

SimTK 1.0 Requirements Overview

Michael Sherman

Version 1.0, September 1, 2005

Abstract

In our planning for SimTK 1.0, SimTK staff produced a series of requirements documents setting out goals to be achieved in several areas. This document provides a quick overview of those goals and references to the more detailed documents. We conclude with a SimTK 1.0 timeline including tentative milestones.

1	Purpose of this document	1
2	SimTK 1.0 Projects	2
2.1	Infrastructure	2
2.1.1	Build System 1.0	2
2.1.2	Installation	3
2.1.3	Usage Tracking 1.0	3
2.2	Computation	4
2.2.1	Numerical Methods	4
2.2.2	Multibody dynamics	5
2.3	Modeling Layer	6
2.4	Applications	6
2.4.1	ISIM-SU	6
2.4.2	NMBL	6
2.4.3	ToRNAdo	7
2.5	Miscellaneous cross-threads	7
3	SimTK 1.0 Preliminary Milestones	7
3.1	Infrastructure	8
3.1.1	Build system	8
3.1.2	Installation	8
3.1.3	Usage tracking	9
3.2	Computation	9
3.2.1	Numerical methods	9
3.2.2	Multibody system (Simbody™)	10
3.3	Modeling	10
3.4	Applications	11
3.4.1	ISIM-SU	11
3.4.2	NMBL	11
3.4.3	ToRNAdo	12
	Acknowledgments	13
	References	13

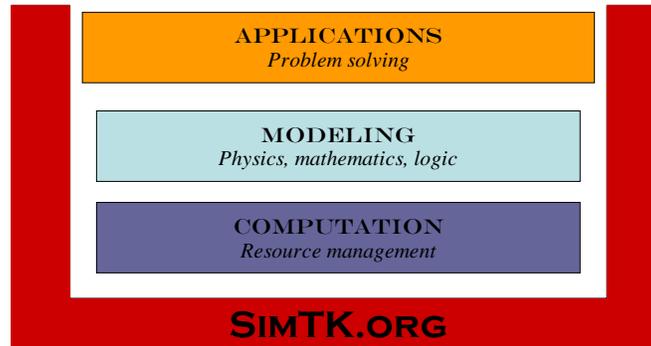
1 Purpose of this document

This document is intended to give a high-level view of the requirements we have set for the March, 2006 release of SimTK 1.0. It should serve both as a stand alone “executive summary” and as a guide to obtaining more detailed information for those who are interested. Note that we are discussing *requirements*, i.e., *what* we hope to achieve, rather than *specifications* which would cover *how* we intend to achieve our goals. We provide milestones at the end which sketch out our initial implementation strategy, but these are subject to change as we learn more.

For projects of this length and level of uncertainty, we will not produce more detailed specifications in advance of implementation. Instead we will work from the requirements, remain in close communication and make changes along the way, and then provide appropriate documentation of the “as built” systems at the end. Anyone interested in more information is invited to read the detailed requirements documents at <https://simtk.org/home/planning> and to talk to the developers and join our discussion lists.

2 SimTK 1.0 Projects

The overall SimTK architecture consists of four broad categories, represented in the figure below. For our purposes here we'll refer to the "SimTK.org" part here by the more general term "Infrastructure."



In SimTK 1.0 we hope to make progress on all four fronts. At this stage, infrastructure development is most critical and is a prerequisite to achieving SimTK's long term goals. Later releases of SimTK will have less infrastructure work and more direct content. Even for the non-infrastructure projects in SimTK 1.0, it is a major goal to employ the newly developed infrastructure so that they can serve as examples, test cases, and enticement to others to do the same later. We recognize that struggling with newly-implemented infrastructure may reduce the amount of functionality we can deliver in the allotted time, but we believe it is a worthwhile tradeoff at this point.

The Requirements Documents for all projects can be found in the SimTK "planning" project, on its home page <https://simtk.org/home/planning>.

2.1 Infrastructure

The projects in this category are Build System, Installation, and Usage Tracking. Our intention for 1.0 is to substantially complete the basic SimTK support structure for project development, delivery, and tracking. The other SimTK 1.0 projects will be converted to make use of this infrastructure as soon as it is available.

2.1.1 Build System 1.0

Source document: "SimTK.org build system" by Bryan Keller.¹

Note that we have been using Subversion instead of CVS for source control and found it very, very good. We have standardized on Subversion now and intend to use it as the only supported SimTK source code control system.

Requirements for SimTK 1.0:

- Build libraries, executables, documents
- Run tests and collect results
- Standardize and publicize build reports in "dashboard" form on the web site
- Support Mac/Windows/Linux and C/C++/Java/Fortran, giving thought to procedures for allowing interested contributors to add to this list
- This system will be in use for all staff projects. We will produce a draft of a document which could be used by others to learn how to use it also, although this will not be final by 1.0.

- Likely modeled after VTK/ITK build systems, meaning we will probably use CMake, CTest, and Dart.

2.1.2 Installation

A robust and easy to use installation system for SimTK projects is absolutely essential for us to be successful with our target user communities. This will be an area of substantial focus for us. We do not want to find ourselves years from now in the position of having built an incredible software system that no one can use because they can't get it operational on their own machines!

Source document: "SimTK installation system requirements" by Chris Bruns.²

Requirements for SimTK 1.0:

- Find, install, update, uninstall
 - SimTK apps, libraries, subsystems, data, documents, images, ...
 - Windows/Mac/Linux; compiled application binaries and Java apps
 - Source code/build are *not* part of installation
 - Plan for Perl/Python/TCL, but these will not be supported by 1.0
- Functionality like Java Webstart
 - Generalized to libraries & non-Java apps
 - One click, auto platform ID, recursive *SimTK* dependencies, updating
 - "Good neighbor" policy: system install or sandbox
 - Explore whether we can use Java Webstart directly
- Provide credit where it is due
 - Support individual projects' requirements for credit allocation
 - Three tier system: unrestricted/must register/intermediate
 - Ensure that SimTK contributors receive credit/acknowledgement for their work by displaying appropriate screens, obtaining commitments, or collecting information from downloaders
 - This will also serve as another enticement to get people to use SimTK and adopt our installation system
 - Note: this is *not* the same as licensing! Licensing terms create legal obligations which can be daunting even to decipher. These must be kept to a minimum if we hope for broad community adoption. Acknowledgements, tracking requirements, and so on are technological and moral obligations which do not require the hiring of attorneys to understand.
- This system will be in use for all staff projects. We will produce a draft of a document which could be used by others to learn how to use it also, although this will not be final by 1.0.

2.1.3 Usage Tracking 1.0

It is an NIH requirement (and good for us anyway) to be able to measure the degree to which our work is being put to use by others, especially outside of Stanford.

Source document: "Tracking and analyzing visits to the Simbios and SimTK websites" by Jeanette Schmidt.³

Requirements for SimTK 1.0:

- Cover SimTK.org only, within our GForge framework – Simbios.stanford.edu and BiomedicalComputationReview.org usage tracking will be dealt with elsewhere
- Collect everything now that we may need later, in a noninvasive and privacy-preserving manner
 - Unique and total visits, registered vs. not
 - Site/project/page/link/download granularity
 - OS/browser stats if possible
- Geographic information, esp. Stanford vs. outside users
- Easy graphical report generation by any interested party without special training (probably restricted to registered users)
- Will include posted SimTK policy statement and possibly posted user responsibilities as well

2.2 Computation

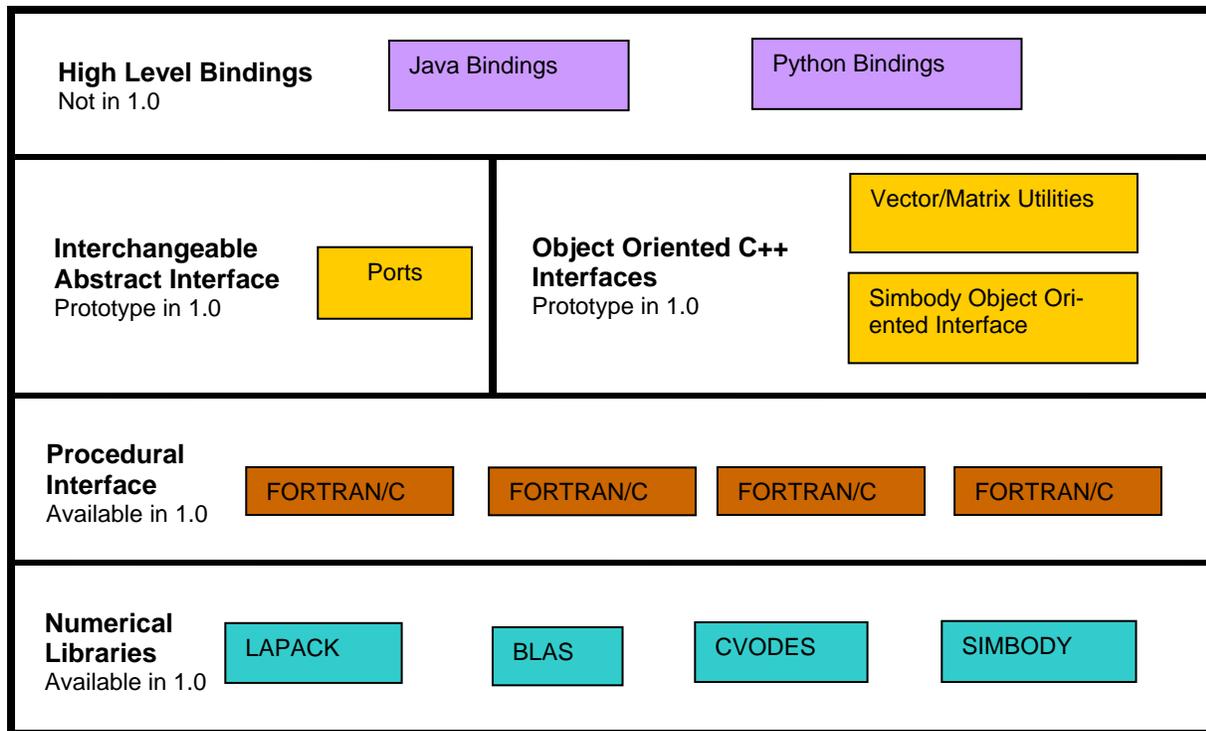
We will provide some basic numerical methods and a multibody dynamics engine. Although we expect these to be useful immediately, in SimTK 1.0 we are most interested in testing our ability to build, test, and deliver robust, high-performance computational methods on multiple platforms. This support infrastructure will be essential to all future computational projects and it is worth the early investment to ensure that this is solid, even at the cost of reducing the amount of numerical code we can deliver in 1.0.

2.2.1 Numerical Methods

Source document: “SimTK 1.0 requirements: numerical methods” by Jack Middleton.⁴

Requirements for SimTK 1.0:

- Linear algebra, ODE integration, multibody dynamics
 - Specifics: LAPACK/BLAS, CVODES, Simbody
 - Possible (in prototype form): root finders & optimizers, nice vector/matrix interface
 - Not in 1.0: PDEs, SWIG wrappers, CCA component discovery
- Multiprocessors: yes, clusters: not yet. Most likely this means only parallel BLAS for now.
- Platform-optimal binaries
 - Static and dynamic libraries plus prototype CCA components
 - Windows 32/64, Linux 32/64, Mac Tiger 32/64 (G4,G5)
- Extensive, automated test suites
- *Prototype* interchangeable APIs (Ports)
- Layered APIs (see diagram below)
 - Supports access at different levels depending on user comfort/preferences/needs.



Layered architecture for SimTK numerical methods

2.2.2 Multibody dynamics

A multibody dynamics (MBD) system is composed of bodies interconnected by joints, influenced by forces and restricted by constraints. MBD *does not* include force models, contact models, geometry, or numerical methods but does provide the hooks needed to use these in conjunction with the MBD system.

Source document: “Simbody™ (the SimTK multibody dynamics toolset)” by Michael Sherman.⁵

Requirements for SimTK 1.0:

- Cover needs of NMBL gait simulation and coarse-grained molecular dynamics
- For 1.0:
 - Forward and inverse dynamics; normal modes; internal ↔ spatial conversions (kinematics)
 - Stateless design with computation divided into stages
 - Layered procedural and object-oriented interfaces; prototype Port interface
 - SD/FAST replacement with compatible interface (imperfect)
 - Draft of user’s manual
- NOT for 1.0:
 - Deformable bodies
 - Analytic dynamic Jacobian
 - Domain-specific model-building support

- High-level wrappers

2.3 Modeling Layer

Here we expect to continue exploring what it will take to build an adoptable scheme for building models from interchangeable, sharable subsystems. This will lead to a second-generation prototype, with validation that some existing applications, especially the NMBL application, can be refactored into this form with a reasonable effort.

Source document: “SimTK Modeling Layer” by Ayman Habib.⁶

Requirements for SimTK 1.0:

- Create refined 2nd prototype
 - Must be usable by modelers
 - Must be adoptable, e.g. compelling benefits, high performance
 - Must enforce correct behavior, especially dependency checking
- Investigate chopping NMBL models into SimTK subsystems
- Prototype interchangeable subsystems
- Final report and draft user’s guide
- Provide some executable examples

2.4 Applications

We would like to have some useful, if narrowly targeted, end-user applications available in SimTK 1.0. In addition to addressing direct DBP needs, we hope to learn lessons and build tools here that will be useful for others in building future applications.

2.4.1 ISIM-SU

The goal here is to provide a user interface to Mark Engelhardt’s improvements to Nathan Baker’s ISIM.

Source document: “Requirements for ISIM graphical user interface” by Chris Bruns.⁷

Requirements for SimTK 1.0:

- Issues for 1.0:
 - Licensing
 - Usability, target users
 - App design
- Expected to be functional but not necessarily easily distributable by 1.0

2.4.2 NMBL

Source document: “NMBL Application Requirements for SimTK 1.0” by Clay Anderson, Ayman Habib, Pete Loan, Allison Arnold, May Liu, Ilse Jonkers, and Scott Delp.⁸

The basic goal for NMBLTS 1.0 is to have the ability to generate analyzable subject-specific simulations of both normal and pathological human gait. The workflow for doing this consists of the following steps: scale a model (Scale), solve the Inverse kinematics problem (IK), reduce residuals (RRA), apply computed muscle control (CMC), and run analyses (Analyses). That is, Workflow = Scale + IK + RRA + CMC + Analyses.

Requirements for SimTK 1.0:

- Migrate away from SIMM towards a SIMM-compatible open source application combining SIMM modeling and more recent work from Delp lab
- Includes subject-specific pipeline
- Note: there are real user demands that may affect the scheduling of this project independently of SimTK 1.0 needs

2.4.3 ToRNAdo

We will continue thinking about and planning for a ToRNAdo daughter application, at least at the beginning of the SimTK 1.0 development period. This may lead to a software deliverable in 1.0, but could also result in our choosing to pursue that in 2.0 instead. For example, the proposed pathway-morphing application could probably make good use of the SimTK Multibody (Simbody) facility, but that will not be available until near the end of 1.0.

Source document: “Requirements for Tornado daughter application #1” by Chris Bruns.⁹

Requirements for 1.0:

- Determine how best to proceed
- Explore collaboration with Mark Gerstein
- Is there a ToRNAdo/rigid body mechanics/Gromacs connection?
- Depending on investigation results, 1.0 deliverable may be a white paper rather than code

2.5 *Miscellaneous cross-threads*

The following 1.0 requirements overlap several projects:

- Build and installation should work together
- Appropriate binaries must be recognizable to the installation system
- Numerical libraries need to be usable from subsystems (e.g. stateless with staged computation)
- Targeted applications should contribute to general application-building toolset
- Capture SimTK best-practices in Coding Guidelines documents (these are kept in the SimTK “resources” project).
- Since many things will be incomplete at 1.0, we need to summarize what is left to be done in a document somewhere.
- Write a document capturing best practice in setting up a SimTK project, including policies regarding privacy (e.g., for Simbios-funded people public project/private source is the most privacy they get!).

3 SimTK 1.0 Preliminary Milestones

The SimTK staff philosophy for setting milestones is to choose the release date first, and then define the projects by that date rather than the reverse. So the release date below is firm, and the milestones (and thus the detailed contents of SimTK 1.0) will be changed as needed to have the 1.0 release on that date. So take this first set with a grain of salt.

We expect the SimTK 1.0 development period to last about 6½ months beginning Thursday, Sept. 1, comprising 4 months spent on new functionality, 1½ months integration with the new build & installation systems, and 3 weeks preparing for the SimTK 1.0 release and rollout presentations. (We're assuming 2 weeks of downtime in December.) The idea is to be completely done and on to planning for SimTK 2.0 by Tuesday, March 21, 2006. There will be one or more SimTK 1.0 presentations with demonstrations at about that time with details to be filled in later. To summarize:

- 9/1/05: Begin implementation
- 1/15/06: Functionality complete; no more new code; start integration
- 3/1/06: Integration with new build/install system complete; start rollout
- 3/21/06: SimTK 1.0 rollout presentation & demos

Individual project milestones follow. Please note that these are simply our initial plans and are expected to change substantially during implementation. We will maintain a spreadsheet elsewhere to reflect these changes while this document will record only our pre-planning state.

3.1 Infrastructure

Here are the milestones for the various infrastructure projects. All the other projects have milestones dependent on Build and Install infrastructure, so these will be completed in the order (1) build system, (2) installation system, (3) usage tracking.

3.1.1 Build system

Date	Milestone	Responsible party
9/1/05	Requirements document complete.	Bryan
10/1/05	Build system functional ⁽¹⁾	Bryan
11/1/05	Build system integration ⁽²⁾	Bryan
12/1/05–2/15/06	Move all staff projects to new build system.	Staff
12/1/05–2/15/06	As-needed build system support & tweaking.	Bryan
3/21/06	Demo.	

- (1) Ability to submit a build report to the server, View dashboards on the Web UI (most recent nightly dashboards), Edit dashboard settings from admin page on Web UI (define platform support, alerts, other). Example project that uses all aspects of the build system (compile, test, coverage, doc generation). Perform nightly builds on 3 platforms (if hardware is available) for the example project. Upload the 3 nightly build reports to server for the example project. Roll up the 3 build reports into a nightly dashboard for the example project. Email alerts of build failures to project admins for the example project.
- (2) Integrate at least one existing project with the build system. Testing and tweaking, beautification of UI. Deploy onto simtk.org. Documentation of build system for internal use.

3.1.2 Installation

Date	Milestone	Responsible party
9/1/05	Requirements document complete.	Chris
9/15	Determine feasibility of running non-Java apps from Java Webstart sandbox.	Chris
10/1	Platform detection system done.	Chris, Jack

10/15	Specify metadata format for all software products (version checksum, dependencies, etc.)	Chris
11/1	Usable application install in sandbox area done.	Chris
11/15	Dependency resolution system working.	Chris, Bryan
12/1	Update system working.	Chris, Bryan
12/15	Installation in system areas/backup of superseded content working.	Chris, Bryan
1/15/06	Registration system working.	Chris, Bryan
2/1	Uninstall process working.	Chris, Bryan
2/15/06	Installation 1.0 done, including first user manual.	Chris
2/15–3/1/06	Integrate all staff projects with install system.	Staff
3/21/06	Demo.	

3.1.3 Usage tracking

Note: because many projects have dependencies on Build & Install, we will make those our first priorities and start the Usage Tracking project during the time when staff members are integrating with Build and Install (between disasters!).

Date	Milestone	Responsible party
9/1/05	Requirements document complete.	Jeanette
1/15/06	Write text for privacy/user responsibility disclaimer.	Jeanette
2/15/06	Gathering appropriate data.	Bryan
3/1/06	Deployed on simtk.org; users can view statistics graphs from Web UI.	Bryan
3/7/06	Usage tracking 1.0 done; privacy disclaimer posted; documentation (if needed) written.	Bryan, Jeanette
3/21/06	Demo.	

3.2 Computation

3.2.1 Numerical methods

Date	Milestone	Responsible party
9/1/05	Requirements document complete.	Jack
10/1/05	LAPACK download/install & builds done	Jack
11/1/05	LAPACK regression tests, multiprocessor examples done	Jack
12/1/05	CVODES download/install & builds done	Jack

12/15	CVODES regression tests, multiprocessor examples done	Jack
1/15/06	Prototype of port definitions for linear algebra and ODE solvers done	Jack, Sherm
2/1	Prototype of vector/matrix utilities done	Jack, Sherm
2/15/06	LAPACK & CVODES integrated with SimTK 1.0 build system.	Jack
3/1/06	LAPACK & CVODES integrated with SimTK 1.0 installation system.	Jack
3/15/06	Numerical methods draft user guide done.	Jack, Sherm
3/21/06	Demo.	

3.2.2 Multibody system (Simbody™)

Date	Milestone	Responsible party
9/1/05	Requirements document complete.	Sherm
10/1/05	Separate atom-centered modeling from underlying rigid body mechanics.	Sherm
11/1/05	Functional C++ interface to underlying rigid body capability.	Sherm
12/1/05	Molecule modeling layer available.	Sherm
1/15/06	Procedural interfaces done. First cut at SD/FAST compatibility layer.	Sherm
2/15/06	Integrated with SimTK 1.0 build system. Some regression test cases installed.	Sherm
3/1/06	Integrated with SimTK 1.0 installation system. Draft user guide available. Simbody 1.0 done.	Sherm
3/21/06	Demo.	

3.3 Modeling

Caveat: Ayman's responsibilities to the NMBL group needs to take precedence over Modeling Layer activities since research and grant renewals may depend on progress in NMBLTS. Also, Ayman is on vacation most of December and early January.

Date	Milestone	Responsible party
9/1/05	Requirements document complete.	Ayman
10/1/05	Identify problems that will be formulated in SimTK modeling layer.	Ayman
11/1/05	Chop up NMBL problems into subsystem and	Ayman

	have preliminary interfaces documented.	
12/1/05	Modify modeling layer to achieve desired goals of simplicity and correctness.	Ayman, Sherm
2/15/06	Interfaces documented, build & test integrated with SimTK 1.0 build system.	Ayman
3/21/06	Demo.	

3.4 Applications

3.4.1 ISIM-SU

Note: there are some tricky licensing issues involving existing GPL-contaminated code which we will work on during the course of this project, but we do not have specific milestones defined.

Date	Milestone	Responsible party
9/1/05	Requirements document complete.	Chris
9/15	Interface for running APBS working.	Chris
10/1/05	Parameters for initial sets of ionic conditions created.	Chris
10/15	Interface extended to run ISIM.	Chris
11/1/05	First user manual published.	Chris
12/1/05	User testing outside of SimTK started.	Chris
2/15/06	Integrated with SimTK 1.0 build system.	Chris
3/1/06	Integrated with SimTK 1.0 installation system.	Chris
3/7/06	ISIM-SU 1.0 done.	Chris
3/21/06	Demo.	

Future plans; i.e., not for 1.0: interactive electrostatic molecular surface created; interactive typical ion environment visualization created; visualization of ion density created.

3.4.2 NMBL

For Delp lab research needs independent of SimTK, the goal here is to be able to execute the scale/IK/RRA/CMC/analysis pipeline by 1/15/06. The remaining time will be used for SimTK integration.

Date	Milestone	Responsible party
9/1/05	Requirements document complete.	Clay, Ayman, Pete
10/1/05	Create utility program to convert SIMM files (.jnt, .msl, .asc) into NMBLTS files (.xml, .vtk). Have code working to serialize and deserialize NMBLTS model with the following components: bodies, joints, coordinates, bones, muscles, constraint functions.	Pete
10/1/05	Complete CMC analyses (perturbation analysis)	Clay, Ayman

	with corrective springs) & have available for group use & experimentation.	
11/1	Have code working to output SD/FAST input file from NMBLTS model. Implement interface between NMBLTS model and SD/FAST-generated dynamics model, to enable running of forward and inverse simulations.	Pete
11/1/05	Complete and have evaluated the Residual Reduction Algorithm (RRA).	Clay, Ayman
12/1/05	Solve the scaling and IK problems for an existing SIMM Pipeline gait model. Have IK working with new kinematics engine.	Clay, Ayman, Pete
1/8/06	Complete NMBLTS models for muscle wrap objects & calculation of muscle moment arms.	Pete
1/15/06	Generate and analyze subject-specific simulations entirely in NMBLTS.	Clay, Ayman
2/8/06	Support for deform objects & constraint objects done.	Pete
2/15/06	Complete testing of the NMBLTS workflow for normal and pathological gait. Have devised and implemented at least one test of the workflow for the SimTK Dashboard. Integrated with SimTK 1.0 build system.	Clay, Ayman
3/1/06	Integrated with SimTK 1.0 installation system.	Clay, Ayman, Pete
3/15/06	Complete documentation for subject-specific simulations in NMBLTS, including Doxygen-generated API reference, user manual, several examples. NMBLTS 1.0 done.	Clay, Ayman, Pete
3/21/06	Demo.	

3.4.3 ToRNAdo

The scope of this project remains to be determined. Depending on the outcome (probably around 10/15/05) we may proceed as indicated below, change plans entirely, or choose to delay until 2.0.

Date	Milestone	Responsible party
9/1/05	Requirements document complete.	Chris
10/1/05	Determine roles & contributions from Gerstein & others.	Chris, others
10/15	Final requirements/schedule done.	Chris
11/1/05	Initial force field & molecular mechanics tool chosen.	Chris
12/15/05	Molecule trajectory created from interface.	Chris

1/15/06	Movie file created from interface.	Chris
2/15/06	Integrated with SimTK 1.0 build system?	Chris
3/1/06	Integrated with SimTK 1.0 installation system.	Chris
3/15/06	First user manual available; ToRNAAdo 1.0 done(?)	Chris
3/21/06	Demo?	

Not for 1.0: User testing outside SimTK.

Acknowledgments

I would like to thank the SimTK team members for diligently tackling the impossible but essential project of selecting the small set of “next steps” from an unstructured universe of possibilities, and for taking the time to document these choices. I believe this ability to focus on specifics (while keeping the big picture in view on the horizon), and commitment to open, transparent development, will be keys to our ultimate success.

This work was funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 GM072970. Information on the National Centers for Biomedical Computing can be obtained from <http://nihroadmap.nih.gov/bioinformatics>.

References

As an alternate to the possibly ephemeral links given below, the following documents (and this one¹⁰) can always be found on the SimTK planning project’s home page, <https://simtk.org/home/planning>.

¹ Keller, B.; SimTK.org Build System.

<https://simtk.org/docman/view.php/43/157/BuildSystemRequirements.doc.pdf>

² Bruns, C.; SimTK installation system requirements.

https://simtk.org/docman/view.php/43/158/simtk_installation_requirements.pdf

³ Schmidt, J.P.; Tracking and analyzing visits to the Simbios and SimTK websites.

<https://simtk.org/docman/view.php/43/171/TrackingWebsites.pdf>

⁴ Middleton, J.; SimTK 1.0 requirements: numerical methods.

https://simtk.org/docman/view.php/43/165/NumericalMethodsRequirements_1.0.pdf

⁵ Sherman, M.; Simbody™ (the SimTK multibody dynamics toolset).

https://simtk.org/docman/view.php/43/155/MultibodyDynamicsRequirements_1.0.pdf

⁶ Habib, A. and Sherman, M.; SimTK Modeling Layer.

https://simtk.org/docman/view.php/43/166/ModelingLayerRequirements_1.0.pdf

⁷ Bruns, C.; Requirements for ISIM graphical user interface.

https://simtk.org/docman/view.php/43/167/ISIM_interface.pdf

⁸ Anderson, C.; Habib, A.; Loan, P.; Arnold, A.; Liu, M.; Jonkers, I; Delp, S.; NMBL Application Requirements for SimTK 1.0.

<https://simtk.org/docman/view.php/43/170/NMBLApplicationRequirements.pdf>

⁹ Bruns, C.; Requirements for Tornado daughter application #1.

https://simtk.org/docman/view.php/43/168/tornado_morph.pdf

¹⁰ Sherman, M.; SimTK 1.0 Requirements Overview.

https://simtk.org/docman/view.php/43/169/SimTK1.0_RequirementsOverview.pdf