A Roadmap for Composable, Customizable, Collaborative Environments for Extreme Scale Science

Ian Gorton,
Computational Sciences and Math Division,
Pacific Northwest National Lab

Summary

This position paper presents a roadmap for the research and development (R&D) required to create customized collaborative environments for extreme scale science. The results of the R&D described below will be transformative for DOE scientific communities, enabling community-specific collaborative environments to be created in days in weeks, rather than months to years. This will be achieved by (1) leveraging state-of-the-art meta-modeling techniques and model-driven technologies that can *generate* collaborative environments (as opposed to custom coding), and (2) integrating the broad range of modeling, simulation and analysis tools produced by DOE and other science communities with robust, open source frameworks that already exist to facilitate collaboration. This will result in a foundational, extensible technology for rapidly developing, deploying and evolving customized scientific collaboration environments for extreme scale science.

Challenges of Building Collaborative Scientific Environments for Extreme Scale Science

Creating and deploying collaboration environments for different scientific user communities is an extremely challenging problem for DOE. The roots of these challenges stem from the diversity of the science that different communities undertake, and the heterogeneity of the data and tools, both simulation and analysis, that each scientific discipline utilizes. This inherent complexity is exacerbated by the need for careful design of customized collaborative user interfaces for each community, in order to produce domain-specific collaboration tools that are suited to the community's needs and culture (e.g. policies and norms for data sharing).

The sheer number of different scientific user communities and their diverse requirements makes building customized, point solutions for each community impractical, especially given expected budget constraints. In addition, some user communities have existing collaborative tools (e.g climate science and the Earth Systems Grid) that must be leveraged and integrated in future developments. Building custom interfaces for these environments is again intractable in any reasonable timeframe and cost.

These problems point to the need for a radically new approach to creating customized collaborative scientific environments. In order to do 'more with less', existing technologies from both DOE and the open source community should be leveraged and integrated using state-of-the-art software engineering approaches. Hand-coding every different collaborative environment is not an approach that scales in terms of development costs and ability to evolve over long periods of time. Hence transformative approaches must be utilized that generate tailored collaborative

user environments based on core frameworks and meta-models, just as we now routinely generate optimized executable code from high-level source code representations.

Composable Collaborative Scientific Environments

This vision encompasses a step change in the way customized collaborative environments for different user communities are created, deployed and evolved. In the next few years, we envisage scenarios where environments for new user communities can be *generated and deployed* within a small number of weeks. These will include customized user interfaces, capabilities for handling community-specific data types, invoking community specific simulation, analysis and workflows tools, and a secure, policy-driven collaborative user environment.

The process for creating such environments will be as follows:

1. Stakeholders from the user community work with an expert from the collaborative technology team to define the data types, simulations, analysis tools and workflows that are needed. Together they form a design team for the customized user environment.
2. The design team develops simple *wrappers* that enable the community's specific resources (data, codes, workflows) to be automatically integrated with the collaborative technology framework.
3. The design team utilizes a graphical design tool to describe and compose the features that the collaborative environment must have. These features include community specific resources and more general features such as security requirements and policies for data visibility
4. The community-specific collaborative environment is *generated* in minutes and automatically deployed so that it is ready for immediate testing, enhancement, regeneration and subsequent use.

This approach is transformative over today's prevailing engineering approach of hand-coding customized environments for every different scientific user community. Specifically, the advantages are:

Massively reduced costs: Customized collaborative environments for scientific user communities are generated in days to weeks, with no custom code needed for each deployment. This contrasts with the months to years that are currently consumed in building community-specific environments.
Massively reduced time-to-deploy: Generated collaborative environments can be deployed instantly, and tested, refined and deployed in a matter of days. This is possible because modifications are made based on high-level descriptions and regeneration, not detailed code changes.
Ease of evolution: Once deployed, a collaborative environment's model can be enhanced and the changed parts of the environment regenerated and deployed within minutes
Ease of leverage of scientific resources: The framework is designed to incorporate community-specific codes. Integrating these requires a one-off effort to create a wrapper for the tool. Once created, this wrapper becomes part of the framework that can be leveraged by all subsequent collaborative environments created for different communities.

Taken together, these advantages can deliver an unprecedented return-on-investment for DOE programs. This will be achieved by increasing scientific productivity in a whole range of diverse

scientific communities that can easily and cost-efficiently leverage these breakthrough technologies.

The Path to Composable Collaborative Scientific Environments

In order to realize this vision, several computer science advances are needed to create the infrastructure and tools that underpin this transformative approach. These are:

A meta-model for collaborative scientific user environments: A meta-model defines the essential elements and abstractions that exist in scientific collaborative environments, for example users, simulations, data sets, policies, provenance, and their inherent relationships. The meta-model provides the basis for tools that can subsequently generate the deployable user environment. It is analogous to a grammar for a programming language.

A lightweight, extensible collaborative user framework: The vast majority of the common, core functionality of collaborative environments exists already in many open source tools. These have been created by both the scientific and business communities. It is therefore possible to design and create an underlying framework that can serve as the common basis for generated user environments. This framework would extensively leverage existing DOE investments in tools (eg Kepler, Paraview, Globus), and integrate them with widely-deployed, robust open source collaborative frameworks. The framework would also define extension points where community-specific tools, data, workflows and policies can be 'plugged in' as black boxes, enabling customization. This framework is analogous to the run –time support system used by compilers that generate executable codes based on high-level language descriptions.

Composition and description tools for collaborative user environments: Designers must be able to easily compose and configure the features and resources required in a collaborative environment for their community. This will result in a customized version of the general meta-model that specifies the precise capabilities that the environment must encompass. Tools to facilitate creating this model, including graphical design tools and interfaces to widely-used scripting languages, must therefore be created. These tools leverage the meta-model (above) to ensure that valid compositions are created, greatly easing the design effort.

Scalable, extensible generation tools: Based on the specific model of the customized collaborative environment created by designers, tools are needed that generate a tailored version of the collaborative user framework. These tools take this model and generate user interfaces, workflows, and access to data repositories and analysis tools that the scientific user community needs. The generated environment is instantly deployable and testable by users.

Figure 1 depicts at a very high level the outcomes of this research and development roadmap. It exploits a taxonomy that collects various tools serving similar purposes under a common abstraction layer and treats each tool as a black box that offers a variety of capabilities. Interfaces for each tool that adhere to this abstraction layer then make it possible for high level models of discipline-specific collaborative environments to specify their needs in terms of this common abstraction layer. Constructing discovery environments based on these abstractions can be done through scripting languages (e.g. Python) by developers or a graphical tool that would be amenable for end-user scientists to customize aspects of an environment (e.g. execute a Paraview visualization after a simulation runs).

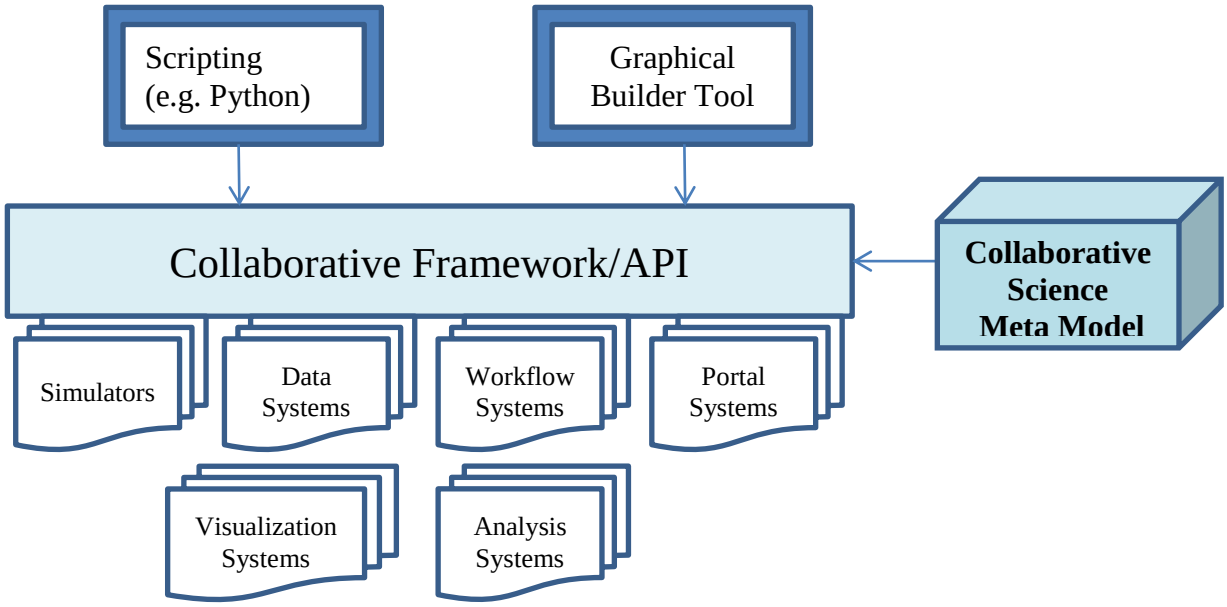

Figure 1 Overview of Scientific Discovery Collaborative Framework

The Building Blocks and Steps for Fulfilling this Vision

The motivation for this roadmap is based on:

1) the availability or a rich ecosystem of standalone tools for visualization, workflow, data management, performance analysis and data analysis that have been developed by SciDAC and other communities, but which are not integrated into a useable framework for scientists to easily exploit;
2) the observation that considerable commonality exists across scientific simulation and modeling disciplines in terms of the steps undertaken to create and execute simulations, and visualize and analyze their results. This commonality can be exploited in designing domain-specific collaborative environments that can be 'customized' for a given scientific discipline based on these common patterns of usage;
3) the availability of advanced, mature, open source model-driven development tools that can be brought together into a high-level framework for designing and generating domain-specific environments. Technologies such as the Eclipse Modeling Framework[1] can be used to create meta-models of scientific discovery environments for a range of science domains and associated analytical tasks. These models will be based on abstractions of the broad collection of available community tools for each phase of the modeling and simulation process.

Meta-modeling tools are widely used in the software industry to generate cohesive applications that exploit and integrate existing components, libraries and tools. Prior work at PNNL and with our partners has given us deep insights into these technologies and approaches. For example, we have developed components for the Kepler workflow tool, and built scientific collaboration

[1] http://eclipse.org/modeling/emf/

environments such as ECCE for computational chemistry and SALSSA for groundwater modeling that integrate a range of tools into a coherent user environment. We have designed the Velo[2] and CAT frameworks, which are domain-independent knowledge management environments that can be easily customized for specific science domain needs; Velo is currently used for carbon sequestration, subsurface, climate and nuclear waste modeling, and in the ASCEM Platform[3]. We have considerable expertise with meta-modeling tools for building and generating complex applications based on frameworks, for example the MeDICi[4] Component Builder is built using EMF and Velo/CAT are built on highly extensible and reusable open source software frameworks.

An ideal approach to undertake this R&D would involve developing these core collaborative frameworks and tools in conjunction with a small number of specific DOE scientific user communities. By working with several different science domains concurrently, a broad coverage of requirements can be obtained that will ensure the meta model, framework and abstractions are designed to be highly extensible and reusable. After an R&D period of 1-2 years, concrete outcomes in terms of initial versions of collaboration environments can be delivered. This will demonstrate the feasibility of the approach, and provide the foundation for reaching out and enhancing the technologies in conjunction with the broad DOE scientific community.

[2] Ian Gorton, Chandrika Sivaramakrishnan, Gary Black, Signe White, Sumit Purohit, Michael Madison, and Karen Schuchardt. 2011. Velo: riding the knowledge management wave for simulation and modeling. In Proceedings of the 4th International Workshop on Software Engineering for Computational Science and Engineering (SECSE '11). ACM, New York, NY, USA,

[3] www.ascemdoe.org

[4] Ian Gorton, Adam Wynne, Yan Liu, Jian Yin: Components in the Pipeline. IEEE Software 28(3): 34-40 (2011)