

Association for Information Systems AIS Electronic Library (AISeL)

PACIS 1993 Proceedings

Pacific Asia Conference on Information Systems
(PACIS)

December 1993

An Overview of Model-assisted Global Query System

Waiman Cheung
Chinese University of Hong Kong

Follow this and additional works at: <http://aisel.aisnet.org/pacis1993>

Recommended Citation

Cheung, Waiman, "An Overview of Model-assisted Global Query System" (1993). *PACIS 1993 Proceedings*. 12.
<http://aisel.aisnet.org/pacis1993/12>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 1993 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

An Overview of Model-assisted Global Query System

Waiman Cheung

Department of Operations and Systems Management
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong

ABSTRACT

Today's enterprises typically employ multiple information systems which are independently developed, locally administered and different in logical or physical designs. Therefore, a fundamental challenge in enterprise information management is the integration of such systems across functional boundaries within an organization. Conventional technologies, such as syntax-based query languages and heterogeneous distributed databases (HDDDBMS) are not sufficient to solve this problem.

This paper presents an overview of the model-assisted global query system, that utilizes enterprise metadata to facilitate global query formulation and processing. The significance of possessing knowledge (via a metadatabase) for the notion of "on-line intelligence and assistance" is analyzed. The model-assisted approach also resolves some interoperability issues (e.g., differences and incompatibilities among local systems) in information sharing. Traditional HDDDBMS approaches handle these differences by imposing an integrated schema on the local systems. This approach, instead, shows that the same objective of resolving differences can be accomplished by utilizing the metadatabase containing data semantics, conversion rules, and other knowledge of the system. This way, not only difficult tasks such as schemata integration at both design time and run-time can be largely avoided, but also a higher degree of flexibility of system evolution can be achieved. The latter is an important goal of open system architecture.

1. Introduction

Information systems in today's enterprises are typically characterized by heterogeneous, distributed environments. For example, a computerized manufacturing enterprise may have a number of functional sub-systems that are implemented with different file management systems and database management systems (DBMS) on different platforms. Integration of multiple systems [Rattner 1990], therefore, is a problem facing these enterprise. A major objective of information integration is to provide a logical structure to integrate these islands of information resources for enterprise-wide information sharing and management. Current efforts on information integration include the emerging Repository technology from IBM and other corporations, the multi-national project of CIMOSA [ESPRIT 1989], the Composite Information Systems project at MIT [Madnick et al., 1988, Wang et al., 1989, 1990], and the industry-sponsored research on the Metadatabase approach at Rensselaer (see, e.g., [Hsu et al., 1990c, 1991, 1992]).

An integrated enterprise information management system must be able to (1) support end users' needs in retrieving information at a global level and (2) facilitate application programs to access, aggregate, and update information maintained in multiple local systems. A key requirement here is what might be called the "on-line intelligence and assistance" capabilities of the system. In particular, the end users should be able to query the system as a whole without having to know the technical details of the local application systems. Similarly, the application programs should be freed from embedding into them the local access paths,

mappings among database schemata, basic business rules, and other excessive metadata of the system. However, such kind of systems is largely unavailable at present.

To illustrate the significance of these envisioned capabilities, we consider herein some basic tasks required of a global query system in a heterogeneous, distributed environment (see, e.g., Reddy, et al., 1988 for a survey on these tasks). A typical global query operation involves two steps; namely, global query formulation and global query processing. In the first step, the user's requests are articulated and represented in a way that the global query system understands. The query formulation is done primarily through the user interface of the system. In the second step, accomplished by the system internally, queries are sent to appropriate local systems to retrieve pertinent information and reassembled for the users.

Towards query formulation, on-line intelligence empowers the user interface to provide assistance in the articulation as well as allow for high-level, intuitive representation of queries. Specifically, for the articulation of queries, the system would utilize its knowledge on the enterprise information resources (or, metadata; including information models, implementation models and business and operating rules) to alleviate semantic ambiguity, facilitate logical analysis, and enhance adaptive construction during the formulation of queries. Similarly, the representation itself could accommodate heterogeneity in local models across the enterprise (through, e.g., the knowledge on data equivalence), without having to impose a single view (e.g., an "integrated schema" of all databases) for all users to comply by.

While supporting high-level and non-syntax-based queries for enterprise users, the system would handle all interpretations and mappings between the globally formulated query and the locally implemented file structures or database schemata such as relations, attributes, formats and keys. Assisted with these capabilities, global query processing would be accomplished in the following fashion:

- (1) query optimization: the global query is first optimally decomposed into local queries taking into account both semantics and performance;
- (2) query translation: the local queries are then coded in their respective data manipulation languages; and finally
- (3) result integration: results from local systems are assembled to answer the global query.

Each of these processing steps makes use of metadata as well. For instance, local database schemata, directory information and their contextual knowledge are required for query optimization; local DBMS information for query translation; and knowledge on equivalent objects, incompatibility and data conversion for result integration. Without the assistance from on-line metadata (or, knowledge), this information would have to be provided by the users or application programs.

Examples from previous systems best testify to the significance of the envisioned capabilities, through illustrating the problems of lacking them. Concerning query formulation, instead of freeing users from cumbersome details, existing technology tends to require users to learn an interface language

and to adjust their own views in order to use the systems ([Astrahan et al., 1976], [Shipman 1981], [Motro 1990], etc.). Moreover, during formulation, users are frequently presented with overwhelming amounts of technical details (e.g., database schemata, data items, names, file locations, etc.) through the query system; they must make numerous decisions on critical technical parameters in a short time frame ([Cattell 1980], [McDonald 1975], [Herot 1980], [Fogg 1984], [Agrawal et al., 1990], etc.). Such situations are often made worse when users lack knowledge and experience. This problem is also manifested in the increasing emphasis on user interface technology (including both computing techniques and cognitive principles for design) as an essential element to the operational success of application systems [Kleinrock 1985, Stonebraker 1988]. A common belief now is that the complexity of the system should be transparent to the users.

In addition to the user interface issues in the global query formulation step, many difficulties also arise in the global query processing step. This is due to the differences in data semantics (models), data manipulation languages, and data structures among local systems. Also, because the same pieces of information could be stored in many locations, different accessing strategies generally exist, which need to be optimized as to, e.g., from where certain information should be retrieved and in what manner. These differences must be resolved either by the users or by the system itself before the information can be shared. Without instructions from users the system would have to rely on the metadata it possesses on-line to deal with any of these issues; while such metadata are not sufficiently supplied previously. Enterprise users, on the other hand, normally do not have the expertise in database technology, nor the knowledge about all of the local systems, to resolve these differences. Therefore, they cannot truly benefit from a global query system without on-line intelligence.

It is worth noting that user interface techniques alone cannot relieve the users from being responsible for providing (technical) knowledge on the enterprise information systems. This knowledge have to be placed on-line into the systems in order to provide a viable solution to the above mentioned problem. Nevertheless, many serious efforts on this problem today follow or utilize results from the field of user interface technology. In this sense the proposed research should be formulated first in the light of these results, as follows.

2. Previous Solution Approaches

The interest in human-computer interface spans end user computing formalism and cognition-based design approaches. Over the past two decades, research efforts have focused on developing user interfaces for interactive use of information processing. The major objective is to free the typical non-programmer user from having to learn sophisticated programming languages. Techniques such as windows, icons, menus, graphics, and other forms of visual or non-syntax-based formalisms are used to bring the computer closer to the human level, rather than expecting users to accommodate to the level of the computer. However, relying on these basic techniques alone to provide user interfaces usually achieve simplicity and convenience at the expense of expressiveness and flexibility. For instance, a predefined menu system offers straight forward but fixed and limited uses of system capabilities. Moreover, this approach virtually precludes the users' "growth" or adaptive use of the systems as they acquire more expertise on the underlying logic. Without also providing or assisting users with the knowledge about system capabilities or contents, these interface techniques tend to fail either to make the systems responsive to users' evolving needs or to enable users to truly tap into the systems' functionality.

2.1 Interface languages for query formulation

Naturally, global query is an area that can benefit in a fundamental way from the advancements of the above general user interface technology, especially for the query formulation tasks. The progress in this area may be considered as an evolution from database programming to user interface for enterprise users, utilizing first technology and cognitive principles and finally knowledge.

The traditional approach to accessing databases is through programming-oriented, syntax-based languages. A query language interface is defined as a high level language in which all data retrieval requests are expressed. It is usually interactive and supposedly designed for non-technical users. The advantages of such language interfaces are their generality (the ability to express arbitrary requests) and their unambiguity (each statement has clear semantics based on the language's syntax). The drawback, clearly, is that using a query language interface requires considerable proficiency. Besides the syntax of the query language itself, users must understand the principles of the underlying data model(s), and they must be familiar with the contents and organization of the particular database(s) being accessed. In the absence of any of these prerequisite knowledge, using such interfaces can become very ineffective and frustrating. Hence, most query language interfaces simply do not fit, and are actually geared away from, end users altogether [Motro 90].

To alleviate some of these problems, the computing needs of end users have been studied by researchers from a man-machine interface perspective. For example, human factors and ease-of-use are among the major concerns in query languages. Therefore, many interface design principles and guidelines [Hansen 1971] [Smith 1981, 1982, 1986] [Shneiderman 1986] have been derived from cognitive theory. There has been a trend that query interface development moves from being novice oriented toward more functionally oriented.

More ambitious attempts toward the same objective are the so-called natural language interfaces. In principle, any systems that use natural languages should not require the users to learn the artificialities of correct command formats or modes of interaction [Norcio 1989]. Unfortunately, few systems (even research systems) have successfully achieved this goal [Hirschman 1987], due mainly to insufficient results in linguistics and artificial intelligence to support this class of user interface. However, even in the ideal case, a natural language interface would not be able to assist on the query formulation itself. In particular, it does not provide users with such knowledge as the enterprise information models (e.g., the local and global data resources, dictionary and directory knowledge, and business rules); all of which are needed before a user interface could ever support non-technical user in a "natural" mode of query formulation.

Addressing the issue of providing metadata to users, as a separate goal, database browsers have been developed to help end users to look through the contents of a particular database. Most of these database interfaces (e.g., [Zloof 1975, 1977] [Cattell 1980] [Herot 1980] [Stonebraker 1982]) are limited to browsing the data instances and therefore they can process only simple and single-tuple queries. The remaining few [Wong 1982], [Smith, et al., 1981], [Motro 1990] still confine the browsing to the database schema per se and do not include other types of enterprise metadata. They, nonetheless, provide interesting evidence supporting the thesis that metadata can lead to a new breakthrough for the problem.

The above discussion applies to both single and multiple information systems as well as to both homogeneous and

heterogeneous environments. Multiplicity and heterogeneity do make the issue of knowledge even more acute since the user would have to deal with more of the same technical detail: (e.g., data models) as in the single system case. However, they incur fundamentally new complexities and challenges in the areas of system architecture and management.

2.2 Integrated schema for global query

A key issue, especially concerning the query processing side, is how to reconcile and consolidate the various system views and representation methods across the enterprise and yet still retain local differences for flexibility. Research efforts (e.g., [Smith 1981] [Deen et al., 1985] [Templeton 1986] [Krishnamurthy 1987] [Chung 1990]) have also revealed the need of enterprise metadata on presenting global views and managing query transactions across heterogeneous, distributed databases.

Most of these efforts emphasize the development of architectures and an "integrated schema" environment to achieve global query operation. However, the employment of integrated schemata tends to contradict or even nullify some of the basic promises of truly distributed, let alone heterogeneous, environments [Litwin et al., 1990]. Therefore, these efforts have again established the critical significance of metadata in global query processing. Their limitations, ironically, have also further pointed to an even bigger goal for enterprise metadata; that is, minimizing the reliance on some controlling global schemata to effect information integration.

2.3 Summary

The previous solution approaches to the problem of global query have been limited by the lack of on-line knowledge in the form of enterprise metadata. The significance of knowledge is best illuminated in the context of query formulation from the perspective of user interface (see Figure 1). In this regard, we have discussed two major aspects in query interface development. The first aspect is concerned with cognitive psychology in which human factors are the major concerns. The other takes roots in the technically oriented formal languages (e.g., DL/I, SQL, QUEL, etc.) and end user computing techniques such as menus, icons, windows, etc. These two aspects together have given rise to functionally powerful and relatively easy to use query languages such as nature language interfaces and database browsers for query formulation. However, using these interfaces alone still does not remove the requirement of expertise on the users part in formulating queries. What has been neglected is the third dimension, "knowledge," which should be put on-line for global query instead of relying on users to supply it. The same argument also applies to query processing, for which enterprise metadata holds a key for the problem and should be incorporated as on-line knowledge into the system. As such, knowledge through enterprise metadata is elevated and explicitly formulated to play the central role in a new solution approach to solving the global query problem in this research.

3. Enterprise Metadata

In general, information in an enterprise is derived from facts (data about the real-world using the knowledge that drives the enterprise functions and specifies the business operating rules [Potter 1988]). Since metadata is information about the enterprise information, it follows that metadata itself is comprised of data and knowledge aspects. These aspects can be classified according to their roles into four categories:

dictionary, directory, application, and contextual metadata [Bouziane 1991].

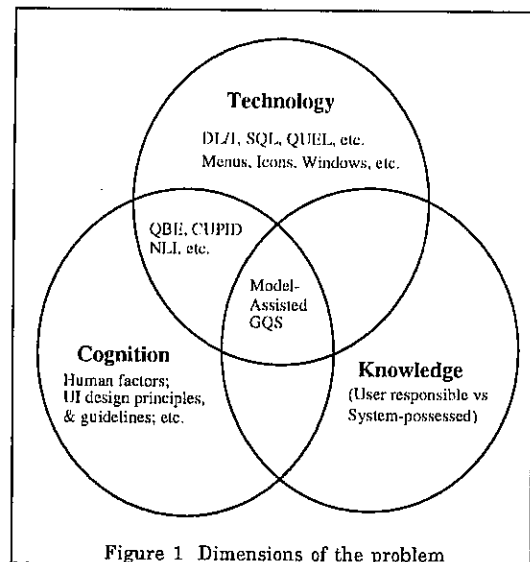


Figure 1 Dimensions of the problem

of expertise on the users part in formulating queries. What has been neglected is the third dimension, "knowledge," which should be put on-line for global query instead of relying on users to supply it. The same argument also applies to query processing, for which enterprise metadata holds a key for the problem and should be incorporated as on-line knowledge into the system. As such, knowledge through enterprise metadata is elevated and explicitly formulated to play the central role in a new solution approach to solving the global query problem in this research.

3. Enterprise Metadata

In general, information in an enterprise is derived from facts (data about the real-world using the knowledge that drives the enterprise functions and specifies the business operating rules [Potter 1988]). Since metadata is information about the enterprise information, it follows that metadata itself is comprised of data and knowledge aspects. These aspects can be classified according to their roles into four categories: dictionary, directory, application, and contextual metadata [Bouziane 1991].

Dictionary metadata is the information that defines the data objects in terms of structure, meaning, and their relationships in an organization. This category includes semantic data models and logical database structures. The second category, directory metadata, refers to the information that identifies where data objects are stored and how to get them. It includes database models and software resources implementation designs. The application metadata category consists of the information describing user groups, programs, and processes that have defined access to the information resources, as who can use what and how. Finally, contextual metadata comprises three classes of knowledge use to (1) describe the intended use of data (e.g., business rules), (2) prescribe information flows, and (3) facilitate decision processes across systems.

Consequently, in order to support information management and information integration functions, the scope of metadata must span all four categories. More importantly, these metadata must be interrelated in a unified way. Insufficient

efforts on any of these capabilities will limit the role and functionality of the system in information processing technology, as evidenced in previous metadata systems and applications.

4. The Model-assisted Global Query System

4.1 The objective

The objective of the model-assisted Global Query System is to develop new fundamental global query capabilities that enable enterprise (end) users to access information maintained in multiple heterogeneous sub-systems without having to either change these existing systems nor having to acquire excessive technical knowledge about them, as required by previous results. The basic approach is to employ a new enlarged class of enterprise metadata as the on-line knowledge and utilize it as the basis to develop new classes of methods for solving the problem. Further, we incorporate the existing metadata technology into these methods and develop them into the new concept of a metadata-assisted (used interchangeable with model-assisted) Global Query System (GQS).

This model-assisted GQS has the following goals: (1) investigate and characterize the role of enterprise metadata on-line knowledge in global query formulation and processing; (2) design a new query method using the on-line knowledge to provide non-technical, non-programming-based query formulation capabilities to enterprise users; (3) develop a metadata-supported approach to resolving some interoperability issues in global query processing (e.g., differences and incompatibilities among local sub-systems) without requiring the control of global integrated schemata; and (4) generalize the findings of this work on enterprise metadata for the design and development of new classes of user interfaces in other traditional information management functions such as information modeling.

4.2 The approach

In order to keep the complexity of global query operation transparent to users, a GQS needs to incorporate a repository of enterprise metadata. Our approach is to use the metadata as the enterprise metadata repository to assist the GQS. This approach utilizes the unique promises of the metadata model in the scope of knowledge it provides and the structure of metadata representation it supports towards satisfying the needs of global query formulation and processing. In comparison, other metadata systems are confined either to homogeneous environments or to data models per se, or both. A prime case in point is the National Institute of Standards and Technology's Information Resources Dictionary System (IRDS) [Goldfine 1988, 1985]. Albeit an improvement over traditional data dictionary systems from single database systems to multiple database systems, it does not extend the scope of metadata beyond data sharing of common homogeneous databases. Knowledge of data heterogeneity, systems interaction, and business rules are not considered, either.

The GQS employs a methodology following the basic engineering paradigm: New methods and a conceptual design for model-assisted global query system (MGQS) are developed from the research, then these new results are reduced to practice. A prototype MGQS is developed to implement this approach, to demonstrate these basic concepts, and to verify the proposed methods and the design.

4.3 Scope of the research

This work focuses on investigating new methods and techniques for information retrieval using the assistance from on-line enterprise metadata. Other data management functionalities such as updating and deleting information are extensions of this research that will be pursued in the future. The developmental process of the prototype MGQS is also a focus of this work in order to prove the basic concepts. Established user interface design guidelines and principles are considered and adopted for the prototype system where appropriate. However, this work does not specifically address the research areas of cognitive science and human factors, nor is it concerned with performance analysis with respect to other designs or mathematical formalisms on query processing. Experiments regarding the comparative efficiency and effectiveness of the proposed MGQS through the prototype system will be conducted as future work and thus are beyond the scope of the thesis.

4.4 MGQS Architecture

When the metadata is employed directly in the processing of global information, it assumes an active role in the integration process. Three progressive levels of functionality are provided; namely, repository (or passive), global query (or semi-active), and systems integration (or active). The metadata approach facilitates such activities without interfering with routine data transfers between sub-systems and sub-system independent operations. Since the model-assisted global query capability is the second of the three levels of functionality, it is developed on the basis of the first level of functionality; enterprise-wide repository [Bouziane 1991]. The contents (i.e., metadata) of the metadata are used to provide model-assisted capabilities for global information retrieval.

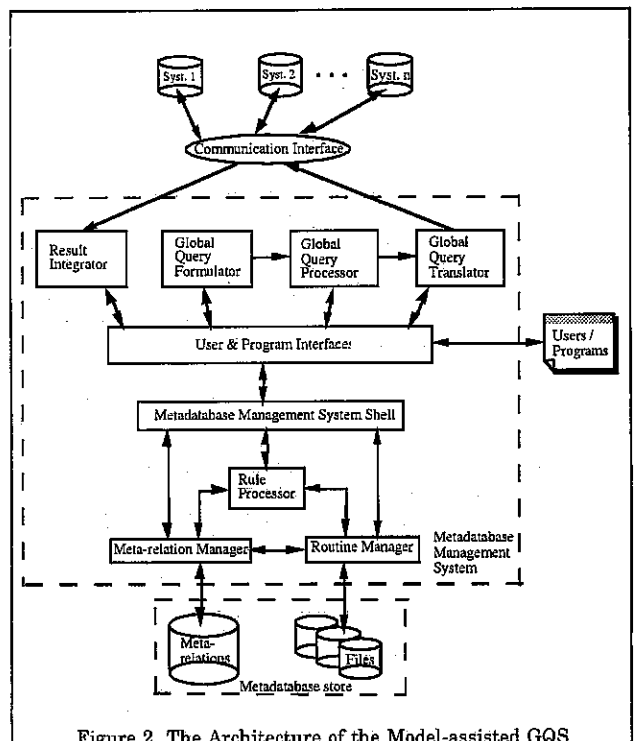


Figure 2 The Architecture of the Model-assisted GQS

The MGQS architecture that complies with the basic requirements and the idea of model assistance is shown in Figure 6. This architecture can be conceptually divided into two major components (1) metadata manager and (2) global query manager. The metadata manager consists of the metadatabase management system shell, rule processor, meta-relation manager, and routine manager. This metadata manager provides the passive mode functionality. It accesses the metadatabase, and provides model assistance for the global query operation through the user interface. The global query manager on the other hand, includes modules, query formulator, processor, translator and result integrator that would fulfill the basic requirements of global query.

4.5 Major results

The MGQS has achieved the following results toward the objective:

- 1) A definitive model of the new approach for the notion of "on-line intelligence and assistance" using enterprise metadata for global query formulation and processing. The model characterizes the various classes of knowledge (metadata) required for specific global query operations and their roles in these tasks. On this basis, it then defines the new approach and its on-line intelligence and assistance in terms of these metadata requirements.
- 2) metadatabase-assisted global query methods. These new methods include model traversal (which supports query formulation through direct manipulation of the metadata), implicit joins determination, local query generation (as opposed to schema/query mapping), global query optimization and conflicts resolution using the rule base contained in the metadatabase.
- 3) A MGQS architecture implementing the concept of model-assisted GQS. The unique properties of this architecture compared to previous results are the metadatabase and a rule processor. The new results from (1) and (2) above are embodied and operationalized through this architecture. A graphical user interface is included to provide end users with query formulation capabilities using these methods.
- 4) A global query language, the metadatabase query language (MQL), as a program interface for MGQS. This language can be used by a technical programmer and be embedded in a high level programming language such as C. With the model-assistance, MQL also provides adaptability to accommodate non-technical users. Many technical details of the local systems are automatically obtained from the metadatabase for formulating a global query. In addition, MQL can also be used as a metadatabase language to formulate ad hoc query for retrieving metadata from the metadatabase itself.
- 5) A prototype global query system based on the architecture in (3) for verifying the metadatabase-assisted approach to the global query problem. This prototype system is written in C language, implemented on a VAX workstation, and tested using Rensselaer's heterogeneous CIM systems (consisting of a shop floor controller, a process planning system and an order entry system). The implementation not only reduces the concept to practice, but also in its own right provides new possibilities of applications in this field.

- 6) An empirical study of global information modeling for the MGQS prototype. While based on existing results in this field, the study addresses the specific needs of MGQS and proves the concepts through the CIM facility at Rensselaer. In particular, this modeling effort provides sufficient metadata to enable the new methods in (2) for the prototype. It also avoids traditional schemata integration by identifying the contextual knowledge of corresponding data items (e.g., conflicting definitions and their resolutions) across the local systems.

5. Conclusions

Based on these results, the model-assisted approach facilitates certain difficult tasks in global query formulation and processing. First of all, it provides a direct approach using the model traversal method to allow users access the enterprise metadata and articulate directly in terms of information models for query formulation. Moreover, this approach utilizes the metadatabase to provide on-line assistance and thereby further alleviates the requirements of knowledge on the user part. This is a new level of enhancement to the user interface technology than previous efforts, which only simplified the language complexities (through, e.g., "uniform interface").

Equally significant is the fact that this approach allows for run-time local query generation with the on-line knowledge, rather than relying on integrating all schemata at compile time. Conflicts are resolved during the global query processing using a rule processor. This approach is, therefore, more flexible (in the spirit of open architecture) and less costly (in terms of mapping, conversion, and other overheads or restrictions) than previous approaches. Traditional distributed DBMS and most heterogeneous distributed DBMSs rely on schema integration to control or resolve conflicting data definitions at the local systems. This inevitably requires excessive schema translations at run-time, as well as imposes constraints against the incorporation of new systems into the global environment.

6. Future Work

A natural extension to this research is to upgrade the model-assisted global query system into a model-assisted data management system which includes other management functionalities such as update and deletion of local information. Two major issues involved are (1) maintaining data integrity and (2) concurrence control across distributed functional subsystems. Combining metadata and knowledge processing capabilities as in the MGQS provides major opportunities for tackling these issues. The extension of the MQL in this direction will lead to a metadata language which promises to support the management of knowledge in software implementation, metadata evolution, and interoperability of multiple systems without global serialization.

References

- Agrawal, R., N.H. Gehani, and J. Srinivasan, "OdeView: The Graphical Interface to Ode," *ACM 089791-365-5/90/0005/0034*, pp. 34-43, 1990.
- Astrahan et al., "System R: Relational Approach to Database Management," *ACM Transactions on Database Systems*, Vol. 1, No.2, June 1976.
- Bouziane, M. "Metadata Modeling and Management," *Ph.D. Dissertation*, Computer Science Department, Rensselaer Polytechnic Institute, Troy, N.Y., 1991.
- Cattell, R., "An Entity-based Database User Interface," *Proceedings ACM-SIGMOD Conference on Management of Data*, Santa Monica, California, pp.144-150, 1980.

- Chung, C.W., "DATAPLEX: An Access to Heterogeneous Distributed Databases," *Communications of the ACM*, Vol. 33, No. 1, pp.70-80, January 1990
- Deen, S.M., R.R. Amin, G.O. Ofori-Dwumfuoh, and M.C. Taylor, "The Architecture of a Generalised Distributed database System - PERCI*," *The Computer Journal*, Vol. 28, No. 3, pp. 282-290, 1985.
- ESPRIT Consortium AMICE, Open System Architecture for CIM, Project 688 AMICE Research Report, Berlin Heidelberg New York London Paris Tokyo Hong Kong: Springer-Verlag, ESPRIT Consortium AMICE (eds.) 1989.
- Fogg, D., "Lessons from a 'Living In a Database' Graphical Query Interface," *Proceedings ACM-SIGMOD*, pp.100-106, 1984.
- Goldfine, A.H. and P.A. Konig, "A Technical Overview of the Information Resources Dictionary System (Second Edition)," *NBS Special Publication NBSIR 88-3700*, National Bureau of Standards, Gaithersburg, MD, January 1988.
- Goulette, A.H., "The Information Resource Dictionary System," *Proceedings of fourth International Entity-Relationship Conference*, IEEE Press, New York, pp.114-122, 1985.
- Hansen, W.J., "User Engineering Principles for Interaction Systems," *AFIPS Conference Proceedings*, New Jersey, pp. 523-532, 1971.
- Herot, C., "Spatial Management of Data," *ACM Transactions on Database Systems*, Vol. 5, No. 4, pp. 493-513, 1980.
- Hirschman, L., "Natural Language Interfaces for Large-Scale Information Processing," First Symposium of Knowledge-Based Integrated Information Systems Engineering, February 1987
- Hsu, C., G. Babin, M. Bouziane, W. Cheung, L. Rattner, and L. Yee, "Metadatabase Modeling for Enterprise Information Integration," *Journal of Systems Integration*, Vol. 1, pp.5-37, January 1992.
- Hsu, C., M. Bouziane, L. Rattner, and L. Yee, "Information Resources Management in Heterogeneous, Distributed Environment: A Metadatabase Approach," *IEEE Transactions on Software Engineering*, Vol. SE-17, No. 6, June 1991.
- Hsu, C., M. Bouziane, L. Rattner, and L. Yee, "A Metadatabase Architecture for Heterogeneous, Distributed Environments: Global Information Resources Dictionary," *Proceedings of the Second International Conference On Computer-Integrated Manufacturing*, *IEEE Computer Society*, pp. 492-499, May 1990c.
- Kleinrock, L., "Distributed Systems," *Communications of the ACM*, Vol. 28, No. 11, pp.1200-1213, 1985.
- Krishnamurthy, R., W. Litwin, and W. Kent, "Language Features for Interoperability of Databases with Schematic Discrepancies," *Proceedings 1991 ACM SIGMOD International Conference on Management of Data*, pp.40-49, 1991.
- Litwin, W., L. Mark, and N. Roussopoulos, "Interoperability of Multiple Autonomous Databases," *ACM Computing Surveys*, Vol. 22, No. 3, pp. 267-293, 1990.
- Madnick, S.E., and R. Wang, "Evolution towards Strategic Application of Databases through Composite Information Systems," *Journal of Management Information Systems*, Vol. 5, No. 2, pp 5-22, 1988.
- McDonald, N., and M. Stonebraker, "CUPID: A User Friendly Graphics Query Language," *Proceedings of ACM PACIFIC*, San Francisco, California, pp. 127-131, 1975.
- Motro, A., "FLEX: A Tolerant and Cooperative User Interface to Databases," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 2, No. 2, pp.231-246, June 1990.
- Norcio A.F., and J. Stanley, "Adaptive Human-Computer Interfaces: A Literature Survey and Perspective," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 2, pp.399-408, March/April 1989.
- Potter, W.D., "KDL Advisor: A knowledge/Data Based System Written in KDL," *Processings of IEEE Conference on System Science*, Vol. III: Decision Support and Knowledge Based Systems Track, January 1988.
- Rattner, L., "Information Requirements for Integrated Manufacturing Planning and Control: A Theoretical Model," *Ph.D. Dissertation*, Department of Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, Troy, NY, 1990.
- Reddy, M.P., B.E. Prasad, P.G. Reddy, "Query Processing in Heterogeneous Distributed Database Management Systems," *Integration of Information Systems: Bridging Heterogeneous Database*, A.Gupta (ed.), IEEE Press, pp.264-277, 1989.
- Shipman, D.W., "The Functional Data Language DAPLEX," *ACM Transactions on Database Systems*, Vol. 6, No. 1, pp.140-173, 1981.
- Shneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley Publishing Company, 1986.
- Smith, J.M., P.A. Bernstein, U. Dayal, N. Goodman, T. Landers, K.W.T. Lin, and E. Wong, "Multibase Integrating Heterogeneous Distributed Database Systems," *Proceedings of AFIPS NCC*, Vol. 50, pp. 487-499, 1981.
- Smith, S.L., "Standard Versus Guidelines for Designing User Interface Software," *Behavior and Information Technology*, Vol.5, No. 1, pp.47-61, 1986.
- Smith, S.L., "User-System Interface Design for Computer-Based Information Systems," *Technical Report ESD-TR-82-132*, Hanscom Air Force Base, MA: USAF Electronic Systems Division, NTIS No. AD A115 853, 1982.
- Smith, S.L., "Man-Machine Interface (MMI) Requirements Definition and Design Guidelines: A progress Report," *Technical Report ESD-TR-81-113*, Hanscom Air Force Base, MA: USAF Electronic Systems Division, NTIS No. AD 096 705, 1981.
- Stonebraker, M., and J. Kalash, "TIMBER: A Sophisticated Relation Browser," *Proceedings of Eighth International Conference on Very Large Databases*, pp. 1-10, September 1982.
- Stonebraker, M., "Introduction for User Interfaces," *Readings of Database Systems*, M. Stonebraker (ed.) Morgan Kaufmann Publishers, Inc. San Mateo, California, pp.337-339, 1988.
- Templeton, M., D. Brill, S.K. Dao, E. Lund, P. Ward, A.P. Chen, and R. MacGregor, "Mermaid - A Front End to Distributed Databases," *IEEE International Conference of Data Engineering*, Los Angeles, CA, February 1986.
- Wang, R., and S. Madnick, "A Polygen Model for Heterogeneous Database Systems: The Source Tagging Perspective," *Proceedings of the 16th International Conference on Very Large Data Bases*, 1990.
- Wang, R., and S. Madnick, "Facilitating Connectivity in Composite Information Systems," *ACM Database*, Fall 1989.
- Wong, H.K.T., and I.Kuo, "GUIDE: Graphical User Interface for Database Exploration," *Proceedings of the Eighth International Conference on Very Large Databases*, Mexico City, pp.22-32, September 1982.
- Zloof, M.M., "Query by Example," *Proceedings on National Computer Conference*, Anaheim, California, pp. 431-437, 1975.
- Zloof, M.M., "Query-by-Example: A Data Base Language," *IBM Systems Journal*, Vol. 4, pp.324-343, 1977.