Chem-Bio Informatics Journal, Vol. 2, No. 1, pp.1-17(2002)

Web-based BEST-KIT：Development of Web-based Biochemical Engineering System Analyzing Tool-KIT

Masahiro Okamoto[a*] , Kenji Sakuraba[b], Jun Yoshimura[b] , Kouji Tanaka[b] , Jun-ichi Ueno[b], Masakazu Mori[b], Tadahiro Shimonobou[b], Tatsuya Sekiguchi[c]

[a]*Laboratory for Applied Biological Regulation Technology, Graduate School of Bioresource and Bioenvironmental Sciences, Kyushu University, 6-10-1, Hakozaki, Higashi-ku, 812-8581, Japan*

[b]*Department of Biochemical Engineering & Science, Faculty of Computer Science & Systems Engineering, Kyushu Institute of Technology, Iizuka-city, Fukuoka, 820-8502 Japan*

[C]*Department of Information Engineering, Faculty of Engineering, Maebashi Institute of Technology, 460-1, Kamisadori, Maebashi-city, Gunma, 371-0816 Japan*

**E-mail*: *okahon@brs.kyushu-u.ac.jp*

(Received April 13, 2001; accepted November 8, 2001; published online February 15, 2002)

Abstract

We have implemented an efficient, user-friendly, and web-based "biosimulator" called BEST-KIT (Biochemical Engineering System analyzing Tool-KIT: http://helios.brs.Kyushu-u.ac.jp/~bestkit/) for analyzing a large scale nonlinear reaction networks such as metabolic pathways. BEST-KIT consists mainly of two kinds of modules, "MassAction" and "EnzymeKinetics." In this study, we have developed a new module, "MassAction++," which can construct and analyze reaction schemes represented by both mass action law (mass balance) and approximated velocity functions of enzyme kinetics at steady state. All modules in BEST-KIT are developed in Java applet style, and can be run on "any" platform machine through the web browser. The "MassAction++" module was developed in Java applet style. It can save data related to the constructed reaction scheme and time course data onto the client disk space and can load it onto the web browser.

Key Words: Web-based simulator, computer simulation, optimization, metabolic pathway, bioinformatics

Area of Interest: Bioinformatics and Bio Computing

1. Introduction

Recently we have implemented an efficient, user-friendly, and web-based "biosimulator" called BEST-KIT [1,2] (Biochemical Engineering System analyzing Tool-KIT) for analyzing large scale nonlinear reaction networks such as metabolic pathways. BEST-KIT mainly consists of two modules: "MassAction" and "EnzymeKinetics. "These two modules have several remarkable properties: (1) They are developed in Java (Java Developer's Kit version 1.1.3), and the client user can use them from "any" platform machine through the web browser. (2) With the "mouse," the user can easily design and update an arbitrary reaction scheme (nonlinear system) in the editing window (working area) through an efficient GUI (Graphics User Interface), even when the number of reaction components is relatively large. (3) After the scheme has been edited, cumbersome simultaneous nonlinear differential equations can be automatically produced without writing troublesome equations. (4) These modules form a "client-server system", whereby, heavy numerical calculations of the constructed scheme, which might require long cpu-time on the client machine, can be carried out on the server machine (virtual cpu-server having a high-performance cpu-capability) through the Internet, and the results are sent back in graphic form to the client..

These two modules in BEST-KIT can be easily used even by those who are unfamiliar with computer technology and with computer programming; however, these two modules differ in the method for mathematically modeling the reaction scheme. In "MassAction", mathematical modeling is represented by using mass action law (mass balance), but in "EnzymeKinetics", it is represented by using approximate velocity functions related to enzyme kinetics under steady state conditions. In this study, we develop a new module named "MassAction++" [3] that can construct and analyze a mixed reaction scheme mathematical equation represented by mass action law and approximated velocity functions of enzyme kinetics at steady state.

2. Concept of BEST-KIT

In the development of BEST-KIT, we have considered some remarkable concepts. All modules of BEST-KIT are developed based on those concepts as follows: 1) So that all users can run these simulators from "any" platform machine, we have developed them in Java applet style by using JDK (Java Developer's Kit) version 1.1.3. There is a requirement, however, that the user's machine has to be equipped with web browser executable JDK1.1 (for example, Netscape Navigator 4.0 or Internet Explorer 4.0, etc.) in order to use these simulators. 2) By using the "mouse," the user can easily design and update an arbitrary reaction scheme (nonlinear system) in the editing window (working area) through an efficient GUI, even if the number of reaction components (state variables) is relatively large. In these modules, each reactant and each reaction step comprising a reaction scheme is represented as a "symbol," and the users can construct an arbitrary reaction scheme by connecting the symbols with lines. 3) Cumbersome simultaneous nonlinear differential equations of a constructed reaction scheme can be automatically produced without writing troublesome equations. Since we have adopted dynamic allocation of memory for simultaneous differential equations, the maximum number of reactants that can be dealt with on BEST-KIT depends on the memory size of the cpu-server. In the current version, the users can define simultaneous differential equations including 240 reactants. 4) We have designed a "client-server system" where the heavy duty work of numerically calculating the constructed scheme, which

might require long cpu-time in the client machine, can be carried out in the server machine (virtual cpu-server having high-performance cpu-capability) through the Internet and the calculated results can be sent back to the client in graphic form. Since the cpu-server issues a unique task ID number for each calculation request from a client, many users can login and request calculation at once. 5) We have ed the Gear method [4] as one of the most efficient numerical calculation methods for "stiff" differential equations and HORB[5] as a communication package that extends Java for distributed object computing for data communication between client and server through the Internet. HORB is the world's first Java ORB (Object Remote Broker) and is 100% compatible with Sun's Java language specification, interpreter, and Java classes. Of course HORB includes the CORBA IDL compiler and CORBA IIOP protocol and supports a very fast and functional proprietary protocol as well. The processing speed of HORB is very fast, reportedly twice the speed of other ORBs. These concepts of BEST-KIT are summarized in Fig.1.

3. MassAction++

Fig.2 is the snapshot of "MassAction++." It consists of the following four areas: 1) Menu bar area, 2) Working area (editing area), 3) Choice area, and 4) Input area. The Menu bar area consists of File (related to file handling such as save, upload, load), Edit (related to editing the constructed scheme), Calculate (related to numerical calculation) and Graph (related to visualization). The user can construct the reaction scheme within the Working area by selecting the suitable reaction type appearing in the Choice area and by setting initial concentrations and kinetic constants in the Input area. Details on the construction of a scheme are given in section 3.2.

3.1 Symbols

In BEST-KIT, each reactant and each reaction step comprising the reaction scheme is represented as a "symbol," and the users can construct arbitrary reaction schemes by connecting the symbols with lines by using the "mouse." In MassAction++, a reactant is represented as a reactant symbol, and a reaction step is represented as a reaction symbol in Fig.3. A reactant symbol has two parts such that (i) the Label area on the upper side of reactant symbol displays the abbreviated reactant's name (changeable in the Input area) and (ii) the Connect button (rectangular button) on the bottom side of the reactant symbol is used to form the connection between the reactant symbol and the reaction symbol. A reaction symbol has three parts such that (i) the Label areaat the center of the reaction symbol displays the abbreviated reaction's name (changeable in the Input area), (ii) the Input button(son the left side of the reaction symbol indicates inflow from the reactant (substrate, inhibitor or activator), and (iii) the Output button on the far right side of the reaction symbol indicates the outflow to reactant (product). For example, in the reaction symbol of "competitive inhibition" ("R4" in Fig. 3), the lower left-side button and the lower center button represent inflow from substrate and from inhibitor, respectively, and the far right lower side button indicates outflow to product. At present, the rate constant of mass action law (mass balance) and 11 kinds of approximated velocity functions of enzyme kinetics at steady state, such as Michaelis-Menten, Hill cooperative, competitive inhibition and so on, are provided.

In this module, cumbersome simultaneous nonlinear differential equations of the reaction scheme can be automatically produced without the need for writing troublesome equations. For this purpose,

these symbols have a data structure in which much information is saved related to the connected interactions between the reactant symbol and the reaction symbol. The main parts of these data structures are shown in Fig.4. As for the data structure of a reaction symbol, each part is represented as follows.

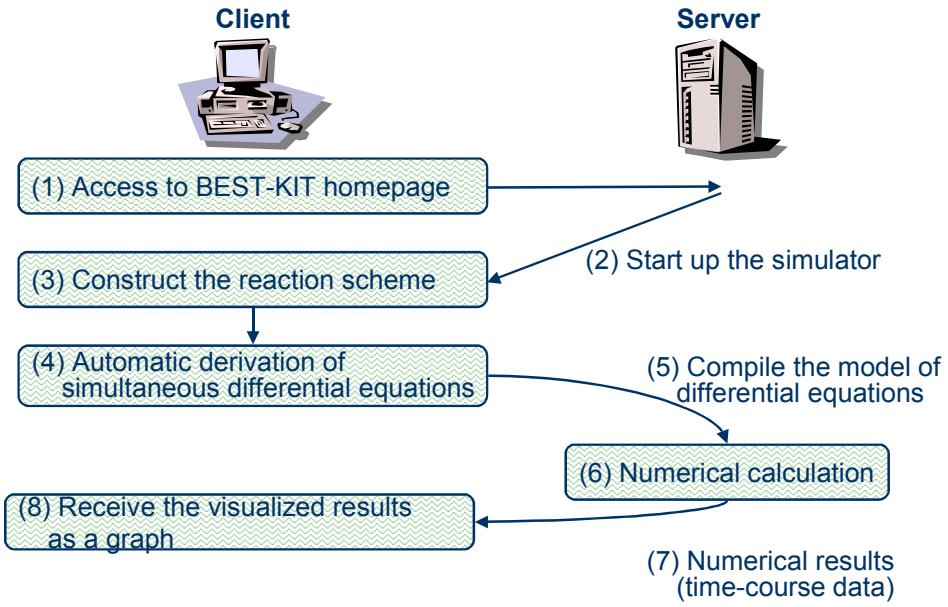

Figure 1. Procedure for simulation in BEST-KIT

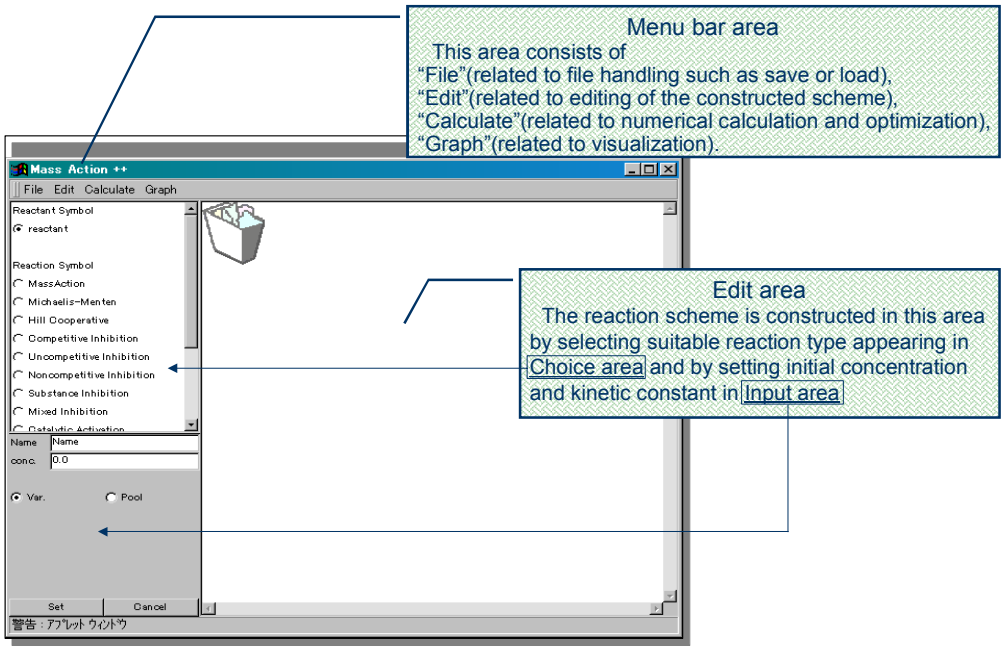

Figure 2. Snapshot of MassAction++

Part a) is the ID number of the reactant symbol that is assigned automatically. Part b) shows the abbreviated name of the reactant. Part c) represents x and y-coordinates in the working area, the value of which is used to re-draw the reactant symbol in the working area. Part d) shows the initial concentration of this reactant, and part e) shows the flag for determining whether the value of the initial concentration remains constant or varies with time. Part f) shows the number of reaction symbols related to the inflow of this reactant symbol. Part g) is a series of reaction symbols related to the inflow. Similarly as for parts f) to g), Part h) shows the number of reaction symbols related to the outflow of this reactant symbol, and Part i) is a series of reaction symbols related to the outflow. As for the data structure of the reaction symbol, parts A), B), and C) are defined in the same way as parts a) to c). Part D) shows the type number of the reaction symbol; for example, the type number of the rate constant (R1 in Fig. 3) is 1, and 2 is for Michaelis-Menten (R2 in Fig. 3). Parts E) and F) represent the number of reaction constants and the values of these reaction constants, respectively. Parts G) and H) show the number and a series of reactant symbols related to the input of this reaction symbol, respectively. Parts I) and J) show the number and a series of reactant symbols related to the output of this reaction symbol, respectively.

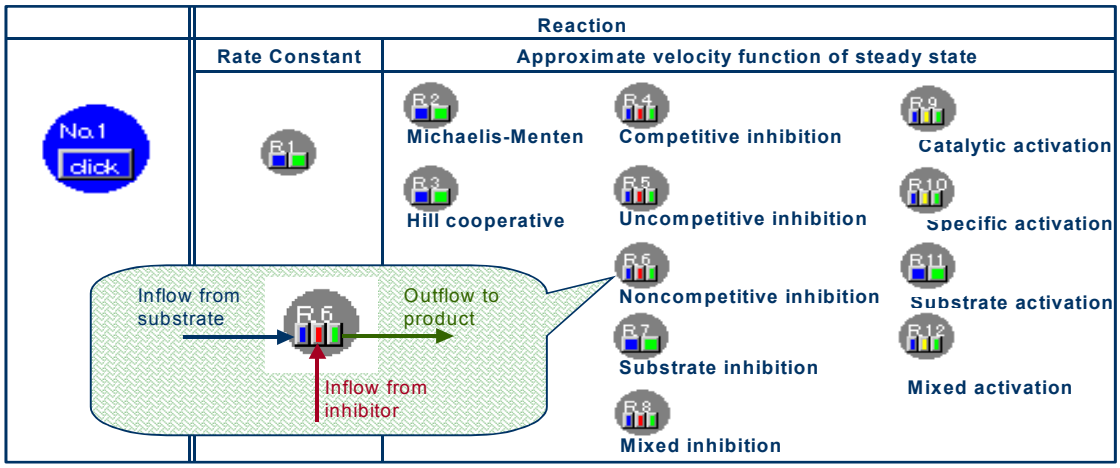

Figure 3. Reactant symbols and Reaction symbols

Reactant symbol

a)	ID number
b)	Name
c)	X, Y-coordinate
d)	Initial concentration
e)	Variable or Constant
f)	Number of reaction symbols related to the input
g)	A series of reaction symbols related to the input
h)	Number of reaction symbols related to the output
i)	A series of reaction symbols related to the output

Reaction symbol

A)	ID number
B)	Name
C)	X, Y-coordinate
D)	Type number of reaction
E)	Number of reaction constants
F)	Reaction constants
G)	Number of reaction symbols related to the inputs
H)	A series of reaction symbols related to the inputs
I)	Number of reaction symbols related to the output
J)	A series of reaction symbols related to the output

Figure 4. Data structure of reactant symbol and reaction symbol

3.2 Construction of reaction scheme

The user can easily construct and edit an arbitrary reaction scheme, the mathematical formalism of which is represented by mass action law and by approximated velocity functions of enzyme kinetics at steady state, without having to write troublesome mathematical equations. The procedure for constructing a reaction scheme is summarized as follows: At first, the kinds of symbols are selected in the Choice area in Fig. 2 and the symbols representing the reactants or reactions involved in the reaction scheme are put onto the Working area. Next, the interactions between reactants and reactions are defined by clicking several buttons in the symbol and by connecting the reactant symbol with the reaction symbol. Finally, the initial values of the parameters (initial concentrations of the reactants and several kinetic constants of the reactions) are set in the Input area.

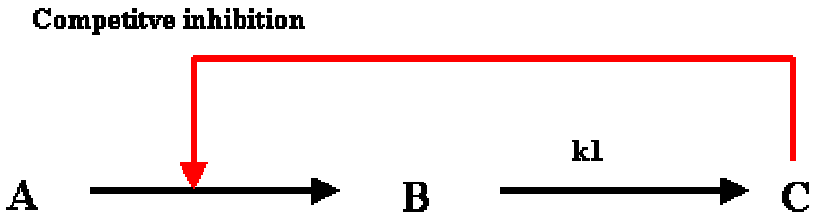

Figure 5. Example of reaction scheme

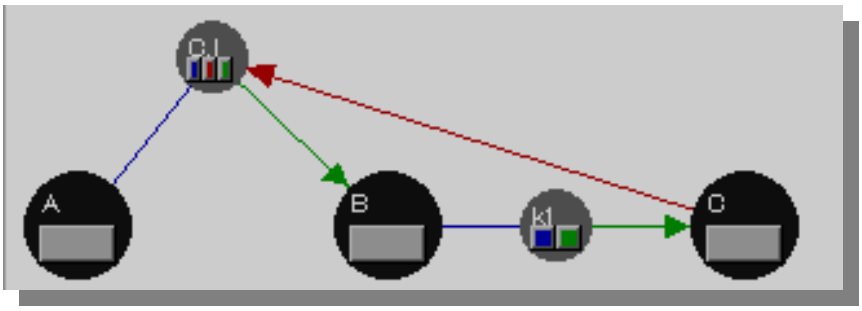

Figure 6. Constructed reaction scheme in MassAction++

The reaction scheme shown in Fig.5 consists of three reactants "A," "B," and "C." Reaction step A to B is considered to be competitive inhibition that is inhibited by C, and reaction step B to C is represented by the first order reaction in mass action law. We shall describe how to construct this reaction scheme as an example (see Fig. 6). In this case, since the total number of reactants is three, after selecting the item "reactant" in the Choice area, arrange three reactant symbols in the Working area by clicking the mouse once on each of three arbitrary positions. Since the two reaction steps, A to B and B to C in this reaction scheme, are competitive inhibition type and mass action type, respectively, after selecting the item "competitive inhibition" or "mass action" in the Choice area, put each reaction symbol at an appropriate position on the working area. In the definition of the reaction step A to B, click the "click" button in A symbol followed by clicking on the input substrate button (left lower side) in the C.I. symbol, and click the "click" button in B symbol followed by clicking on the output button (far right lower side) in the C.I. symbol. In this way, the reactant symbols A and B are connected with lines through the reaction symbol C.I., and information of the connection between reactant symbol and reaction symbol is added to the data structures of each symbol automatically. In the same manner, the remainder of the reaction scheme can be constructed as follows: In the case of reaction step B to C, click the "click" button in B symbol followed by clicking on the input button (left lower side) in the k1 symbol, and click the "click" button in C symbol followed by clicking on the output button (right lower side) in the k1 symbol. In the case of reaction step A to B, since C an inhibitor of this reaction step, click the "click" button in C symbol followed by clicking on the input inhibitor button (lower center) in the C.I. symbol.

The user can also modify the scheme. Any symbols on the Working area can be deleted by dragging and dropping them into the Trash box.

3.3 Automatic derivation of differential equations and Numerical calculation

After the scheme has been edited, by clicking the menu "Calculation" in the menu bar, derivation of differential equations and numerical calculations are carried out automatically (the detailed procedures for this automatic derivation were described elsewhere [2]). The calculation procedures are summarized in Fig.7. (1) Cumbersome simultaneous nonlinear differential equations describing the scheme can be automatically produced without writing troublesome equations. At this time, if the type number of the reaction symbol is "rate constant," differential equations produced will be based on mass action law (mass balance). If the type number of the reaction symbol is not a rate constant, differential equations will be produced based on an approximated equation of enzyme kinetics at steady state. (2) The derived differential equations and the parameter

values of initial conditions are packed into a data class for calculation as shown in Fig.8. The data class is then sent to the server through the Internet by using HORB, a communication tool between client and server. (3) The data class received for calculation by the server is saved as a file by using the Java application program. The server produces a C-language source code for the definition of differential equations based on the data to be calculated. (4) And the server compiles and links it to the Gear method numerical calculation program. (5) After the numerical calculation has been executed, numerical results (time course data) are saved in a file. (6) The result file is loaded by the Java application program followed by packing it into the result data class shown in Fig.9. (7) This result data class is sent back to the client via the Internet by means of HORB. (8) These data (time courses of reactants) are visualized in graphic form on the client's machine. The data structure for calculation consists of ten parts, a—j, as shown in Fig. 8 The parts include, respectively, (a) the total number of reactants, (b) initial concentration of each reactant, (c) the total number of reaction parameters, (d) reaction parameters of each reaction step, (e—i)the five conditions for numerical calculation, and (j) the differential equations derived from the constructed reaction scheme. The data structure of the result consists of three parts (see Fig. 9); (a) the total number of reactants, (b) the number of each reactant's data points for drawing, and (c) the result data (time course data).

3.4 Save and Load function

The simulators in BEST-KIT are developed in Java applet style, which can neither a file to the client nor load a client's file directly because of Java applet's security restrictions. Nevertheless, in MassAction++, the user can save a file of the constructed scheme data or time course data to the user's disk space and can let Java applet load its data file through the server. The procedures of "Save " in MassAction++ are summarized in Fig.10; (1) First the data of the constructed scheme are packed into a Data class (Fig. 8) following which they are sent and submitted to the server through the Internet by using HORB. (2) The server saves this received class as a temporary file by using the Java application program. (3) After saving these data as a temporary file on the server, Java applet opens a new window browser, and (4) a temporary data file on the server is shown on the client's newly opened browser. (5) Finally the client saves the data as a "text" file on the client's disk space by using save file function of the browser. Thus the user can save the data of the constructed scheme and time course on the user's disk space. Such saved data can be loaded into the module "MassAction++". This loading procedure is strictly divided into two parts: "Upload" and "Load." "Upload" sends the data file from the client to the server, and "Load" reconstructs the file based on the sending data file. These procedures are summarized in Fig.11; (1) The data file of scheme or time course constructed on the client are sent to the server by using the file uploading function of "PHP/FI," which is module combined with the WWW server "Apache". (2) The received data file is saved on the disk space of the server as a temporary file. Steps (1) and (2) steps are the "Upload" procedure. (3) The data file saved on the server is loaded into the Java application and packed into a Data class. (4) This data class is sent from the server to the Java applet on the client's machine. (5) The Java applet reconstructs the reaction scheme based on the sending data of the constructed reaction scheme. Thus, the user can save this simulator load data on the client's disk space.

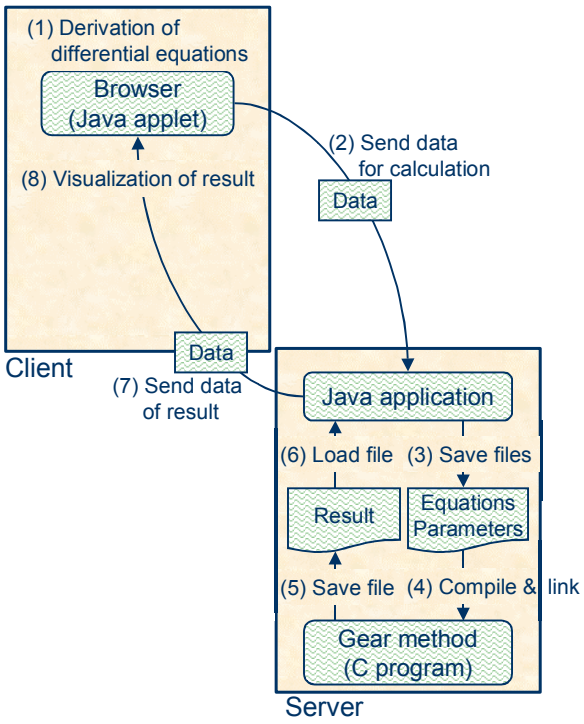

Figure 7. Calculation procedures in the client/server system

a)	Total number of reactants
b)	Initial concentrations of reactants
c)	Total number of reaction parameters
d)	Reaction parameters
e)	Calculation start time
f)	Calculation end time
g)	Calculation step time
h)	Calculation step number
i)	Calculation time scale
j)	Differential equations

Figure 8. Data structure for calculation

a)	Total number of reactants
b)	Number of data points
c)	Result data (Time course data)

Figure 9. Data structure of result

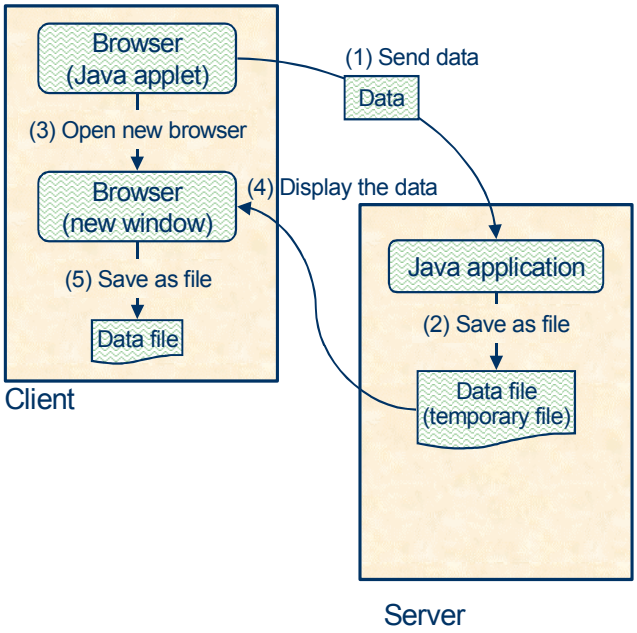

Figure 10. Procedure of "Save" in MassAction++

3.5 Time-course calculation and parameter-fitting module

As shown in Fig.7, the server machine performs all numerical calculations. After a set of simultaneous differential equations has been calculated, the numerical results are sent back to the client's machine through the Internet and are visualized on the web browser (see Fig. 12(4)). In the case of a long execution job, the user can receive the results (numerical data of obtained time-course) by e-mail. Furthermore, in MassAction ++, by using a parameter-fitting module we can estimate unknown kinetic parameters based on the observed time-course data. Fig. 13 represents the situation where both the inhibition constant (Ki) in the competitive enzymatic reaction A → B and the Michaelis constant (Km) in the Michaelis-Menten reaction B → C are unknown, and the experimental time-course data of variables A, B, and C shown in Fig. 14 are given. By assigning "unknown" to the estimated parameters and setting initial guess values for these parameters, the simulator estimates the kinetic values of these parameters that can explain the experimental time-course data by using nonlinear numerical optimization techniques such as a modified Powell method or genetic algorithm [6] (see Fig. 15). In this case, as shown in the left-hand figure of Fig. 15, the estimated values are 0.00936 for Ki at A → B, 0.0978 for Km at B → C, and the mean squared relative error between the calculated values and experimentally observed ones is 6.48 %. The right-hand figure of Fig. 15 represents the calculated time courses of A, B, and C by using estimated kinetic values and experimentally observed data.

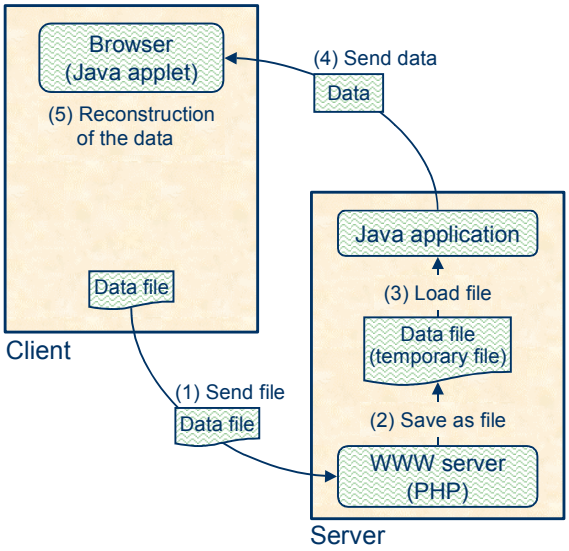

Figure 11. Procedure of "Upload" and "Load" in MassAction++

4. Discussion

Researchers in the fields of biotechnology and biochemical engineering are increasingly interested in the development of methodologies and computer-aided tools for mathematical modeling and dynamic simulation of metabolic pathways controlled by gene expression. On the one hand, both established and novel methods of molecular biology allow the sequence analysis and functional interpretation of genes, leading to an exponential growth in the amount of available biological data. On the other hand, however, to devise detailed mathematical models of the metabolism and dynamic simulation of metabolites, the stoichiometric parameters, thermodynamical parameters, and reaction mechanisms of every single biochemical reaction step in metabolic pathways are required. In this study, we have developed BEST-KIT for analyzing biochemical engineering systems including metabolic pathways. Several modeling and simulation tools for biological cells or neurons have been reported; A-cell by Ichikawa [7], BioSpice by Arkin [8], DBsolve by Goryanin [9], E-cell by Tomita [10], Gepasi [11], MIST by Ehlde [12], SCAMP by Mendes [13], Virtual Cell by Schaff [14] and so on. Compared with these simulators, our web-based BEST-KIT has the following key features: 1) large scale modeling based on mass action law and velocity functions under steady-state approximation, 2) large scale automatic derivation of simultaneous differential equations, 3) client-server style, 4) large size numerical optimization of kinetic parameters, 5) possibly text-based compatibility with other simulators such as E-cell.

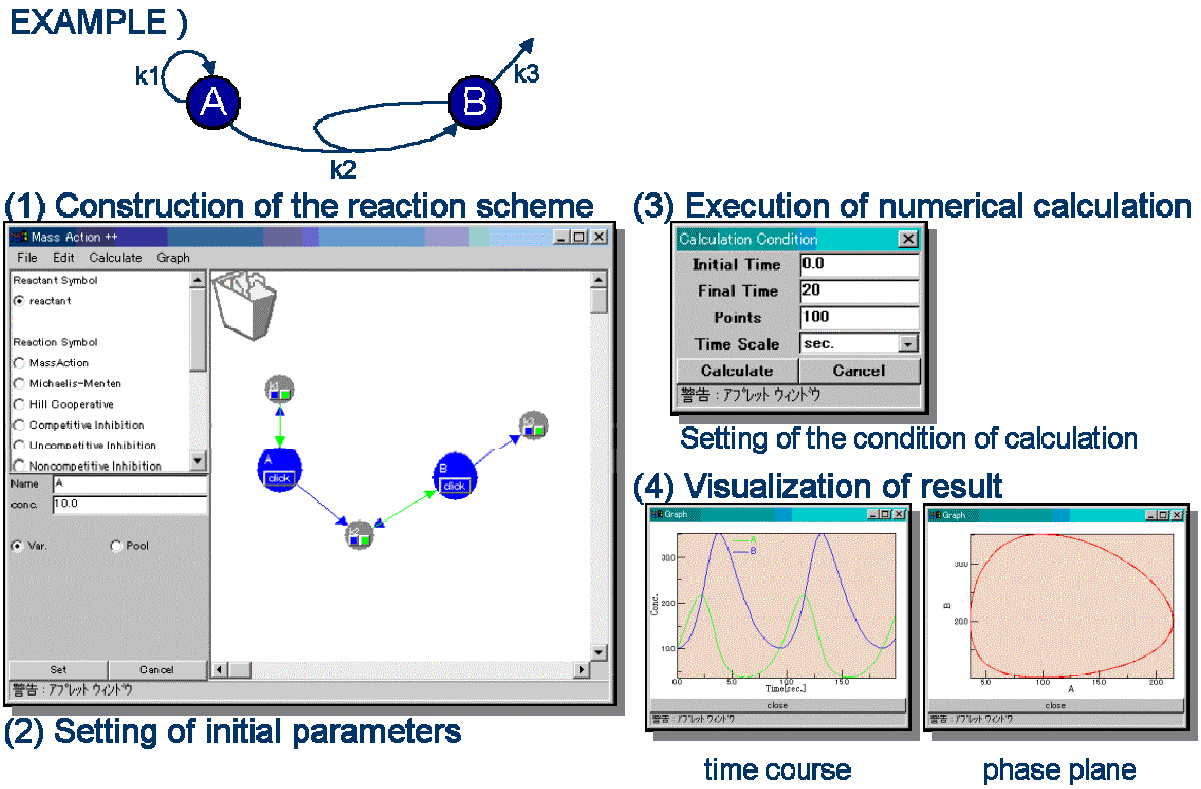

Figure 12. Time-course simulation

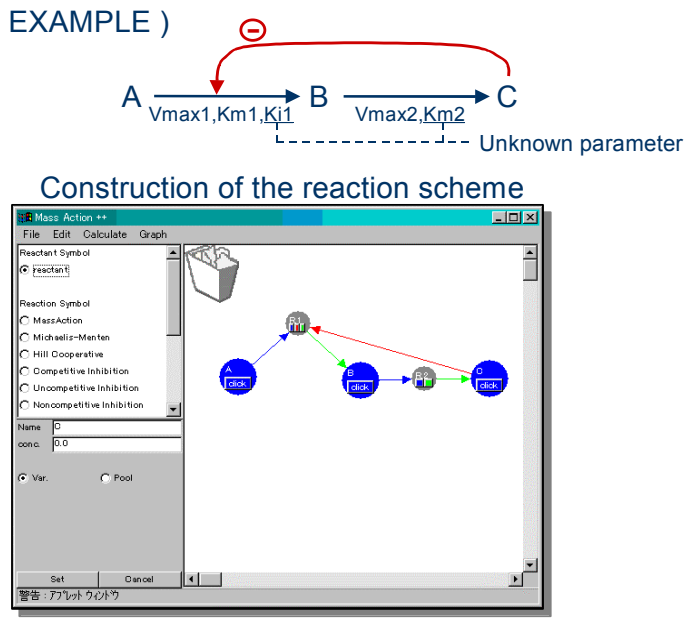

Figure 13. Parameter-fitting module

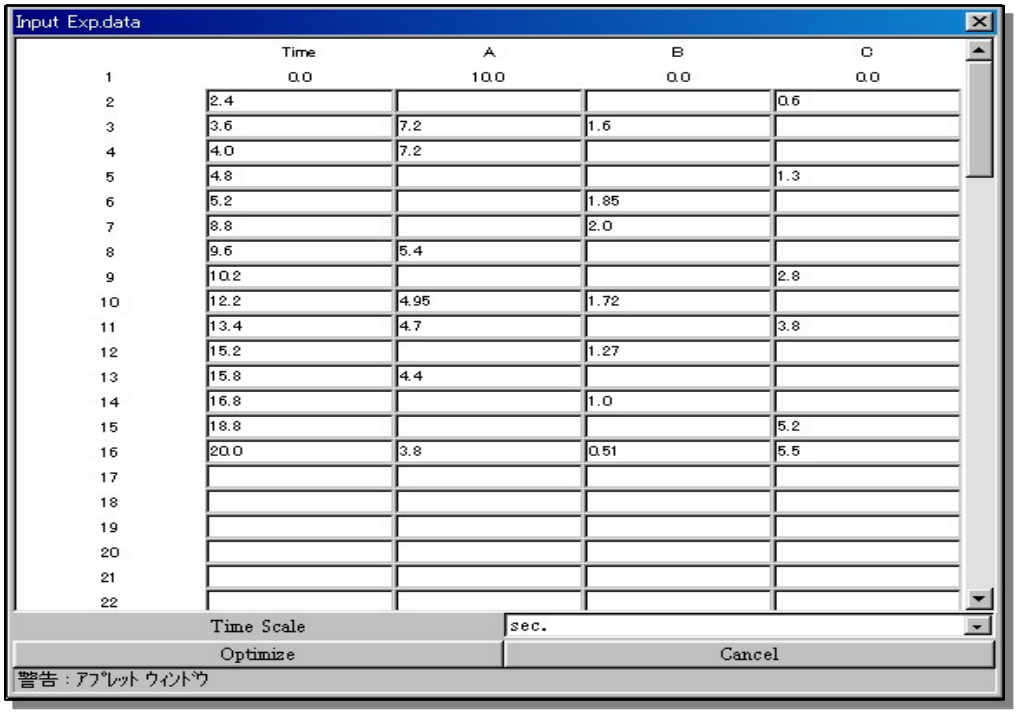

Figure 14. Experimentally observed time-course data

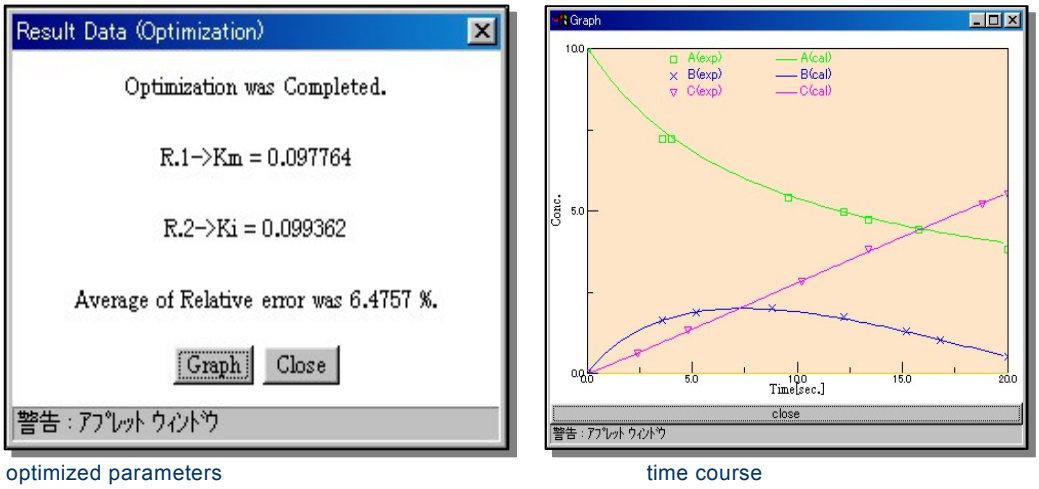

Figure 15. Estimation of the kinetic values of unknown parameters

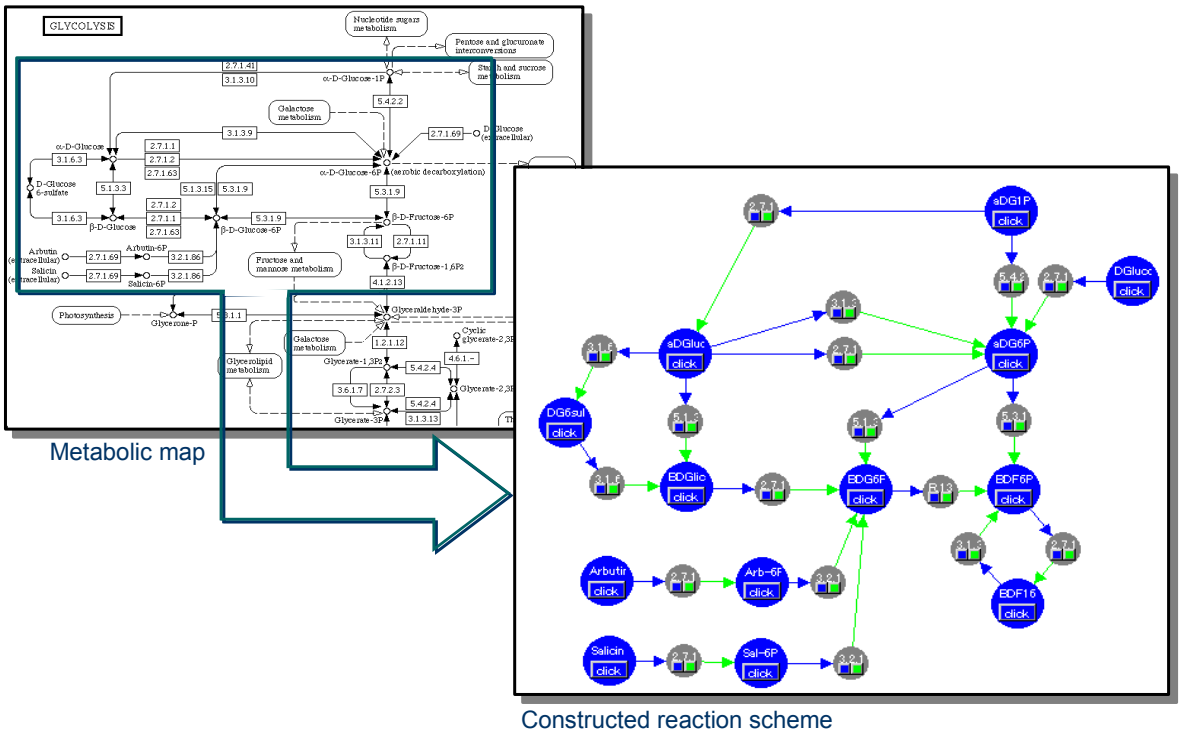

Figure 16. Snapshot of "Metabolic module"

Since all modules in BEST-KIT are developed in Java applet style and can be executed through the web browser, the user is unable to define original function boxes in addition to the prepared ones shown in Fig. 3. However, the user can modify differential equations based on mass action law by reconnecting reactant and reaction symbols. We are now developing a Window's version of BEST-KIT that will enable the user to define and make original function boxes. Also, in the very near future, as shown in Fig. 16, we will open the "Metabolic module" in which the a user can easily construct a certain reaction network system within a metabolic pathway only by clipping the system from a metabolic map with the "mouse." At present, kinetic parameters and reaction mechanisms of the corresponding reaction step cannot be retrieved from the database through the Internet, because there is a lack of such a database for collecting those data of metabolic pathways; many of the kinetic data, however, can be found in textbooks such as "Enzyme Handbook." Thus, by developing such a database and setting the environment to retrieve relevant kinetic data of a given enzymatic reaction from such a database through the Internet, a virtual-lab-system for metabolic engineering such as that shown in Fig.17 will be realized on the Internet. Then, those who want to examine the dynamics of a metabolic system can access a "biosimulator" such as BEST-KIT and easily carry out computer simulation by means of a virtual-lab-server (cpu-server) that provides for the retrieval and collection of relevant kinetic data through the Internet. The development of an efficient virtual-lab-system for metabolic engineering will suggest new types of research and will bring together highly motivated scientists from both research fields of molecular biology and computer science.

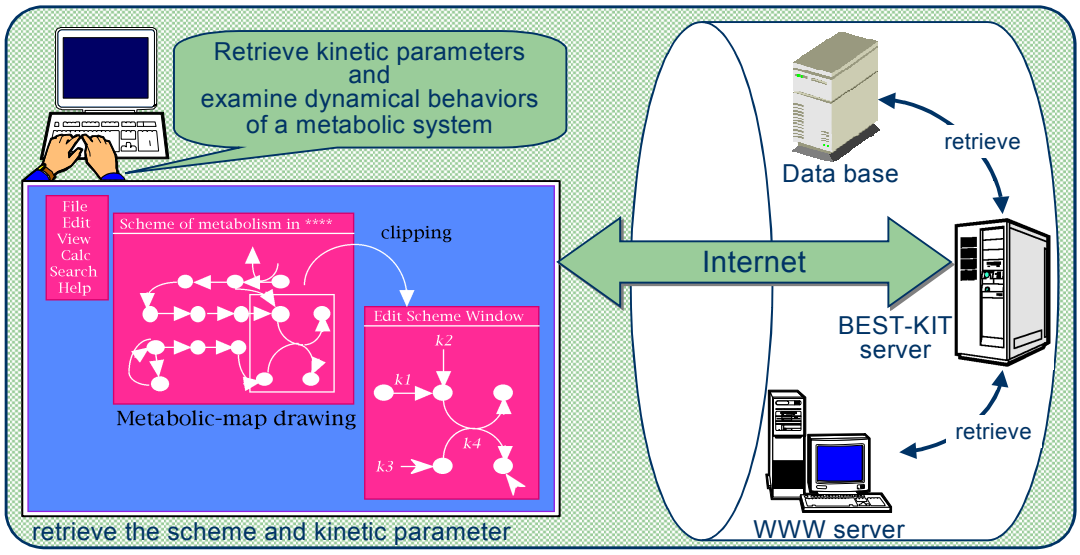

Figure 17. Conceptual figure of the Virtual-lab-system for metabolic engineering

Acknowledgement

This study was partly supported by a Grant-in-Aid for Scientific Research (B) (No. 10555128) from the Ministry of Education, Science, Sports and Culture, Japan, by the Scientific Funds of the Basic Research 21 for Breakthroughs in Info-Communications of the Ministry of Posts and Telecommunications, Japan, and by a Grant-in-Aid for Research and Development for Applying Advanced Computational Science and Technology (ACT-JST) from the Japan Science and Technology Corporation (E-CELL Project: Towards computer modeling of the whole cell (project leader: Masaru Tomita in Keio University, Japan).

References

[1] M. Okamoto, Y. Morita, Proceeding of 6th. Intl. Conf. On Comput. Applic. in Biotech., 383-386. (1995).
[2] M. Okamoto, Y. Morita, D. Tominaga, N. Kinoshita, J-I. Ueno, Y. Miura, Y. Maki, Y. Eguchi, Pacific Symposium on Biocomputing'97 (PSB97), 304-315. World Scientific, New Jersey, (1997).
[3] Home Page of MassAction++ in BEST-KIT, http://helios.brs.kyushu-u.ac.jp/~bestkit/bestkit/MassAction.html
[4] C.W. Gear, Numerical initial value problems in ordinary differential equations, Prentice-Hall, New Jersey, (1971).
[5] S. Hirano, Springer Lecture Notes in Computer Science, **1274**, 29. (1997).
[6] M. Okamoto, T. Nonaka, S. Ochiai, D. Tominaga, Applied Math. & Comput., **91**, 63-72 (1998).
[7] Home Page of A-Cell , http://www.fujixerox.co.jp/crc/cng/A-Cell/
[8] Home Page of BioSpice , http://www.lbl.gov/~aparkin/
[9] Home Page of Dbsolve, http://websites.ntl.com/~igor.goryanin
[10] Home Page of E-Cell , http://www.e-cell.org/
[11] Home Page of Gepasi , http://www.gepasi.org/
[12] Home Page of MIST , http://www.cse.ogi.edu/DISC/projects/mist/
[13] Home Page of SCAMP, http://members.tripod.co.uk/sauro/biotech.htm
[14] Home Page of Virtual Cell , http://www.nrcam.uchc.edu/main_index.html

Chem-Bio Informatics Journal, Vol. 2, No. 1, pp.1-17(2002)

Web-based BEST-KIT: Webをベースにした生物化学工学システム解析用ツールキットの開発

岡本正宏[a,*], 櫻庭健司[b], 吉村淳[b], 田中康司[b], 上野純一[b], 森正和[b], 下之坊匡弘[b], 関口達也[c]

[a] *九州大学大学院農学研究院生物機能科学部門生物機能制御学講座*
〒812-8581 福岡市東区箱崎6-10-1
[b] *九州工業大学情報工学部生物化学システム工学科*
〒820-8502 福岡県飯塚市大字川津680-4
[c] *前橋工科大学工学部情報工学科*
〒371-0816 群馬県前橋市上佐鳥町460-1

**E-mail: okahon@brs.kyushu-u.ac.jp*

要旨

質量作用則に基づく微分方程式と定常状態近似式が混在した数理モデルでも非線形連立微分方程式が導出できる、新しいバージョンのシミュレータMassAction++を開発した。利用者は理論的アプローチにおける数理モデルの導出や数値計算について全く意識することなく、反応系の動的挙動を快適にシミュレートすることができる。また、JAVAアプレットとして開発したことにより、インターネットに接続されているマシンであれば、機種に関係なく利用できる。使用する際には、WebブラウザでBEST-KITのホームページ（http://helios.brs.kyushu-u.ac.jp/~bestkit/）にアクセスするだけでよい。シミュレータ内に予め反応物質や反応ステップを表す各種のシンボルを用意しておくことにより、利用者がマウスを使ってそれらのシンボルを自由に組み合わせて、任意の反応系を視覚的に構築できる。シミュレータをクライアント・サーバ型に設計し、全ての数値計算をC言語で開発された計算プログラムを用いてサーバ側で行うことにより、JAVA言語の処理速度に関するデメリットを解消した。これにより、解析対象が大規模なものであっても、利用者が快適にシミュレーションを実行することができる。シミュレータにセーブ・ロード機能を実装させたことにより、JAVAアプレットでありながら、利用者は構築した反応スキームや計算結果を自身のマシン上に保存でき、また保存したデータをシミュレータに読み込ませることが出来るようになった。さらに、タイムコースシミュレーション機能だけでなく、パラメータフィッティング機能（最適化による未知パラメータの推定）も備わっている。

キーワード: ウェッブベースシミュレータ、コンピュータシミュレーション、代謝経路、最適化、バイオインフォマティクス

領域区分：分子生物学における情報計算技術