Demand−Driven Database Integration for Biomolecular Applications

Karl Aberer

GMD−IPSI, Dolivostr. 15, D−64293 Darmstadt, GERMANY

email: aberer@darmstadt.gmd.de

Introduction

As a member of the consortium for the "Computation and Prediction of Receptor−Ligand Interaction" the Integrated Publication and Information Systems Institute, GMD−IPSI , Darmstadt, participates in the national joint project RELIWE. Docking−D is the part of RELIWE which considers heterogeneous database support and in which GMD−IPSI takes the leading role. In the current situation the receptor and ligand data used within the project, either raw data or data derived during analysis, is extremely heterogeneous. Many of these databases are supported by autonomous systems which employ different data management facilities with heterogeneous data models, in particular dedicated file systems with specialized retrieval and presentation functionality (e.g. PDB [1]) or a relational model (e.g. Whatif [20]). In addition, the information is represented at different levels of detail (e.g. sequence vs. structural data), with mutual inconsistencies in structure, naming, scaling, and behavior, whereby much of this behavior is hidden in the implementation of the autonomous systems. Thus the database system must enable integrated access to the underlying, autonomous, heterogeneous information bases, but also has to allow the integration of new datatypes (e.g. sequence and spatial data) and has to support associative retrieval of the data. Different tools, like receptor−ligand docking algorithms, model building tools for receptors or visualization tools, which are developed or provided by the other partners within the project (e.g. Whatif, LUDI [2]), must be connected to the DBMS.

Database Integration

There exist several approaches and projects which address interoperability or integration of information bases. For an extensive discussion of related work see [3][11][12][20], which give good overviews and present fundamental concepts including the terminology of the different approaches, e.g., multidatabase systems, multidatabase languages, and federated database systems. GMD−IPSI takes what is called the federated database approach. The tools and techniques developed for semantic integration assist incremental integration driven by actual information requests of end users and the dynamic maintenance of integrated schemas driven by external schema evolution. This approach tries to meet the requirements of realistic situations with a big number of external information bases. For example, currently there are at least 100 databases known providing biomolecular information. Due to their autonomy they are subject to schema change, which can not be controlled globally. Therefore completely integrated views valid for all users can hardly be achieved with reasonable effort. The complexity of the macromolecular CIF dictionary definition [18], which is aimed at establishing a universal schema for molecular biology, is a vivid illustration for that.

Database integration steps can be partitioned into two conceptual layers: a syntactic transformation phase and a semantic integration phase.

Syntactic transformation phase. In the syntactic transformation phase, heterogeneous data models have to be mapped to a uniform data model. This requires the translation of manipulation languages and the transformation of diverse data formats as well as the connection of external database management systems or other systems providing external data. This step overcomes model heterogeneities by generating export schemas.

To this end VML [9] , the data model of the open, object−oriented database system VODAK , is used as the canonical data model into which the external schemas are mapped. To use an object−oriented data model as the canonical model is widely recognized as the right choice for easier representation of external data models as well as for schema integration purposes (see [8],[20]). Also, an object−oriented data model is considered to be very well suited to allow a natural representation of the complexly interrelated biomolecular data, see e.g.[5].

A particular feature of the VML data model is the concept of metaclasses, which are containers for classes. Metaclass schemas allow to adapt the data model to the needs of particular application areas like database integration or biomolecular applications. For example, one might introduce new semantic relationships between classes, like generalization. The generalization relationship will turn out to be a central construction for database integration. Another example is the sequenceof relationship between classes, which allows to model a situation where an instance of one class is always related to a sequence of instances of another class. This is a situation occuring typically in biomolecular applications. The metaclass schemas are developed by model designers and are hidden from the application schema developers, such that they do not have to bother with the details of the realizations of such data model extensions.

To provide functions to access an existing external database system from VODAK we provide mechanisms using the openness of the data model. For example such mechanisms have been developed to access data stored in the relational database system Postgres [10] . Similar mappings can be provided for other relational systems thus providing access to data like that stored in SESAM [6] or BIPED [7]. However most biomolecular databases do not provide fine grained, explicitly structured data like relational databases do. They are available in form of flat files, like PDB, Swissprot, PIR or PRF. Usually this requires structural enrichment of the external schemas or file formats. This can be just parsing the files, but in many cases the situation is more intricate. For example, the Whatif system provides a substantial amount of code for analyzing PDB files in order to come to a well−defined structural representation of their contents. It is also known that some of the file formats used in biomolecular databases cannot be analyzed by means of a context−free grammar, e.g. the present state of DDL/mmCIF [14].

Semantic integration phase. The semantic integration phase is needed to combine several export schemas. On top of the bottom layer, i.e., on the basis of the uniform data model, implicit structure and semantics have to be made explicit, inconsistencies in structure, naming, and scaling have to be overcome, and semantic interrelationships between data have to be acquired in order to establish integrated views onto the external resources. Finally appropriate user views can be determined in order to specify one (or a couple of) integrated schemas and/or individual application schemas.

Semantic integration includes semantic enrichment which makes implicit structure and semantics explicit and associates additional behavior, which is hidden in local application programs or even worse in informal local conventions. A typical example of semantic enrichment is resolving the literature citations, that are present in many protein databases, like PDB, Swissprot etc., to actual references to database objects representing this literature. This approach is realized in ENTREZ [16] and successfully employed for retrieving relevant information for proteins from the literature.

A declarative methodology to overcome heterogeneities typical for object−oriented schemas, where the same concepts can be represented by different schema constituents (e.g., classes, types, attributes) was developed. This allows the user to declare correspondences between constituents of different kinds. Then the consistency of the user−defined correspondences is checked and the heterogeneity of corresponding subschemas is overcome by schema unification. This leads to augmentation of the structural granularity, a form of structural enrichment. In addition, we develop concepts to assist the detection of possibly corresponding subschemas using fuzzy terminological knowledge about the application domain. Details about these techniques employed for the final semantic integration steps are also given in [4],[13],[17],[19].

In the last step the integrated schema is generated by generalization. That is, classes are constructed that are containers for the union of the instances of different classes carrying information about the same real world aspect. For example a class for protein sequences might be the generalization class of several classes in export schemas carrying sequence information from different biomolecular databases. For overlapping instances correspondence predicates have to be specified. This turns out to be a difficult problem in biological databases as well−defined key properties are not available. For example, although in principle an amino acid sequence uniquely defines a protein, this property cannot be used for identification without careful examination, due to errors that might be present in the sequences. Also often identifiers used in biomolecular databases are not stable like the mnemonic names used in GenBank , DDBJ and EMBL. For an attempt to resolve this problem see, e.g., [16]. Furthermore, for the correspondences between properties appropriate methods have to be generated, which treat data conflicts for overlapping instances. There are several strategies to overcome data conflicts, namely prefer data from one database, aggregate conflicting data or ask the user to resolve the data conflicts intellectually at query time.

Transaction Management. In addition to solutions related to the various integration steps an operational database system providing access to integrated views onto external resources must also offer appropriate global transaction management. As an example consider the scenario envisaged by NCBI’s GenBank [16] where researchers independently access and update a biomolecular information base as a means for information exchange. Then a transaction model able to support the concurrent access of these multiple users to the integrated data has to address two main problems: At first, it has to provide for a high degree of concurrency compared to conventional models, as transactions in an integrated system are relatively long−lasting and complex. We solve this problem by utilizing semantic information about methods, that is available in the VODAK data model, in the VODAK open nested transaction concept [15] . The second problem is the integration of probably different concurrency control and recovery schemes of the existing systems into a single global transaction management. The open nested approach allows us to integrate without changes the existing transaction management modules.

Conclusion

We do not aim at a complete, global integrated schema, which overcomes all heterogeneities, but rather want to assist the incremental design and maintenance of integrated schemas, according to the specific needs of an integrated application. For this purpose we developed an open object−oriented database management system, which is operational, and a declarative methodology for schema integration. Nevertheless, this kind of demand−driven database integration may contribute on the way of finding a global integrated schema for biomolecular information bases by allowing to investigate integration problems for partially integrated working systems.

References

[1] Bernstein F.C., Koetzle T.F., Williams G.J.B., Meyer E.F. Jr., Brice M.D., Rodgers J.R., Kennard O., Shimanouchi T., Tasuni T.: "The Protein Data Bank: a computer based archival file for macromolecular structures", J. Mol. Biol. 112, pp. 535 − 542, 1977.

[2] Boehm, H.J.: " The computer program LUDI: A new simple method for the de−novo design of enzyme inhibitors", J.Comput.Aided Molec.Des. 6, pp 61−78, 1992.

[3] Bright, M.W.; Hurson, A.R.; Pakzad, S.H.: A Taxonomy and Current Issues in Multidatabase Systems. IEEE Computer, Vol. 25, No. 3, March 1992.

[4] Fankhauser, P.; Kracker M.; Neuhold, E.J: Semantic vs. Structural Resemblance of Classes. In: Special SIGMOD RECORD Issue on Semantic Issues in Multidatabase Systems, Vol. 20, No 4, ACM Press, Dec. 1991.

[5] Gray P.M.D., Paton N.W., Kemp G.J.L., Fothergill J.E.: "An object−oriented database for protein structure analysis", Protein Engineering, 3(4), pp. 235 − 243, 1990.

[6] Huysmans,M., Richelle,J., Wodak,S.J.: "SESAM: a relational database for structure and sequence of macromolecules", Proteins: Structure, Function and Genetics, 11, pp. 59 − 76, 1990.

[7] Islam, S.A., Sternberg, M.J.E.: "A relational database of protein structures designed for flexible enquiries about conformation", Prot. Engin, 2, pp. 431−442, 1989.

[8] Kaul, M.; Drosten, K.: ViewSystem: Integrating Heterogeneous Information Bases by Object−Oriented Views. Proceedings Data Engineering 1990.

[9] Klas, W., Aberer K., Neuhold, E.J.: Object−Oriented Modeling for Hypermedia Systems using the VODAK Modelling Language (VML) to appear in: Object−Oriented Database Management Systems, NATO ASI Series, Springer Verlag Berlin Heidelberg, 1993.

[10] Klas, W., Fischer G., Aberer K.: Integrating a Relational Database System into VODAK using its Metaclass Concept. Technical Report GMD No. 738, GMD Sankt Augustin, March 1993.

[11] Klas, W., Fankhauser, P., Muth, P., Rakow, T.,Neuhold, E.J.: "Database Integration Using the Open Object−oriented Database System VODAK", to appear in: Object−Oriented

Multidatabase Systems, Eds. Elmagarmid, A., Bukhres, O., Prentice Hall, 1994.

[12] Litwin, W.; Mark, L.; Roussopoulos, N.: Interoperability of Multiple Autonomous Databases. ACM Computing Survey, Vol. 22, No. 3, September 1990.

[13] Mehta A.; Geller, J.; Perl,Y.; Fankhauser, P.: Algorithms for Computing Access Relevance in Object−Oriented Databases. Proc. of the First Int. Conf. on Information and Knowledge Management, Maryland, Nov. 1992.

[14] Murray−Rust, P.: "Analysis of the DDL/Dictionary Parsing Problem", in [18].

[15] Muth, P.; Rakow, T.C.; Weikum, G.; Broessler, P.; Hasse, C.: Semantic Concurrency Control in Object−Oriented Database Systems. Proc. IEEE Data Engineering, Vienna, Austria, 1993.

[16] NCBI Software Development ToolKit, Version 1.8, August 1, 1993.

[17] Neuhold, E. J.; Schrefl, M.: Dynamic Derivation of Personalized Views. Proc. of the 14th Int. Conf. on Very Large Data Bases, Los Angeles, CA, 1988.

[18] Proc. of the First Macromolecular Crystallographic Information File (CIF) Tools Workshop, Ed. Bourne, P., October 15−18, Tarrytown NY, obtainable via anonymous ftp from cuhhca.hhmi.columbia.edu, 1993.

[19] Schrefl, M.; Neuhold, E.J.: Object class definition by generalization using upward inheritance. Proc. 4th Int. Conf. on Data Engineering, 1988.

[20] Sheth, A. P.; Larson, J. A.: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Survey, Vol. 22, No. 3, September 1990.

[21] Vriend G., Sander C., Stouten P.F.W.: "A novel search method for protein sequence−structure relations using property profiles", Protein Engineering, 7(1), pp. 23 − 29, 1994.