largely on grounds that many scientists would not want to use it under EPL.

## 5.3 User Interface

Kepler extends Vergil, Ptolemy II's Swing-based user interface, to provide editors for block diagrams. Vergil provides many facilities such as block diagram editors, icon editors, parameter editors, animation and state machine editors.

Eclipse offers several editor development frameworks, for example Graphiti which is used in Triquetrum, but Vergil has many years of focused development on providing a solution to specific set of use cases. The risk here is that while it is easy to quickly develop a Graphiti-based block diagram editor, it is much more difficult to develop an editor with all the commonly used features in Vergil.

In the near term, it would be possible to port Vergil to use the Triquetrum classes and retain the remaining Vergil code while the Triquetrum editors are developed. Alternatively, Vergil could be integrated into Triquetrum using the SWT-AWT bridge, a translation widget that allows for Swing to be embedded into Eclipse's SWT-based UI. In the long term, moving toward a more common framework like Graphiti would help ensure the long-term usability of Kepler.

---

<sup>2</sup>In fact, over 200,000 lines of ICE were updated with a script when it was moved from Sourceforge.net.

<sup>3</sup>Note that the Apache 2.0, Mozilla Public License, LGPL, GPLv.3, and other licenses have similar issues.

## 6 Acknowledgements

The authors would like to thank Erwin de Ley for his input and review of this material. Mr. Brooks is contributing to this project as an individual. Mr. Billings was supported by the US Department of Energy, Office of Nuclear Energy. Oak Ridge National Laboratory is managed by UT-Battelle, LLC, for the US Department of Energy under contract no. DE-AC05-00OR22725.

## 7 Conclusions

Modifying Kepler to use Triquetrum has benefits: larger community, a COTS build/module system, and maintenance of the Ptolemy II core. There are also several risks: forking, licensing issues and an immature user-interface. In the near term, by porting Kepler to use the Triquetrum execution engine and using MoML filters to retain readability of older models, it could be possible for Kepler to gain some of the benefits and avoid some of the risks. Ultimately the community must decide on the future direction of the project, but arguably integrating and consolidating with related Ptolemy II-based projects has its advantages.

## References

- [1] Erwin de Ley. Web-based scientific workflows in Passerelle. In *NOBUGS 2012, RAL, UK*, September 2012.
- [2] Davide Dozza. Recito: An open source graphical ide for automatic document and dataflow manipulation. In *OpenOffice.org Conference 2007, Barcelona*. OpenOffice.org, 2007.
- [3] Johan Eker, Jörn W. Janneck, Edward A. Lee, Jie Liu, Xiaojun Liu, Jozsef Ludvig, Stephen Neuendorffer, Sonia Sachs, and Yuhong Xiong. Taming heterogeneity—the Ptolemy approach. *Proceedings of the IEEE*, 91(2):127–144, 2003.
- [4] Thomas Huining Feng and Edward A. Lee. Incremental checkpointing with application to distributed discrete event simulation. In *Winter Simulation Conference (WSC)*, Monterey, CA, 2006.
- [5] Hauke Fuhrmann and Reinhard von Hanxleden. On the pragmatics of model-based design. Technical Report Bericht Nr. 0913, Department of Computer Science, Christian-Albrechts-Universitaet Kiel, May 2009.
- [6] Matt Gerring. Rich client platform for synchrotron science. In *EclipseConNA 2012, Reston, VA, USA*. Eclipse, 2012.
- [7] Matt Gerring. How to use the DAWNSci Eclipse project. In *EclipseCon Europe 2014, Ludwigsburg, Germany*. Eclipse, 2014.
- [8] B. Ludscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao. Scientific workflow management and the Kepler system. *Concurrency & Computation: Practice & Experience*, 18(10):1039 – 1065, 2006.
- [9] Alex McCaskey, Taylor Patterson, and Jay Jay Billings. Modernizing simulation input generation and post-simulation data visualization with Eclipse ICE. In *EclipseConNA 2015, San Mateo, CA*. Eclipse, 2015.
- [10] Christian Schneider, Miro Spömann, and Reinhard von Hanxleden. Transient view generation in Eclipse. In *Proceedings of the First Workshop on Academics Modeling with Eclipse (AcME)*. Kgs. Lyngby, Denmark, July 2012.
- [11] Hallvard Traetteberg. Modeling user interfaces with Cal and Ptolemy. In *10th Ptolemy Miniconference, Berkeley, CA, U.S.A.* Ptolemy Project, 2013.