A Web-based Computer Architecture for the Virtual Soldier

Line C. Pouchard, Richard C. Ward
Oak Ridge National Laboratory
Michael N. Huhns, Laura Zavala, Karthik Iyer
University of South Carolina

http://www.virtualsoldier.net

What is the Virtual Soldier Project?

- Develop methods that will revolutionize medical care for the soldier in the battlefield and elsewhere.
- Phase I:
  - Coupling physiological, electro-mechanical, and anatomical properties of the heart to predict chances of survival after injury.
  - An interactive 3D display that includes time series, a searchable semantic network and complex visualization of a virtual human heart.
- http://www.virtualsoldier.net

What is the project trying to do?

Build a Virtual Soldier on an Electronic "dog tag" From Which to Diagnose and Predict Combat Injury

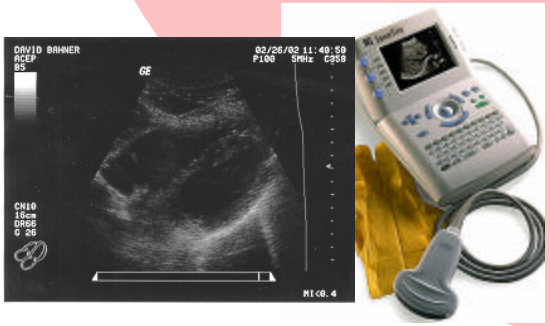
Why?

Instantly & Accurately diagnose internal combat injury (heart)

How?

3-D model from total body scan on "dog tag" (anatomy & physiology)

Compare to data acquired on the battlefield after wounding (Ultrasound)



Predict likelihood of battlefield mortality

Holographic Medical Electronic Representation

Holomer

ORNL mandate:

- **Develop a middleware computer architecture in view of a demonstration of the Holomer.**
- **Includes a graphical representation of human anatomy enriched with medical indicators obtained in physiological modeling.**
- **Seamlessly integrate components provided by six universities, one lab, and several companies.**

Why a Web Services approach for the Virtual Soldier Project?

- Six universities, one lab, several companies.
- A web-based distributed approach is advantageous to expose a programmatic interface for running services, programs, and software components between the different teams.
- Developers own their code: each version or change in the service does not affect the client: ex JSIM updated 30 times a year.
- New data can be automatically available.
- An underlying principle of making these services and data available to Web users in the future.

Defining Requirements for Integrating Project Components

A Provider/Consumer table

- Who provides data or a service that another will use?
  - Human users, other software components, APIs.
- What is the required functionality of the necessary interactions within the project?
- What data formats and platforms are likely to exist?
- **VSKB**: The Virtual Soldier Knowledge Base, an ontology of anatomy (Digital Anatomist Foundation Model).
  - approximately 70,000 concepts and over 110,000 terms; 168 relationship types; over 1.5 million relationship instances

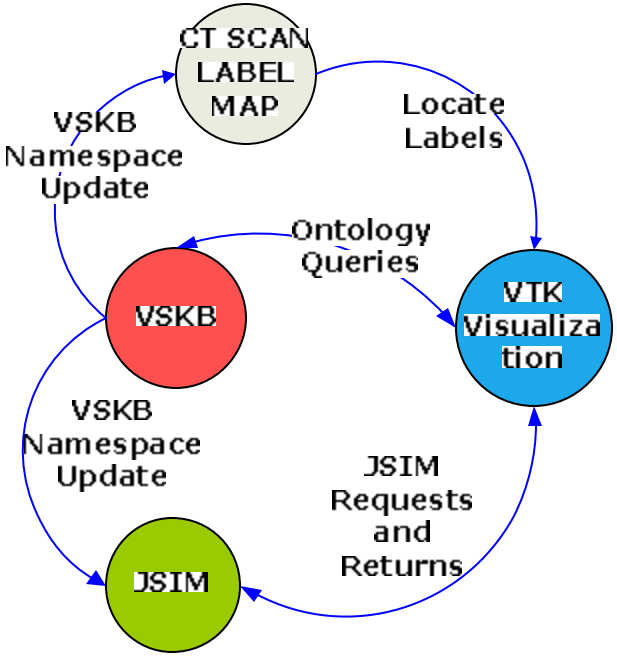
PROVIDER	SERVICE NAME	CONSUMER
VSKB Knowledge Base	MIDDLEWARE Knowledge Update Services	HIP Model Simulation (JSim)
VSKB Namespace	MIDDLEWARE Ontology Query Services	VTK Visualization
HIP Model Simulation (JSim)	MIDDLEWARE Dynamic Invocation Services	VTK Visualization
FE Heart and Thorax Simulation	MIDDLEWARE Data Location Services	SCIRun Visualization

- **HIP**: High-level Interactive Physiological model run by JSim.
  - **JSIM**: component that runs the model.

Identifying Necessary Web Services

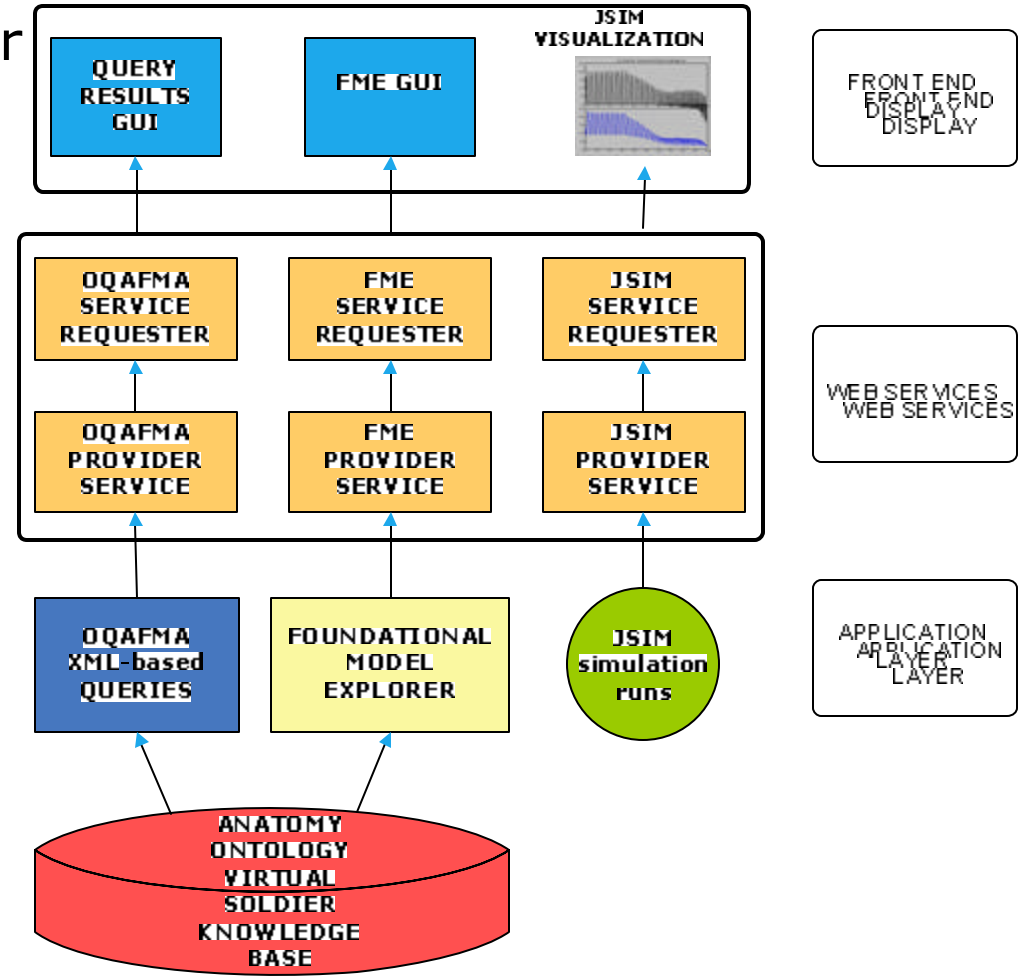
- Services for collecting, accessing, querying and presenting the data made available by models and experiments to others within the project.

- Process diagrams identifying the web services supporting requirements.
  - Locate
  - Update
  - Query
  - Run



VSP Web Service Architecture

- Display: a Web or other graphical front-end
  - Independent from the WS
- Universal Data Format
  - XML
- Service Description Language
  - WSDL
- Service Interaction
  - SOAP
- Communication
  - http, tcp/ip

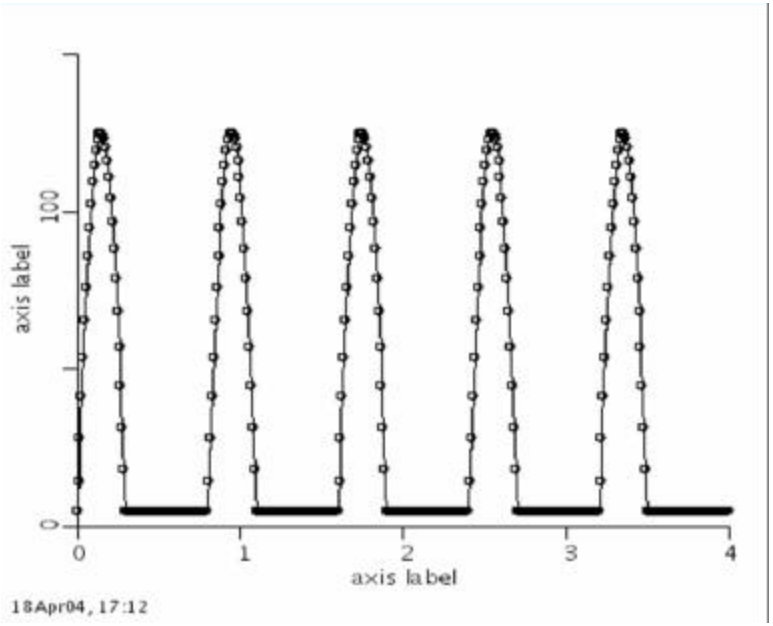


High-level Interactive Physiological Model (U. Washington)

Goal: simulate a patient monitor with heartbeat, ventricular pressure, blood flow. Run by JSim software

```
t = {0 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08…} mmHg*sec*ml^-1
Rvc(t) = {(0,0.03192521) (0.01,0.03192828) (0.02,0.03193117)…} mmHg*sec*ml^-1
Plv(t) = {(0,5) (0.01,14.28506947) (0.02,28.02342933))…} mmHg
Paop2(t) = {(0,65.10416667) (0.01,65.0309892) (0.02,64.90119688)…} mmHg
Paop(t) = {(0,65.10416667) (0.01,65.0309892) (0.02,64.90119688)…} mmHg
   etc…
```

Plv(mmHg)

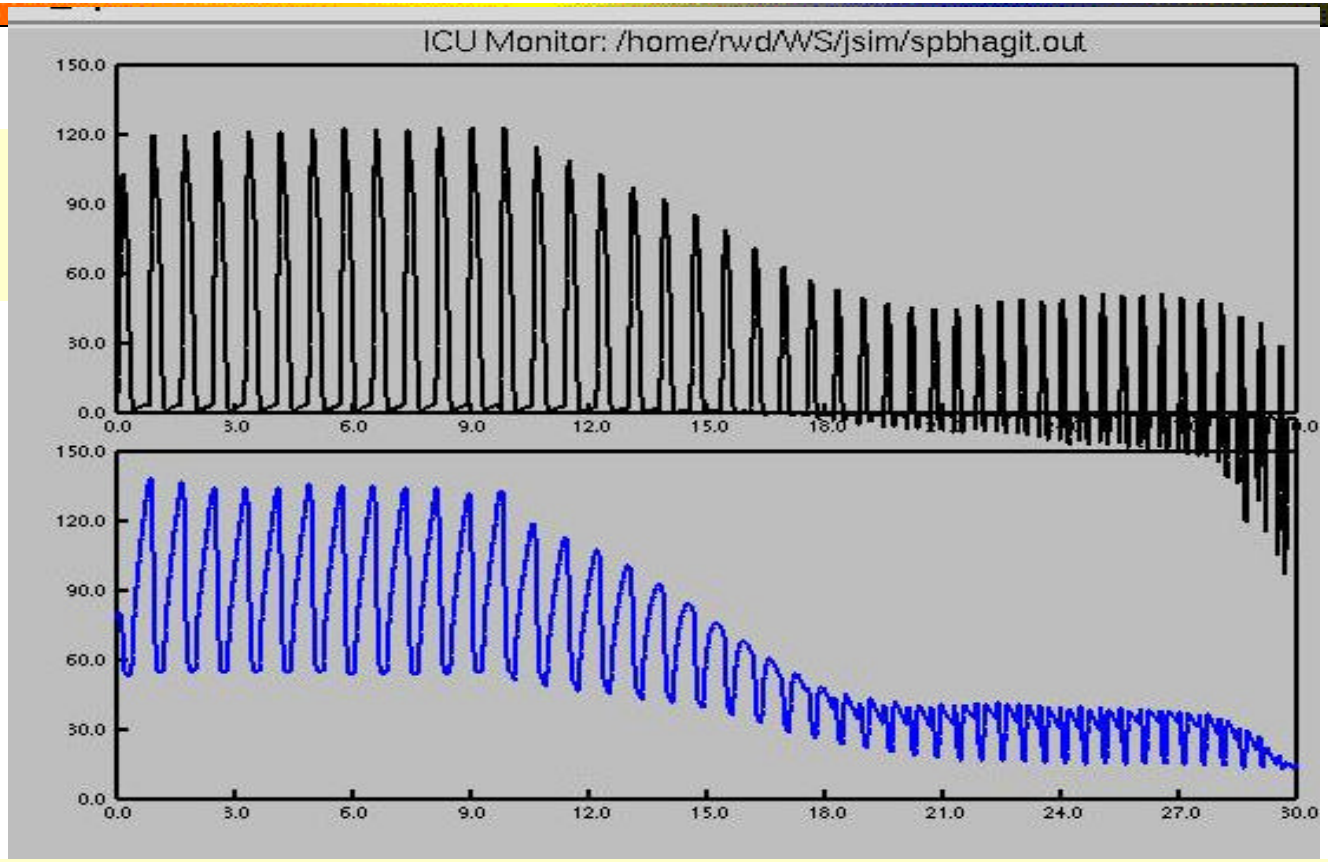


Time (min)

Output of HIP/JSIM model

Plv (mmHg)

Qlv (ml)

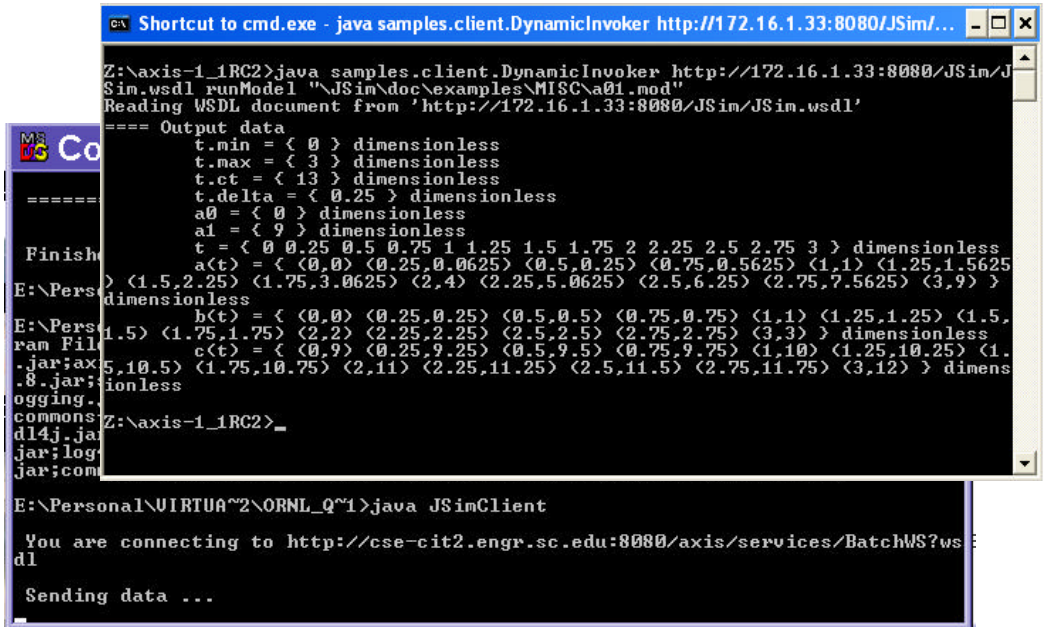


Left ventricle pressure (mmHg) & volume of blood (ml) for injury model (spbhagit) with a 2.5 mm hole in the left ventricle at 10s.

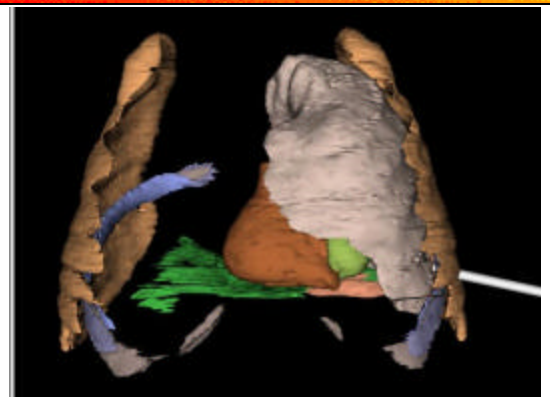
WS Integration of components

Initiating the service

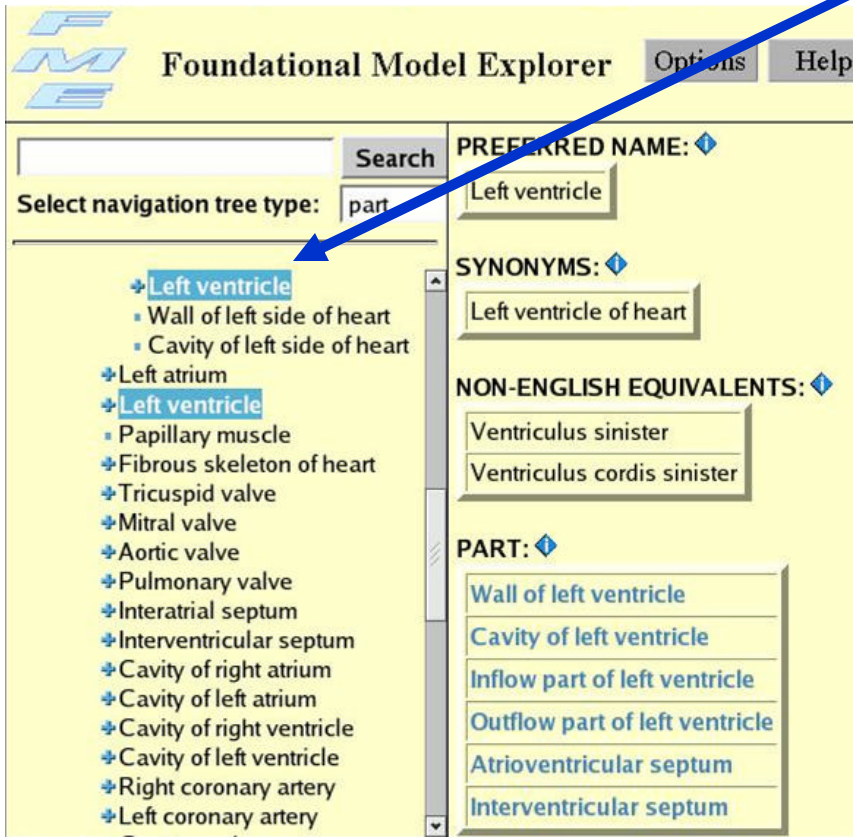
- Running the spheart model with a batch input file corresponding to an injury scenario invoking the JSim web service
- The batch input file also invokes a vtk visualization

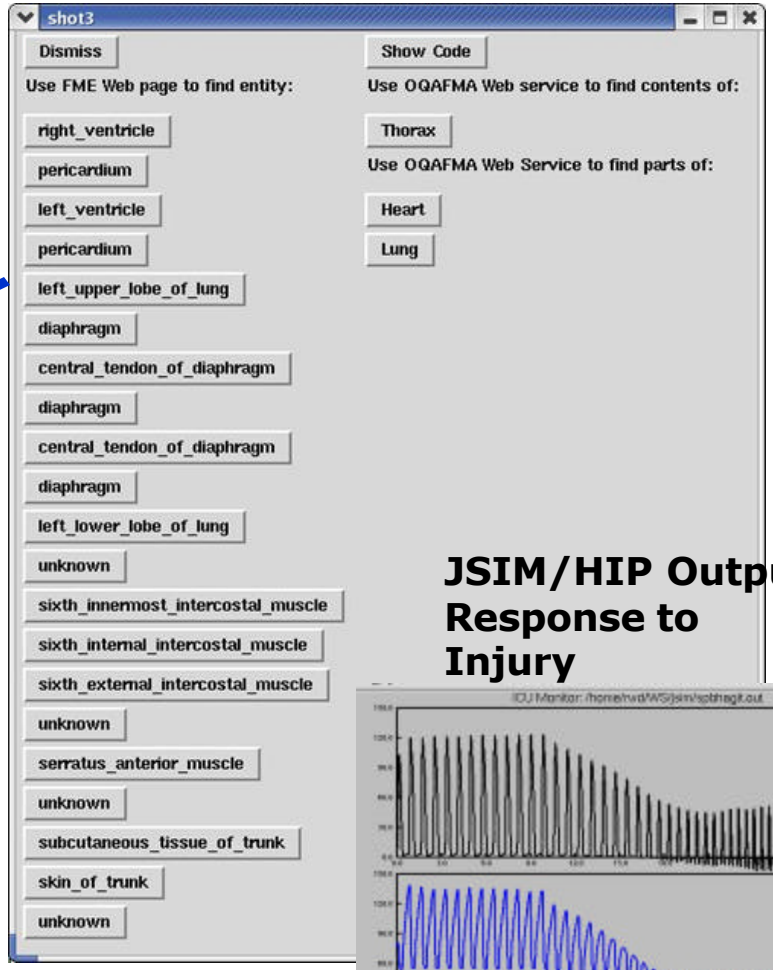


WS Integration of components

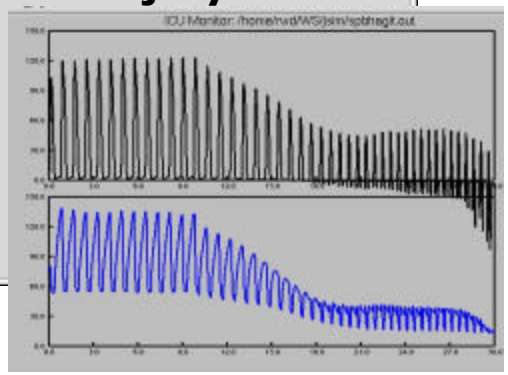


CT Label Map





JSIM/HIP Output Response to Injury



Performance of JSim WS

- Conditions
  - The Web services installed on a Web server located at USC
  - Compared WS to direct invocation of JSim for several values of the simulation parameters.
  - Measurements, which included latency and throughput, made both locally and remotely.
- Results show that use of Web services adds relatively small overhead.
- In some cases, running the batch program directly with JSim is slower than running it through WS.

Performance of JSim WS

- Each case involves running JSim for 30 seconds of simulation time using the spheart.mod model and a batch input file.
- 1.8 GHZ CPU, with 512 MB RAM.
- LAN for case #4 is 1 Gigabit Ethernet

- The Java VM has to be loaded into memory every time **jsbatch** runs, whereas Tomcat is always up and running.
- **jsbatch** will read from and write to files, whereas the data for the Web service resides in data structures in memory.

Running jsbatch only	**26.75 s**	**31.08 s**	**27.14 s**	
Running jsbatch wrapped as a WS	**23.89 s**	**22.15 s**		
Running jsbatch as a WS, from a client residing on the server machine	**31.173s**			
Running jsbatch as a WS, from a client residingon a remote machine	**29.30 s**	**32.28 s**		

Where do we go from here?

- Continue development of services (location, metadata look-up, VSP data retrieval).
- Refine the JSim web service based on componentization of JSim code.
- Integrate with SCIRun visualization environment.
- Build a grid portal for the Virtual Soldier Application (Phase II).

Acknowledgements

- DARPA
  - Rick Satava, MD, Program Manager.
- USArmy Medical Research and Materials Command
  - Gerry Moses, PhD.

- University of South Carolina, Center for Information Technology
  - Pr. Mike Huhns, Laura Zavala, Karthik Iyer
- University of Washington
  - Jim Bassingthwaighte
  - Cornelius Rosse, Daniel Cook (Structural Informatics)