Proceedings of the 26th Annual International Conference of the IEEE EMBS
San Francisco, CA, USA • September 1-5, 2004

Development of Clinician-Friendly Software for Musculoskeletal Modeling and Control

R. Davoodi, C. Urata, E. Todorov, and G. E. Loeb
A.E. Mann Institute and Biomedical Engineering Department, University of Southern California
Los Angeles, CA 90089, USA, http://ami.usc.edu

Abstract: ***Research and development in various fields dealing with human movement has been hampered by the lack of adequate software tools. We have formed a core development team to organize a collective effort by the research community to develop musculoskeletal modeling software that satisfies the requirements of both researchers and clinicians. We have identified initial requirements and have developed some of the basic components. We are developing common standards to facilitate sharing and reuse of musculoskeletal models and their component parts. Free distribution of the software and its source code will allow users to contribute to further development of the software as new models and data become available in the future.***

I. INTRODUCTION

Both scientists and clinicians desire to understand normal and pathological movement in animals and human subjects in order to advance basic science and to devise and implement methods to treat and repair dysfunction. Studies of sensorimotor control of movement require measurements from neurons, muscles, and limbs in moving subjects, but these measurements are difficult and limited to only a subset of movement variables. Researchers are forced to make assumptions and inferences about what is happening in the unmeasured parts of the system. Mathematical models of the individual components and their interactions can provide an objective basis for these inferences. Their formal structure makes it easier to see what is really known and how accurately it is known rather than relying on subjective and perhaps erroneous assumptions. This benefit, however, usually requires many models of many components to be tested systematically in many different combinations in an integrating environment. Computerized models of neuromusculoskeletal systems can provide such an integrating environment. They can extend and complement experimental studies and provide a big picture view of the system while allowing the user to inspect and modify any model parameter to understand its role in the control of movement.

Developing realistic musculoskeletal models with the currently available tools is a difficult task requiring a great deal of effort and expertise not possessed by most researchers. The currently available software packages for musculoskeletal modeling have limited functionality, their use require high-level programming expertise, and they are expensive [1]. To overcome these limitations, many researchers have expended great effort to develop their own models for specific applications. But these models are usually developed in programming and simulation environments most familiar to the developer and are often difficult to share or reuse in other modeling studies even in the same laboratory.

In our laboratory, we have a tradition of developing generic musculoskeletal modeling software and sharing it freely with the research community. For example, we developed generic add-on software to complement and get around limitations of SIMM (the only commercial software for musculoskeletal modeling; available from Motion Analysis Corp., USA), but the combined package is still expensive, limited in speed and functionality and difficult to maintain.[1;2]. We believe that the availability of affordable and easy to use musculoskeletal modeling software is essential for advancement of research in control of human movement and its clinical applications, but this market is too small to attract commercial entities. Therefore, such software must be developed by the collective effort of the research community.

We have decided to organize this collective effort from our laboratory at Alfred Mann Institute (AMI) of the University of Southern California. We are motivated to lead this effort because AMI has pioneered the development and clinical application of BIONs™ [3;4]. This technology platform of wireless injectable stimulators and sensors can be applied to a wide range of sensorimotor disabilities, but only if clinicians have accessible and reliable software tools to fit them systematically to individual patients. Here, we will describe the procedures for specification of requirements and

0-7803-8439-3/04/$20.00©2004 IEEE

development and testing of the musculoskeletal modeling software (MSMS). We will then report on the progress to date and discuss ways in which the research community can get involved in this effort.

II. STATEMENT OF REQUIREMENTS

Specification of requirements is an important first step in which all the required capabilities and features of MSMS are discussed and documented before writing a single line of code. A series of meetings among the software developers and researchers from different fields has produced documents that describe the MSMS requirements in different levels of detail.

The first is the *vision document,* which is a high-level analysis of the key features that are needed to address the most critical problems. This high-level analysis justified the decision to develop MSMS because it showed there are no products on the market that satisfy the key requirements. It identified the main users (e.g. model builders, basic researchers and clinicians) and their key requirements such as the need for anatomically accurate models of individual patients, interfaces to models of feedback control systems, fast simulations for controller optimization, and real-time simulation for human-in-the-loop simulations and training exercises. We also collected examples of model components, user interfaces and description languages in order to learn from best current practices. It was concluded that MSMS must be open source to promote its adoption by the research community and ensure its continued evolution and growth in the future as the new models and data become available.

The next step in analysis of requirements is the *use case analysis.* Here, different scenarios such as "building a model" or "forward dynamic simulation" are analyzed. Each use case analysis identifies the preconditions, the primary users and stakeholders, and the goal of the analysis. It then describes the step-by-step sequence of actions for the main success scenario and provides remedies for all alternate scenarios such as failures.

In the last stage of the requirement analysis, important decisions on the *software architecture* are made based on the key requirements and use cases. The architectural block diagram of the MSMS is shown in Fig. 1. It has three top-level blocks: graphic user interface (GUI), Modeling and Simulation, and Database. Application program interfaces (API) are interleaved allowing any package to be replaced without requiring rewriting of the other packages. For example, multiple GUIs may be written to address various requirements of the users such as clinicians and other power users such as researchers.

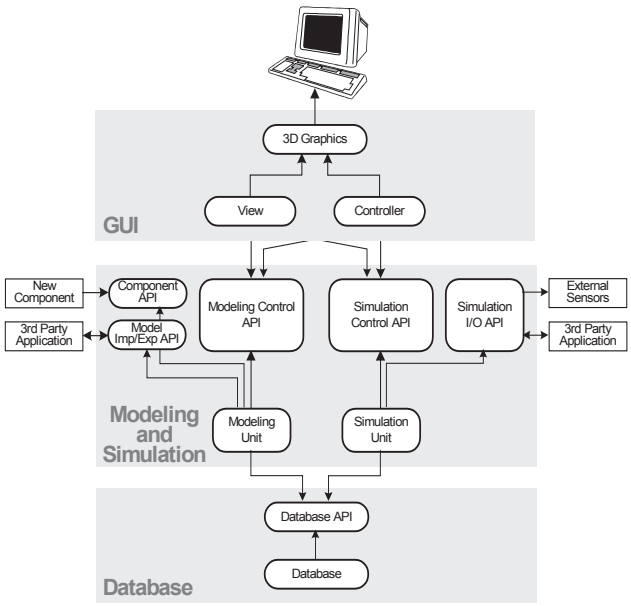

Figure 1. Architectural diagram of MSMS. Standard APIs, flexibility to add new components, interface to 3rd party applications and devices are among the main features.

III. DEVELOPMENT AND TESTING

We are using a method known as *iterative development* to develop and test MSMS according to the requirements. We use consecutive three-week iterations to add incremental functionality to the MSMS until all requirements are met. We test as we go, not in one push at the end as was the case in traditional software development. Each iteration produces an executable program that enables us to test the developed components for functionality and integration. As a result, we can discover design and integration problems early in the process, which are then easier to remedy.

Both manual and automated test methods are used to test and validate MSMS during and after development. In each iteration lower-level unit tests and system-level integration tests are performed. The repeated iterations thereby test each component's functionality and integration several times.

Another test-bed during the development is provided by ongoing musculoskeletal modeling projects in our laboratory that enable us to test the critical features of the MSMS on realistic problems. As the MSMS is developed, a diverse group of researchers who are consulting on the project (see Acknowledgement below) will test different features of MSMS by applying it to realistic problems in their own fields.

We are using Java as the main programming language to develop the GUI, the modeling unit and the database, but the computationally intensive simulation unit will be developed using more efficient environments. MSMS will allow the users to simulate their models within MSMS or export them to standalone simulation models in C or Simulink. Different simulation environments will serve the demands of different applications and user groups. For example, simulations in C will target applications requiring fast and real-time simulations and users with higher levels of programming expertise while simulations in Simulink will target academic researchers and applications that are not time-critical. To facilitate sharing and reuse of the MSMS models, we are using the eXtensible Markup Language (XML) to define the standard formats for MSMS models. These standard formats will be reviewed and refined by the research community.

IV. PROGRESS TO DATE

The implementation effort to date has resulted in a basic graphic user interface that can load and visualize musculoskeletal models developed in SIMM (Fig. 2). The loaded model can be graphically manipulated or animated by motion data from a saved file such as those produced by SIMM or streamed in real-time from a motion capture system. The main components now available include bones and muscle paths. Muscle wrapping around bony surfaces is currently modeled by cylindrical objects. We have also developed Matlab programs to test other wrapping objects based on the algorithms in [5], which are now being implemented in MSMS. In previous work, we have developed a library of highly realistic mathematical models of muscle excitation and contraction, tendon elasticity, and proprioceptive transduction; these and other components will be gradually added to the XML library.

We have performed a comprehensive test to select dynamic engines for dynamic simulations of the MSMS models [6].

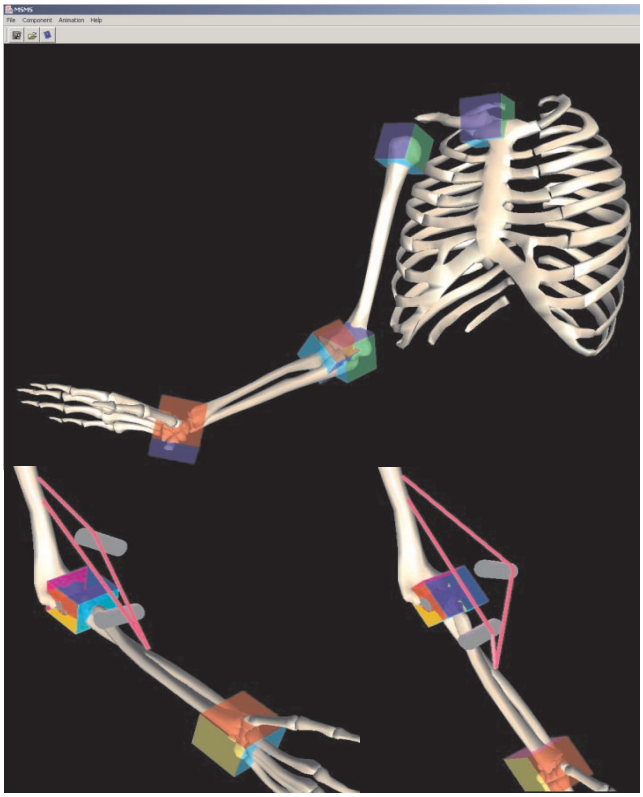

Figure 2. The graphic user interface of MSMS under development. Muscle wrapping around cylindrical surfaces are shown.

Sample models with open- and closed-loop topologies were used to benchmark the speed and accuracy of several free and commercial dynamic engines. The main criteria for selection were the speed and accuracy but features such as support for closed-loop topologies that are important in musculoskeletal systems, and implementation concerns such as ease of integration with MSMS code were also considered. The results showed that SD-Fast (Parametric Technology Corp., USA) is the best choice for fast and real-time simulations in C because it produces the most efficient code, provides many additional utilities and features, has a good collection of joint types, and could handle different constraints encountered in musculoskeletal systems. SD-Fast is also used by SIMM, but it is expensive and may not be the best choice for all users. Therefore, we are designing MSMS so that it can interface with other dynamics engines, such as the free Dynamechs [7] and the SimMechanics toolbox (Mathworks Inc., USA), which is designed specifically to work with the Simulink suite of numerical integrators.

V. CONTRIBUTIONS BY THE RESEARCH COMMUNITY

MSMS is a collective effort to develop a software tool that is essential for research and development in musculoskeletal modeling and control of movement. As such, community participation is essential for the success of the MSMS. Therefore, our number one priority is to involve the research community in all stages of MSMS development. To organize such a large effort, we have formed a core team of researchers and developers in our laboratory. This team is using the latest software engineering practices to organize the collective effort and to make sure that the completed software satisfies the requirements and is easy to use and maintain.

During the development, the core development team consults with a group of thirteen prominent researchers in different musculoskeletal modeling fields in their area of expertise. These researchers were drawn from the attendees of two preparatory conferences sponsored by the Mann Institute to define the need for and requirements of MSMS. These consultants define standards for musculoskeletal models and will test the alpha releases of the MSMS in their own applications.

The larger community will be involved in the development through a central web page. They will provide feedback on the standards for musculoskeletal modeling drafted by the development team and consultants and test the beta releases of the MSMS.

Once developed, the MSMS will be distributed freely to the public. The distribution will include the MSMS executable, its source code and the documentation. Procedures and guidelines will be provided to enable end users to expand the MSMS with new features as new models, methods and data become available. We think this is essential because rapid advances in movement science and software algorithms produce new models and data that must be incorporated into pre-existing models with minimal effort.

For the latest news and to participate in MSMS development, please visit MSMS web page at: http://ami.usc.edu/msms/

ACKNOWLEDGEMENT

We acknowledge the enthusiastic advice and support from the MSMS consulting team: Behzad Dariush, Francisco Valero, Garry Yamaguchi, Ian Brown, Marcus Pandy, Robert Kirsch, Scott Delp, Stefan Schaal, Steven Arms, Scott Selbie, and Victor Ng-Thow-Hing. Funded by Alfred E. Mann Institute for Biomedical Engineering, University of Southern California.

REFERENCES

[1] R. Davoodi and G. E. Loeb, "A Software Tool for Faster Development of Complex Models of Musculoskeletal Systems and Sensorimotor Controllers in Simulink," *Journal of Applied Biomechanics*, vol. 18, pp. 357-365, 2002.

[2] R. Davoodi, I. E. Brown, and G. E. Loeb, "Advanced modeling environment for developing and testing FES control systems," *Med Eng Phys*, vol. 25, pp. 3-9, 2003.

[3] G. E. Loeb, C. J. Zamin, J. H. Schulman, and P. R. Troyk, "Injectable microstimulator for functional electrical stimulation," *Med. Biol. Eng Comput.*, vol. 29, pp. NS13-NS19, 1991.

[4] G. E. Loeb, R. A. Peck, W. H. Moore, and K. Hood, "BION system for distributed neural prosthetic interfaces," *Med. Eng Phys.*, vol. 23, pp. 9-18, 2001.

[5] B. A. Garner and M. G. Pandy, "The Obstacle-Set Method for Representing Muscle Paths in Musculoskeletal Models," *Comput. Methods Biomech. Biomed. Engin.*, vol. 3, pp. 1-30, 2000.

[6] Montazemi, P. T., Davoodi, R., and Loeb, G. E. Comparison of dynamic engines for musculoskeletal modeling software MSMS. Proceedings of the American Society of Biomechanics Conference. 2004.

[7] S. McMillan, D. E. Orin, and R. B. McGhee, "A Computational Framework for Simulation of Underwater Robotic Vehicle Systems," *Journal of Autonomous Robots on Autonomous Underwater Robots*, vol. 3, pp. 253-268, 1996.