

BY MICHAEL SHERMAN

Putting Technology In Its Place

When you step on the gas pedal, you expect acceleration (and lots of it). Stomp on the brake to come safely to a stop in the rain. Finger the power-assisted steering wheel and the car obeys. Make a serious mistake and find yourself safely ensconced in deflating airbags. There is an immense amount of technology at your service as a driver, most notable for being practically invisible as you use it to get where you want to go. Each new generation of automobile is more

need to employ the immensely complex and ever-improving computational technologies of physics-based simulation. They want to step on the computational gas pedal and investigate biomolecular structure formation, neuromuscular gait disorders, or the turbulent flow of blood through the circulatory system. They need to get there fast, reliably, in reasonable comfort, and confident that inevitable errors will result in deployment of gentle safety devices rather than



Granny is off to the store.

To get to that level of simplicity and utility takes the intervention of vehicle designers who understand both drivers' needs and the available technology. Designers don't have to understand the

BOX 1. TYPICAL ABSTRACTIONS AVAILABLE TO VEHICLE DESIGNERS.

- Body, Chassis, Paint, Glass
- Engine, Transmission
- Electrical, Cooling, Safety
- Tires, Axles, Suspension
- Brakes, Steering
- Fuel, Exhaust, Emission Controls
- Security, Comfort, Entertainment



internal details of every technology; instead, they work with a different set of more sophisticated abstractions, such as those listed in Box 1. Similarly, for us to best serve the varied needs of biomedical researchers, it is the developers of biomedical application software we need to reach. They are the "vehicle designers" in our field, uniquely situated to understand both the needs of researchers in a particular biomedical domain, and the appropriate uses of physics-based simulation technology to serve those researchers. Application developers need to work with meaningful abstractions of our technology rather than its ever-changing intricate details. The applications they produce can in turn provide tightly focused interfaces presenting the abstractions that make sense to the people for whom they were designed. SimTK, the Simbios Toolkit (including Simbody, OpenMM, Molmodel, OpenSim Core, and other tools), implements such a set of technology abstractions for application designers, who can then effectively employ physics-based simulation technologies without having to be specialists in them. In turn, they hide these "under

sophisticated and complex than the last, chasing the ever-improving technological state of the art. Yet as a driver all you see is better transportation.

Would that software could be like that! At the Simbios Center, like all the NCBCs, we support researchers who are going important places in biomedical research. To get there, our "drivers"

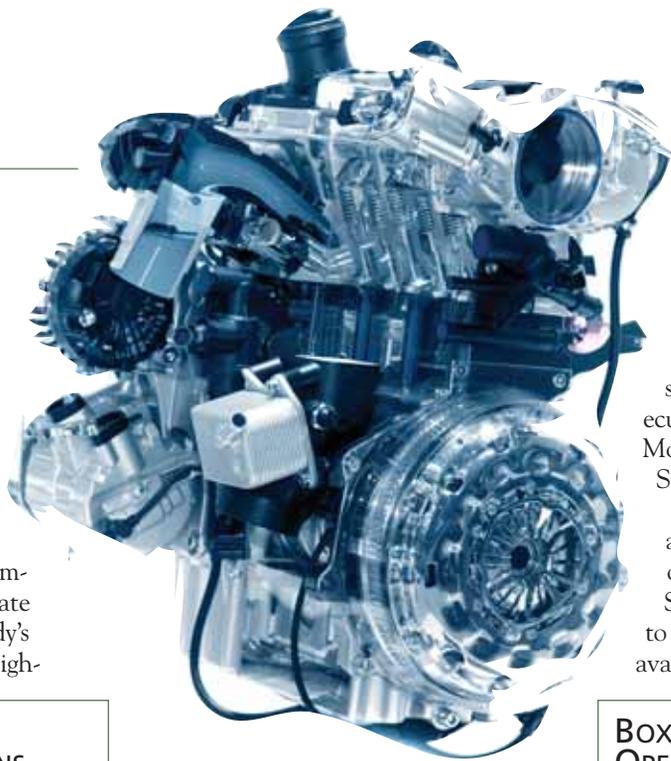
bloody career-mangling disasters.

In this column I'm going to talk about how we at Simbios approach the problem of delivering powerful, high-performance computational software to users who have better things to do than worry about what's under the physics-based simulation hood. The key to keeping technology in its place, whether for vehicles or software, is: *abstraction*. Users shouldn't interact with the technology itself but with suitable abstractions of that technology. In a car, the engine, transmission, and fuel system are reduced to a few amazingly simple concepts: ignition (on or off), gear (reverse and drive), and gas pedal (more and less). Add in similar abstractions for steering and brakes and

DETAILS

Michael Sherman is Chief Software Architect for the Simbios Center.

For more information about the Simbios Toolkit, go to <https://simtk.org/home/> {Simbody, OpenMM, Molmodel, OpenSim}



the hood” in their applications, to best serve their own research communities.

For example, multibody dynamics is a highly specialized discipline necessary for both musculoskeletal dynamics and internal coordinate molecular dynamics. Simbody’s abstractions (Box 2) enable high-

Box 2.
COMMON ABSTRACTIONS AVAILABLE TO APPLICATION DESIGNERS USING SIMBODY.

- **Vector, Matrix, Factor**
- **System, Subsystem, State**
- **Mass, Inertia, Geometry**
- **Gravity, Contact, Measure**
- **Time, Position, Force, Energy**
- **Body, Mobilizer, Constraint**
- **Integrator, Optimizer**
- **Time Stepper, Event**

performance multibody simulation of complex systems of interconnected articulated bodies at any scale.

Graphics Processing Units (GPUs) offer an immense amount of low-cost numerical computation hardware, with year-to-year speed improvements substantially outstripping CPUs. Yet programming GPUs for scientific computation is an exacting, time-consuming, and esoteric skill, differing among GPU manufacturers and even from version to version of the same product line. OpenMM’s abstractions (Box 3) allow the application developer to express molecular dynamics problems in familiar terms, while hiding the details of the particular hardware platform on which the problem is currently being solved. The result is dramatically accelerated molecular dynamics computations that take advantage of whatever GPU hardware is available, and can easily be incorporated into new applications and retrofitted into existing ones as we have done with Gromacs.

Application developers need to work with meaningful abstractions of our technology rather than its ever-changing intricate details. The applications they produce can in turn provide tightly focused interfaces presenting the abstractions that make sense to the people for whom they were designed.

Our higher-level modeling libraries for articulated skeletal and molecular systems also use abstractions to manage complexity (Box 3). OpenSim Core enables modeling and analysis of neuromuscular systems composed of abstractions like skeleton, muscle, tendon, and neurological control, using Simbody to calculate the articulated dynamics. Molmodel handles modeling of complex biopolymers as coarse grained articulated

systems with abstractions like molecule, bond, residue, and protein. Molmodel in turn employs both Simbody and OpenMM for speed.

There are excellent toolkits already available for many aspects of application development; Simbios was specifically chartered to tackle a substantial gap in the availability of suitable high-perform-

Box 3. SOME OPENMM, OPENSIM CORE, AND MOLMODEL ABSTRACTIONS.

GPU Acceleration (OpenMM)

- **Platform, Context**
- **Particle, Bond, Constraint**
- **Force, Integrator**

Neuromuscular Modeling (OpenSim Core)

- **Muscle, Attachment, Fiber, Path**
- **Controller, Activation**
- **Bone, Joint, Ground Reaction**

Molecule Modeling (Molmodel)

- **Element, Atom, Bond, Molecule**
- **Residue, Protein, RNA**
- **Force Field, PDB File**

ance, high-quality, open source toolkits for physics-based simulation of biological structures. SimTK addresses the computationally intense, physics-based simulation aspects of biomedical applications. These are extremely involved technologies, but surely no worse than the thousands of moving parts and dozens of computers that Toyota already packs meekly into a Prius. By putting meaningful abstractions of our technology into the hands of domain-knowledgeable application developers both within Simbios and in the community at large, we have made significant progress toward the goal of delivering them into the hands of biomedical researchers who can now navigate the open road ahead rather than struggle to get the engine running. □