A Web Based Tool for Integration of Molecular Pathway Models

Eva Sciacca, V. A. Shiva Ayyadurai, C. Forbes Dewey, Jr.

Abstract— **Developing complex models of cellular function requires the collaboration of multiple teams of researchers remotely distributed worldwide. A challenge of computational systems biology is to find easy and accessible mechanisms to enable such collaboration to construct higher level models of cellular function.**

This paper presents the development of an on-line web portal for enabling open access to Cytosolve, an existing, proven and scalable computational architecture for integrating quantitative molecular pathways. The developed graphical user interface allows ease-of-use for developers of quantitative molecular pathway models to remotely collaborate to build larger and more complex models using the Cytosolve infrastructure.

The on-line web portal will be accessible and it will allow users to remotely collaborate with the existing Cytosolve computational environment that supports integration of models in a parallel manner without geographical restrictions. A creator of a model will be able to integrate their model from their local location to an ensemble of distributed models through this on-line web portal.

I. INTRODUCTION

A grand challenge of Systems Biology is to model the whole cell. A cell consists of a set of organelles that interact to provide cellular functions such as protein synthesis, metabolism, apoptosis, or motility. Systems Biology aims to develop a model of the cell by connecting the biochemical kinetics of these interactions at the molecular mechanistic level to derive the quantitative descriptions of higher level cellular functions [12].

There is a worldwide movement in the computational systems biology community to find powerful ways to integrate the growing number of biological pathway models. However, the current approaches do not provide flexibility and scalability in integrating multiple models [3].

The open-access on-line web portal discussed in this paper is based on the scalable computational architecture of Cytosolve which allows the integration of an ensemble of distributed biological pathway models [1]. The Cytosolve architecture has been validated on well known biological models in the systems biology community by comparing solutions obtained with the existing approaches. Moreover, new integrative models, such as the interferon (IFN) response to virus infection, were developed using Cytosolve.

Cytosolve architecture [2] directly addresses the integration and scalability problems of coupling multiple biological pathways by providing a parallel and distributed architecture. Individual pathway model can be expressed in any one of a number of formats (SBML, CellML, MML, etc.) and be computed on different computers. The architecture removes the need to manually load, understand and interconnect each individual pathway into a single monolithic program as is required in other systems.

In order to open access to the broader community of researchers seeking to collaborate and integrate remote and distributed biological pathway models it is essential to design and implement a web based GUI for worldwide use. The goal of this research was to provide the Cytosolve computational architecture with a Web-enabled GUI.

II. ARCHITECTURE

Current architectures used for integrating biological pathway models are essentially based on two methods. The first method proposes to use direct computation to solve the problem, i.e. developing a program from scratch for each set of coupled reactions. The second method, the *monolithic approach*, takes individual component models in a single supported mathematical syntax such as SBML and manually integrates them to create one monolithic software program. A variation on this approach is to use semi-automation tools that help to automatically read and integrate source codes together to create one monolithic software program.

Currently the most common architectures, such as Cell Designer [9], Jarnac/JDesigner [17] or Gepasi [16], use the *monolithic approach*. This approach presents some drawbacks such as difficulty in scaling to large numbers of models. They also do not support multiple language standards without conversion to a single format. All models have to reside in the same geographical location, and they require that the person integrating the models is intimately acquainted with all the multiple pathways to be merged.

The computational architecture used by Cytosolve is based on the *dynamic messaging approach*. The dynamic messaging approach implies that the models remain independently-executing programs that interact only by exchanging data via message passing during execution. Architectures that use this approach can include or not an independent application (a controller) that mediates the execution and messaging between the models.

The logical software architecture of Cytosolve (shown in Figure 1) provides a Controller, a Presentation and Com-

This work was supported by the MIT-Italy Program 2007/2008 and the Singapore-MIT Alliance Program.

E. Sciacca is the corresponding author. She is with the Department of Mathematics and Computer Science, University of Catania, Viale A. Doria 6, I 95125 Catania, Italy sciacca@dmi.unict.it and with the Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307 , U.S.A. sciacca@mit.edu

V. A. S. Ayyadurai and Professor C. F. Dewey, Jr. are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307 , U.S.A. {vashiva,cfdewey}@mit.edu

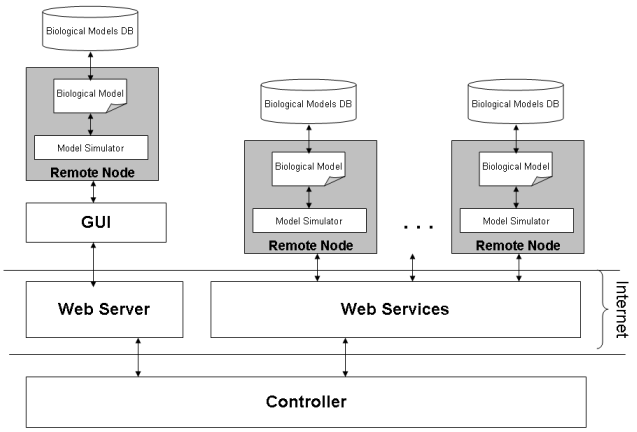

Fig. 1. Generalized Logical Architecture of Cytosolve

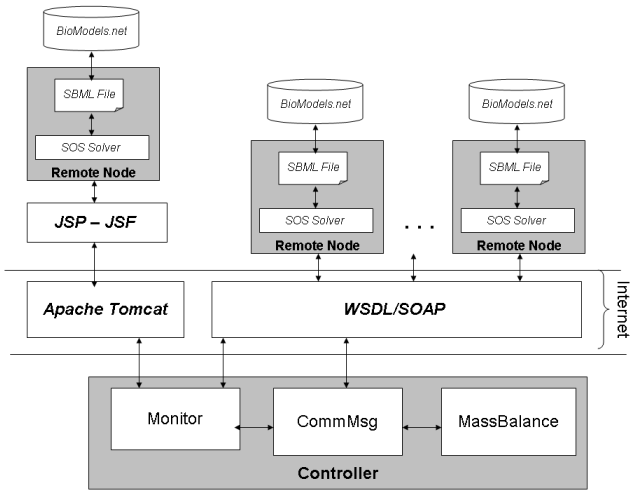

Fig. 2. Implementation of the Logical Architecture of Cytosolve

munications Layer that includes a Web Server with its Graphical User Interface and the Web Services, and finally the Remote Node machines. In this work, we developed a new Graphical User Interface (GUI) to exploit previous architectural components.

The *Controller* mediates the execution and messaging between the models. The primary role of the controller is to transform exchanged data, which typically involves data type conversions, but the controller can also control the startup of the models and track the global state of the integrated model as well. Architectures that do include a controller have messaging libraries that support direct model-to-model messaging as well as model-to-controller messaging.

The *Presentation and Communications Layer* provides the implementation of the dynamic messaging approach to support communications through the Internet network among the remote nodes which run the simulation of the models as well as with the User Interface.

The *Remote Node* machines involve the presence of a model simulator that communicates to the Controller through the Web Services. Each Remote Node includes a biological model as an input to the simulator. The biological models can be downloaded from a biological models database, such as BioModels Database [14], or can be produced locally by a scientist and have never been published or made publicly available.

The *Web Services* are designed for remote model simulations. During a run, the web service can be instructed by a remote computer to perform two major operations. First, the web service can be instructed to simulate a local model over a single time step. After the simulation, the service sends back new concentration values calculated by the model. In the second operation, the web service can be instructed to insert new species concentration values into the model simulation. This allows external control of the simulation. Using these two operations, the centralized controller can couple multiple models together, with each model running on different computers.

III. IMPLEMENTATION

The CytoSolve Web application was implemented using open source tools to reduce expense and to ensure that future researches could be pursued with minimal reliance on proprietary tools. Figure 2 shows the implementation of the Logical Architecture of Cytsolve illustrated in Figure 1 for the particular case of using SBML-based models.

The Controller of Cytosolve consists of three main components : the *Monitor* that serves to track the progress of each pathway's solution time, the *Communications Manager* (Comm Mgr) and the *Mass Balance* algorithm. The Controller serves two purposes. First, by means of the Comm Mgr, it mediates communication across all pathway models. Second, using the Mass Balance algorithm, it provides computational steering by ensuring mass conservation across all integrated models for each time step.

The Web Services were implemented using the Simple Object Access Protocol (SOAP) [4]. This XML-based messaging format established a transmission framework for inter-application (or inter-service) communication via HTTP. SOAP is a vendor-neutral technology thus it provided an attractive alternative to traditional proprietary protocols, such as CORBA or DCOM. The Web Services Description Language (WSDL) [7] supplied a language for describing the interface of the web services.

The Web Server software in use is Apache Tomcat. The Graphical User Interface (GUI) was developed by means of the JavaServer Pages (JSP) [20] technology that is based on the Java language and that enables the development of dynamic web sites. JSP are supported with Java Server Faces (JSF) that provides tag libraries with custom actions for representing the standard UI components in the JSP pages.

The model simulator on the Remote Node machines is the SBML ODE Solver Library (SOSlib) [15] that is a programming library for symbolic and numerical analysis of chemical reaction networks. This library takes as input model

files encoded in the Systems Biology Markup Language (SBML) [11] and computes the steady-state solution of species concentrations for a given number of time steps. Other solvers supporting CellML [8], MML [6], and other pathway description dialects can be used interchangeably with SOSlib. Even Matlab programs have been used.

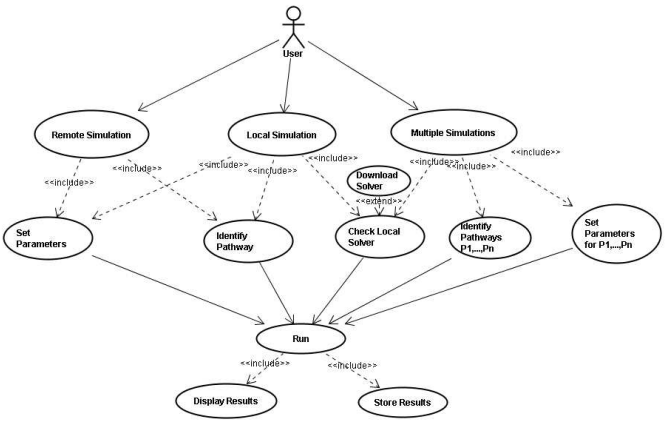

Fig. 3. Use-Case diagram of the Cytosolve web application

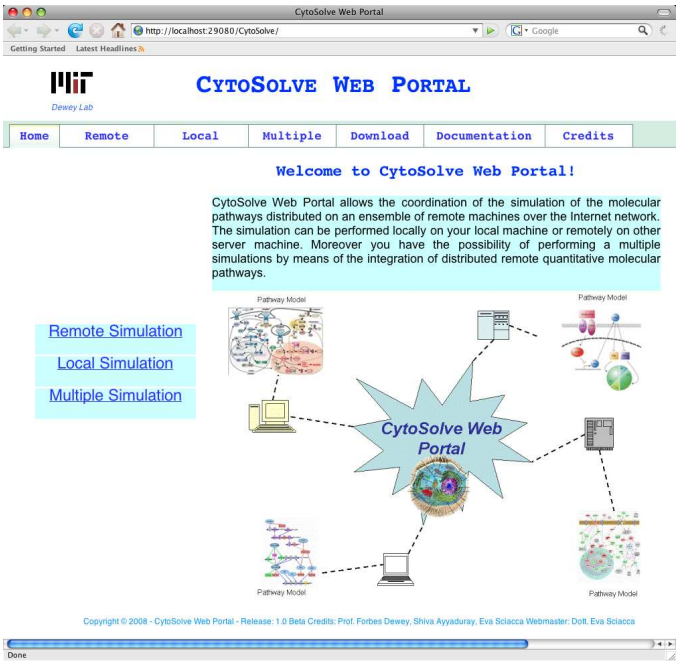

Fig. 4. Home page of the CytoSolve web application

Figure 3 illustrates the use-case diagram of the web application. A User from a Remote Node machine, by means of the GUI and the interaction with the Web Server, can:

1) Download the version of the model simulator to simulate a model on the local machine;
2) Submit a model to the Web Server and simulate it remotely;
3) Download the version of the model simulator to simulate a model locally coupling this simulation with the other remote nodes machines on the network; and
4) Submit a model to the Web Server and simulate it remotely coupling this simulation with the other remote nodes machines on the network.

Figure 4 shows the home page of the web application.

A remote user by means of the web GUI is driven step by step on the simulation (see Figure 5). When the topology of simulation is chosen, the user will be asked to submit the model (or the models in the case of multiple simulations) that should be a file resident on the local machine, a URL linking to a specific model or a model inside the database on the remote server. The local simulation requires that a solver is installed on the machine, if it is not available, the user will be asked to download it. The local solver is also needed when a multiple simulation is chosen and the user wants to simulate a model locally coupling the simulation with other remote node machines on the network. After the setting of the parameters for each model, the simulation can be executed and the results will be displayed and stored.

The local simulation of the model implies a use of the local computational resources on the Remote Node machines. An advantage is that it ensures protection of proprietary models (models where the source code is inaccessible). Many pharmaceutical companies, for example, will not want to share the inner source code of their particular proprietary models; however, they are interested in coupling their models with other models to gain better understanding of a larger cellular process. On the other hand the black-box simulations of the models allow researchers to integrate Public models with existing Proprietary models to learn some new aspect of science, without violating confidentiality issues.

The biological pathway models can reside anywhere geographically on the planet because the world wide web is used for communication with the controller. One can decide to have all models centralized on one computer and the computational environment remains unchanged; the individual models can be run as separate processes and the same infrastructure will support the local communication between each model.

IV. METHODS

A. Mass Balance

The Mass Balance serves to provide the calculation of species concentration for each time step n across the ensemble of M models. Each model was treated as a black box with the input and output being a vector of species concentrations denoted by the following two variables:

$$S_n^{j,i}$$

which denotes the species concentration at time step n, of the ith model and the jth species, and

$$S_{n+1}^{j,i}$$

which denotes the jth species concentration at time step n+1, of the ith model. A new variable

$$S_{g,n}^{j}$$

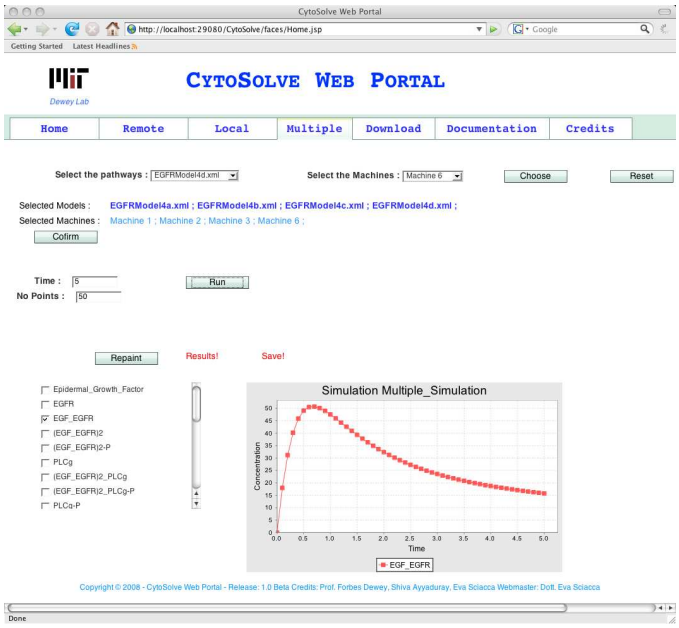

Fig. 5. The web page to perform multiple simulations of the CytoSolve web application

which denotes the jth species concentration of the integrated model in the global vector (denoted by subscript g).

Using the above notations, mathematically the formalism for the mass balance is represented as follows:

$$S^{j}_{g,n+1} = S^{j}_{g,n+1} + \sum_{i=1}^{M} (S^{j,i}_{n+1} - S^{j,i}_{n}) \qquad (1)$$

V. VALIDATION OF CYTOSOLVE ARCHITECTURE

In this section results from CytoSolve are presented in the solution of a concrete biological model: the Epidermal Growth Factor Receptor (EGFR) pathway model published by Kholodenko et al [13]. The EGFR pathway is selected since known solutions exist for this problem thus enabling direct confirmation of the CytoSolve approach. Snoep et al [19] have instantiated the Kholodenko EGFR model into the SBML language so that the model can be simulated using software programs such as Cell Designer which adopts a monolithic approach.

The EGFR model of Kholodenko shown in Fig. 6 can be considered to be derived by integrating a set of smaller pathways. There are many such smaller pathways. In Fig. 7 and Fig. 8, diagrammatic representations of one set of such smaller pathways are created, and denoted as Model 1, Model 2, Model 3 and Model 4, which, if integrated would derive the whole EGFR pathway shown above in Fig. 6.

In reviewing Model 1, Model 2, Model 3 and Model 4, one will recognize that the species $(EGF\_EGFR)2-P$ is shared by all four models. Model 3 and Model 4 share the common species SOS.

Below in Table I, the results of executing each of the four sub-models: Model 1, Model 2, Model 3, Model 4, first in Cell Designer then in CytoSolve individually are presented.

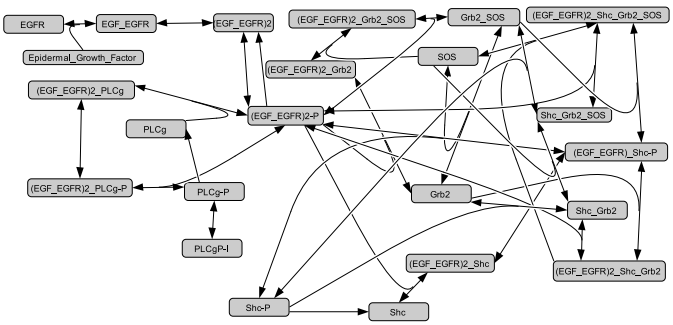

Fig. 6. Diagrammatic description of the whole EGFR pathway as published by Kholodenko et al

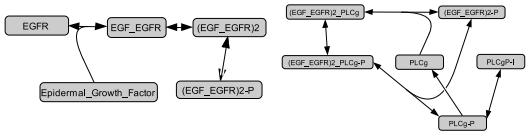

Fig. 7. Diagrammatic description of Model 1 and Model 2, two portions of the whole EGFR model.

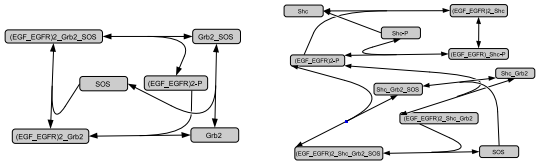

Fig. 8. Diagrammatic description of Model 3 and Model 4, other two portions of the whole EGFR model.

Model	Cell Designer	CytoSolve	Difference
Model 1	1310 ms	4271 ms	0.021 %
Model 2	1752 ms	4615 ms	0.034 %
Model 3	1763 ms	4714 ms	0.015 %
Model 4	2133 ms	5102 ms	0.017 %

TABLE I

TIME SPENT BY CELL DESIGNER AND CYTOSOLVE FOR COMPUTING EACH SUB-MODEL.

For Cell Designer, each model was loaded in one at time and then executed. For CytoSolve, CytoSolves central controller was implemented on one server and each model was implemented on another server. The results in Table I for columns 2 and 3 are a result of averaging five different test runs. The Difference is calculated as the RMS average across those five test runs for various species concentrations in each sub-model. The difference in compute times is primarily due to network latency required for CytoSolves central controller to contact and receive information back from each model. Cell Designer has no network latency since each model runs on the same server as Cell Designer.

In the following case study, the full integration of all four models is performed to derive the complete EGFR model shown Figure 6. For Cell Designer, all four models were

loaded into the Cell Designer system and had to be connected by hand to recreate the diagram in Fig. 6. This process took several hours to perform and ensure consistency and accuracy of the pathway as described by Kholodenko. For CytoSolve, the central controller was run on one machine and four separate computers were setup, each running one independent model. The goal in this exercise was to evaluate the difference in solution between CytoSolve and Cell Designer as well as computational time differences for deriving the whole EGFR model. The results are shown in Table II.

Cell Designer	CytoSolve	Difference
3217 ms	9685 ms	0.026 %

TABLE II

TIME SPENT BY CELL DESIGNER AND CYTOSOLVE FOR COMPUTING EACH SUB-MODEL.

The above discussion focused on comparing the computation times of the two different approaches. Figure 9 illustrates the comparison of actual solutions for the bound EGF-EGFR concentration profile, respectively, from CytoSolve and Cell Designer. The results from CytoSolve where carried out using 50 and 100 number of points for the solution of the ODE systems. The error of the concentrations computed by CytoSolve with respect to the concentrations computed by Cell Designer is 1.76% when 50 points are used and decrease to 0.9% when 100 points are used.

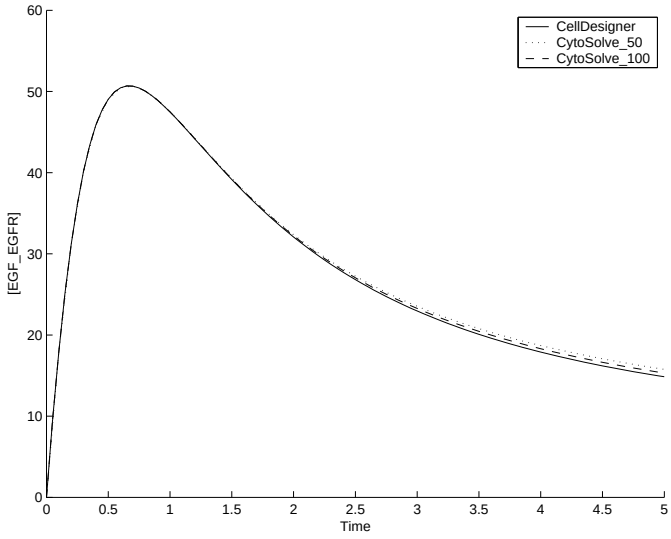

Fig. 9. Concentration of the bound EGF-EGFR using CytoSolve and Cell Designer.

Initial results from the EGFR example has demonstrated that CytoSolve can serve as an alternative to the monolithic approaches for integrating and solving biomolecular pathways. Most important is CytoSolves core feature for integrating multiple pathway models, which can be distributed across multiple computing systems, without ”hand wiring” of each model. While such a manual approach may be viable for a handful of models, it will not scale to support the integration of all pathway models necessary to model the whole cell.

VI. DISCUSSION

Apart from computational tools that use the monolithic approach there are two existing systems based on the messaging approach to integrate multiple biological pathways. These two systems are CellAK [22] and Cellulat [10]. Both of them use a static messaging approach through the Agent-based modeling. In the static messaging approach, the models remain independent programs and do not affect each other as they are executing. Any one model accepts as input a dataset and executes to completion to generate an output dataset. That output dataset is then given to another model which that model uses it as input and also executes through to completion. This process can then be continued with other models, and they can be executed concurrently if there are no dependencies between their datasets. CellAK and Cellulat treat each biological pathway model as a single entity (or agent) obeying its own pre-defined rules and reacting to its environment and neighboring agents accordingly. These approaches offers many positive ways for integrating biological pathway models; however, a non-specialist has very high learning curve in preparing a set of biological pathway models for use with this approach because the integrator has to understand deeply the biology and architecture behind them. Furthermore these tools do not use ordinary differential equations to determine the time evolution of cellular behavior, since differential equations find it difficult to model directed or local diffusion processes and sub-cellular compartmentalization and they lack the ability to deal with non-equilibrium solutions. Most common biological modeling systems use traditional ODEs to simulate the models. Finally their architectures are not designed to perform simulations on a distributed computational environment, which Cytosolve offers.

The computational architecture of Cytosolve, now available via an on-line web portal as shared in this paper, offers researchers a computational environment to collaborate and integrate quantitative molecular pathways with greater ease using the GUI. The user does not have to be a specialist on the computational architecture neither an expert of all the multiple sub-models to be merged. Model simulation by means of ordinary differential equations does not require the user to change the mode of solution between an individual pathway and a collection of pathways acting in parallel. Common languages found in modeling (SBML, CellML, MathML and certain ODE solvers like Matlab) are supported. Finally, distributed control allows the maintenance of each model at the local level, not at a central level. Any creator of a model will be able to integrate the model from their local location to an ensemble of distributed models. This means that if the owners of a model wish to quickly test or integrate their model with a set of other models they will not have to download each of the other models to their local computer. The Cytosolve computational environment will enable the owners to integrate their model with the other models with little to no effort.

VII. CONCLUSIONS AND FUTURE WORKS

This paper presents the development of a web-portal, with a GUI front-end to the Cytosolve architecture for integrating quantitative molecular pathways. The on-line web portal will enable users to access the existing functions of the Cytosolve environment through a GUI making collaboration with other researchers and integration of biological pathway models easier. Using the on-line portal, users can simulate a quantitative biological pathway model by accessing the Cytosolve remote controller through a straightforward graphical user interface. The users will be able to load a biological pathway model from the GUI and run the simulation of a biological model on the local machine or remotely on a server machine, and will be able to couple the simulation of their model with other models running on an ensemble of remote machines.

The main features of the on-line GUI-based web portal to Cytosolve presented in this paper are:

- *Ease of access to the distributed and scalable architecture.* The architecture is able to integrate new pathway models with the same ease as it is to integrate the first one and any creator of a model should be able to integrate their model from their local location to an ensemble of distributed models.
- *The multiple platform availability for the simulations.* The computational system allows models developed on different hardware and computing environments to be integrated with ease.
- *The open accessibility.* The computational system supports integration of models across geographical boundaries. While each model may be on different computers, they may also be physically at different locations anywhere in the world. In fact the architecture and the implementation of the web application support protocols for communicating with models anywhere without regard to geographical location.

Future work we will include a more sophisticated Ontology to manage nomenclature and species identification across all individual biological pathway models to be integrated by means of the web application, and automated searching for related biological pathways.

REFERENCES

[1] Ayyadurai, V.A.S. (2007) Scalable Computational Architecture for Integrating Biological Pathway Models, *PhD Thesis*, MIT University.

[2] Ayyadurai, V.A.S., Dewey, C.F., Jr. (2008) Cytosolve: Scalable method for dynamic integration of a distributed ensemble of biological pathway models, *BMC Bioinformatics*, in review

[3] Ayyadurai, V.A.S. (2008) Challenges to Computational Modeling of the Cell , *BMC Syst Biol*, in review

[4] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H., Thatte, S., Winer, D., (2007) Simple Object Access Protocol (SOAP) 1.2. [http://www.w3.org/TR/SOAP/]

[5] Bulatwicz, T.F. (2006) Support for Model Coupling: an Interface-Based Approach, *PhD Thesis*, University of Oregon.

[6] Carlisle, D., Ion, P., Miner, R., Poppelier, N. (2003) Mathematical Markup Language (MathML) Version 2.0 (Second Edition). [www.w3.org/Math]

[7] Christensen, E., Curbera, F., Meredith, G., Weerawarana, S. (2001). Web Services Description Language (WSDL) 1.1. [www.w3.org/TR/wsdl]

[8] Cuellar, A.A., Lloyd, C.M., Nielsen, P.F., Bullivant, D.P., Nickerson, D.P., Hunter, P.J. (2003) An Overview of CellML 1.1, a Biological Model Description Language, *Simulation*, **79**, 740-747.

[9] Funahashi,A., Matsuoka,Y., Jouraku,A., Kitano,H., Kikuchi,N., (2006) Cell Designer: a Modeling Tool for Biochemical Networks, *Proceedings of the 38th conference on Winter simulation*

[10] Gonzalez, P.P., Cardenas, M., Camacho, D., Franyuti, A., Rosas, O. and Lagunez-Otero, J. (2003) Cellulat: an agent-based intracellular signalling model, *Biosystems*, 68, 171-185.

[11] Hucka,M., Finney,A., Sauro,H. M., Bolouri,H. , Doyle,J. C. , Kitano, H. et al. (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19** (4) , 524-531

[12] Kitano, H. (2002) Computational systems biology, *Nature*, 420, 206-210.

[13] B. N. Kholodenko, O. V. Demin, G. Moehren, J. B. Hoek, Quantification of short term signaling by the epidermal growth factor receptor, Journal of Biological Chemistry, vol. 274, no. 42 , pp. 30169-30181, 1998.

[14] Le Novere, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H., Schilstra, M., et al. (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res* **34**, 689-691.

[15] Machné,R., Finney,A., Müller,S., Lu, J., Widder, S. and Flamm,C. (2006) The SBML ODE Solver Library: a native API for symbolic and fast numerical analysis of reaction networks. *Bioinformatics* **22**(11),1406-1407

[16] Mendes, P. (1993)GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Bioinformatics* **9**(5),563-571

[17] Sauro, H.M. (2000) Jarnac: a system for interactive metabolic analysis. *Animating the Cellular Map: Proceedings of the 9th International Meeting on BioThermoKinetics*, 221 - 228.

[18] Schulz, M., Uhlendorf, J., Klipp, E., Liebermeister, W. (2006) SBMLmerge, a system for combining biochemical network models. *Genome Inform.* **17**(1), 62 - 71.

[19] J. L. Snoep, L. Lu, Epidermal growth factor receptor signaling pathway, SBML software program file. European Bioinformatics Institute, 2005. December 5, 2005. ¡http://www.biomodels.net¿.

[20] Sun Microsystems (2002). Java Server Pages, [http://java.sun.com/products/jsp]

[21] Vaidehi, N., Goddard, W.A. (2001) Atom-Level Simulation and Modeling of Biomacromolecules, *Computational Modeling of Genetic and Biochemical Networks*, 161 - 163.

[22] Webb, K. and White, T. (2005) UML as a cell and biochemistry modeling language, *Biosystems*, 80, 283-302.