International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:5, 2008

Visualisation Techniques Connecting VRML and GENESIS Environments

Eduard Kuriščák, and Jiří Chludil

***Abstract*—**We created the tool, which combines the powerful GENESIS (GEneral NEural SImulation System) simulation language with the up-to-date visualisation and internet techniques. Our solution resides in the connection between the simulation output from GENESIS, which is converted to the data-structure suitable for WWW browsers and VRML (Virtual Reality Modelling Language) viewers. The selected GENESIS simulations are once exported into the VRML code, and stored in our neurovisualisation portal (webserver). There, the loaded models, demonstrating mainly the spread of electrical signal (action potentials, postsynaptic potentials) along the neuronal membrane (axon, dendritic tree, neuron) could be displayed in the client's VRML viewer, without interacting with original GENESIS environment. This enables the visualisation of basic neurophysiological phenomena designed for GENESIS simulator on the independent OS (operation system).

***Keywords*—** GENESIS, neurosimulation, VRML, Java3D.

I. Introduction

IT is about 20 years ago the software environment and computer language GENESIS (GEneral NEural SImulation System) was introduced at Caltech [1]. It was created mainly for the scientific purposes, but following years of its use and development proved, it is also suitable tool for visualization of neuronal processes in education [2], [3]. Despite its impact on neurosimulation community of that time and still up-to-date simulation language, the ongoing advance in visualisation, data processing, and internet technologies cause, the GENESIS do not offer the connectivity and flexibility needed for the current educational standards. It must be installed on the functional Unix-based system, and its installation is not intuitive as required for the today's educational applications. Although it is a very powerful tool for simulation and analyzing of various neuronal models [4]-[6], its control requires the tutors and students be familiar with Linux based systems and have nontrivial knowledge of GENESIS software environment. The distribution of GENESIS simulation packets to the client machines running GENESIS neither do not comply with the declared independence of the OS platform (GENESIS requires the Unix-based OS system) nor with the requirements for minimal installation, easy control and expected knowledge at client-side.

Therefore we decided to create a tool which combines advantages of GENESIS (modularity, structural language, object programming approach), with the capabilities of VRML (Virtual Reality Modelling Language). Our solution converts the simulation output from GENESIS into the VRML data-structure. Particular, it represents the distribution of electrical potentials of neuronal membrane in time and 3D space (along the neuronal membrane). The neuronal membrane (made up of patches of dendrites, axons and soma) is divided into the cylinders of various dimensions that are optimal for demonstration of propagation of electrical signal along the neuronal membrane. In the VRML viewer, the voltage at defined place of neuronal membrane is represented by corresponding colour. The 3D VRML representation of neuronal model (or its part), could be rotated, zoomed in or out and the speed of simulation could be adjusted according to the various requirements.

Manuscript received September 30, 2006. This work was supported by the grant FRVŠ/2006/345/F3d and partly by GAUK grant no. 203212 32/05 and GAČR grant no. 305/05/P198.

Eduard Kuriščák is with the Institute of Physiology, Charles University of Prague, Prague, Albertov 5, 128 00, Czech Republic (phone: 00420 608 611 835; fax: 420 224 918 816; e-mail: ekuri@lf1.cuni.cz).

Jirka Chludil is with the Department of Computer Science and Engineering, Czech Technical University of Prague (CTU FEE), Praha 2, Karlovo náměstí 13, 120 00, Czech Republic (e-mail: ChludilJ@seznam.cz).

II. Technical Background

To comply with the desired functionality outlined in the Introduction, we defined four requirements we tried to accomplish:

- preparation and configuration of the server hosting all applications and services needed for linking the GENESIS with appropriate VRML services
- platform independence: the visualisation of VRML data must be OS independent (operation system) on the client-side
- user-friendly control of presented simulations
- low traffic exchange between server and client machine

The information offered by our server consists of multimedia data (pictures, video sequences) and GENESIS simulations in the form of 2D and 3D VRML animations. Based on the server requirements and respecting the character of stored data, we decided to build the server providing mainly the WWW services. These are platform independent (there exist multiple web-browsers under various OS) and require minimal installation effort on the client-side. However, due to the nature of GENESIS development (it was created in programming language C and Unix-based systems) the

conversion of the GENESIS into the WWW environment is highly problematic.

The first solution to make the simulation environment of GENESIS fully accessible through WWW services resides in creating the java extension (java applet, java interface interface), communicating with GENESIS simulator on central server (Fig. 1). Using this applet the GENESIS would be fully controlled and simultaneously, the simulation results, of any form (visual or other output) would be in real time displayed by WWW services to the client.

The second solution assumes some services (user interface) based on the HTML, PHP or Java applet interface would communicate between the user at client-side and GENESIS at server-side (Fig. 2). The mentioned user interface passes the appropriate parameters to the GENESIS simulator. After the simulation is completed, the simulation output is converted into the animation of required form and displayed to the user.

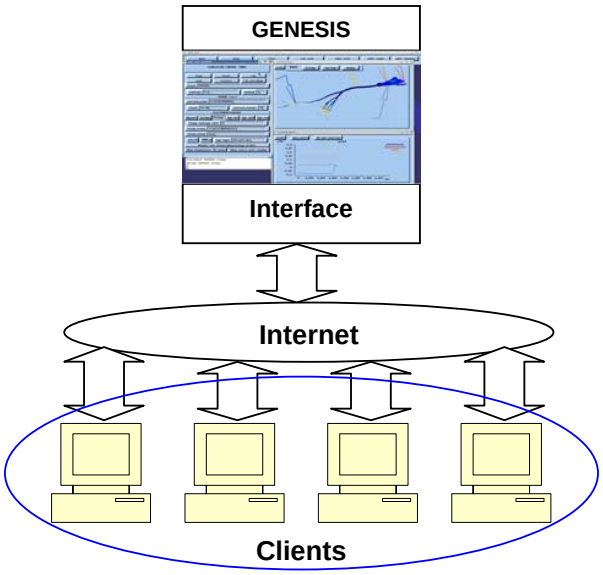

Fig. 1 Scheme of a possible solution enabling the fully transparent control of GENESIS. The depicted interface (Java applet) renders the users (internet clients working with WWW browser) full functionality of GENESIS located at server-side

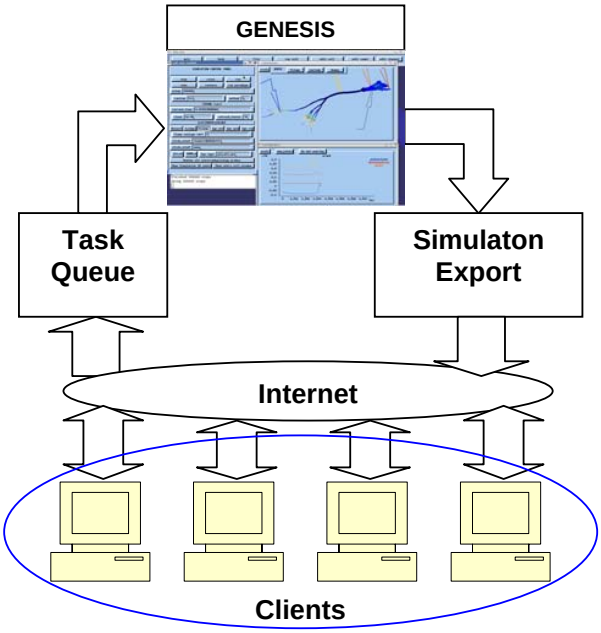

Fig. 2 The scheme of implemented solution. User requests are relayed to services (HTML, PHP or JAVA applet interface) processing task request queues. The simulation result is returned as 2D (video sequences) or 3D (VRML code) animations

Although the first solution is very elegant and transparent, in case more users simultaneously utilize the GENESIS computational resources, the offered services would be slow down and detoriated. The second solution is less transparent, but more users can utilize the computational resources of the server; by sending their simulation requests, the server decides whether it must be resimulated, specially, in case new parameters are entered. Otherwise, the simulation result (the 2D or 3D animation) is reloaded from a data-storage and sent to the user. Later on, even without the connection to the server, the downloaded animations will be useable at any time. We decided to implement the second methods, because of some advantages and easier implementation.

III. Implementation

Passing the simulation parameters using the HTML/PHP protocol and following management of request queue is the trivial programming problem. The export of GENESIS graphical output to the form viewable by WWW, specially by VRML, is much more complicated and desires some explanation.

A. Used Technology

1) VRML [7]

Virtual Reality Modelling Language (VRML) is a computer language designed for the description and representation of virtual scenes. It defines the graphical appearance of virtual scene and also the behaviour of individual graphical elements.

Thanks to the Avatar (the representation of user in a shared virtual reality), the users can freely move through virtual scenes and also interact with a virtual world (buttons, touch sensors). There are some software tools that can visualize virtual scene of VRML, called VRML viewers. The most common are Cortona or BlackSun, which are also built in the form of plug-ins into the common HTML viewers.

2) Vrmlworld Library [8]

This library serves the communication between the VRML scene and the Java code. It encapsulates a standardized programming interface called EAI [9] (External Authoring Interface - Interface between VRML and Java). Thus, the virtual scene may be controlled via Java application code in case of web browser hosting the VRML viewer. A typical web application consists of VRML browser window and additional controls in Java applet.

Vrmlworld is an object-oriented library with declarations of VRML nodes and variables. Each type of node in a VRML scene has a mirror (prototype of a Java class) in Vrmlworld library. Class initialization corresponds with a new VRML node creation or with a connection to existing named Node. Change of class attribute is represented by relevant response in a VRML node. It can also watch all changes in scene and update relevant attributes in Java classes. Vrmlworld has protection against above mentioned runtime errors. This solution enables simple access to the tree structure of VRML. In addition, Vrmlworld allows obtaining of hidden information about the scene e.g. unnamed child nodes. This library dramatically reduces the size of the source code in Java (to 25%).

3) Java3D [10]

Java 3D is a scene graph-based 3D application programming interface (API) for the Java platform. It runs on top of either OpenGL or Direct3D. Since the version 1.2, the Java 3D is developed under the Java Community Process. The window of Java3D could be very easily integrated into the HTML viewer as the Java Applet.

B. Export of GENESIS Output

With respect to our solution, each simulation process requires the mathematical representation of simulated model and the representation of simulation results. By the process of simulation we study the behaviour of model (e.g. neuron), which could be visualized in limited aspects by implemented graphical interface. Although a mathematical model could be exported (converted) to various simulation environments (softwares), in many cases it is relatively complicated problem. Moreover, such a process can detoriate some relevant characteristics of the model and influence its behaviour. The export of graphical output seems to be more transparent process, in case parameters of original model and simulation could still be manipulated by some external interface that is controlled by user. Depending on the extent these parameters are controlled, three approaches were considered:

The simplest way to export the graphical representation of simulation process is the recording of GENESIS visualization window and its conversion to the 2D animation. This approach omits the interaction between the user and various parameters of a model (only simulation speed can be changed).

The better, however complicated way requires some transformations of recorded output to the VRML data-structure. This approach improves interactivity between the user and simulated model (the model can be rotated, moved, zoomed in or out, the perspective can be changed).

The fully transparent approach would preserve the complete interference between the user and simulation parameters, but in our case imposes the undesired transformations changing the mathematical representation of simulated model.

For our purposes, respecting the optimal computational and traffic load on the server, the first two transformations were implemented in our solution.

1) Export to 2D animation

The graphical output in GENESIS window is recorded by available utilities (the tool HyperCam). Consequently the animation in form of mpeg, avi or flash is created.

2) Export to 3D

The transformation of the simulation output to the 3D representation is realized using the two graphic engines VRML and Java3D. The VRML (Virtual Reality Modelling Language) and Java3D are fully functional in the WWW environment.

The transformation to the VRML is based on a model approximation utilizing basic graphical primitives (in our case the cylinders of various dimensions). Each primitive is assigned the time sequence representing the voltage course at particular place of neuronal membrane in time. The sequences belonging to the corresponding primitives are passed to Java environment, which by means of VrmlWorld library manipulates the model in VRML viewer. The advantage of this approach resides in using the VRML viewer which is fully implemented in WWW browser.

The transformation to the Java 3D do not require the approximation by graphical primitives, therefore the same graphical definition of graphical elements as in GENESIS is used. However, there do not exist apparent support for the manipulation with 3D objects as it is common in the VRML. This feature must be implemented in the future.

IV. Conclusion

Our solution extends the access to the GENESIS simulator and its resources (simulation capabilities, programming language), without the need for mastering the original GENESIS environment. On the side of the client it requires only a functional WWW browser connected to the internet. On the side of the server the user requests are processed, sent to the GENESIS simulator, and returned to the client in form

of 2D (video sequences) or 3D (VRML sequences) animations. The animations could be stored at server-side for later access, and at any time reloaded without interacting with GENESIS computational resources.

REFERENCES

[1] J. M. Bower, D. Beeman, *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System*. New York: Springer-Verlag, 1994

[2] D. Beeman, "Simulation-based tutorials for neuroscience education" *Computation in Neurons and Neural Systems.* F. Eeckman, Ed., Kluwer Academic Publishers, 1994, pp. 65-70.

[3] The main GENESIS simulator page. www.bbb.caltech.edu/GENESIS

[4] E. De Schutter, J. M. Bower, "Purkinje neuron simulation on the Intel Touchstone Delta with GENESIS". *Proceedings of the Grand Challenge Computing Fair.* T. Mihaly and P. Messina (eds). CCSF publishing, Caltech, Pasadena, CA, 1992, pp. 268-279

[5] R.D. Traub, G.R. Jefferys, R. Miles, M.A. Whittington, K. Toth, "A branching dendritic model of a rodent CA3 pyramidal neurone" *Journal of Physiology* 481.1: 1994, pp 79-95

[6] E. Kuriščák, S. Trojan, Z. Wünsch, "Model of spike propagation reliability along the myelinated axon corrupted by axonal intrinsic noise sources" Erratum in *Physiol Res* 51(3), 323. *Physiol Res* 51(2), 2002 pp. 205-215

[7] The Virtual Reality Modeling Language. International Standard ISO/IEC 14772-1:1997. http://www.web3d.org/technicalinfo/specifications/vrml97

[8] J. Chludil, J. Žára, „Nautilus - The Environment for Training and Testing" *Proceedings of the Sixth IEEE International Workshop on Distributed Simulation and Real-Time Applications*. Los Alamitos: IEEE Computer Society Press, 2002, pp. 134-139.

[9] The Virtual Reality Modeling Language External Authoring Interface. Committee Draft ISO/IEC 14772-2, EAI is now under final vote for approval by ISO as VRML 97 Amendment 1, Part 2.) http://www.vrml.org/WorkingGroups/vrml-eai/Specification

[10] TSun's Java3D Homepage http://java.sun.com/products/java-media/3D