The Eclipse Integrated Computational Environment

Jay Jay Billings[1,2], Alexander J. McCaskey[1], ADD ALL OTHER AUTHORS

[1] *Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA*

[2] *The Bredesen Center for Interdisciplinary Research and Graduate Education, University of Tennessee, Knoxville, TN 37996-1200, USA*

Keywords: ,, supercomputing

PACS: 07.05.Tp

---

1. Introduction

High performance modeling and simulation software is hard to use. Much of the challenge comes from the inherent nature of the science under consideration, but perhaps just as much of the problem comes from focusing very heavily on raw compute performance. Focusing on performance is understandable, given the field, but it leaves the rest of us to figure out how to actually use the software in a way that scales for a much larger set of users, many of whom may be novices or migrate between software packages regularly.

In previous work, Billings et. al., discovered through requirements gathering interviews that many of the difficulties using high performance modeling and simulation software fall broadly into five distinct categories, [**?**]. This includes input generation and preprocessing (also called "model setup"); job execution and monitoring; postprocessing, visualization and data analysis; data management; and customizing the software. There are many tools that address these problems individually, but the same work found that the excess number and specialization of these tools also contribute to the learning curve.

Efforts to address these five issues typically fall in with general purpose scientific workflow tools like Kepler, [**?**], or are reduced to myopic tools that satisfy some set of requirements for a single piece of software or platform. That is, the proposed solution to this problem is often to shoe-horn it into existing workflow tools that are so general that they focus on nothing in particular or to ignore the general problem entirely and deploy a completely tailored solution for the given application. These are opposing extremes, but a middle-of-the-road solution is also possible. A workflow engine could be developed that limits its scope to High-Performance Computing (HPC) and to the set of possible workflows that come from the previously mentioned five activities. A rich enough Application Programming Interface (API) could be exposed so that highly customized solutions could still be made based on this limited workflow engine with only a relatively minor amount of additional development required.

It is not clear that one of these solutions is better than the others. Practical requirements will ultimately dictate which way projects go. This work considers the middle ground sollution and presents the Eclipse Integrated Computational Environment (ICE) as proof that it is possible to create such a system. Specifically, we show

- an architecture for such a workflow system that accomplishes all five of the required activities outlined above in a simple, extensible way, **??**.
- that such a system can be cross-platform for workstations and simultaneously deployable on the web, **??**.
- that smart design decisions enable not only authoring code for simulation software, but also make it possible for the system to extend itself, thereby enabling heavy customization, **??**.
- that the system can be easily integrated with other tools, including the general-purpose workflow engines and single-focus tools on opposite ends of this tool space, **??**.

2. Summary and Conclusions

3. Acknowledgements

The authors are grateful for the assistance and support of the following people and institutions without which this work would not have been possible.

FIXME!

This work has been supported by the US Department of Energy, Offices of Nuclear Energy (DOE-NE) and Energy Efficiency and Renewable Energy (DOE-EERE), and by the ORNL Undergraduate Research Participation Program, which is sponsored by ORNL and administered jointly by ORNL and the Oak Ridge Institute for Science and Education (ORISE). This work was also supported in part by the Oak Ridge National Laboratory Director's Research and Development Fund. ORNL is managed by UT-Battelle, LLC, for the US Department of Energy under contract no. DE-AC05-00OR22725. ORISE is managed by Oak Ridge Associated Universities for the US Department of Energy under contract no. DE-AC05-00OR22750.

LDRD ACK?!