N.1 Introduction

Molecular Biology, Biotechnology and Bioinformatics started to focus on the problem of gene regulated metabolic network control. This problem cannot be circumvented, because no open reading frame (ORF) can be expressed without the appropriate regulatory sequences. Moreover, some genes code for proteins which turn other genes on and off. Groups of these genes constitute networks with complex behaviors. These networks control other genes whose protein products catalyze specific biochemical reactions. Hence the small molecules which are substrates or products of these reactions can in turn activate or deactivate proteins which control transcription or translation. For that reason, gene regulation can be said to indirectly control biochemical reactions in cellular metabolism, and cellular metabolism itself exerts control of gene expression.

For these reasons, the interdependent biochemical processes of metabolism and gene expression can and should be interpreted and analyzed in terms of complex dynamical networks. Therefore modeling and simulation are necessary. To solve this problem, we have to bring together information from gene regulation and metabolic pathways. These data have been and will be stored systematically in specific databases which nowadays are

accessible via the Internet. Recently many firms have been established which provide essential data for the solution of scientific and industrial problems, and even more importantly the corresponding infrastructure. As a result, there are more than 200 databases available via the Internet for all known sequenced genes (e.g. EMBL), proteins (e.g. SWISS-PROT, PIR, BRENDA), transcription factors (e.g. TRANSFAC), biochemical reactions (KEGG) and signal induction reactions (e.g. TRANSPATH, GeneNet). As well as databases, simulators for metabolic networks, which employ most of the currently popular modeling methods, are also available via the Internet. In addition to the classical methods of differential equations, discrete methods have become quite important. The integration of relevant molecular database systems and the analysis tools will be the backbone of powerful information systems for Biotechnology.

In our chapter we will introduce the basics of database and information systems for molecular biology. Based on this introduction we will discuss the important topics of database integration for the automatic fusion of molecular knowledge. Furthermore computer science is developing and implementing tools for the analysis of molecular data. The analysis of molecular data and the analysis of metabolic networks are of equal importance.

Section 5 of our chapter will present the key idea of the rule based model of metabolic processes. This method is the core of our simulation shell. Section 6 will shows the prototype of our MARG-Bench system (MARG – Modeling and Animation of Regulative Gene Networks), which integrates basic molecular database systems and our simulation tool.

N.2 Databases and Information Systems

For the physical storage and maintenance of data by means of computers, specific software is necessary. In the simplest case, the functions of the operating system are used. Accordingly, the data is stored in simple files. Therefore the biological software tools themselves specify the internal structure and sequentially write the data into related files. The tools are as different as the used file formats. These range from unreadable binary format to well standardized XML (Extensible Markup Language). In this respect the software engineer of each tool is responsible for the data storage. He has to implement methods for data access, updates and backup (see Figure N.1).

This kind of data storage and access is often mistaken for a Database Management System (DBMS). Some important disadvantages of such accumulatively acquired data collections are e.g. lack of performance, data

redundancies, no standardized query languages, synchronous data updates, no explicit scheme information that can be called up etc. These disadvantages are often unimportant for individual uses. But in the case of multi tool usage and massive global and public use, they present a problem. Nevertheless, the necessity of their use is generally accepted, because all collected data have to be analyzed, using several tools and techniques. It is an important task to enable efficient data access, that is independence from the implementation of the storage mechanism as illustrated in Figure N.2.

Consequently, for a common and wide range usage of data, a homogeneous kind of data storage and access is essential. Therefore the main tasks of a DBMS are (Codd, 1982):

- uniform management of all data,
- provision of data operations like storage, update and searching,
- covering a unique data description of all stored data (scheme),
- consistence check,
- authorization control,
- transaction management and
- backup mechanism.

Today a lot of different DBMS are available, which differ in price, data model (hierarchical, relational, object oriented, object relational etc.), features (supported operators, transactions, indexes etc.), query mechanism and languages (SQL, OQL, QBE etc.). Popular examples for such DBMS are: ORACLE, INFORMIX, DB2, Microsoft SQL Server, POET, Object Store.

The main advantages of such DBMS are often pointed out to be very efficient searching and access opportunities, common usable data, few data storage redundancies, easy data access by software tools. Thus, DBMS are increasingly used for biological data (Kemp et al., 1999; Xie et al., 2000).

In most cases stored biological data are not independent. Rather this data is processed by related biological tools. That means, tools and related data represent a coupled system, which is called an information system (IS). An IS is a complex coupled software system for information processing. In other words we can define IS as a logical union of the data itself, tools and the import of external data. Consequently, a DBMS may be identified as the central substructure of persistent data storage of such a system.

N.3 Molecular Databases

In molecular biology a large number of research projects are currently producing an exponentially increasing amount of data. A popular example is the Human Genome Project. Only in this context about $3*10^9$ base pairs and mapping data must be stored (see http://www.ornl.gov/hgmis/project/progress.html). The most popular public sequence database EMBL includes 4.7 million entries of primary nucleotide sequences and related data (Baker et al., 2000). If these databases are set in relation to the count of molecular databases, an impression of the data volume may be given. This data is often published in publicly accessible data sources. Often the categories are very different. An investigation divides sequence and related data into 16 categories (Burks, 1999). In correspondence with this, about 200 WWW based data sources are listed. And this is meant to be only a fraction of the overall number of databases available. In most cases this data is stored for processing, analyzing etc. A common approach is to collect all available data, and then it is still possible to decide which one is dispensable. However storage capacity is not a problem today.

Nowadays the majority of the molecular database providers store their research results into "quick-and-imperfect" systems. In other words, in most

cases simple flat files are used, which are managed by the directory and file system structures of the operating system. The data access is often achieved with software, which can handle the individual file format. As an interface for an external access the usage of the WWW is popular. In those cases HTML-forms coupled with CGI-scripts are used and the data obtained by this method are presented in HTML-pages, using tables or other formats. A popular example is for instance the KEGG system (Ogata et al., 1999).

One of the reasons why the content of these databases increases constantly is clarified in a special edition of the *Nucleic Acid Research* Journal which is published annually in January and gives an overview of databases as well as presenting selected systems in more detail. In order to provide a rough imagination of the extent of currently available databases, a selection of important databases is listed in the appendix. A quite detailed collection on this topic has been compiled under the following URL http://www-bm.cs.uni-magdeburg.de/iti_bm/marg/dataacquisition/data_sources.html. The most important systems in the field of molecular network control are listed below:

Genes:

- EMBL (http://www.ebi.ac.uk)
- GDB (http://gdbwww.gdb.org/)
- GenBank (http://www.ncbi.nlm.nih.gov)

Proteins and Enzymes:

- ENZYME (http://www.expasy.ch)
- LIGAND (http://www.genome.ad.jp)
- PDB (http://www.pdb.bnl.gov/)
- PIR (http://pir.georgetown.edu)
- SWISS-PROT (http://expasy.hcuge.ch)

Pathways:

- ExPASy (http://www.expasy.ch)
- KEGG (http://www.genome.ad.jp)
- WIT (http://wit.mcs.anl.gov/WIT2)

Gene Regulation:

- EPD (ftp://ftp.ebi.ac.uk)
- RegulonDB (http://www.cifn.unam.mx)
- TRANSFAC (http://transfac.gbf.de)
- TRRD (http://www.bionet.nsc.ru)

N.4 Database Integration and Information Fusion

In order to use these special databases, the user has to connect to each database separately. However, the

integration of databases can help to detect new information. An example is the relationship between aligned gene sequences of a different organism to supplement an unknown part of a metabolism or suggest and predict alternative pathways (Luttgen et al., 2000).

At first sight, two basic problems may arise, when handling such a distributed data retrieval:

- How does the user access each relevant database?
- How can the several query results be merged into a joined data set?

These problems lead directly to the field of database integration. The integration of databases includes the following parts:

Data merging

This task describes the overcoming of the distributed data storage. On the one hand, data overlaps between the several databases and on the other hand, the data are totally different. To produce a global data set, it becomes necessary to implement a unique global data store. To achieve this there are two different methods: One is to copy all needed data into one large database (materialized). The other is to leave the data where they are and merge them virtually (non-materialized).

Derivation of an integrated scheme

Every local database provides its own data scheme. But in the case of global access, a unique scheme over all

databases is needed. The method used to solve this problem specifies the degree of scheme integration. Consequently, it is possible to define one global scheme over all local schemes. That means, all locally modeled data is integrated into a global one (bottom up). The other kind of scheme integration is to specify which data are needed by individual application scenarios. Thus several partly integrated global schemes could be modeled (top down).

Unique access mechanism and control

For adequate access to the integrated data it is necessary to provide a unique data access and querying method. For this, the mapping of local data models and access methods to global ones has to be carried out. In the case of read only data access this is no problem, but for writing operations problems occur. This comprises of transactions of global management methods for simultaneous writing on the same object, data consistency mechanism, global integrity policies etc.

Quality of data

Besides the data retrieval, the quality of this integrated data is important. Therefore it should decided how reliable the data actually is (Bork and Bairoch, 1996). It is generally accepted that databases include faulty or low quality entries like incorrect sequences, missed annotation, wrongly assigned enzyme

numbers, missed out annotations etc. Consequently, these problems with quality problems are propagate during the database integration to the global data set. Hence, a mechanism for quality control of the databases and their entries must be established.

Thus a software layer has to be provided, which offers methods for data integration. Figure N.3 illustrates this situation.

These approaches deal with several disadvantages, such as inadequate support of standardized query languages, non appropriate programming interfaces, insufficient consideration of individual user requirements to the integrated data. These reasons are the catalyst to the implementation of an alternative approach, called MARG-Bench and which is described later in this chapter.

Moreover, new concepts of so called “information fusion” have been developed. Therefore, in this context, the new methods are focused an “information discovery” and on the basis of database integration. The notion of transformation includes various aspects such as integration, filtering, analysis and preparation of data aimed to discover and represent the hidden knowledge.

N.4.1 Related Works

Today, several ways of realizing a database integration for biological data exist. The main difficulties are the unique access to proprietary flat files, WWW interfaces and the large amount of data. In fact, several approaches are currently in use:

- Hypertext Navigation: KEGG (Ogata et al., 1999)
- Data Warehouse: SRS (Etzold et al., 1996), PEDANT (Frishman and Mewes, 1997), HUSAR (Senger et al., 1995)
- Multi Database Queries: BioKleisli (Davidson et al., 1997), OPM (Topaloglou et al., 1999)
- Mediator techniques: Multiagents (Matsuda et al., 1999)

One of the most developed technologies of WWW integration of molecular databases uses the SRS - Sequence Retrieval System (Etzold et al., 1996). It is based on local copies of each component database, which have to be provided in a text-based format. The results of the query are sets of WWW-links. The user can navigate through these links. Up to now, more than a hundred databases on molecular Biology are integrated under SRS. However, within the limitations of this approach, data fusion is still the task of the user. We also do not find real data fusion; i.e. data for one real life object

(e.g. an enzyme) coming from two different databases (e.g. KEGG and BRENDA) is represented twice by different WWW pages.

Therefore, research groups try to integrate molecular databases on a higher level than the SRS approach. Results of current database research, e.g. federated database systems, data warehousing architectures or data mining techniques (Conrad, 1997), are applied. Many problems in bioinformatics require:

(1) Access to data sources that are large in volume, highly heterogeneous and complex, constantly evolving and geographically dispersed.
(2) Solutions that involve multiple, carefully sequenced steps.
(3) Information to be passed smoothly between the steps.
(4) An increasing amount of computation.
(5) An increasing amount of visualization.

E. g. BioKleisli (Davidson et al., 1997) is an advanced technology designed to handle the first three requirements directly. In particular, BioKleisli provides the high-level query language CPL that can be used to express complicated transformation across multiple data sources in a clear and simple way. In addition, while BioKleisli does not handle the last two requirements directly, it is capable of distributing

computation to appropriate servers and initiating visualization programs.

N.4.2 The Idea of our Integration Approach

In consideration of the advantages of an integrative data access and the existence of inflexible approaches a new system for database integration was developed (see http://www-bm.cs.uni-magdeburg.de/iti_bm/marg/).
According to the latest findings, our system offers a scalable and flexible approach. This is achieved by the concept of wide range database access by a wrapper (adapter) technology. The data merging is done non-materialized using set operations. Specific data integration related schemes can be defined for the user and the standardized access to the integrated data can be performed by a SQL like language. For comfortable use, a JDBC driver is available.

N.5 Modeling of Metabolic Networks

Based on the rule based modeling of metabolic processes, we implemented the simulation environment MetabSim for the analysis and visualization of gene controlled metabolic processes. The advantage of our concept is the integration of relevant molecular database systems which are available via the Internet.

N.5.1 Related Works

The availability of the rapidly increasing volume of molecular data on genes, proteins and metabolic pathways improves our capability of studing cell behavior. To understand the molecular logic of cells, we must be able to analyze metabolic processes and gene networks in qualitative and quantitative terms. Therefore, modeling and simulation are important methods.

Classification of mathematical models may be subdivided into two categories: analytical and discrete. Analytical models perform the processes of elements, acting as functional relations (algebraic, integral-differential, finite-differential, etc.) or logical conditions. An analytical model may be studied by qualitative, analytical, or numerical methods. Analytical models are generally based on integral and differential systems of equations. The paper published by Waser et al. presents a computer simulation of phosphofructokinase. This enzyme is a part of the glycolysis pathway. Waser and co-workers model all kinetic features of the metabolic reaction using computer simulation (Waser et al., 1983). This computer program is based on the rules of chemical reaction rules, which are described by differential equations. Franco and Canelas simulate the purine metabolism by differential equations, where each reaction is

described by the relevant substances and the catalytic enzymes, using the Michaelis-constant of each enzyme (Franco and Canela, 1984).

Discrete models are based on state transition diagrams. Simple models of this class are based on simple production units, which can be combined. Overbeek presented an amino acid production system. A black-box with an input-set and an output-set displays a specific production (Overbeek, 1992). The graphical model of Kohn and Letzkus, which allows the discussion of metabolic regulation processes, is representative for the class of graph theoretical approaches (Kohn and Letzkus, 1982). They expand the graph theory by a specific function which allows the modeling of dynamic processes. In this case, the approach of Petri Nets is a new method. Reddy et al. presented the first application of Petri Nets in molecular Biology (Reddy et al., 1993). This formalism is able to model metabolic pathways (Hofestadt and Thelen, 1998). The highest abstraction level of this model class is represented by expert systems and object oriented systems (Brutlag et al., 1991; Stoffers et al., 1992). Expert systems and object oriented systems are developed by higher programming languages (Lisp, C++) and allow the modeling of metabolic processes by facts/classes (protein and enzymes) and rules/classes

(chemical reactions).

The grammatical formalization is able to model complex metabolic networks. Within this class of models one may consider the cell model (E-CELL), developed by Tomita et al. This E-CELL is the generic computer software environment for modeling and simulation of whole cell systems (see Tomita et al., 1999 and chapter ***TOMITA et al.***). E-CELL is an object-oriented environment for the simulation of molecular processes in user-definable models, equipped with interfaces that allow observation and intervention, written in C++. Using E-CELL one could construct a hypothetical cell with a definite number of genes sufficient for transcription, translation, energy production and phospholipids synthesis.

N.5.2 Rule Based Modeling

Our model is an extension of the Semi-Thue system. Using a universal rule, this formalization allows the representation of genetic, biosynthetic and cell communication processes. Furthermore, it is necessary to expand this discrete model by adding concentration rates for each metabolite. Metabolites are substances or substance concentrations which can be modified by biochemical reactions. Enzymes are specific proteins which catalyze biochemical reactions. Inducers and

repressors are metabolites which are able to accelerate or slow down/prevent biochemical reactions. The biochemical space (cell state) of a cell is a mixture of these components. The set of all cell states will be denoted by Z. By these definitions the abstract metabolism is defined by the actual cell state and the biochemical reaction rules. The metabolic rule is the basic unit of the metabolic system.

Let Z be a finite set of cell states. A 5-tuple (B,A,E,I,p) with $p \in [0,1]$. and $B,A,E,I \in Z$ is called a metabolic rule. p is called rule probability, B (Before) a set of preconditions, A (After) a set of postconditions, E (Enzyme) a set of catalyzed conditions and I (Inhibitor) a set of inhibitor conditions.

Example 1

The reversible biosynthesis product Glucose-6-phosphat <-> Fructose-6-phosphat will be catalyzed by the enzyme Glucosephospat-Isomerase. This process can be described by two rules:

- ({Glucose-6-phosphat}, {Fructose-6-phosphat}, {Glucosephosphat-Isomerase}, {}, p) and
- ({Fructose-6-phosphat}, {Glucose-6-phosphat}, {Glucosephosphat-Isomerase}, {}, p).

Based on the metabolic rule we are able to define the

basic model.

G = (Z,R) is called metabolic system. Z is a finite set of cell states, S ∈ Z is called start state and R is a set of metabolic rules which is called metabolic rule set.

In the following paragraph, we define the semantics of the metabolic system. The integration of the analyzed metabolic features is the basic idea of this formalization. This is the reason for specifying a stochastic parallel derivation mechanism which will describe the change of actual cell states, depending on the specified rule set. Therefore, the set of all activated rules must be fixed. This will be the first step of the derivation process. A rule is called activated, if the preconditions of this rule are elements of the actual state z ∈ Z. Moreover, effects of inducer and inhibitor elements must be considered. If such metabolites are elements of the actual state z, then the probability of this rule will be modified by inhibitor and inducer effects (the rule probability will be modified by these elements). A special function CALCULATE(z,r) will determine the absolute probability value as rule r depending on state z. A random generator (RANDOM), using the absolute probability value of the

input, works as a boolean function and will produce either positive or negative results (true or false). Regarding the boolean value true/false, rule $r \in R$ is described as activated (deactivated) and goes into action.

Let $G = (Z,R)$ be a metabolic system, $r = (B,A,E,I,p) \in R$ a rule and $z \in Z$ a cellular state. r will activated by z (in symbols r_z), iff $\forall x \in B\ x \in z$. $A(z) = \{ r \in R : r \text{ is activated by } z \}$ is called the set of activated rules by z.

Example 2

Let $G = (Z, R, z_0)$ be a metabolic system and $z_0 = \{S,D\}$ and $R= \{r_1, r_2\}$ with

- $r_1=(\{S\}, \{H,S\}, \{D\}, \{L\}, 0.8)$
- $r_2= (\{D\}, \{X\}, \{E\}, \{D\}, 0.6)$

Regarding the configuration z_0 $A(z_0) = \{r_1\}$. The rule r_2 is not activated because the repressor is available.

Any activated rule $r \in R$ can go into action. The action of r will modify the actual cellular state of the metabolic system. Elements of the actual cellular state, which are elements of the before set of rule r, will be eliminated in z and all elements of the after component will be added to z. Therefore, the action of rule r can

produce a new state z' ∈ Z.

Let G = (Z,R) be a metabolic system, z ∈ Z the actual cellular state and r_z = (B,A,E,I,p) ∈ R. The action of r_z is defined by: If RANDOM(CALCULATE(z,r)) = true then z' = (z-B) ∪ A. The action of r_z will be described in symbols by z ->$_r$ z'.

According to the metabolic system defined in Example 2 the action of r_1 will produce the state z' = {H,S,D}.

The one step derivation of a metabolic system is defined by the (quasi) simultaneous action of all activated rules. Therefore, we consider the set of all activated rules and determine two new sets: the before set and the after set. The before set includes all B elements of the activated rules. The definition of the after set is analogous. Using these sets, the one-step derivation could be interpreted as an addition and subtraction of elements.

Example 3

Let G=(Z,R,z_0) be a metabolic system and z_0 = {B,C,E} and R = {r_1,r_2,r_3} with

- r_1 = ({B}, {S,B}, {C}, {}, 0.9)
- r_2 = ({C}, {F,C}, {E}, {}, 0.3)

- r_3 = ({C,F}, {B,C}, {E}, {}, 0.3)

$A(z_0) = \{r_1, r_2\}$ is the set of activated rules. Regarding z_0 we can identify different one-step-derivations, which will produce the following states:

- {B,C,E,S} (action of r_1),
- {B,C,E,F} (action of r_2),
- {B,C,E,S,F} (action of r_1 and r_2) and
- {B,C,E} (empty action).

Let G = (Z,R) be a metabolic system, $z \in Z$ the actual cellular state, A(z) the set of all activated rules under z and $B_z = \{B : \exists\ r \in A(z) \bullet B \in r_z\}$ and $A_z = \{A : \exists\ r \in A(z) \bullet A \in r_z\}$. The simultaneous action of A(z) is called one-step derivation, if $z' = (z - B_z) \cup A_z$. The one-step derivation is described in symbols by z => z'.

Each action could be interpreted as an independent event. Therefore, the probability of each one-step derivation can be calculated, regarding the absolute probability values of all activated and deactivated rules. In our simulation system, this will be done by multiplying these values.

However, based on the one-step derivation we can define the derivation inductively. Furthermore, based on the

one-step derivation, a probability can be calculated for any derivation.

Let G = (Z,R) be a metabolic system. $x \in Z^+$ is called derivation in G, iff $|x| = 1$ or $|x| > 1$ and $\exists\, y' \in Z^*$ z',z'' $\in$ Z: x=z'z''y and z''y is a derivation and z'->z''.

Example 4

Let be G the metabolic system defined in Example 3. The following sequence of cellular states describes a derivation:

{B,C,E} => T {B,C,E,S} => T' {B,C,E,S,F} => T'' => {B,C,E,S,F} ... with T = $\{r_1\}$, T' = $\{r_1, r_2\}$ and T''=$\{r_2, r_3\}$.

In the case of analytical modeling, it is necessary to expand our model, using abstract concentration rates. To realize this requirement for each component (metabolite), specific integer values must be assigned. These values can be interpreted as abstract concentration rates. Regarding the metabolic system, these effects can be included, using the formalization of multi-sets. Therefore, the definition of the metabolic system must be modified. Regarding the rule activation, the concentration rate of any before component must be satisfied in connection with corresponding metabolites of the actual state. The

concentration rate of this metabolite must be higher or equal in comparison with the corresponding before component of this metabolic rule. Moreover, the function CALCULATE must be modified. In this case, the influence of all concentration rates of inductor and repressor metabolites will determine the absolute rule probability. All activated rules can go into action simultaneously. With regard to corresponding metabolites of the actual state (integer values), the addition and subtraction of the concentration rates of all before and after components is needed. In this chapter, we present only the fundamental part of this formalization.

Let $z \in Z$ be a state. The multi-set $k: z \rightarrow \bullet_0$ is called metabolic concentration. K denotes the set of all metabolic concentrations.

Example 5

Let $z = \{Glucose, Lactose, RNA\text{-}Polymerase\}$. [34 Glucose, Lactose, 15 RNA-Polymerase] defines a metabolic concentration of 34 molecules Glucose, one molecule Lactose and 15 molecules RNA-Polamerase.

Based on the formalization of multi-sets, the analytical metabolic system, which enables the discussion of kinetic effects, can be defined.

G = (Z,R,k) with A ∈ Z the start state, k ∈ K a multi-set (K: A -> •) and R a finite set of metabolic rules, where r = (B,A,E,I,p) ∈ R with p ∈ [0,1]. and B,A,E,I are metabolic concentrations, is called metabolic system.

The definition of activation is fundamentally necessary. Regarding multi-sets, the activation of any rule r ∈ R depends on the specified concentration rates of each rule component, the concentration rates of the actual state and the absolute rule probability.

Let G = (Z,R,k) be an analytical metabolic system, r ∈ R a rule and z ∈ Z a cellular state. r is activated by z (in symbols r_z), iff ∀ x ∈ B x ∈ z and k(x) ≤ k(z_x) A(z) = {r ∈ R : r is activated by z} is called the set of activated rules by z.

Based on this definition, the one-step derivation can be modified. All activated metabolic rules can go into action simultaneously. Regarding the set of activated rules, all after concentration rates must be added to the actual state and all before concentration rates must be subtracted from the actual state.

G = (Z,R,k) is an analytical metabolic system, z the actual cellular state, A(z) the set of all activated rules under z and

$B_z = \{B : \exists\ r \in A(z)\ \ B \in r_z\}$ and $|B_z| := \sum_{b \in B} k(b)$,

$A_z = \{A : \exists\ r \in A(z)\ \ A \in r_z\}$ and $|A_z| := \sum_{a \in A} k(a)$.

The simultaneous action of A(z) is called one-step derivation, iff $z' = \{ x : x \in A_z$ or $\forall\ x \in z, y \in B_z\ x=y$ and $k(x) - k(y) > 0\}$. The one-step derivation is described in symbols by z -> z'.

Using the one-step derivation operator, we can define the derivation of an analytical metabolic system inductively.

Example 6

Let G be a analytical metabolic system with k = [6 A, 8 B, 3 E] and the rule set R with

- r_1 = ([2 A], [3 B], [E], [X], 0.8)
- r_2 = ([2 B], [3 D], [E], [X], 0.5).

$A(k) = \{r_1, r_2\}$ and the set {[6 A, 8 B, 3 E], [4 A, 11 B, 3 E], [6 A, 6 B, 3 D, 3 E], [4 A, 9 B, 3 D, 3 E]} describes the different one-step derivations.

N.5.3 Application

The implemented universal metabolic rule allows the

formalization of biosynthesis, gene expression, gene regulation and cell communication processes. Regarding biosynthetic processes, the before, after, inducer and repressor components are used. For example:

Enzyme E_1 will catalyze the biochemical process S_1 into S_2.

This can be expressed by:
$B = S_1$, $A = S_2$ and $E = E_1$. Moreover, we can add any concentration rates. For example:

$B = 15S_1$, $A = 12S_2$, $E = 2E_1$. The probability value will model the flux of this biochemical rule which can be influenced by specific inducer and repressor metabolites, depending on their concentration rates.

In the case of simple cell communication processes, only the before or after component will be used. From this, we obtain the following interpretation: substance A enters the cell by endocytotic processes. Therefore, we have to define a rule, where only the after component will be assigned specific substances. Moreover, such processes can be influenced by specific events which can be formulated regarding inducer and repressor components.

Normally, metabolites will disintegrate after a specific time interval. This can be expressed by a rule which represents only the specific before component. Moreover, concentration rates and specific influence

components can be defined.

In the case of the gene regulation, the activity of operons can be modeled easily. If we choose an operon which represents two structure genes (S_1,S_2), two operator genes (O_1,O_2) and one enhancer sequence, then this can be expressed by:

B = [RNA-polymerase, ribosome, amino acid, tRNA]

A = [S_1,S_2], E = [IO_1,IO_2] and I = [O_1,O_2].

However, this model allows the simulation of complex metabolic networks, and the grammatical formalization allows the definition and implementation of different languages. These languages represent specific biological aspects. For example, it is possible to produce the set of all possible pathways, to produce metabolic pathways, depending on specific conditions (as for example the probability value), to search for the appearance of specific substances (as for example toxic substances) and so on.

N.5.4 Theoretical Aspects

The set of the attainable configurations is an infinite set, and the set of all derivations is enumerable. Moreover, the set of all configurations is undecidable (Hofestadt, 1996). Use of concentrations (multi-sets) is the main reason for the undecidability. However, this result implies that no interesting question can be

solved in the research field of biotechnology.

In practice, biochemical systems are restricted. In our model, we can restrict the depth and width of the derivation process. Therefore, important questions are decidable, and we have to discuss the complexity of the derivation algorithm. If we restrict the derivation depth the language L(G,i) is decidable. L(G,i) is the set of all configurations which can be produced from the start configuration by the application of up to i derivation steps. Hence, for a metabolic system with the generation depth i it is decidable, if k is a member of L(G,i). Based on the exponential complexity of the derivation process, this question cannot be solved in practice if i is high. This is the reason that the calculation of a derivation tree is not possible in practice, because the one-step derivation represents an exponential time complexity. However, using our simulation tool, we have to restrict the derivation depth and/or width.

N.5.5 MetabSim

The simulation system MetabSim is the current implementation of our rule based model described above. It is consisting of two main parts. The first part, the object-oriented data structure of MetabSim, describes the metabolic grammar. The main structure here is the

metabolic rule. A rule contains the stoichiometry of substrates and products, enhancers, inhibitors, factors and the elasticity coefficients of one complex reaction. The user can instantiate the rule type to build up rules according to reactions of a metabolic network. The second data type is the metabolic state. The user can define a metabolic state to act as a root configuration. All calculations, that are later done will base on this state. The whole data structure is mapped into a database so that all rules and states are stored herein.

The second part of MetabSim is the derivation logic. Because the system has been designed modular, several derivation modules can be implemented and applied independently. Figure N.4 is showing the information flow in the rule based simulation system. After defining the rule set and the root configuration (default cell states) the derivation logic can be applied to the data. In the first step, the „Rule Selection" module accesses the current state and calculates the rules which can be applied, because their premise is becoming true related to the current state. The „Rule Application" module calculates the following configuration(s). Optionally, the reaction time is applied by a „Rule Kinetics" module. The new configurations (states) are the input for the next derivation step.

In biochemistry, for a set of single metabolic

systems, the systems behavior is described by the usage of differential equations. But the problem is, that this approach does not discuss all interactions in large networks. We decided to apply a more simplified and abstract method, as a compromise between the mathematical modeling and the data outcome of molecular databases. The BRENDA database is one database storing kinetic parameters of enzymes. What we can get from this system is the Michaelis-Menten value and the turnover rate and the type of regulation of the enzyme. In MetabSim, there are different kinetic behaviors implemented (see Figure N.5): the constant flux and the linear, hyperbolic and sigmoid dependency from the substrate concentration. What we do need for the calculation is the Michaelis-Menten constant (Km), the maximum reaction rate (V_{MAX}) and the enzyme type (allosteric, hyperbolic, ...). The types read from the databases are mapped into the according kinetic behavior and so the calculation is done using the given parameters.

As an example, we have built up a rule network for the glycolysis pathway to simulate the consumption and the production of ATP. The following rules illustrate the stoichiometrie of the metabolic system.

- r_0 = ({D-Glucose, ATP}, {Glucose-6-phosphate, ADP}, {Hexokinase, hyp})

- r_1 = ({Glucose-6-phosphate},
  {(Fructose-6-phosphate},
  {Phosphohexoseisomerase, hyp})
- r_2 = ({Fructose-6-phosphate,ATP},
  {Fructose-1,6-bisphosphate,ADP},
  {Phosphofructokinase, sgm})
- r_3 = ({Fructose-1,6-bisphosphate},
  {Glycerinaldehyd-3-phosphate,
  Dihydroxyacetonephosphate}, {Aldolase, hyp})
- r_4 = ({Dihydroxyacetonephosphate},
  {Glycerinaldehyd-3-phosphate},
  {Triosephosphatisomerase, hyp})
- r_5 = ({Glycerinaldehyd-3-phosphate,Pi},
  {1,3-Biphosphoglycerat},
  {Phosphoglycerinaldehyddehydrogenase, hyp})
- r_6 = ({1,3-Biphosphoglycerat,ADP},
  {3-Phosphoglycerate,ATP},
  {Phosphoglyceratkinase, hyp})
- r_7 = ({3-Phosphoglycerate}, {2-Phosphoglycerate},
  {Phosphoglyceratmutase, hyp})
- r_8 = ({2-Phosphoglycerate}, {Phosphoenolpyruvate},
  {Enolase, hyp})
- r_9 = ({Phosphoenolpyruvate,ADP}, {Pyruvate,ATP},
  {Pyruvatekinase, hyp})

In this ruleset, metabolites and enzymes are included. What we can see is, that the metabolite D-Glucose is consumed by the two rules r_4 and r_5. One ATP is consumed in the first part of the glycolysis, while 2 ATP are produced in the second part. This concludes by the double application of the rules r_7 and r_9. In Figure N.6 the system is drawn as a graph by the MetabSim environment. The application of the operator leads to the substance concentration development shown in Figure N.7.

N.6 MARGBench

The idea of our project, which is supported by the German Research Council (DFG), is to present a virtual laboratory for the analysis of molecular processes (Hofestadt and Scholz, 1998). Therefore, we provide a full scalable system for a user specific integration of different heterogeneous database systems and different interfaces for accessing the integrated data with special analysis tools (e.g. the simulation environment MetabSim, which was described earlier in this chapter).

N.6.1 Architecture of MARGBench

The architecture of our prototype is shown in Figure N.8 and is available in the WWW under the address: http://www-bm.cs.uni-magdeburg.de/iti_bm/marg/. The system is divided into two main parts, the integration

layer and the application layer. Where the integration layer consists of three different modules: data acquisition (BioDataServer), data storage (BioDataCache) and graphical data management (BioDataBrowser).

In general, the BioDataServer realizes a logical non-materialized database integration based on the concept of federated databases. A workable Internet access to the molecular biological databases is the main prerequisite for a database integration. Within this context several problems must be solved:

- different interfaces (e.g. CGI, JDBC, ...)
- different query languages (SQL, OQL, non-standardized, ...)
- different data structures (HTML, flat files, database objects, ...)
- different data models (ERM, OO, ...)

For such a homogeneous access to the several data sources, a functional interface was defined, which is implemented by special software modules, called adapters. With the technique of semi-automatic adapter generation it will be possible to dynamically connect relevant data sources. In the case of a HTML data source the adapter accesses the specific URL and parses the resulting HTML-page. For this generation a description file is necessary. Such a description file contains

structure and syntax information about the data source, which should be integrated into our system. This information enables a mapping between the data fields of the docked data sources and the attributes of an integrated user scheme.

For the consideration of the user requirements to the integrated data it is possible to specify integrated data schemes. These schemes can be defined or manipulated, using a special language. They describe the accessible attributes of integrated data sources in transparent form.

In order to obtain a complete and wide spectrum of data, it is recommended to integrate as many databases as possible. Using this special integrated user schemes, the BioDataServer combines the outcomes of adapter queries into integrated global results, known as information fusion. The scheme is relational and defines the source for each attribute and its internal dependencies. This is the basis to perform logical data integration and provide a relational view on the fusioned data. Thus it is possible to retrieve integrated data by a sub-set of SQL queries.

As can be seen, an automatic mechanism is necessary to merge the partitioned data values from the various databases. This is one task of the BioDataServer and can be solved, using mathematical set operations.

The premise to access the database integration server by computer programs is the definition of an interface. Because the server should be accessible via the Internet, a communication protocol and a query language must be specified. Nowadays, lots of database systems exist, which support a subset of SQL as query language, which in turn is based on the relation model and is well standardized. This was the reason to support SQL elements by the BioDataServer. Different techniques in the field of interfaces for remote database access have been established e.g. JDBC and ODBC. ODBC is by now only supported by Microsoft platforms. Therefore, the BioDataServer currently offers a JDBC driver, which provides a standardized database access to JAVA applications. Consequently, any JAVA platform can simply access the BioDataServer by related JAVA programs.

The main advantages of this BioDataServer are the transparent physical database access, the dynamic building of new non-materialized, logical integrated databases, a standardized access interface, a client-server capability and the platform independence. With the implemented JDBC driver the integration service of the BioDataServer is also independently useful for other external Java applications. A demonstration of the BioDataServer is available under the address:

http://www-bm.cs.uni-magdeburg.de/BDSDemo/.

The next level in the integration layer is the BioDataCache, which handles the local storage of the fusioned data in a user specific integration database. With this method it is possible to build individual integrated databases, which reflect the individual users respective application requirements. Thus it becomes possible to perform data analysis, cleaning, improvements, enrichment etc. The user is able to interactively specify and create the integration database in interface definition language (IDL) syntax. If the IDL is ready, the service modules will be generated automatically. The individual data import is based on specific integrated user schemes of the BioDataServer, which have to be defined previously. The access to these integration databases is possible by the Common Object Request Broker Architecture (CORBA) (OMG, 1991) and OQL (Cattell, 1994).

The BioDataCache provides the materialized layer for the data integration which is based on CORBA. The access for importing integrated data from the BioDataServer is possible using the JBDC driver of the integration layer. Furthermore, the BioDataCache uses an integrated user scheme for the selection of attributes, which should be integrated. An empty database related to the integrated scheme is generated

automatically.

Once data from the integration service is read, it will be stored in the underlying standard object-oriented database.

By storing the fusioned information in the cache, an separate new data source will be created. This new user specific integration database represents a quality of a meta database and is comparable to a data warehouse. The offered CORBA interface, similar to the BioDataServer, enables other software tools to access the BioDataCache.

The third part in the integration layer is represented by the BioDataBrowser. It is automatically generated during the generation of an integration database. This module allows the user a graphical managing and browsing of his fusioned data. Its functions are similar to a windows file browser. Furthermore, a JAVA interface is offered to access this component within JAVA applications. The development of DBMS-supported applications forces the programming of database-related user-interactive components to establish database connections, query the data, transmit the results, store the data and so forth. The BioDataBrowser provides this feature and can be included as a component in different Java-applications.

N.6.2 Application of MARGBench

The application of MARGBench is done at different levels, because the system consists of plugable components. Every user is able to access the components which are essentially for his specific integration problem.

One possibility is the usage of the online integration provided by the BioDataServer component. Here a uniform SQL interface with client/server and multi-user/multi-client feature is available.

A case study for this kind of data analysis tools is given under: http://www-bm.cs.uni-magdeburg.de/phpMetaTool

The software METATOOL (Pfeiffer et al., 1999) is an independent system which has been written in the C programming language. A separate program is necessary to generate the input file and to read the output files. Regarding our integration architecture the input is obtained by the BioDataServer. This client program has been implemented, using the PHP programming language and enables the program to be used via the Internet.

The second possibility to use MARGBench is to access automatically generated integration databases. Once the integration database (BioDataCache) has been established, the user can load his integration database with integrated data. For coupling application

programs, the CORBA interface and the BioDataBrowser can be used.

As a reference application, we are using MARGBench together with our simulation tool MetabSim. It accesses a special configured BioDataCache to obtain data and to build the rule network. At first, we configure a BioDataCache for storing information about metabolic networks. After the data structures for the cache are denoted, the BioDataCache Installation Tool is used to compile and install an integration database (cache) called MetabNets. From the tool we get a CORBA interface for accessing the data structures defined before. We can use this interface to submit database queries and to operate on cache objects by creating, modifying and deleting operations. Data structures of MetabNets are for instance the classes metabolism, pathway, reaction and enzyme. Once the cache scheme is defined, the user can start the integration procedure. During this process the BioDataServer is queried and objects of MetabNets are instantiated with the query results.

The data structures in the MetabSim program are defined in CORBA too. What we have to do now is to implement a mapping algorithm to produce rules out of the MetabNets objects. The program containing this algorithm accesses the BioDataBrowser and the BioDataCache. In this context, the browser is used for

the interactive selection of metabolic pathways and single reactions. An example for enzyme information of the BioDataBrowser is shown in Figure N.9. For the data exchange between BioDataBrowser and application an active interface is available. The application program (MetabSim) implements this data exchange interface. When the user selects one or more objects from the cache the data exchange interface is called and the algorithm in MetabSim processes the transformation into MetabSim rules.

In order to handle large networks, all entities of the MetabSim simulation model are stored in a standard object-oriented database system. The MetabVis program is the front-end of the simulation. With this graphical user interface, one can initiate the import of MetabNets data, modify the rule, run the simulation and display the results. As an example for the result of an access to different databases Figure N.10 presents the gene regulation system of the CRP-operon in E.coli.

N.7 Discussion

We are living at the beginning of the century of Biotechnology. The progress of this new technology depends on the application of methods and concepts of computer science, because the exponential growth of experimental data must be handled. In other words, we

have to implement molecular database systems and analysis tools. Apweiler et al. shows in chapter ***APWEILER et al.*** that more than 200 molecular database systems and hundreds of analysis tools are available via the Internet. For the analysis and synthesis of molecular processes the integration of database systems is important. Therefore, the main goal of Bioinformatics is to develop and implement the information technological infrastructure for Molecular Biology.

Different companies are already on the market and the most important are: Human Genome Sciences (USA), Celera Genomics (USA), INCYTE Genomics (USA), Lion Bioscience (D) and Informax (USA). The backbone of such information systems is the integration of heterogeneous molecular database systems and analysis tools.

In our chapter we describe the current methods of database integration. Moreover, we describe the architecture of our molecular information system for the analysis and synthesis of gene controlled metabolic networks. Therefore, we implemented an integration shell which allows the semi-automatic implementation of a user specific integration database which represents an information fusion process based on the integration of different molecular database systems. In our implementation we integrated, as a case study, seven different molecular database systems and our rule based

simulation tool MetabSim, which allows the simulation of gene controlled metabolic networks.

Our rule based method is easy to handle and allows also the abstract simulation of analytical effects. We do not have the vision of the virtual cell like Tomita (see chapter ***TOMITA et al.***), because our theoretical results show that the complexity of the simulation of the complete metabolic processes is exponential. That means that only parts of the metabolism can be calculated. However, simulation is and will be one important point to understand the function of gene regulated metabolic pathways. Moreover, the algorithmic analysis of metabolic networks must be achieved. Chapter ***DANDEKAR et al.*** and chapter ***KOLCHANOW et al.*** are showing the state of the art of the algorithmic analysis of metabolic networks. To understand the logic of life we have to develop algorithms for the calculation of alignments of metabolic pathways or for the prediction of metabolic pathways based on rudimentary knowledge. Finally the analysis process needs information systems which integrate analysis tools and simulation environments based on the integration of molecular database systems.

Acknowledgements

This work is supported by the German Research Council (DFG) and by the German Volkswagen Foundation.

References

Baker, W., van den Broek, A., Camon, E., Hingamp, P., Sterk, P., Stoesser, G., and Tuli, M.A. (2000). The EMBL Nucleotide Sequence Database. *Nucleic Acids Research*, 28(1):19-23.

Bork, P., and Bairoch, A. (1996). Go hunting in sequence databases but watch out for the traps. *Trends in genetics*, 12(10):425-427.

Brutlag, D.L., Galpher, A.R., and Millis, D.H., (1991). Knowledge-based simulation of DNA metabolism: prediction of enzyme action. *CABIOS*, 7(1):9-19.

Burks, C. (1999) Molecular Biology Database List. *Bioinformatics*, 27(1):1-9.

Cattell, R.G. (1994). *The Object Database Standard: ODMG-93*. Morgan Kaufmann Publishers, San Mateo, CA.

Codd, E.F. (1982). Relational Database: A Practical Foundation for Productivity. *Communications of the ACM*, 25(2):109-117.

Conrad S. (1997). *Federated Database Systems: Concepts of Data Integration*. Springer-Verlag, Berlin/Heidelberg, (In German).

Davidson, S.B., Overton, C., Tannen, V., and Wong, V.

(1997). BioKleisli: a digital library for biomedical researchers. *International Journal on Digital Libraries*, 1:36-53.

Etzold, T., Ulyanow, A., and Argos, P. (1996). SRS: Information Retrieval System for Molecular Biology Data Banks. *Methods in Enzymology*, 266:114-128.

Franco, R., and Canela, E. (1984). Computer simulation of purine metabolism. *European Journal of Biochemistry*, 144:305-315.

Frishman, D., and Mewes, H.W. (1997). PEDANTic genome analysis. *Trends in Genetics*, 13(10):415-416.

Hofestadt, R. (1996). *Theorie der regelbasierten Modellierung des Zellstoffwechsels*. Aachen: Shaker, (In German).

Hofestadt, R., and Scholz, U. (1998). Information Processing for the Analysis of Metabolic Pathways and Inborn Errors. *BioSystems*, 47(1-2):91-102.

Hofestadt, R., and Thelen, S. (1998). Quantitative Modeling of Biochemical Networks. *In Silico Biology*, 1(1):39-53.

Kemp, G.J., Robertson, C.J., and Gray, P.M. (1999).

Efficient access to biological databases using CORBA. *CCP11 Newsletter*, 3.1(7).

Kohn, M.C., and Letzkus, W. (1982). A Graph-theoretical Analysis of Metabolic Regulation. *Journal of Theoretical Biology*, 100:293-304.

Luttgen, H., Rohdich, F., Herz, S. Wungsintaweekul, J., Hecht, S., Schuhr, C.A., Fellermeier, M., Sagner, S. Zenk, M.H., Bacher, A., and Eisenreich, W. (2000). Biosynthesis of terpenoids: YchB protein of Escherichia coli phosphorylates the 2-hydroxy group of 4-diphosphocytidyl-2C-methyl-D-erythritol. *Proc. National Academy of Science USA*, 97(3):1062-1067.

Matsuda, H., Imai, T., Nakanishi, M., and Hashimoto, A. (1999). Querying Molecular Biology Databases by Integration Using Multiagents. *IEICE TRANS. INF. & SYST.*, E82-D(1):199-207.

Ogata, H., Goto, S., Kazushige, S., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto Enzyclopedia of Genes and Genomes. *Nucleid Acids Research*, 27(1):29-34.

OMG - Object Management Group, (1991). The Common Object Request Broker: Architecture and Specification, OMG

Document Number 91.12.1.

Overbeek R. (1992). Logic Programming and Genetic Sequence Analyzis: a Tutorial. In *Proceedings of Logic Programming*. Chicago: MIT Press.

Pfeiffer, T., Sánches-Valdenebro, I., Nuño, J.C., Montero, F., and Schuster, S. (1999). METATOOL: for studying metabolic networks. *Bioinformatics*, 15(3):251-257.

Reddy, V.N., Mavrovouniotis, M.L., and Liebmann, M.N. (1993). Petri Net Representations in Metabolic Pathways. In L. Hunter, D. Searls, and J. Shavlik, editors, *ISMB-93 Procceedings: First International Conference on Intelligent Systems for Molecular Biology*, pages 328-336. Menlo Park et al: AAAI Press / The MIT Press.

Senger, M., Glatting, K.-H., Ritter, O., and Suhai, S. (1995). X-Husar, an X-based graphical interface for the analysis of genomic sequences. *Computer Methods and Programs in Biomedicine*, 46(2):131-142.

Stoffers, H.J., Sonnhammer, E.L., Blommestijn, G.J., Raat, N.J., and Westerhoff, H.V. (1992). METASIM: object-oriented modeling of cell regulation. *CABIOS*,

8(5):443-449.

Tomita, M., Hashimoto, K., Takahashi, K., Shimizu, T.S., Matsuzaki, Y., Miyoshi, F., Saito, K., Tanida, S., Yugi, K. Venter, J.C., and Hutchison, C.A. (1999). E-CELL: software environment for whole-cell simulation. *Bioinformatics*, 15(1):72-84.

Topaloglou, T., Kosky, A., and Markowitz, V. (1999). Seamless Integration of Biological Applications within a Database Framework. In T. Lengauer, R. Schneider, D. Boork, D. Brutlag, J. Glasgow, H.-W. Mewes and R. Zimmer, editors, *The Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB '99), Heidelberg, Germany, August 6-10*, pp. 272-281.

Waser, M., Garfinkel, L., Kohn, C., and Garfinkel, D. (1983). Computer Modeling of Muscle Phosphofructokinase. *Journal of Theoretical Biology*, 103:295-312.

Xie, G., DeMarco, R., Blevis, R., and Wang, Y. (2000). Storing biological sequence databases in relational form. *Bioinformatics*, 16(3):288-289.

Suggested Reviews / Further Reading

Bell, D., and Grimson, J. (1992). *Distributed database systems*. Addison-Wesley, Wokingham et al.

Elmasri, R., and Navathe, S.B. (2000). *Fundamentals of database systems*. Addison-Wesley, Reading, Mass. et al., 3rd edition.

Middelton, J., Jones, M.L., and Pande, G.N. (1996). *Computer Methods in Biochmechanics & Biomedical Engineering*. Gordon and Breach Publishers Amsterdam.

Stephanopoulos, G.N., Aristidou, A.A., and Nielsen, J. (1998). *Metabolic Engineering: Principles and Methodologies*. Academic Press, London.

Figure Captions

Figure N.1: Scenario of individual independent data access

Figure N.2: Scenario of data access using a database management system

Figure N.3: Integrative data access using database integration software

Figure N.4: Mechanism of the derivation process in MetabSim operators

Figure N.5: Examples for abstract rule kinetic types; from hyperbolic behavior to sigmoid dependency function

Figure N.6: Screenshot of Glycolysis graph in MetabVis

Figure N.7: Development of Substance Concentration in MetabSim

Figure N.8: Architecture of the MARGBench

Figure N.9: Enzymes stored in BioDataCache viewed with BioDataBrowser

Figure N.10: The CRP-Operon in E.coli shown with MetabVis

Glossary

CORBA

Common Object Request Broker Architecture is an open, vendor-independent architecture and infrastructure that computer use to work together over networks. It uses the Interchange Interorb protocol for communication for interoperation between CORBA programs.

DBS

A Database System (DBS) is a DBMS which includes at least one real database.

DBMS

A Database Management System (DBMS) is a collection of all special software for managing one or more databases. It is situated between databases and applications and acts as a filter, which has to fulfill a common set feature, e.g. data persistency, data scheme, efficient query and manipulating language, transactions, data consistency etc.

Formal Language

A Formal Language L over the alphabet A is a subset of A*, where * denotes the star operator. Production Systems are used for the description of Formal Languages.

IDL

Interface Definition Language is the language for describing interfaces of CORBA programs. For every CORBA implementation (binding), there is a pre-compiler to produce client and server interfaces for a given document written in IDL

IS

An information system (IS) describes a coupling between a data storage system and further data processing applications. Such an IS can be characterized by its main functions: data storage, information retrieval, data interconnection and analysis of information.

JAVA

Java(TM) is a programming language designed for the development of computer system independent software. To execute in Java programmed applications an interpreter (virtual machine) is required. Today, Java programs are frequently used in the context of the WWW.

JDBC

The JDBC (TM) technology is an application programming interface that lets you access any relational data source from the Java (TM) programming language. It

provides cross-DBMS connectivity to a wide range of SQL databases.

OQL

Object Query Language is an object query language, which supports the data model of the Object Database Management Group (ODMG) and is comparable with SQL.

PHP

PHP, which stands for "PHP: Hypertext Preprocessor", is a HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features included. The goal of the language is to allow web developers to write dynamically generated pages quickly.

Production System

A Production System is defined by a finite set of production rules and a finite alphabet A. Production rules are pairs of words over the Alphabet A*, which define the derivation of the production system. A word w over A* will be derived by a production rule (u, v), iff the first component of the production rule (u), which will be a sub-word of w, will be substituted by v.

QBE

This non-procedural language developed by IBM(TM) was designed for textual querying and manipulating of relational DBS. QBE means "Query by Example" and is implemented as an intuitive query construction using tables.

Semi-Thue System

A Semi-Thue system is a specific Production System, which is equivalent to the Chomsky Typ-0 grammar.

SQL

The Structured Query Language (SQL) is a non-procedural language for interacting with a relational DBMS. It allows the user to query, manipulate and define data, respectively data structures.

XML

The Extensible Markup Language (XML) is the universal format for structured documents and data. It is well defined by the World Wide Web Consortium (W3C) and has become a standard for text based data exchange.