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

Kenji Sakuraba*, Jun-ichi Ueno*, Masakazu Mori*, Kouji Tanaka*, Jun Yoshimura*, Masahiro Okamoto**[a]

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

**Laboratory for Control Design of Biological Functions, Graduate School of Bioresource and Bioenvironmental Sciences, Kyushu University, 6-10-1, Hakozaki, Higashi-ku, 812-8581, Japan

Recently we have implemented an efficient, user-friendly "biosimulator" named BEST-KIT (Biochemical Engineering System analyzing Tool-KIT) for analyzing a large scaled nonlinear reaction network such as metabolic pathways. The BEST-KIT mainly consists of two kinds of modules, "MassAction" and "EnzymeKinetics". In this study, we have developed new module, "MassAction++", which can construct and analyze reaction scheme 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 carried out on the "any" platform machine through the web browser. The "MassAction++" was developed by Java applet style and can save data related on the constructed reaction scheme and time course data on the client disk space and can load it to the web browser.

1 Introduction

Recently we have implemented an efficient, user-friendly "biosimulator" named BEST-KIT[1,2,3] (Biochemical Engineering System analyzing Tool-KIT) for analyzing a large scale nonlinear reaction network such as metabolic pathways. The BEST-KIT mainly consists of the following two modules: "MassAction" and "EnzymeKinetics". In these two modules, there are several remarkable properties; (1) These modules in BEST-KIT are developed in JAVA[4] (Java Developer's Kit version 1.1.3), and client user can use these from "any" platform machine through web browser. (2) Using the "mouse", the users 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 is relatively large. (3) After editing the scheme, cumbersome simultaneous nonlinear differential equations can be automatically produced without writing troublesome equations. (4) These modules realize "client-server system", where, numerical calculation of the constructed scheme that might be heavy duty requiring long cpu-time in a client machine, can be carried out in the server machine (virtual cpu-server having high-performance in cpu-capability) through the Internet and the results are sent back to the client and are visualized as graph.

These two modules in BEST-KIT can easily be used by even those who are unfamiliar with computer technology and with computer programming, however, a difference between these two modules is a method of mathematical modeling of reaction scheme. In the "MassAction", mathematical modeling is represented using mass action law (mass balance), but in the "EnzymeKinetics", it is represented using approximate velocity functions related to enzyme kinetics under the steady state conditions. In this study, we develop new module named "MassAction++" which can construct and analyze mixed reaction scheme mathematical equation of which is represented by mass action low 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 carry out these simulators form "any" platform machine, we have developed in Java applet style by using JDK (Java Developer's Kit) version 1.1.3. However, since we have used JDK version 1.1.3, 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) Using the "mouse", the users 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 reactants and each reaction steps which consist in reaction scheme are represented as "symbol", and the users can construct arbitrary reaction scheme by connecting a line between the symbols. 3) Cumbersome simultaneous nonlinear differential equations of a constructed reaction scheme can be automatically produced without writing troublesome equations. 4) We have designed "client-server system" where numerical calculation of the constructed scheme, which might be heavy duty work requiring long cpu-time in client machine, can be carried out in the server machine (virtual cpu-server having high-performance in cpu-capability) through the Internet and the visualized results can be sent back to the client after calculation. We adopt the Gear method[5] as one of the best efficient numerical

[a] To whom correspondence should be addressed. (e-mail): okahon@bse.kyutech.ac.jp

calculation method for "stiff" differential equations, and adopt HORB[6], a communication package that extends Java for distributed object computing, for data communication between client and server through the Internet.

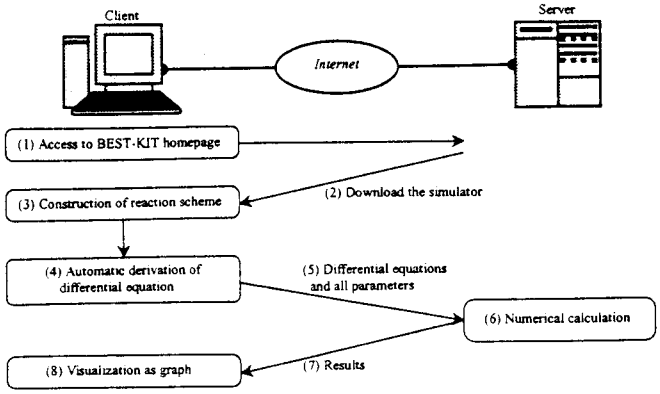

Figure 1 : Procedure for simulation in BEST-KIT

These concepts of BEST-KIT are summarized in Fig.1.

3 MassAction++

Fig.2 is the snapshot of "MassAction++" we have developed. This consists of the following four areas: 1) Menu bar area, 2) Working area (editing area), 3) Choice area, and 4) Input area. 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 users can construct the reaction scheme within Working area by selecting suitable reaction type appearing in Choice area and by setting initial concentration and kinetic constant in Input area. Details on the construction of scheme will be in the section 3.2.

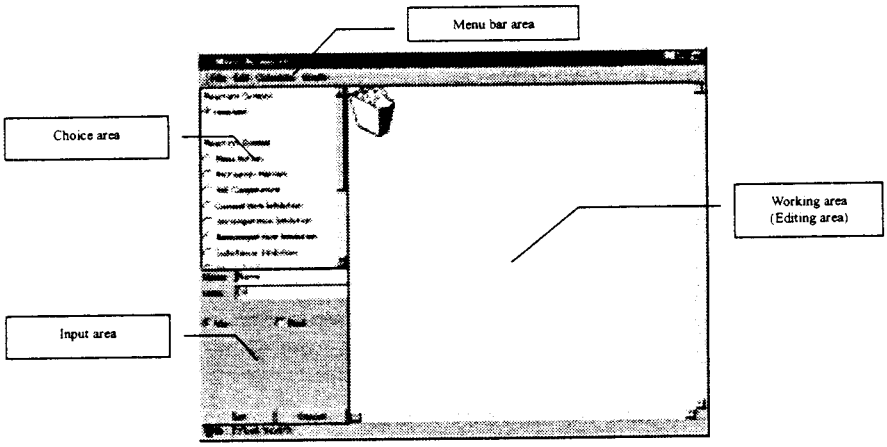

Figure 2 : Snapshot of MassAction++

3.1 Symbol

In the BEST-KIT, each reactants and each reaction steps consisting the reaction scheme are represented as "symbol", and the users can construct arbitrary reaction scheme by connecting a line between the symbols with using mouse. In the MassAction++, reactant is represented as reactant symbol, and reaction step is represented as reaction symbol in Fig.3. Reactant symbol has two parts such that, i) Label area: this area, upper side of reactant symbol, displays abbreviated reactant's name (changeable in Input area), ii) Connect button: this button (displayed "connect"), bottom side of reactant symbol, is used at the connection between reactant symbol and reaction symbol. Reaction symbol has three parts such that, i) Label area: this area, center of reaction symbol, displays the abbreviated reaction's name (changeable in Input area), ii) Input button(s): this button, left side of reaction symbol, indicates inflow from reactant (substrate, inhibitor or activator). iii) Output button: this button, right side of reaction symbol, indicates outflow to reactant (product). For example, reaction symbol of "competitive inhibition" ("R4" in Fig. 3), the left upper side button

and the left bottom side button represent inflow from substrate and from inhibitor, respectively, and the right side button indicates outflow to product. At this present, 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 prepared.

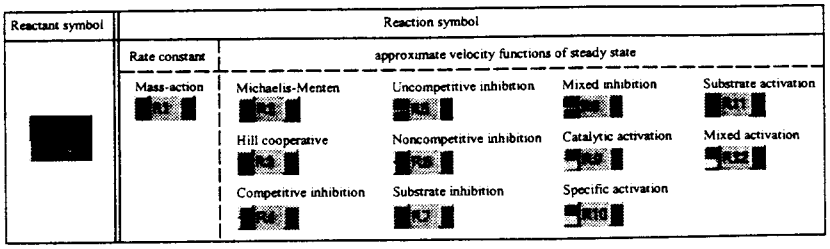

Figure 3 : Reactant symbol and Reaction symbols

In this module, cumbersome simultaneous nonlinear differential equations of the reaction scheme can be automatically produced without writing troublesome equations. For this purpose these symbols have data structure which is saved several information related to the connected interactions between reactant symbol and reaction symbol. Main parts of these data structures are shown in Fig.4.

As for the data structure of reaction symbol, each part represents as follows: Part a) is ID number of reactant symbol that is assigned automatically. Part b) shows the abbreviated name of reactant. Part c) represents x and y-coordinate in working area the value of which is used in case of re-drawing reactant symbol in working area. Part d) shows the initial concentration of this reactant, and part e) shows the flag on determining whether the value of initial concentration keeps constant or variable with time. Part f) shows the number of reaction symbols related to the inflow of this reactant symbol, and part g) is a series of reaction symbols related to the inflow. Similar to part f) to g), the h) shows the number of reaction symbols related to the outflow of this reactant symbol, and the i) is a series of reaction symbols related to the outflow.

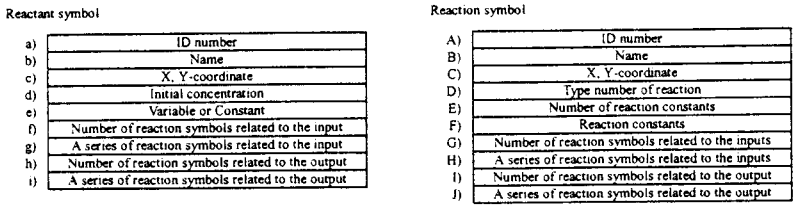

Figure 4 : Data structure of reactant symbol and reaction symbol

As for the data structure of reaction symbol, parts A), B) and C) are the same definition as part a) to c). Part D) shows the type number of reaction symbol; for example, the type number of rate constant (R1 in Fig. 3) is 1 and 2 is for Michaelis-Menten (R2 in Fig. 3). Part E) and F) represent the number of reaction constants and the values of this reaction constants respectively. Part G) and H) show the number and a series of reactant symbols related to the input of this reaction symbol respectively. Part I) and J) show the number and a series of reactant symbols related to the output of this reaction symbol, respectively.

3.2 Construction of reaction scheme

The users can easily construct and edit arbitrary reaction scheme mathematical formalism of which is represented by mass action law and by approximated velocity functions of enzyme kinetics at steady state without writing troublesome mathematical equations. The procedure of constructing reaction scheme is summarized as follows: At first, the users have to select a kind of symbols in the Choice area in Fig. 2 and put symbols representing reactant or reaction consist in reaction scheme on Working area. Next, the users have to define interactions between reactants and reactions by clicking several buttons in symbol and by connecting reactant symbol with reaction symbol. Finally the users have to set initial value of parameters (initial concentration of the reactants and several kinetic constant of reactions) in the Input area.

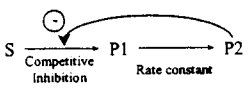

Figure 5 : Example of reaction scheme

Figure 6 : Constructed reaction scheme in the MassAction++

Reaction scheme shown in Fig.5 consists of three reactants "S", "P1" and "P2". Reaction step S to P1 is supposed to be competitive inhibition which is inhibited by P2, and reaction step P1 to P2 is represented by 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 three times on arbitrary position. Since two reaction steps S to P1 and P1 to P2 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 appropriate position on the Working area. In the definition of reaction step S to P1, click the connect button in S symbol followed by clicking input substrate button (left upper side) in C.I. symbol, and click the connect button in P1 symbol followed by clicking output button (right side) in C.I. symbol. Then the reactant symbols S and P1 are connected with lines through 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. As the same manner, the remainder of reaction scheme can be constructed as follows: In the case of reaction step P1 to P2, click the connect button in P1 symbol followed by clicking input button (left side) in k1 symbol, and click the connect button in P2 symbol followed by clicking output button (right side) in k1 symbol. In the case of reaction step S to P1, since P2 is the inhibitor of this reaction step, click the connect button in P2 symbol followed by clicking input inhibitor button (left lower side) in C.I. symbol.

The users can also modify the scheme. Any symbols on the Working area are deleted by dragging and dropping to the Trash box.

3.3 Automatic derivation of differential equations and Numerical calculation

After editing the scheme, by clicking the menu "Calculation" in the menu bar, derivation of differential equations and numerical calculation are carried out automatically. These procedures are summarized in Fig.7; (1) Cumbersome simultaneous nonlinear differential equations of the scheme can be automatically produced without writing troublesome equations. At this time, if type number of reaction symbol is "rate constant", differential equations will be produced based on mass action law (mass balance). If type number of reaction symbol is not rate constant, differential equations will be produced based on approximated equation of enzyme kinetics at steady state. (2) Derived differential equations and parameter values of initial conditions are packed in data class for calculation as shown in Fig.8, which is followed by sending and submitting to the server through the Internet with using HORB that is a communication tool between client and server. (3) Received data class for calculation, the server saves this class as a file style by using Java application program. The server produces C-language source code for the definition of differential equations based on the data for calculation. (4) And the server compiles and links it to the numerical calculation program of Gear method. (5)(6) After the execution of numerical calculation, numerical results (time course data) area saved in a file style. (7) The result file is loaded by Java application program followed by packing it to data class of result shown in Fig.9. (8) This data class of result is sent back to the client through the Internet with using HORB. (9) Received data class of result, these data (time courses of reactants) is visualized as graph on the client machine.

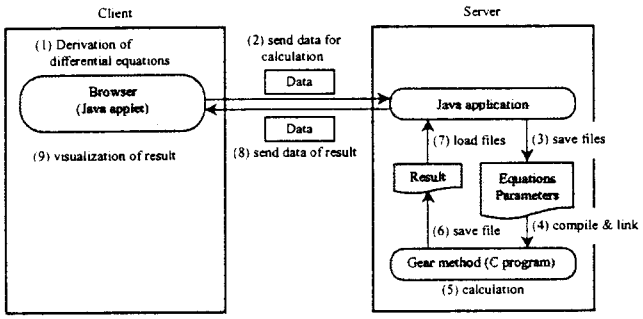

Figure 7 : Procedures of the calculation in client/server system

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

The data structure for calculation consists of ten parts as shown in Fig. 8; Part a) is the total number of reactants and part b) is initial concentration of each reactant. Part c) is the total number of reaction parameters and part d) is reaction parameters of each reaction step. Part e) to i) are the conditions for numerical calculation. Part j) is differential equations derived from constructed reaction scheme. The data structure of result consists of three parts (see Fig. 9); Part a) is the total number of reactants and part b) is the number of each reactant's data points for drawing. Part c) is the result data (time course data).

3.4 Save and load function

The simulators in BEST-KIT are developed in Java applet style. Java applet can not save a file to the client and can not load client's file directly because of Java applet's restriction for security. But in the MassAction++, the users can save 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 the MassAction++ are summarized in Fig.10; (1) First the data of the constructed scheme or time course are packed in Data class followed by sending and submitting to the server through the Internet with using HORB. (2) The server saves this received class as a temporary file by using Java application program. (3) After saving a data as a temporary file on the server, Java applet opens new window browser and (4) temporary data file on the server is shown on the client's new opened browser. (5) Finally client saves the data as a "text" file on the client's disk space by using save function of the browser. Thus the users can save data of the constructed scheme and time course on the user's disk space.

Saved data on the client's disk space can be loaded into the module "MassAction++". This load procedure is strictly divided into two parts; "Upload" and "Load". "Upload" is the procedure of sending the data file from the client to the server and "Load" is the procedure of reconstructing based on sending data file. These procedures are summarized in Fig.11; (1) Data file of constructed scheme or time course on the client are sent to the server with using file uploading function of "PHP/FI" which is combined module with WWW server "Apache". (2) Received data file is saved on the disk space of the server as a temporary file. These (1) and (2) steps are "Upload" procedure. (3) Saved data file on the server is loaded into Java application and packed in Data class. (4) This data class is sent from the server to the Java applet on the client. (5) If the received file is the data of constructed reaction scheme, Java applet reconstructs reaction scheme based on the sending data. Or if received file is the time course data, Java applet sets it inside. Thus, the users can let this simulator load data saved on the client's disk space.

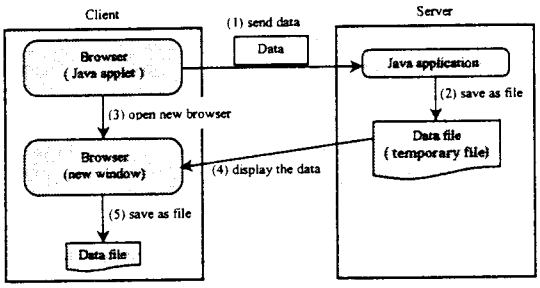

Figure 10 : Procedure of "Save" in the MassAction++

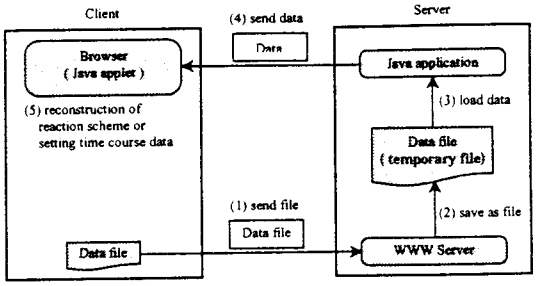

Figure 11 : Procedure of "Load" in the MassAction++

The data structure of constructed reaction scheme is shown in Fig.12; Part a) is the data of reactants listed in data structure of each reactant symbol (left side of Fig. 4) and part b) is the data of reactions listed in data structure of each reaction symbol (right side of Fig. 4).

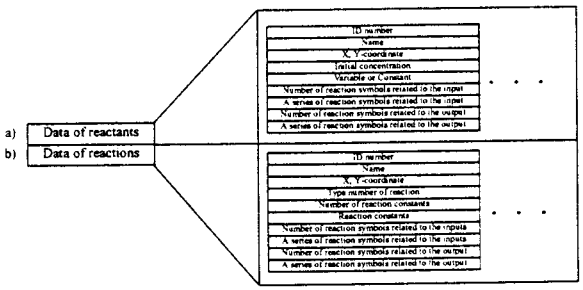

Figure 12 : Data structure of constructed reaction scheme

4 Discussion

Scientific research in biotechnology and biochemical engineering are increasingly interested in the development of methodologies and computer-aided tools for mathematical modeling and dynamics simulation of metabolic pathways controlled by gene expressions. On the other hands, novel and established methods of molecular biology allow the sequence analysis and functional interpretation of genes, leading to an exponential growth in the available biological data. However, detailed mathematical models of the metabolism and dynamic simulation of metabolites require stoichiometric parameters, thermodynamical parameters and reaction mechanism of every single biochemical reaction step in metabolic pathways. In this study, we have developed BEST-KIT for analyzing biochemical engineering systems including metabolic pathways, however, at this present, kinetic parameters and reaction mechanism of the corresponding reaction step cannot be retrieved from database through the Internet because of the lack of such a

database collecting those data of metabolic pathways; many of kinetic data are in the textbook 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, virtual-lab-system for metabolic engineering shown in Fig.13 will be realized on the Internet, where those who want to examine the dynamics of the metabolic system can access to "biosimulator" such as BEST-KIT and easily carry out computer simulation by way of virtual-lab-server (cpu-server) with retrieving and collecting relevant kinetic data through the Internet. The development of an efficient virtual-lab-system for metabolic engineering will propose a new type of research and will bring together highly motivated scientists of both research fields of molecular biology and computer science.

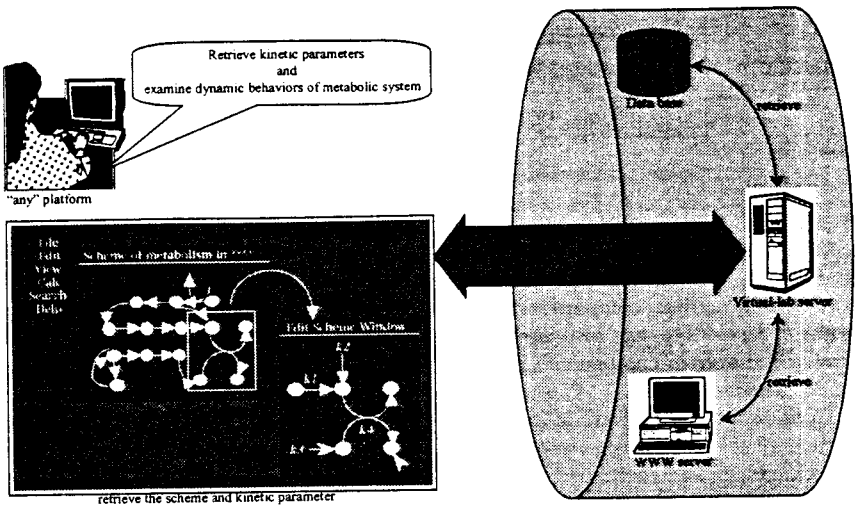

Figure 13 : Conceptual figure of Virtual-lab-system for metabolic engineering.

5 Acknowledgement

This study was supported by Grant-in-Aid for Scientific Research (B) (No. 10555128) from the Ministry of Education, Science, Sports and Culture, Japan.

References

1. M. Okamoto, Y. Morita, BEST-KIT; Development of biochemical engineering system analyzing tool-kit. Proceeding of 6th. Intl. Conf. On Comput. Applic. in Biotech., 383 (1995).
2. M. Okamoto, Y. Morita, D. Tominaga, N. Kinoshita, J-I. Ueno, Y. Miura, Y. Maki, Y. Eguchi, Toward a virtual-lab-system for metabolic engineering; Development of biochemical engineering system analyzing tool-kit (BEST-KIT). Pacific Symposium on Biocomputing'97 (PSB97), 304 (World Scientific, New Jersey, 1997).
3. J-I. Ueno, M. Mori, K. Tanaka, M. Okamoto, Web-based BEST-KIT; Development of Web-based Biochemical Engineering System Analyzing Tool-KIT. Pacific Symposium on Biocomputing'99 (PSB99), (World Scientific, New Jersey, 1999).
4. Java quick reference. (O'REILLY, 1998).
5. C.W. Gear, Numerical initial value problems in ordinary differential equations (Prentice-Hall, New Jersey, 1971).
6. S. Hirano, HORB: Distributed object execution of Java programs. World wide Computing and Its Applications, Springer Lecture Notes in Computer Science, **1274**, 29, (1997).