TM-1080

Protein Function Database as a Deductive and Object-Oriented Database

by
H. Tanaka

August, 1991

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

Protein Function Database as a Deductive and Object-Oriented Database

Hidetoshi Tanaka

Institute for New Generation Computer Technology (ICOT)
1-4-28, Mita, Minato-ku, Tokyo 108 Japan
htanaka@icot.or.jp

Abstract

This paper describes an experiment on a knowledge base approach to molecular biological databases, especially focusing on a protein function database.

Although there are many databanks of DNAs and proteins to support research in molecular biology, their integrated database is urgently requested, because redundancies and gaps among them prevent effectiveness of such data and knowledge.

For the integrated database, we take an approach to write various data and knowledge in a single knowledge representation language, *QUIXOTE*, which is designed at ICOT for deductive and object-oriented database (DOOD). As a protein function database is a typical one with complex data and inference rules, we start to describe it in *QUIXOTE* as a part of the integrated database.

This paper describes motives for our experiments, status quo of biological database researches, features of *QUIXOTE*, the concept of a protein function database, examples of its representation in *QUIXOTE*, and evaluation.

1 Introduction

New concepts of database management systems (DBMS) are anxiously awaited in molecular biology. DBMS are used for storing and statistically analyzing data, both of which play important roles in molecular biological research. When we treat molecular biological data, we find that the traditional relational database model is not suitable for representing them. DNA sequences vary greatly in their length, and the feature descriptions of their regions overlap and are complicated. Amino acid sequences are generally shorter than DNA, but they have complex secondary and 3-D structures. The traditional model has many restrictions in itself to prevents natural representation of such data.

Besides, an integrated database of molecular biology is urgently required. There are various kinds of databanks* in molecular biology so that every time biologists have to use a new one, they are annoyed at understanding characteristics of each attribute: name, definition, syntax, and access methods.

In order to design an effective integrated database by means of a new concept, we have to do two works. Firstly, we should translate each existing databank into a database in new DBMS, so that syntactic mismatches would disappear. It is also important to discuss what kind of schema we should choose for them, or how to parse valuable knowledge written in their comments. Secondly, we have to construct a knowledge base which has various knowledge items supplementary to them, so that semantic mismatches would be reduced. For example, if a certain attribute had its own rule to access data, or another table which explains meanings of its values, they act as knowledge to make its semantics clear.

At ICOT, we have several projects related to molecular biology, including the development of DBMS and the design of a knowledge representation language [12]. We have developed a DBMS, Kappa[17], based on a nested relational model, where GenBank and PIR are stored, and will develop some applications to help biologists. We are also developing a deductive and object-oriented database (DOOD)[16] language named *QUIXOTE* [13].

Our research shows that it is not difficult to translate databanks whose schema are so concrete as ones mentioned above, into both Kappa and *QUIXOTE*.

Thus I proceed to the second issue, through providing formal function descriptions of DNAs or proteins as knowledge items. Function descriptions are what biologists really want to get and store, though it is not usefully acceptable in the existing databanks. These are so important to be written in a formal language, to be used in reasoning.

I choose a protein function database to build in a DOOD concept, as a first step to see following two points. First point is what kinds of knowledge are really useful for molecular biologists. I should provide an example knowledge base to get answers of this question. A protein function database seems useful in itself, in biochemistry or protein engineering. Second point is how efficiently we could adapt this new concept to molecular biological knowledge. One of the features of *QUIXOTE* is natural representation and inference of complex objects. It seems a good example that representing chemical reactions which weave complicated networks.

This paper describes a tiny example of a protein function database in *QUIXOTE*, as an ingredient of an integrated knowledge base of molecular biology. Through this experiment I show the possibility of realization of integrated knowledge base as well as the usefulness of *QUIXOTE*.

I make a brief survey of molecular biological databases in Section 2, and explain how to represent the functions of proteins and show the configuration of the protein function database in Section 3. Introduction to the concepts and the features of *QUIXOTE* is shown in Section 4, and an experimental description of protein functions, electron transfer through cytochromes for example, in Section 5. Section 6 is for the evaluation of the protein function

*In order to avoid semantical overloading of a term "database", I use a term "databank" for a collection of data and "database" for data related to a data model or DBMS.

database in *QUIXOTE* at the present and the future works, and Section 7 is for concluding remarks.

2 Molecular Biological Databases

I make a brief survey of researches in molecular biological databases. There are two points as I mentioned: new data models and integration of databases.

2.1 Data Models for Existing Databanks

Existing databanks [6] fall into roughly four categories: sequence, structure, map, and function.

The main problem in building the data model for sequence databases is how to represent feature descriptions of sequences. A relational model[2], CYC and interval calculus[9], and a nested relational model[17] have been tried.

How to represent "motifs" is another important issue. A motif is a sub-sequence of DNAs or amino acids representing a certain feature. It is usually represented in a syntax like regular expression at the present[11].

The data model for protein structure databank is also discussed: a deductive database[4] and an object-oriented database with a functional data model[3].

No adequate data model has been proposed for either map or function databases. Relational DBMS are usually used for them, enduring inconveniences.

Among data models mentioned above, object-oriented one is the most suitable to represent all databanks commonly. But provided that we require not only syntactic integration but also semantic one, the data model needs a certain mechanism to describe various relations between any attributes or any values easily.

I propose to use a deductive and object-oriented data model, which allows us to write any rules of attributes or values. It could be used efficiently in representing most data and rules in molecular biology.

2.2 Researches on Integrated Database

There are two kinds of trials to conquer impedance mismatches and realize an integrated database.

One is standardization. CODATA (Committee on Data for Science and Technology) in ICSU (International Council of Scientific Unions) proposed standardization of attributes so that it seems like one large database, which really consists of many databases[10]. NLM (National Library of Medicine) provides GenInfo Backbone Database[8]. They are built as a standardized primary databases, which are assumed to be a basis for secondary, value-added databases for each interest of biologists. In that sense, really convenient databases should be built by each biologist.

I choose another approach, to make an integrated knowledge base. It consists of two stages: to represent all facts in one language, and to supplement rules necessary to access facts, in the same language. The former corresponds to standardization, and realizes a syntactically integrated database. In the latter stage, supplementary knowledge items such as access methods for attributes or another table which explains its values as to relations, types, and other constraints or rules are provided.

A DOOD concept used in *QUIXOTE* has enough power to represent every data in existing databanks and most rules for them. Building a protein function database belongs to both stages. It is an ingredient of integrated database, as well as supplementary knowledge base for protein sequence or structure database.

3 Protein Function Database

A protein function database is the first step in the second stage toward an integrated knowledge base of molecular biology. I focus to represent chemical reactions, not only reactions among compounds but also relations among reactions. I show the reason why I choose chemical reactions, outline of an example, and how I design a total protein function database in *QUIXOTE*.

3.1 Functions of Proteins

Proteins grouped by functions[5] and the corresponding properties to describe their functions are shown as follows:

Proteins:	Properties:
Enzymes	substrates, products, coenzymes, environments (ions, temperature, ...)
Transport	object, from, to, environments
Nutrient and Storage	place, environments
Contractile or Motile	place, environments
Structural	place, environments
Defense	object, place, environments
Regulatory	object, place, environments
Others	(?)

Function descriptions of proteins other than enzymes are expressed by names and environments, though the names vary in their meanings. I choose a methodology for representing enzyme function as a main issue. It is no more than chemical reaction expressions and the relations among them. They include the information of enzyme functions implicitly.

3.2 Reactions and Relations of Reactions

A scheme of a chemical reaction among compounds, concerning enzymes and co-enzymes is shown in the expression in Fig. 3.1. As all the attributes are multi-valued, even with an infinite structure, it is necessary to represent in complex objects. A proper frame-like language allows chemical reactions to be represented in the scheme in Fig. 3.1. It is enough to describe its static information.

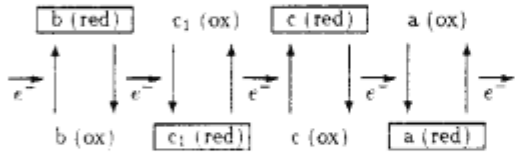
$$\text{Substrates} + \text{Coenzymes} \xrightarrow[\text{Environments}]{\text{Enzymes}} \text{Products}$$

```
Chemical Reaction : Substrates = { ... }
                    Coenzymes  = { ... }
                    Enzymes    = { ... }
                    ....
                    Products   = { ... }
```

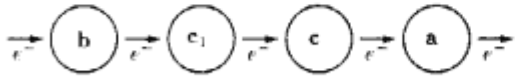
Fig. 3.1 Scheme of Chemical Reactions

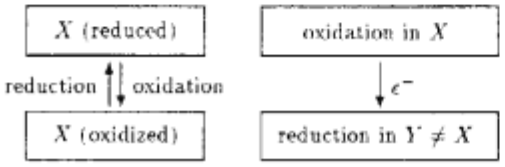
The relations among reactions, such as electron transfer through cytochromes or the relation between catabolism and anabolism, is harder to describe than the reactions themselves. If

we regard only compounds as entities, it would be a complicated description (see Fig. 3.2-(1)). It becomes easier however, once we regard each sort as an entity, such as cytochrome b or cytochrome c_1, and then regard each reaction as an entity, and so on (see Fig. 3.2 (2)). And it is sure that both representation are necessary, if we expect to get answer in both lower level (such as "oxidized cytochrome b") and higher level (such as "cytochrome b").



(1) All-in-one description





(2) Step-by-step description

Fig 3.2 Electron Transport through Cytochromes

Chemical reactions and the relations among them can be written in a frame-like language or scheme if we focus on the ability to answer questions on the function of the enzymes, namely static information as shown above. But once we have to get the relations among compounds or proteins, deductive features are mandatory, in tracing their connections or deducing from primitive knowledge such as a "step-by-step description" to appropriate knowledge such as an "all-in-one description" as shown in Fig. 3.2.

Hence we think DOOD is suitable, for it requires both object-oriented (or frame-like) and deductive features to represent molecular biological knowledge.

3.3 Design of a Protein Function Database

Fig. 3.3 shows the configuration for a protein function database which I plan to build in *QUIXOTE*. *QUIXOTE* has a concept of modules (see Section 4), so each database module is designed to be independent.

In this system, GenBank and PIR are provided in the simplest schema, as near to the schema of original databanks as possible. The protein function database consists of several sub-modules: chemical reactions, relations among them, and the others that were enumerated before. The interface module shown in Fig. 3.3 inherits all other modules and is engaged in query processing. It has rules for getting questions, transforming them into proper queries, getting knowledge items in the implicit forms, and transforming those into proper answers.

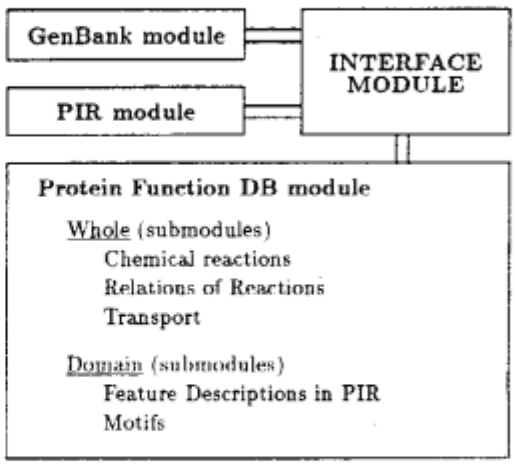


Fig 3.3 Protein Function Database in *QUIXOTE*

4 *QUIXOTE* : a DOOD Language

In [13] a new database language *QUIXOTE* has been proposed as a knowledge representation language, whose ancestors are languages for deductive and object-oriented databases [15] and natural language processing applications. It is shown more precisely in [14] and [7].

4.1 Object Terms and Properties

In *QUIXOTE*, an object consists of the identifier named an object term and the attributes. Each attribute of them is a triple of a label, an operator and a value. An object term consists of the head and the attributes. For example:

$$head[lo1 = vo1, lo2 \leftarrow vo2, \ldots].$$

where $lo1$ and $lo2$ are labels, $vo1$ and $vo2$ are values, and $=, \leftarrow$ are operators. Values are also object terms so that complex objects can be represented. Object terms are partially ordered.

An object term can have attributes as follows:

$$o/[la1 = va1, la2 \rightarrow va2, \ldots].$$

where o is an object term, $la1$ and $la2$ are labels, $va1$ and $va2$ are values, and $=, \rightarrow$ are operators. Such a term is called an attribute term. The right hand side of / is a list of attributes. According to the order of object terms, attributes with $=$ or $\rightarrow$ are inherited from upper objects to the lower objects. As attributes in an object term cannot be updated, exception of inheritance is occurred at the object.

4.2 Rules

In *QUIXOTE*, rules to create new objects can be written as follows:

$$new\_object \Leftarrow object, object, \ldots.$$

A new object is created if all the objects on the right hand side exist.

4.3 Constraints

In *QUIXOTE*, an object term, an attribute term, and a rule can be written in a cannonical constraint form. For example, any rule can be written as follows:

$$o \mid \{c_0, \ldots\} \Leftarrow o_1, o_2, \ldots \mid\mid \{c_1, c_2, \ldots\},$$

where $\{c_0, \ldots\}$ is a set of constraints of o, and $\{c_1, c_2, \ldots\}$ is a set of constraints of $o_1, o_2, \ldots$. Conditions in a chemical reaction can be written as a part of such constraints.

4.4 Modules

QUIXOTE has a concept of modules, each of which is a set of objects and rules. A submodule relation between modules makes rule inheritance possible. An object can have different states or properties in each module. Modules are written as follows:

module_id :: object.
module_id :: rule.

4.5 Features

A DOOD language *QUIXOTE* has various features as follows:

(1) DDB	deductive features
(2) Inheritance	inheritance mechanism in objects
(3) Module	features of modules
(4) Object-ID	representation of object identity
(5) Constraints	representation of constraints
(6) Infinite Terms	representation of circular structures
(7) Exceptions	representation of exceptions

This list is explained in more detail in Section 6.

5 Representation in *QUIXOTE*

I have tried several experimental descriptions of molecular biological knowledge, especially protein function, in *QUIXOTE*. In this section, I show a small one which includes a lot of issues to discuss. I choose electron transfer through cytochromes for this purpose.

5.1 Electron Transfer

The respiratory chain of mitochondria contains a large number of electron carrying proteins, such as cytochromes. They act in sequence to transfer electrons from substrates to oxygen. The cytochromes are iron-containing electron transferring red or brown proteins that act in sequence to carry electrons from ubiquinone to molecular oxygen. They undergo oxidation and reduction as they carry electrons. There are three classes of cytochromes, a, b and c, distinguished by differences in their light-absorption spectra. Each cytochrome in its ferric [Fe(III)] form accepts one electron to become the ferrous form [Fe(II)]. [5]

I simply showed the sequence and the mechanism in Fig. 3.2.

5.2 Electron Transfer in *QUIXOTE*

I described the electron transferring through cytochromes in *QUIXOTE* as simply as possible in Fig. 5.1.

```
-------------------------------------------------------
m_1 :: {{
  e_transfer [l_donor= cyto_b, l_acceptor= cyto_c1].
  e_transfer [l_donor= cyto_c1,l_acceptor= cyto_c ].
  e_transfer [l_donor= cyto_c, l_acceptor= cyto_a ].

  cyto_b  =< cytochrome. %% cyto_b is a cytochrome.
  cyto_c1 =< cytochrome.
  cyto_c  =< cytochrome.
  cyto_a  =< cytochrome.
}}

m_2 >- m_1. %% m_1 is a submodule of m_2.
m_2 :: {{
  oxidation [l_object = X]
     <=e_transfer [l_donor = X, l_acceptor = Y].
  reduction [l_object = X]
     <=e_transfer [l_acceptor = X, l_donor = Y].

  oxidation =< reaction.
  reduction =< reaction.
  reaction.

  oxidation [l_object = X] /
       [l_source+ <- Y, l_product+ <- Z]||
       {Y =< X, Y!l_type = reduced, Z =< X,
        Z!l_type = oxidized}.
                        %% Y / [l_type = Y!l_type]
  reduction [l_object = X] /
       [l_source+ <- Y, l_product+ <- Z]||
       {Y =< X, Y!l_type = oxidized, Z =< X,
        Z!l_type = reduced}.

  cytochrome [l_type = reduced ].
  cytochrome [l_type = oxidized].
}}
-------------------------------------------------------
```

Fig. 5.1 Electron Transfer through Cytochromes

The module m_1 describes only the flow of an electron, while m_2 shows its mechanism in detail.

In m_1, there are declarations on objects in this module: three objects under "electron_transfer" and four objects under "cytochrome". Attributes in a lower object (ex. cytochrome_b) inherit their values from the corresponding attributes in the upper object (ex. cytochrome).

In m_2, there are four groups of rules or facts. The first group contains rules that the oxidation occurs in the donor of electron transfer and the reduction in the acceptor. The second group contains declarations of oxidation, reduction and reaction. The third group shows simplified mechanisms of oxidation and reduction in cytochromes. The last group contains declarations of reduced type and oxidized type of cytochromes. The declaration that objects under cytochrome have both types is included.

If we ask module m_1 how the electron flows, it answers:

```
-------------------------------------------------------
<Q>
? m_1 : e_transfer [l_donor = X, l_acceptor = Y].

<A>
e_transfer [l_donor = cyto_b,  l_acceptor = cyto_c1].
e_transfer [l_donor = cyto_c1, l_acceptor = cyto_c ].
e_transfer [l_donor = cyto_c,  l_acceptor = cyto_a ].
-------------------------------------------------------
```

And if we ask m_2 its mechanism, it answers as follows: (Temporary rules can be put to queries)

```
------------------------------------------------
<Q>
? m_2 : e_transfer / [l_fromR = X, l_toR = Y].
  m_2 :: {{
    e_transfer [l_fromR = reduction [l_object = X],
                l_toR   = oxidation [l_object = Y] ]
          <= e_transfer [l_donor = X, l_acceptor = Y].
  }}
<A>
e_transfer [l_fromR = reduction[l_object = cyto_b],
            l_toR   = oxidation[l_object = cyto_c1] ].
e_transfer [l_fromR = reduction[l_object = cyto_c1],
            l_toR   = oxidation[l_object = cyto_c] ].
e_transfer [l_fromR = reduction[l_object = cyto_c],
            l_toR   = oxidation[l_object = cyto_a] ].
------------------------------------------------
```

6 Evaluation

I evaluate this work from two points of view: the needs, namely from a molecular biological point of view and the seeds, namely from a viewpoint of *QUIXOTE*.

6.1 Influences on Molecular Biology

This paper describes only about chemical reactions in *QUIXOTE*, which is so small part of our target, an integrated knowledge base, that I cannot discuss a lot from that point. Thus I evaluate only the result of Section 5.

For the molecular biology, the major benefit is that data which was difficult to store in databases can now be stored. It is also easy to retrieve whatever data we want when we use *QUIXOTE*. It is powerful in knowledge description and enables us to retrieve complex data by means of the deductive mechanism.

The following are instances that can now be stored, other than chemical reactions.

(1) Formal descriptions of features in sequence databases

The feature descriptions in existing databanks are written mostly in English, not in formal languages. We can put an object term of *QUIXOTE* instead of English words, such as:

English: "essential myosin light chain, exon 1"

QUIXOTE: "myosin[l_1 = essential, l_2 = light_chain, l_3 = exon, l_4 = 1]." or "essential_myosin[l_2 = light_chain, l_3 = exon, l_4 = 1]."

(2) Positions in features

Another issue is in describing positions of regions. Recent GenBank allows operators such as "join" or "one-of" to describe a region separated in several subregions. It is hard to represent naturally in both relational and nested relational models, while deductive representation is very easy in *QUIXOTE*. Overton et al. also show the need for hierarchical representation in feature description and efficiency of an object-oriented database [9].

(3) Motifs

Each protein has domains which have certain functions. One of the main issues related to protein research is what kind of amino acid sequence represents a certain function. The sequences whose functions are known are called functional motifs. We also have structural motifs, which represent the local structure of proteins.

Many of both structural and functional motifs are known so far, but they are also written in some patterns, which are difficult for programs to use. [11].

Motifs have their own syntax which we have to remember to use them. So we could easily use motifs if we have the methods to access them or how to read them in a class and store motif data as its subclasses.

6.2 Features of *QUIXOTE*

This table shows the features of *QUIXOTE* (see Section 4) and their usages and aims in this experiment.

Features	Usages	Aims
(1) DDB	relations among reactions reaction paths	primitive →appropriate trace paths
(2) Inheritance	sorts	reduce amount
(3) Module	knowledge modules	check inconsistency
(4) Object-ID	objects & modules	share objects
(5) Constraints	body of rules queries	represent a domain
(6) Infinite Terms	—	—
(7) Exceptions	—	—

Generally speaking, deductive mechanism enables to deduce appropriate knowledge from primitive knowledge, constraint representation is used in writing conditions efficiently, object identity is useful in sharing objects, and inheritance and the module reduce the amount of knowledge description, while the representation of exceptions and complex structure are not used well at the present.

(1) Deductive feature

We can deduce appropriate knowledge from primitive knowledge, so that we can store knowledge in primitive style. As I mentioned in Section 3, it reduce complicated knowledge such as electron transfer into several simple facts.

It is also useful when we search various paths, especially together with infinite term to represent a path including circles. But I defer this issue.

(2) Inheritance mechanism in object terms

The inheritance mechanism is used to reduce the amount of description of each object. For example, "cytochrome c" inherits all the properties of "cytochrome" so we can omit those properties in all objects under "cytochrome".

(3) Modules

It is important to make small modules of knowledge, so that we can easily check them and keep inconsistency within the module. As I mentioned in Section 3, we can store GenBank and PIR as modules, provide an interface module for them, add our own protein function database as another module and so on, without any trouble.

(4) Object identity

The object identifier in *QUIXOTE* has its name and property with which users can find a specific object from a pool of objects. It enables sharing a persistent object among programs. It makes compound objects to be represented efficiently.

(5) Constraints

Constraints are useful for writing complex attribute values such as specification of the range of a value, and condition to be satisfied in the body of a rule.

(6) Infinite Terms

It would be useful when we represent various paths including circles, for we can use infinite terms. But I defer this issue.

(7) Exception

I have not found any proper examples in molecular biology, though the well known example of birds and penguins are in a very near subject.

6.3 Future Works

There are three directions to proceed our researches.

(1) An useful tool for biologists

It is important for biologist's use to process quantity information. The chemical reaction has various quantity information, or stoichiometrical issues such as coefficient of each compounds or differences of free energy. Brutlag et al. employ the KEE system for this purpose[1].

Besides, We should estimate the performance of *QUIXOTE*, and increase it.

(2) A more useful protein function database

Motifs are indispensable to describe functions of domains. We should look for the best way to represent them, to search the corresponding sequences efficiently.

Representation of complex structures, especially circular structures, should be tried and evaluated to see whether they are useful or not, for representing the result of searching various paths.

(3) An integrated knowledge base

It is important to increase its ingredients. For example, a genome map should be the next target to make an experimental representation.

And it is also important to increase its supplementary knowledge. We should research the way to extract such knowledge or rules, from biologists or from facts in databanks.

7 Concluding Remarks

I show the effectiveness and the possibility of an integrated knowledge base of molecular biology, through an example of a protein function database in *QUIXOTE*.

I also introduced a DOOD language *QUIXOTE*, designed at ICOT. Its great expressive power allows facts to be described in so many ways, that we are discussing a standard format for description. *QUIXOTE* is suitable for describing data and knowledge which could not be described before, and enables us to use them through programs.

Acknowledgements

The author wishes to thank Kazumasa Yokota, Hideki Yasukawa,and other people in the *QUIXOTE* project for their valuable comments on earlier versions of the paper, and people on the computer-biology mailing list for their many suggestions from the viewpoint of biology.

References

[1] Brutlag, D.L., Galper, A.R. and Millis, D.H.: "Knowledge-based Simulation of DNA Metabolism: Prediction of Enzyme Action", *CABIOS* No.7, Vol.9 (1991).

[2] Cinkosky, M.J. et al.: "GenBank/HGIR Technical Manual", *LA-UR* 88-3038 (LANL) (1988).

[3] Gray, P.M.D. et al.: "An Object-Oriented Database for Protein Structure Analysis", *Protein Engineering*, vol.3 no.4, pp.235-243 (1990).

[4] Kuhara, S., Satou, K., Furuichi, E. and Takagi, T.: "A Deductive Database System PACADE for Three Dimensional Structure of Protein", *Draft* (1990).

[5] Lehninger, A.L.: *Principles of Biochemistry*, Worth Publishers Inc. (1982).

[6] Lesk, A.M. (editor) : *Computational Molecular Biology, Sources and Methods for Sequence Analysis*, Oxford Univ. Press (1988).

[7] Morita, Y., Haniuda, H. and Yokota, K.: "Object Identity in *QUIXOTE* ", *IPSJ SIGDBS*, No.80 (Nov 1990).

[8] NCBI: "GenInfo Backbone Database", Version 1.59, *Druft* (Apr 1990).

[9] Overton, G.C., Koile, K. and Pastor, J.A.: "GeneSys: A Knowledge Management System for Molecular Biology", *Computers and DNA, SFI Studies in the Science of Complexity*, vol.VII, Addison-Wesley, pp.213-239 (1990).

[10] PIR-International (JIPID): *PIR Newsletter*, No.3 (June 1990).

[11] Trifonov, E.N. and Brendel, V.: *Gnomic - A Dictionary of Genetic Codes*, Balahan Publishers.

[12] Uchida, S. and Yoshida, K.: "The Fifth Generation Computer Technology and Biological Sequencing", *Proc. of Workshop on Advanced Computer Technologies and Biological Sequencing* (Nov 1988).

[13] Yasukawa, H. and Yokota, K.: "The Overview of a Knowledge Representation Language *QUIXOTE* " *Draft*, (1990).

[14] Yasukawa, H. and Yokota, K.: "Labeled Graphs as Semantics of Objects", *IPSJ SIGDBS*, No.80 (Nov 1990).

[15] Yokota, K.: "The Outline of a Deductive and Object-Oriented Language: Juan", *IPSJ SIGDBS*, No.78 (Jul 1990).

[16] Yokota, K. and Nishio, S.: "Toward Integration of Deductive Databases and Object-Oriented Databases: A Limited Survey", *Proc. of Advanced Database System Symposium*, Kyoto (Dec 1989).

[17] Yokota, K. and Tanaka, H.: "GenBank in Nested Relation", *Joint Japanese-American Workshop on Future Trends in Logic Programming* (Oct 1989).