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/

Project Focus

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

Simulation for Medical Training - MICCAI 2003

Technical Issues

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

Simulation for Medical Training - MICCAI 2003

Test Bed: Heart Model for Surgical Simulation

Simulation for Medical Training - MICCAI 2003

Heart Model

Simulation for Medical Training - MICCAI 2003
**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

---

**Modeling Tools – Deformable Object Models**

- Finite element (linear and nonlinear) and lumped element models implemented

- Finite element model:
  - Geometric and material nonlinearity
  - Multi-grid integration
  - Adaptive mesh refinement using dynamic progressive meshes

---

**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

---

\*FEM implementations adapted from Wu & Tendick (2003), and Wu, Goktekin, & Tendick (2001).
**I/O Interfaces – Visualization**

- Sim. Object
- Sim. Object
- Sim. Object

Triangle Graphics Buffer

Visualization Engine

**I/O Interfaces – Haptics**

- Sim. Object

Haptic Interface

Local Physical and Geometric Model

Instrument Constraints

User

- 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**

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

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

**Simulator Architecture**

**Models**
- Physiology of Heart Beat Regulation
- Heart Tissue Deformation
- Blood Dynamics

**Simulation Kernel**
- Collision Detection
- Support Utility Functions

**Computational Tools**
- Ordinary Differential Eq.
- Differential Algebraic Eq.
- NL Algebraic Eq.
- Discrete Event Systems
- Hybrid Systems

**Input/Output**
- Haptic Interface
- Visualization
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


References – Physical Modeling and Simulation Frameworks


References – Components of GiPSi