Case Study: GiPSi – An Open Source / Open Architecture Software Development Framework for Surgical Simulation

M. Cenk Cavusoglu, PhD
cavusoglu@case.edu

Case Western Reserve University
Dept. of Electrical Eng. and Computer Sci.

http://vorlon.cwru.edu/~mcc14/
http://simcen.usuhs.mil/miccai2003

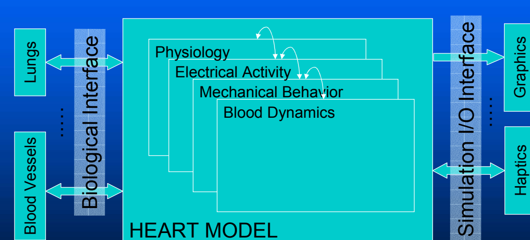
Project Focus

- Organ level simulation
  - Heterogeneous physical processes within the organs
  - Multiple organ interactions
  - Hierarchy of models
- Surgical simulation
  - Training
  - Preoperative planning
  - Intraoperative assistance

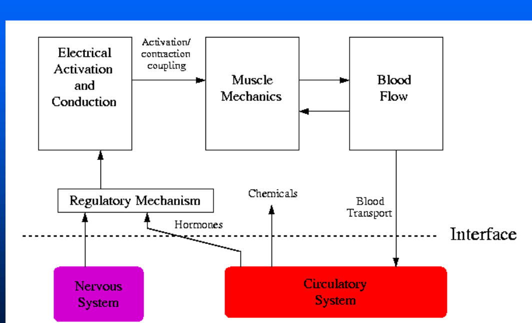
Technical Issues

- Abstraction
- Heterogeneous Physical Mechanism and Models of Computation
- Customization with Patient Specific Models
- Verifiability
- Modularity through Encapsulation and Data Hiding

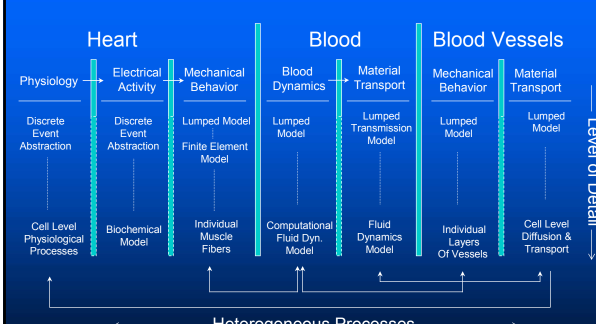
Test Bed: Heart Model for Surgical Simulation



Case Study: Heart Model



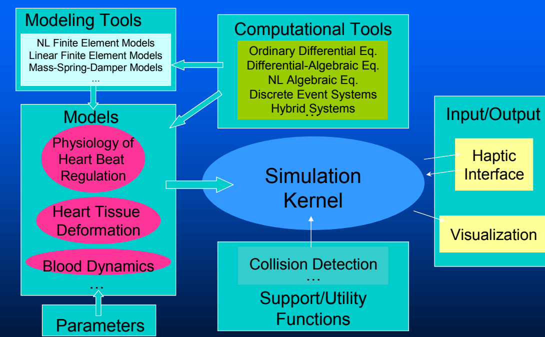
Heart Model



GiPSi – General Interactive Physical Simulation Interface

- An open source / open architecture software development framework for surgical simulation
- Define APIs, Implement selected models and tools
- Focus:
  - Support for heterogeneous models of computation
  - Interfacing between heterogeneous physical processes
  - Standardized I/O interfaces for visualization and haptics
  - Real-time interactive simulation applications
- Goals:
  - Least restrictive and most general APIs
  - Allow a variety of simulations

Simulator Architecture



Modeling Tools – Deformable Object Models

- Finite element (linear and nonlinear) and lumped element models implemented
- Finite element model*:
  - Geometric and material nonlinearities
  - Multi-grid integration
  - Adaptive mesh refinement using dynamic progressive meshes

*: FEM implementations adapted from Wu & Tendick (2003), and Wu, Goktekin, & Tendick (2001).

Modeling Tools – Deformable Object Models

- API
  - Geometry for computation
  - Separate boundary geometry for boundary conditions: meshless physical models possible
  - Boundary geometry also used for collision detection/response
  - Separate display geometry: display of information other than surface geometry possible
  - Local model for multi-rate haptic interaction: high fidelity haptic interaction
  - Material properties (stress-strain relationship) is isolated from the FEM model (strain calculations): Non-isotropic and heterogeneous materials possible

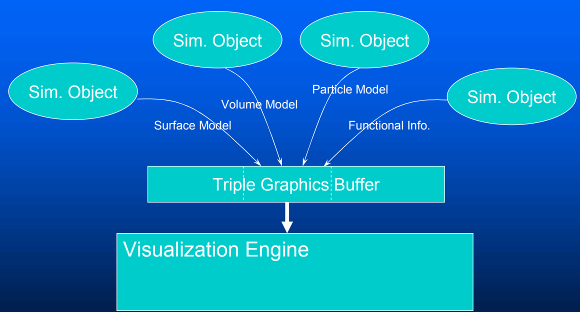
Modeling Tools – Fluid Models

- FEM based incompressible viscous fluid model (2D and 3D implemented)
  - Supports models with unstructured domains
  - Facilitates interfacing with models that have arbitrary boundary
- Stable
  - Advection solved using Semi-Lagrangian
  - Incompressibility imposed by Pressure Correction-Projection method
- Moving grids (ALE)
- API
  - Fluid / Solid Interface (in progress)
  - Fluid model accepts velocity boundary conditions from the solid/liquid boundary and returns force boundary conditions to the solid object

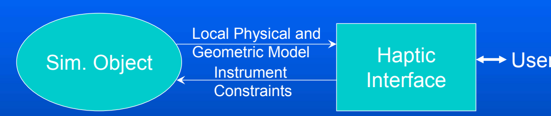
Support and Utility Functions – Collision Detection / Collision Response

- API
  - Each *collidable* object exports its boundary surface to Collision Detection (CD) Module
  - CD Module detects collisions and reports colliding pairs to Collision Response (CR) Module
  - CR Module computes the necessary penalty forces or the displacements and passes the results to objects as surface tractions or as Dirichlet boundary conditions
- Collision Detection
  - Exact or inexact
  - Current implementation is based on Axis Aligned Bounding Boxes
- Collision Response
  - Penalty Forces
  - Displacements

I/O Interfaces – Visualization



I/O Interfaces – Haptics



Slow update rate:
Global mechanical deformation computation.

Fast update rate:
Local collision detection and force feedback computation.

Multi-rate simulation for high fidelity and stable haptic interaction.

Computational Tools

- Object based implementation of Matrix and Vector operations
  - Uses BLAS for highly efficient linear algebra operations
- Solvers for linear algebraic equations
- Standard interfaces for explicit numerical integration
  - Number of numerical integration methods implemented (Euler, Midpoint, Modified Euler, Runge-Kutta (2,3,&4), Heun)
- Planned:
  - Discrete event systems
  - Hybrid systems
  - Differential-Algebraic equations
  - Nonlinear algebraic equations

Object API

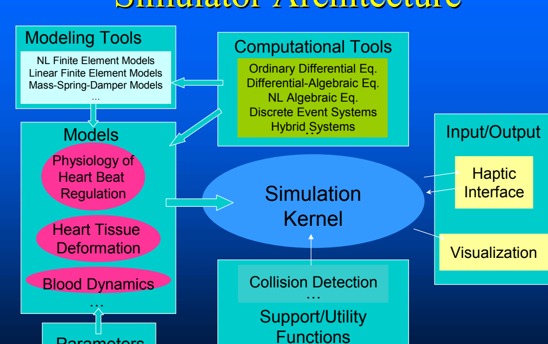
- API
  - Display Geometry
  - Physical Model(s)
    - » Computation geometry (mesh)
    - » Boundary condition geometry (surface)
    - » Display geometry (typically surface)
    - » Local model generation
  - Local haptic model
  - Integration

Sample Object Model

```
class Heart : SimObject {
  Geometry              HeartGeometry;
  NonLinearFEM          Muscle;
  LumpedFluidModel      Blood;
  …

  Integrate ();
  LocalHapticModel();
  Display();
}
```


Simulator Architecture



Team and Collaborators

- Prof. M. Cenk Cavusoglu (CWRU)
- Tolga Goktekin (UC Berkeley)
- Prof. Shankar Sastry (UC Berkeley)
- Prof. Frank Tendick (UC San Francisco)
- Dr. Xunlei Wu (UC Berkeley – now at MIT)
- Prof. Kathy Yelick (UC Berkeley)

References – General Surgical Simulation Frameworks

- M. C. Cavusoglu, T. G. Goktekin, F. Tendick, and S. S. Sastry. GiPSi: An Open Source/Open Architecture Software Development Framework for Surgical Simulation. In J. Westwood et al., editor, *Medicine Meets Virtual Reality (MMVR 2004)*, Amsterdam, 2004. IOS Press.
- S. Cotin, D.W. Shaffer, D. A. Meglan, M. P. Ottensmeyer, P. S. Berry, and S. L. Dawson. CAML: A general framework for the development of medical simulations. In *Proceedings of SPIE Vol. 4037: Battlefield Biomedical Technologies II*, 2000.
- K. Montgomery, C. Bruyns, J. Brown, S. Sorkin, F. Mazzella, G. Thonier, A. Tellier, B. Lerman, and A. C. Menon. Spring: A general framework for collaborative, real-time surgical simulation. In J. Westwood et al., editor, *Medicine Meets Virtual Reality (MMVR 2002)*, Amsterdam, 2002. IOS Press.

References – Physical Modeling and Simulation Frameworks

- J. T. Buck, S. Ha, E. A. Lee, and D. G. Messerschmitt. Ptolemy: A framework for simulating and prototyping heterogeneous systems. *Int. Journal of Computer Simulation special issue on Simulation Software Development*, 1994.
- A. Joukhadar and C. Laugier. Dynamic simulation: Model, basic algorithms, and optimization. In J.-P. Laumond and M. Overmars, editors, *Algorithms For Robotic Motion and Manipulation*, pages 419–434. A.K. Peters Publisher, 1997.
- Mathworks, Inc. Simulink. http://www.mathworks.com/products/simulink/.
- *Modelica — A Unified Object-Oriented Language for Physical Systems Modeling; Language Specifications 2.0*. The Modelica Association, 2002. http://www.modelica.org/.

References – Components of GiPSi

- M. C. Cavusoglu and F. Tendick. Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2000)*, pages 2458–2465, April 2000.
- F. Tendick, M. Downes, T. Goktekin, M. C. Cavusoglu, D. Feygin, X. Wu, R. Eyal, M. Hegarty, and L. W. Way. A virtual environment testbed for training laparoscopic surgical skills. *Presence*, 9(3):236–255, June 2000.
- X. Wu, M. S. Downes, T. Goktekin, and F. Tendick. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In *Proceedings of the EUROGRAPHICS 2001*, September 2001.
- E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, Jack J. Dongarra, J. Du Croz, S. Hammarling, A. Greenbaum, A. McKenney, and D. Sorensen. *LAPACK Users' guide (3rd ed.)*. SIAM, 1999.