Integrating Heterogeneous Rules-Engine Technologies with caGrid

Joel H. Saltz, M.D. Ph.D.; Scott Oster, M.S.; Shannon L. Hastings, M.S.;
Stephen Langella, M.S.; David Ervin; Philip R.O. Payne, Ph.D.
The Ohio State University, Department of Biomedical Informatics, Columbus, OH

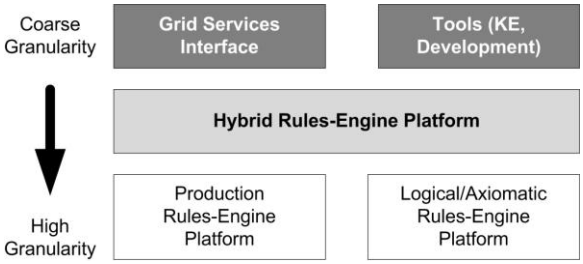
Abstract
The use of rules-engines spans multiple computational and biomedical domains. Within the NCI's caBIG program, the orchestration of grid-based computational workflow has used the BPEL standard. However, recent strategic planning within caBIG has raised questions about the applicability of BPEL for other rule definition and execution scenarios. In response, we have reviewed the current state of rules-engine technologies, and have formulated an architectural model for the integration of heterogeneous rules-engines with caGrid.

Introduction & Background
Recent strategic planning within the NCI's cancer Biomedical Informatics Grid (caBIG™) project has included a focus on the use and applicability of various rules-engine technologies intended to automate workflow at multiple levels. In response to this focus, we have initiated a review of current rules-engine technologies. This review has been situated within the specific context of assessing the feasibility of integrating these technologies with caBIG's grid-bases computing middleware (caGrid) [1]. Based upon our initial findings, we define a rules-engine platform as a combination of an execution engine and knowledge representation scheme intended to allow for the evolution and maintenance of rule-bases independent of application code. Furthermore, we have classified such rule-engine platforms as falling into three major categories:

1) Production rules: used to express declarative, first-order logic in order to automate the actions of computational agents. Examples include CLIPS, JESS, and JBOSS Rules.
2) Logical/axiomatic rules: used to express high-order or multi-part/branching logics, usually in the context of decision support systems. A primary example is the Arden syntax and associated clinical alert systems.
3) Hybrid rules: used to combine either lower-level axiomatic or production rule representation schemes with higher-level workflow abstractions. Examples include BPEL, GLIF, and SAGE.

Finally, based on our review we have concluded that the appropriate choice of a rule-engine platform is extremely context-specific, and therefore, an architectural model that accommodates heterogeneous rules-engines is needed for caGrid.

Proposed Model
Based upon the preceding findings, we propose a conceptual systems architecture model for the deployment of heterogeneous rules-engine platforms within the caGrid environment that consists of three layers:



Key features of this model include: 1) the continued use of BPEL for high-level workflow abstractions and process orchestration, 2) the use of both production and logical/axiomatic rules-engine platforms for highly granular analyses, mediated by the hybrid rules-engine platform, 3) the exposure of this combined rules-engine platform via a grid-services interface, and 4) the provision of tools for both knowledge engineering (KE) by subject matter experts and grid-service enabled rule-based application development.

Discussion
We believe that the proposed model provides for appropriate, context-specific selection and use of rules-engine technologies, while simultaneously supporting rapid application development of rules-based systems via the provision of rules-engine functionality through a model-driven grid service interface. We are currently implementing and evaluating the proposed model to enable a tissue collection protocol monitoring tool for use by the NCI funded CLL Research Consortium.

References
1. Saltz, J., et al., *caGrid: design and implementation of the core architecture of the cancer biomedical informatics grid.* Bioinformatics, 2006. **22**(15): p. 1910-6.