A Middleware-based Computing Architecture for Virtual Medicine*

Line C. Pouchard, Richard C. Ward
Oak Ridge National Laboratory

Michael N. Huhns, Laura Zavala, Karthik Iyer
University of South Carolina

Problem

As the use of computer modeling spreads in medical research, smarter models of the human anatomy are needed in a broad range of applications, including clinical diagnosis, surgical planning, cancer treatment using radiation therapy, and other applications [1,2]. The Defense Advanced Research Projects Agency Virtual Soldier Project investigates methods to improve mathematical models of individual soldiers against in vivo experiments. Phase I focuses on coupling physiological, electro-mechanical, and anatomical properties of the heart in a single interactive 3D display that includes time series, a searchable semantic network and complex visualization of a virtual human heart. Large teams of researchers distributed nationwide among six universities, one national laboratory, and several companies collaborate. The complexity of the project requires a flexible, distributed, computing architecture that supports independent models for seamless integration of software components and ensuring interoperability.

Method

The architecture must support programmatic interaction in “near real-time” between the Integrator/Executive, an error prediction engine measuring the accuracy of the models [3] and the High-level Integrative Physiological Model (HIP) run through the JSIM batch program [4]. Additional requirements include accessing semantic concepts from the Foundation Model Anatomist [5], automating data movement, enabling programmatic interfaces between pre-computed simulation and experimental data stored in databases and file systems, and testing performance. “Near real-time” is defined by human perception and currently estimated as the rate of new data to be presented every second. We are building a middleware-based, flexible, computing architecture that provides services for bringing output data of one system to input of another upon request through programmatic interfaces. This architecture provides services for collecting, accessing, querying and presenting the data made available by models and experiments to the Executive. An inventory of producers and consumers of data were identified by iteration with members of each team. A consumer of data is a software system (typically designed by one team) that runs a program/model using the output data from another system. The same system can be both a producer and consumer of input and output data. Interactions were defined using the Unified Modeling Language [6] and data flow diagrams.

Results

A software component architecture was emphasized for adaptability and extensibility of the architecture in later phases of the project. Prototype services for programmatic access to the JSIM software that run a HIP simulation over the Web have been deployed. These services are implemented using the Simple Object Access Protocol, a Web Consortium standard recommendation for Web services. Other services have been designed but not yet implemented at the time of this writing: a file discovery service based on metadata, file requestor and locator services, and a file mapper service that parses logical to physical file names. Performance testing of the JSIM/FE model web service using the spheart model was done at USC. Timing tests for a single computation resulted in near real time performance including data transfer between client and server. Data volume for JSIM batch output was 936K per run.

Conclusion

Middleware services for running the HIP model over the Web have been implemented and tested, and can now serve as a blueprint for the planned services in the architecture.

Discussion

A large effort went into collecting project and system requirements in view of the Phase I demonstration. The effort will continue throughout the project as it evolves but the design of an architecture based on modules interacting through independent interfaces pays off as new model versions, access methods, and number of systems become available. The effort of migrating from a stand-alone to a distributed environment will also be reduced. An immediate benefit was to make available the computational and modeling capabilities of the HIP model to a wide audience through the Web while developers retain control of the model. While the performance is currently insufficient for serving the needs of the Integrator/Executive, a significant improvement must be made in the JSIM client as this appears to be a bottleneck. Service performance also needs to be improved.

References

[1] Satava, RM, "Cybersurgeon: Advanced simulation technologies for surgical education", *Medical Simulation and Training*, 1:6-9 (1996)

[2] Satava, RM, "Medical Virtual Reality", Medicine Meets Virtual Reality: 4 Health Care in the Information Age – Future Tools for Transforming Medicine, IOS Press, Amsterdam, pp. 100-105 (1996).

[3] Bookstein, F. et al. Tracking physiological models by Kalman filters. MMVR13, forthcoming.

[4] Kellen MR and Bassingthwaighte JB. An integrative model of coupled water and solute exchange in the heart. Am *J Physiol Heart Circ Physiol* 285: H1303-H1316, 2003.

[5] Rosse, C and Mejino, JLV. (2003) A Reference Ontology for Bioinformatics: The Foundational Model of Anatomy. *Journal of Biomedical Informatics* 36:478-500.

[6] Rumbaugh, J, Jacobson, I, Booch, G, *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999.

*This work was supported by a grant from the DARPA, executed by the U.S. Army Medical Research and Materiel Command/TATRC Cooperative Agreement, Contract # W81XWH-04-2-0012.