Constituting a Receptor-Ligand Information Base from Quality-Enriched Data

Klemens Hemm*, Karl Aberer*, Manfred Hendlich**

*GMD–IPSI, Dolivostr. 15, 64293 Darmstadt, Germany,
e–mail: {hemm, aberer}@darmstadt.gmd.de
**current address: E. Merck, Preclinical Research, 64271 Darmstadt, Germany,
e–mail: hendlich@merck.de

Abstract

Many different resources are needed for analyzing relevant experimental data in drug design. Currently this data is difficult to access, because it is stored in heterogeneous databases, spread over many platforms, poorly interconnected, incomplete, erroneous, or just not electronically available. In order to establish a high quality database for drug design we have developed a new demand-driven methodology for integrating and semantically enriching heterogeneous data from different research areas and for migrating the data into an object-oriented database management system. In this way we have established a database containing well-prepared, relevant data needed for drug design and offering the advantages of modern database technology, like a comprehensive object-oriented data model, a flexible declarative query language and support for persistent storage and sharing of data in a multi-user environment.

1 Introduction

New biomolecular and computational methods have led to an explosion in the available biomolecular data. Thus the electronic management of large biomolecular databases has become an important application area of computational tools. The development of biomolecular databases is typically driven by the immediate needs of a particular research task, e.g. structure elucidation of proteins. Therefore the emerging databases are specialized and not necessarily ready to use for other researchers. For example, drug designers are interested in obtaining new insights into the interaction of proteins and ligands by analyzing existing experimental data. The relevant data for this purpose is spread over 3D structure databases, ligand databases or protein mutation databases, just to name a few.

Within the national joint project RELIWE[1], a research consortium for the "Computation and Prediction of Receptor-Ligand Interaction", in which EMBL Heidelberg and GMD participate as research institutions and BASF, Ludwigshafen and E. Merck, Darmstadt, participate as industrial partners, we aim at an integrated and flexible access to the underlying, autonomous, heterogeneous information bases and the integrated usage of different algorithmic tools. One goal is to provide an integrated database that satisfies the information needs of drug designers when they investigate the receptor-ligand docking problem.

1. This work is supported by the German "Bundesministerium für Bildung und Wissenschaft" BMBW, grant number 01IB302E.

During the analysis phase of the project it has turned out, that the existing data is spread over different platforms in different formats or is not available in electronic format at all. A common data model as well as flexible and efficient access mechanisms are missing. The data itself is often erroneous or incomplete, data in different databases is overlapping and sometimes inconsistent. Existing solutions like hard-wired applications over file-based databases do not solve these problems. We have decided to develop in close collaboration with biomolecular scientists and drug designers a system exploiting modern database technology. In this way we achieve a unique high quality database, which is of high interest for researchers in drug design and relieves them from dealing with technical details as far as possible.

As a platform we use an object-oriented database management system for two reasons. Using database management technology provides different services for free, like the support of a structured data model, flexible retrieval mechanisms and persistent storage and sharing of data in a multi-user environment. Object-oriented data models provide flexibility, expressiveness and extensibility. These features are needed to adequately represent the complex data occurring in research.

In this paper we want to report on the approach we are taking in order to integrate the existing databases and other available data. Existing approaches are not sufficient to cover our requirements. In the computer science area of database integration typically strong assumptions are made on the resource databases, like the availability of database schemas or query languages for these databases. The integration and enrichment of data is mainly treated on a schema level, while the main difficulties in our project are encountered when integrating existing data on an object

level. On a physical level these approaches are directed to virtual access to the integrated data, while we need for performance reasons direct physical access.

In the practice of biomolecular databases different approaches exist to integrate existing file-based databases. Some integrate and enrich existing databases producing a file-based database covering a particular specialized aspect, like SWISSPROT (Bairoch & Boeckmann 1991) for protein sequences. Others integrate heterogeneous databases using database management systems without semantically integrating or enriching the data, like IGD (Ritter et al. 1994) for genomic data. So we have developed our own methodology for integrating and semantically enriching heterogeneous data from different research areas, and for migrating the data into an object-oriented database management system. Conceptually we follow the so-called federated database approach (Sheth & Larson 1990), where existing databases are integrated by mapping their schemas into one common integrated schema. During the integration process additional information in terms of complex transformations, enrichments and relationships of the existing data need to be specified, which can only be done using a lot of specialized expert knowledge. Our methodology allows the experts to specify this knowledge in an easy yet formal way, such that an automated process can derive the integrated database from this information.

We use the object-oriented DBMS VODAK developed at GMD-IPSI as the basis for our prototype. It is the first database that exploits the advantages of object-oriented database management technology for realizing a value-added, integrated database for research on drug design. First results show that the major benefits for the user are the provision of a semantically rich data model, mechanisms to support the integration and enrichment of existing data, and the possibility of efficient and flexible access to the data by means of a declarative query language. In particular some of the advanced features of VODAK, like extensibility of the data model as well as query language and processing, prove to be useful.

The paper is organized as follows. In Section 2 we review the current state in biomolecular databases. In Section 3 we discuss the requirements that have arisen in the RELIWE project. In Section 4 we give a short account of the VODAK database management system. Section 5 contains the core of the paper by presenting the integration methodology. In Section 6 we illustrate the approach by means of examples. We conclude in Section 7.

2 Data management in biomolecular chemistry

Biomolecular databases. Intensive research in the field of molecular biology has lead to massive amounts of heterogeneous data. Due to the fact that many of these databases have emerged from the immediate needs of particular research tasks, they often have a strong thematic binding to a specialized area. Over hundred biomolecular databases are known today. Their number and size is rapidly growing.

Typically these databases are file based, like PDB (Nishikawa et al. 1977), SWISSPROT, or PIR (Barker et al. 1991), and make no use of the benefits that modern database technology offers. A large part of the functionality, like the support of a data model, the maintenance of integrity constraints or the retrieval of data, are hard wired into application programs. With regard to these functionalities the resulting systems thus lack flexibility against modifications. This is a major drawback in a rapidly evolving research area like molecular biology. In order to advance beyond purely file-based solutions, there have been attempts to use relational database technology. SESAM (Huysmans et al. 1991) and BIPED (Islam & Sternberg 1989) are built on top of commercial relational database systems. A first step in the direction of using object-oriented database technology was made with P/FDM (Kemp et al. 1994), that is based on a functional object oriented data model and is implemented on top of PROLOG. Other object-oriented approaches use C++ class libraries for accessing flat file databases, like PDBTool (Chang et al. 1994), special purpose object storage systems, like ACEDB (Durbin & Thierry-Mieg 1994), or commercial C++ based object-oriented database management systems, like MapBase (Goodman 1995).

Integration of biomolecular databases. Most of the databases mentioned above focus on a particular kind of data, like PDB on protein structures. On the other hand these databases contain complementary and overlapping information. Thus the integration of these databases becomes a major issue.

In the database community there exist several approaches, that address the issue of database integration. Overviews can be found in (Sheth & Larson 1990), where basic terminology is introduced, e.g. multidatabase systems, multidatabase languages or federated database systems. Besides the approaches from the more general database-oriented viewpoint, there have been many projects that focus directly on the biomolecular application domain. In (Sillince & Sillince 1993) an overview over integration efforts in the field is given, especially for integrating protein structure and sequence databases.

One approach is CAN/SND (Wood et al. 1989), the Canadian Scientific Numeric Database Service, that allows common access to available online databases. Another approach is ISIS (Akrigg et al. 1988), the Integrated Sequence Structure Database, that allows access to two integrated databases, namely OWL and BIPED. OWL is an integrated sequence database from NBRF and SWISSPROT and contains translated sequences from GenBank. BIPED contains

sequence and structure data by using the underlying relational database system ORACLE. Nentrez from NCBI offers a client for local workstations that contacts an integrated database server via the internet. The server database integrates many biomolecular, medical and literature databases and offers common access to these. IGD provides an integrated genomic database (Ritter et al. 1994). Different approaches provide integration on user interface level, like XBio (Kamel et al. 1993).

3 Environmental characteristics

For the realization of our integrated database it is crucial to consider the requirements of the users, both from industrial and academic research. Our aim is to build up a *high-quality* integrated protein-ligand database to support the analysis of data, which is relevant for the investigation of the receptor-ligand docking problem. The attribute high-quality comprises two aspects of equal importance:

1. The *contents* of the database must be of high quality. This is achieved by combining different information resources, correcting the data from these resources and enriching it by data, that is electronically not available.
2. The *access* to the database must be of high quality. This is achieved by of enabling an easy understanding of the available data and operations, providing flexible retrieval capabilities to support user requests, and efficient processing of these requests.

From the existing environment the following requirements have been derived.

Use of preexisting data. We want to use preexisting data as far as possible. A minimal requirement is to provide data from the PDB, SWISSPROT, PIR protein databases and from the (proprietary) MACCS small molecule database used in pharmaceutical research. Additionally we use DSSP-files for secondary structure definitions and HSSP-files for alignments. One common property of all these public and proprietary databases is, that they are *autonomous*. This means that we cannot exert any control on these databases. We have to consider updates and versions of the external databases.

Enrichment and preparation of data. Due to lack of structure, different nomenclatures and possibly errors, for our purposes the publicly available data is not ready for use. Thus we must perform transformations, e.g. for different numbering schemes for residues in sequences and structures. Some of them can be done automatically, others must be carried out manually. Correcting and subsequently combining data from different databases is a challenging, intellectual task, that can only performed by an expert from the area. This process also includes an appropriate selection of data, as the available databases may also contain data we are not interested in.

Addition of data. Data that is not available in electronic form has to be supplemented. Some of the data must be added manually, e.g. from the scientific literature, other data is provided by applying computational analysis to existing data, e.g. computing bond information from spatial coordinates and atom types.

Considering the whole integration process *human interaction is an expensive subtask*, compared to other subtasks, that can be dealt with in an automated fashion.

Data access. The ultimate goal of the whole process is to enable a flexible and powerful access to the data of interest. This access must relieve the user from details of how the data is physically accessed. It must be supported by powerful processing mechanism, that can also efficiently deal with unanticipated requests. This is not possible in most current systems, where queries cannot freely be formulated. Data access must be based on an easily understandable, yet expressive, model of the data.

Updates. Improving, augmenting and processing the preexisting data leads to derived data in the database. Updates on external databases have to be maintained in the integrated database in a consistent way. We have also to consider updates on the integrated database, e.g. users who are storing the result of their work and who wish to use those in queries in combination with the standard database content. These updates have to be preserved in the case of propagation of updates of an external database. Immediate propagation of updates is a minor issue. As we are not aiming at a publicly available online database, it is reasonable to close down the database for expensive update operations.

4 The OODBMS VODAK

The protein ligand database is implemented on top of the object-oriented database management system VODAK release 4.0 (VODAK 1995) developed at GMD-IPSI. The object-oriented data model is accepted to be appropriate for flexible modelling of the complex structures and operational semantics, that are needed for non-standard database application scenarios like molecular biology. VODAK provides full database management system functionality, including schema compiler, transaction management, object management, data dictionary, communication management and query processing. It runs on top of ObjectStore, which is used for low-level storage.

Different features make VODAK particularly attractive for our purposes. The object-oriented data model VML (Klas et al. 1994) provides a metaclass concept for tailoring the datamodel to specific application domains. Thus semantic concepts that are characteristic for an application domain can be provided as new data modelling primitives, which extend the core data model. The declarative SQL-like query language allows the usage of method calls and path expressions within queries, and thus enables flexible and intuitive access to the database. A powerful, extensible

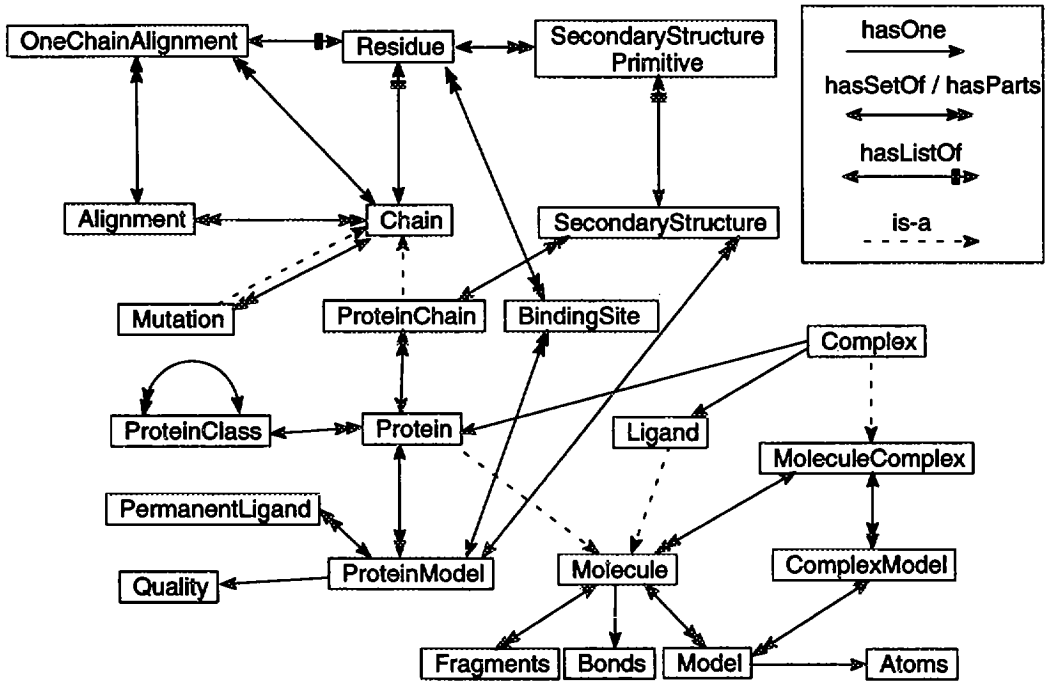

Figure 1: The conceptual database schema

query optimization module, based on algebraic optimization techniques, allows to incorporate application-specific optimization strategies, which are particularly important when complex calculations or expensive access operations to external databases are involved in declarative queries.

5 Database Integration Approach

Schema Integration. In the following we adhere to the terminology that is used in the five-level architecture of the *federated database systems approach.* Within our project first a *conceptual schema* has been defined intellectually in a collaboration of computer scientists and biomolecular and biochemical experts (Figure 1). Due to the expressiveness of the object-oriented data model it is relatively straightforward to map this model into an object-oriented *integrated database schema.* This is the target schema against which users pose queries. The different *local database schemas* of the external databases need to be translated to the common data model. This leads to the *component schema,* which provides for each external database a corresponding representation in the database system. These classes provide database methods for accessing information from external files. Finally a mapping from the component schema to the integrated schema is defined.

General Integration Methodology. The integration process is decomposed into two phases, as depicted in Figure 2. In the first phase, the *biomolecular-oriented specification and preparation phase (BP),* data is prepared under biomolecular and biochemical aspects and in the second phase, the *data-management-oriented integration phase (DP),* the data is integrated within a database management system. In the BP we want to remain as flexible as possible, thus we perform processing outside the particular DBMS in use. In the DP we make full use of the processing capabilities of the DBMS. The result of the BP is used as input to the DP and consists of a set of intermediate files. These files contain (1) *corrected data* from existing file databases, (2) *value-added data* that has been prepared computationally from existing data or made available electronically and (3) a specification of how to integrate this data into one integrated database, the so-called *import specification.* This data is independent of the database system used. Processing in the BP is inherently semi-automatic – although eventually most of the steps are handled algorithmically and manual intervention is needed for few special cases – while processing in the DP can be completely automated and is uniquely specified by the data provided in the BP.

This two phase integration methodology leads to a clear separation of application-specific and database-specific problems, which meets existing spatial and personal constraints and allows independent evolution of the techniques used in the two phases. This approach turned out to be necessary in order to achieve within the project rapid progress.

Biomolecular and biochemical data enrichment and integration. At the beginning of the BP data is either available in electronic form, on paper or in the heads of the researchers. The task within the BP is to prepare and combine

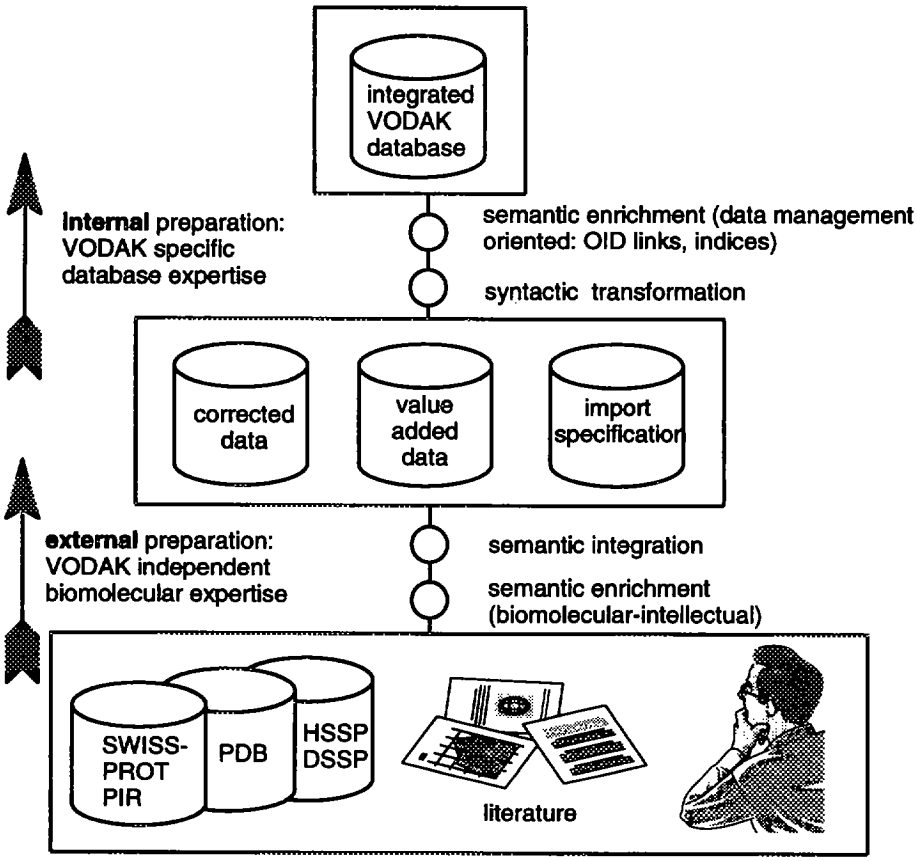

Figure 2: Two phase database integration

this information in a form and quality, allowing to apply data management techniques. This involves the subtasks of definition of the database content, i.e. identifying and selecting the data that will be needed in the integrated protein-ligand database, the addition and correction of data (needed for semantic enrichment), and the identification of relationships between data (needed for semantic integration).

The BP is a combination of intellectual and automatable steps. The question what can be handled algorithmically, and what particular algorithms can be applied, can only be answered by a person with deep expertise in the field. Therefore, from a data management point of view, at the current stage one can support this process only by providing a well-defined interface for specifying the knowledge gained in this process and provide it as input to the DP. When the project has progressed, one is in the position to look for patterns that are usable to reach a general methodology for the BP.

The syntactical representation of the corrected data and value-added data is predetermined by pragmatic constraints. Whenever there preexists a format we use this, e.g. in the case of corrected PDB data, and for the value-added data we use formats that are similar to those of existing file databases.

The import specification. The import specification is needed to define the mapping of the data in the external databases to the integrated database on an object level. For the import specification we have defined a specification language. The language is tailored to the needs of later processing, i.e. it can be easily parsed, can be edited by humans and can be easily extended. It resembles in its syntactical format the formats of the existing databases, which simplifies handling by the expert from the field. An example is given in Figure 3. It allows to express the different functionalities needed for integration. Possible actions that can be specified are

1. Generation of a new object in the integrated database.
2. Mapping of an external object to an object of the integrated database.
3. Establishment of relationships between objects of the integrated database.
4. Addition of data to objects of the integrated database.
5. Correction of attributes of objects of the integrated database.

```
# specification of a protein ligand complex
Ligand:     FCA
LigandModel:     FCA   lm1   FCA.mol2

Protein:   1abf
ProteinModel:   1abf  pm1   pdb1abf.ent     1
Chain:     1abf  -     ARAF_ECOLI.sw
SecondaryStructure:  1abf - m1 pdb1abf.dssp

Complex:   c1
ComplexProtein: c1    1abf
ComplexLigand:  c1    FCA

# specification of additional data
ProteinOrganism:pdb1abf ESCHERICHIA COLI
ProteinKW: pdb1abf TRANSPORT; PERIPLASMIC;

# specification of corrections
ModelCorrection:1abf m1    1     2315 23
SSCorrection:   1abf -     m1    1     304 25
```

Figure 3: Import specification

The user who is fluent with the integrated schema and the available external databases can easily capture the semantics of the different actions. Certain constraints have to be observed within the import specification, e.g. only objects may be interlinked or corrected which have been previously generated. Conflicts occurring due to overlapping instances in external databases must be resolved manually within the import specification. In order to support the user when editing this data appropriate tools are envisaged, e.g. for maintaining consistency constraints.

Generation of the import specification. Besides the preparation and correction of existing file-based databases the generation of the import specification is the most important step in the BP. In the following we describe a concrete example of how data is prepared for generating the import specification.

A considerable fraction of the PDB entries also contains the coordinates of bound ligands. In order to identify these receptor-ligand complexes we have realized a parser which scans the entire PDB database for small molecules such as substrates, inhibitors, cofactors, prosthetic groups and metal ions. Solvent molecules and molecules which are covalently linked to the receptor are ignored. Identified receptor-ligand complexes are entered in the import specification and cross-referenced with other objects in the database (Figure 4).

For processing the receptor proteins in a first phase the parser extracts the coordinates of each receptor atom and the amino acid sequence. Missing atoms, chain breaks are identified and reported. Solvent accessibility and secondary structure are calculated by executing the DSSP-program (Kabsch & Sander 1983). In a second phase the parser produces for each chain of the receptor cross-references to the following databases: SWISSPROT, the protein

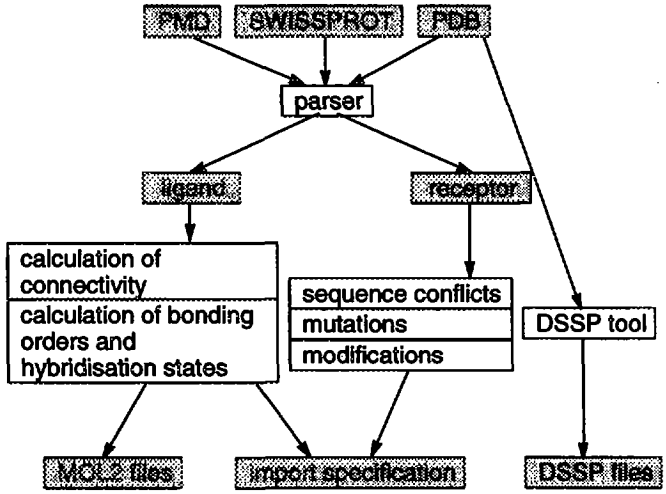

Figure 4: Generation of the import specification in the BP (shaded boxes indicate data whereas white boxes indicate processes).

sequence database, PIR, the protein sequence database of the Protein Identification Resource, EMBL (Higgins et al. 1992), PROSITE, the dictionary of Protein Sites and Patterns (Bairoch 1992) and PMD, the protein mutant database. Cross-references are used to scan the SWISSPROT and PMD databases for the following additional information:

- Sequence conflicts: The parser compares the sequence of each chain of the receptor with the corresponding sequence from SWISSPROT for conflicts such as insertion and deletions, residues missing due to disorders, conflicting and ambiguous experimental results (e.g. amino acids typed as ASX, GLX or UNK).
- Mutations and modifications: For each receptor chain the parser searches SWISSPROT and PMD for information about mutations and modifications (e.g. acylation, glycosylations or phorphorylations). Sequential position, type of mutation or modification and influence on stability or function are extracted. Sequential positions of mutations and modifications from the SWISSPROT and PMD databases are automatically converted to the PDB sequence numbering system.

For processing the ligands, they are identified by searching PDB-files for HETATM-records. Trivial names are extracted from the HET-records. If present, the CONECT-records are used to determine the connectivity of the ligand atoms. If CONECT-records are absent, the connectivity is derived from the interatomic distances. In order to facilitate a detailed analysis of the molecular interaction between ligands and receptors bonding orders and hybridisation states are derived from bond lengths and bond angles and checked for correctness manually (Barber

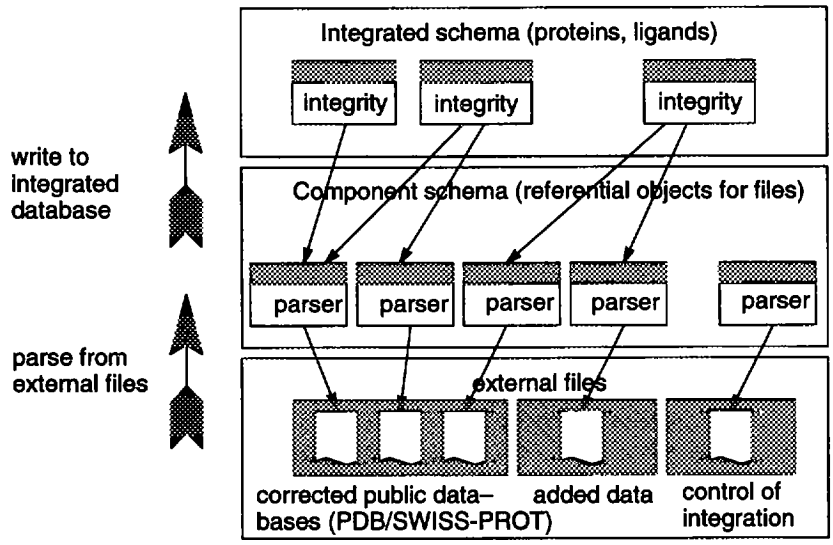

Figure 5: Realisation of the database integration (DP)

& Hodgkin 1992) (Meng & Lewis 1991). Finally the ligands are transformed to the MOL2-format of MDL.

Data integration. In the DP the following tasks have to be performed.

(1) syntactical transformation, i.e. parsing of the textual representations, in order to generate object-oriented data structures according to the component schemas, that can be processed by the DBMS.

(2) semantic enrichment, i.e. using the modelling mechanisms that the DBMS offers, in order to provide a semantically rich representation of the data to improve the efficiency of processing for retrieval.

In most realizations of the federated database approach the existing databases are only virtually integrated, i.e. the integrated database obtains data from the external databases on request ('call by need'). In our case there exists no fine-grained access to the external databases. Thus iterated parsing of external files for obtaining small units of information or repetitive computation of complex transformations would be required. As the import specification is declarative, it does not specify how data integration is to be performed operationally. So we choose to physically replicate the data, i.e. actively populate the integrated database in a 'forward driven' manner. (For other reasons supporting physical migration of data in the context of federated database systems see, e.g., (Radeke & Scholl 1994)).

The population of the integrated database proceeds as follows. A top level procedure realized as database method in the component schema interprets the import specification. Each statement in the import specification triggers an appropriate method call in the database. For example a mapping from an external object to an object of the integrated database is realized by a method initiating a parsing procedure on the corresponding external file, creating the representative object in the component schema and then instantiating the corresponding object(s) in the integrated schema. Within this method index structures may be set up and integrity constraints within the database are checked. Objects of the component and integration schema are linked appropriately, such that for every data item its origin can be derived. This is essential for keeping track of updates in the external databases. These updates lead to a reload of the replicated data, occur however only rarely.

Implementation. The implementation of the integrated database is done on top of the OODBMS VODAK using VML for specifying the database schema. In several method implementations external C++ code was used. Currently the database contains 1495 proteins, 2613 chains and secondary structures, 559 ligands in 1906 protein-ligand complexes and 1536 metal-ion complexes. There are 1195 chain modifications and 5096 chain mutations loaded. The integrated database containing all replicated objects occupies about 500 MB diskspace.

6 Examples

A sample population step. Figure 6 illustrates the semantic enrichment of chain and protein data. The implicit knowledge, that a particular SWISSPROT chain occurs twice in a protein is formulated explicitly in the specification file (load.vcl in Figure 6). This results in the following representation in the integrated schema: the chain object derived from SWISSPROT, is specialized to two protein-chain objects (specialization is indicated by a broken arrow). The two protein-objects can then occur as part of a protein. They are existence-dependent on the protein they belong to and cannot be shared by other proteins.

A sample query. In this section we want to give an impression of the querying capabilities one obtains with our

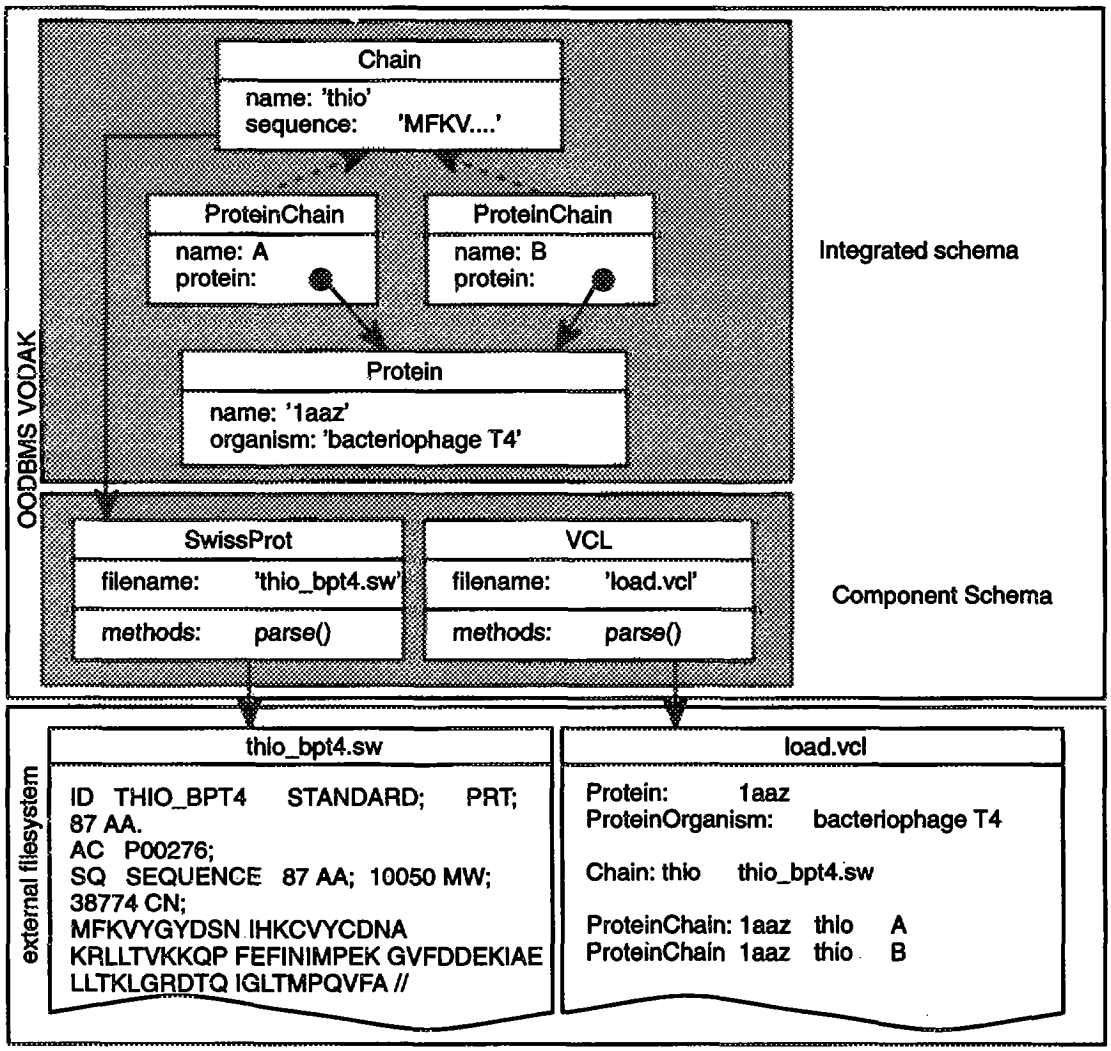

Figure 6: Populating the database with a multichain protein.

approach. The example query illustrates (i) the possibility to access protein and ligand data that has been previously stored in separated databases, (ii) the ability to navigate in the integrated schema, and (iii) the usage of complex algorithms within the query.

The task of the query is to find all complexes that are similar to a particular protein-ligand complex. Similarity between complexes is expressed by a high chain homology of the involved protein chains and a common substructure of the involved ligands. The computation of the homology (by the Needleman/Wunsch algorithm) and substructure search is done via algorithms that are available within the database system. The result of the query are the names of complexes found.

```
ACCESS complex.name
FROM protein IN db::Protein, chain IN protein.chains,
     ligand IN db::Ligand,complex IN db::Complex,
     complexchain IN complex.protein.chains
WHERE protein.name == '1abf' AND
     chain->NWalignment(complexchain) >= 0.8 AND
     complex.ligand->hasSubstructure('CC(=O)O')
```

The ACCESS clause specifies the desired output (i.e. the complex name). The FROM clause allows to bind variable names to sets of objects, either specified by classes or reached by navigation. The WHERE clause provides the selection condition. We are interested in a special protein (protein.name == '1abf') and we are searching the homologous ones (chain->NWalignment(complexchain) >= 0.8). The ligand of the complex must have the specified substructure given in Smiles code format (i.e. ligand->hasSubstructure('CC(=O)O')). Path expressions allow to navigate along related classes by following links, e.g. complex.protein.chains (compare also Figure 1).

7 Conclusion

We have presented an approach for building a high quality receptor-ligand information base for drug design. This is the first approach that makes data from heterogeneous resources from biomolecular and biochemical research available in an integrated and enriched way within an object-oriented database management system. For this purpose we had to develop our own integration methodology.

We have arrived at a working system that opens the way for further research. The process of data preparation needs to be further investigated in order to be able to support this phase with appropriate tools. With regard to data processing, in particular the investigation of application-specific optimizations strategies for queries will be in the center of interest. For supplementary information, like literature databases, where performance is a minor issue, external databases will be integrated virtually.

Acknowledgement. We would like to thank all the project partners involved in the project, including G. Vriend from EMBL, F. Rippmann and G. Barnickel from E. Merck, H.J. Böhm and G. Klebe from BASF, and T. Lengauer and M. Rarey from GMD for providing an exciting research environment. We also want to express gratitude to P. Fankhauser for valuable discussions on database integration and to K. Böhm for comments on the paper.

8 References

Aberer, K., and Fischer, G. 1995. Semantic Query Optimization for Methods in Object-Oriented Database Systems. In Proceedings of the Eleventh International Conference on Data Engineering, 70–79. Taipei, Taiwan.

Akrigg, D., et al. 1988. A Protein Sequence/Structure Database. *Nature* 335, 20:745–748

Bairoch, A., and Boeckmann, B. 1991. The SWISS-PROT Protein Sequence Data Bank. *Nucleic Acids Res.* 19 (Sequences Suppl.):2247–2249.

Bairoch, A. 1992. PROSITE: A Dictionary of Sites and Patterns in Proteins. *Nucleic Acids Res.* 20, suppl.: 2013–2018.

Barber, J.C., and Hodgkin, E.E. 1992. Automatic Assignment of Chemical Connectivity to Organic Molecules in the Cambridge Structural Database. *Journal of Chemical Information and Computer Sciences* 32: 401–406.

Barker, W.C., George, D.G., Hunt, L.T., and Garavelli, J.S. 1991. The PIR protein sequence database. *Nucleic Acids Res.* 19 (Sequences Suppl.): 2231–2236.

Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, E.F. Jr., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T., and Tasuni, T. 1977. The Protein Data Bank: A Computer Based Archival File for Macromolecular Structures. *J. Mol. Biol.* 112:535–542.

Chang, W., Shindyalov, I.N., Pu, C., and Bourne, P.E. 1994. Design and Application of PDBlib, A C++ Macromolecular Class Library. WWW page http://www.cse.ogi.edu/DISC/PDBTool.

Durbin, R., and Thierry-Mieg, J. 1994. The ACEDB Genome Database. WWW page http://probe.nalusda.gov:8000/acedocs/dkfz.html.

Goodman, N. 1995. An Object-Oriented DBMS War Story: Developing a Genome Mapping Database in C++. In *Modern Database Systems*, Kim W. ed., ACM Press, New York, 216–237.

Higgins, D.G., Fuchs, R., Stoehr, P.J. and Cameron, G.N. 1992. The EMBL Data Library. *Nucleic Acids Res.* 20 (suppl.):2071–2074.

Huysmans, M., Richelle, J., and Wodak, S.J. 1991. SESAM – A Relational Database for Structure and Sequence of Macromolecules. *Protein Structure and Genetics* 11(1):59–76.

Islam, S.A., and Sternberg, M.J.E. 1989. A Relational Database of Protein structures Designed for Flexible Inquiries about Conformation. *Protein Engineering* 2:431–442.

Kabsch, W., and Sander, C. 1983. Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-bonded Geometrical Features. *Biopolymers* 22:2577–2637.

Kamel, N.N., Song, T., and Kamel, M. 1993. An Approach for Building an Integrated Environment for Molecular Biology Databases. *Distributed and Parallel Databases* 1:303–327.

Kemp, G.J.L., Jiao, Z., Gray, P.M.D., Fothergill, J.E. 1994. Combining Computation with Database Access in Biomolecular Computing. In Proceedings of Applications of Databases, Vadstena, Sweden, Lecture Notes in Computer Science 819:317–335.

Klas, W., Aberer, K., Neuhold, E.J. 1994. Object-Oriented Modeling for Hypermedia Systems using the VODAK Modelling Language (VML). In *Advances in Object-Oriented Database Systems*, A. Dogac, T. Ozsu, A. Biliris, T. Sellis eds., NATO ASI Series F 130:389–434, Springer, Berlin, Heidelberg.

Meng, E.C. and Lewis, R.A. 1991. Determination of Molecular Topology and Atomic Hybridisation States from Heavy Atom Cordinates. *J. Comp. Chem.* 12:891–898.

Nishikawa,K., Ishino, S., Takenaka, H., Norioka, N., Hirai,T., Yao,T. and Seto,Y. 1994. Constructing a Protein Mutant Database. *Protein Engineering* 7:733.

Radeke, E., and Scholl, M. 1994. Federation and Stepwise Reduction of Database Systems. In Proceedings of Applications of Databases, Vadstena, Sweden, Lecture Notes in Computer Science 819:381–399.

Ritter, O., Kocab, P., Senger, M., Wolf, D., and Suhai, S. 1994. Prototype Implementation of the Integrated Genomic Database. *Computers and Biomedical Research* 27:97–115.

Sheth, A.P., and Larson, J.A. 1990. Federated Database Systems for Managing Distributed, Hetereogenous, and Autonomous Databases. *ACM Computing Surveys* 22(3):183–236.

Sillince, M., and Sillince, J.A.A. 1993. Sequence and Structure Databanks in Molecular Biology: the Reasons for Integration. *Journal of Documentation* 49(1):1–28.

VODAK V 4.0, User Manual 1995. GMD technical report No. 910.

Wood, G.H., Rodgers, J.R., and Gough, S.R. 1989. Canadian scientific numeric database service. *Journal of Chemical Information and Computer Sciences* 29:118–123.