**Peter Dadam**
University of Ulm, Faculty for Informatics
Dept. Databases and Information Systems
dadam@informatik.uni-ulm.de

# BUSINESS INFORMATION SYSTEMS: TRENDS AND TECHNOLOGICAL CHALLENGES

**ABSTRACT**

*Although many companies are using computer-based business information systems (BIS) for more than 30 years, this area is still characterized by a very rapid technological development, „technological jumps", and changing user requirements. Especially during the last years new technological trends and solutions like object-oriented and object-relational database management systems, decentralization, client/server, data warehouse, multi-media support, and workflow management systems have been developed, offering new possibilities and chances but also challenges and risks. This paper describes and characterizes these developments, explains the driving forces behind them, and discusses some of the impacts and potentials of these trends and technologies on the development of business information systems.*

## Introduction

„New trends" are not always really something new, and a „new" technology is not always a real progress. For a better estimation of the present technological developments and trends in the field of Business Information Systems (BIS), let us take a short look back on the development up to now.

In the sixties, when more and more companies began to use computers, dedicated isolated applications like bill of material or payment accounting were realized. With the increasing number of those applications, more and more the problem appeared that either the same data had to be entered several times (each time for the dedicated application), or that new applications had to access data of existent applications that were not optimally adjusted to the new application.

Witold Abramowicz (ed.), Business Information Systems – BIS'97, International Conference, Poznań 9–11, April 1997, 509–524

Often, the used files first had to be recopied, restructured, and resorted, in order to be usable for the new application at all. In course of time, this had caused a complex network of dependencies between the existing data (see Fig. 1). In the seventies, the interest in such integrated applications increased heavily, and so it became the main problem to keep the data consistent. This led to the development and the first application of database management systems (DBMS), where were either based on the hierarchical data model or on the network data model (or variants of it). These DBMS offered tree-oriented or network-oriented data structures and corresponding database operators for retrieval and manipulation of the stored data. Thus they offer a higher abstraction level for the access on the data records to the application programmer compared to file systems (see Fig. 2) and therefore contributed to an increased data independence of the application programs.

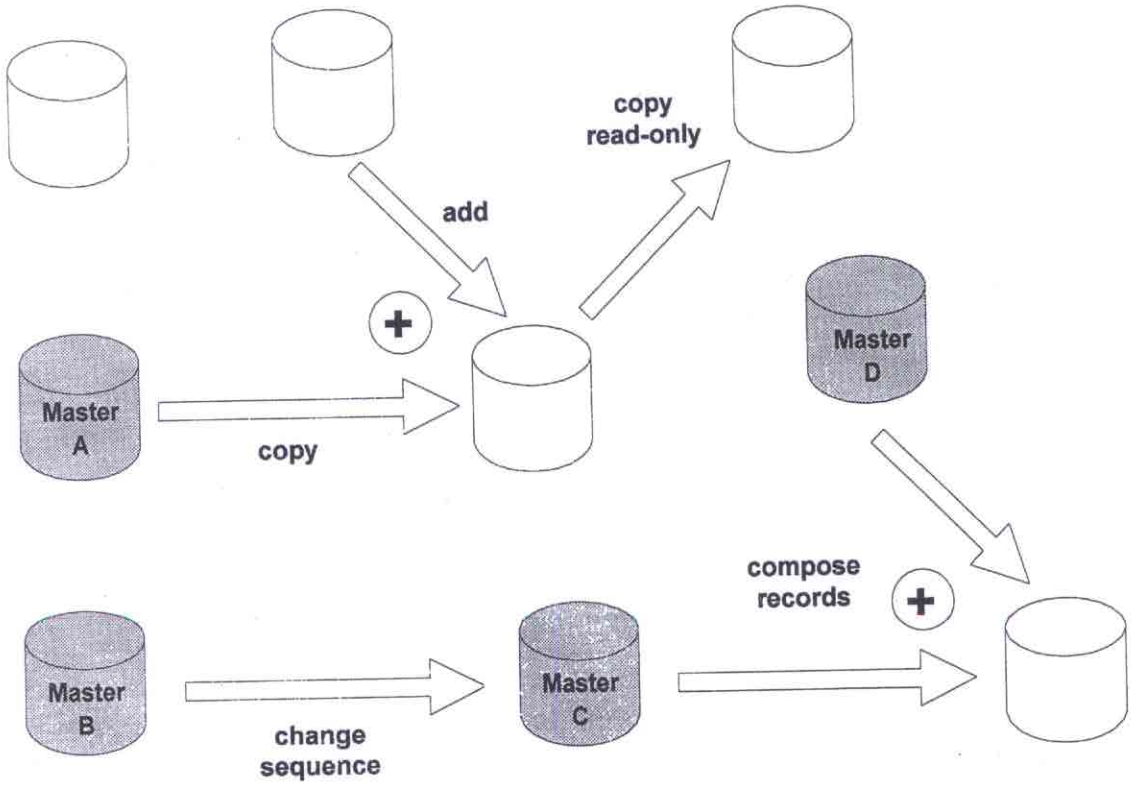
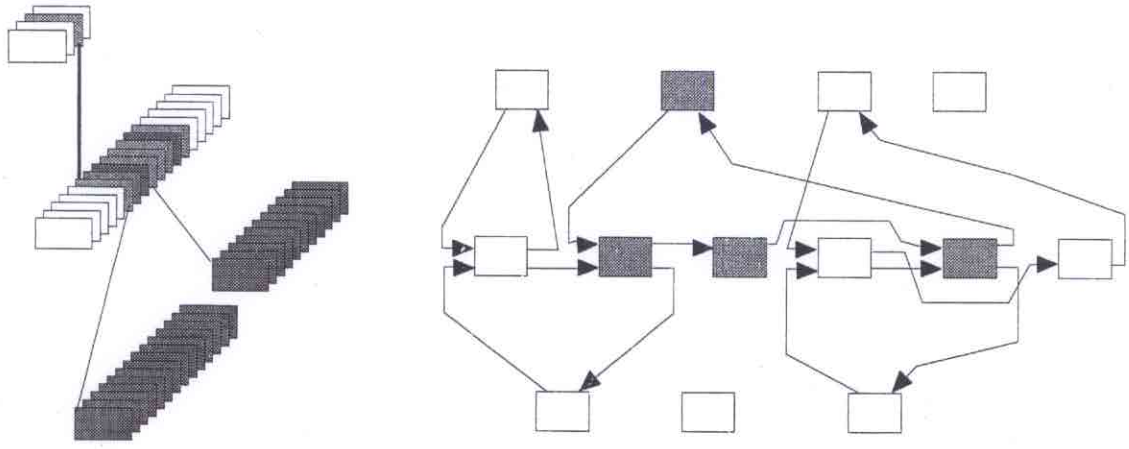



**Fig. 1. File dependencies**

But the real push for the introduction of database technology came at the end of the seventies and at beginning of the eighties with the massive change to on-line information systems. These systems could not use the typical batch processing anymore but required concurrent read and write access on the data. Concurrent read and write access in turn means that techniques to avoid lost

updates and to avoid resp. to detect deadlocks have to be implemented. These techniques are too complicated to be implemented by the normal application programmer and were therefore integrated in the database systems, thus essentially simplifying the development of on-line applications.



**Hierarchical Data Model**
(IMS)

**Network Data Model**
(CODASYL/DBTG)

**Fig. 2. First Generation DBMS**

Today, database-oriented BIS are the standard. Nearly all business areas manage their information with such systems. Very often, however, the information is not managed by one common BIS but by several specialized information systems which have been developed for the different business areas resp. functions. Consequently, the access to this information is only possible by using special application programs or with the knowledge of various database query languages. So it is one of today's challenge to unite these different information sources so that a fast and current access to all relevant information of the enterprise is possible. These and further approaches and trends are treated in the sections of the next chapter. Chapter 3 concludes the paper with a summary and outlook.

## 2. Current Trends and Driving Forces

In the following we want to examine the following trends and technologies more detailed:

- Relational, Object-relational, Object-oriented, and Multi-media DBMS
- Data Warehouse
- Decentralization and Client/Server-Systems
- Business Process Reengineering and Workflow Management

## 2.1 Relational, Object-relational, Object-oriented and Multi-media DBMS

The first relational database systems appeared on the market at the beginning of the eighties. Thanks to the standardization of SQL and to the consequent development of this technology, today, the relational database technology is the most important database technology. The high-level query language SQL has essentially simplified the definition of database queries thus allowing the development of powerful portable development tools. Thanks to the theoretic basis SQL-queries can be optimized and parallelized by the DBMS thus making relational database systems usable for the high-end on-line transaction processing as well as a basis for decision support systems.

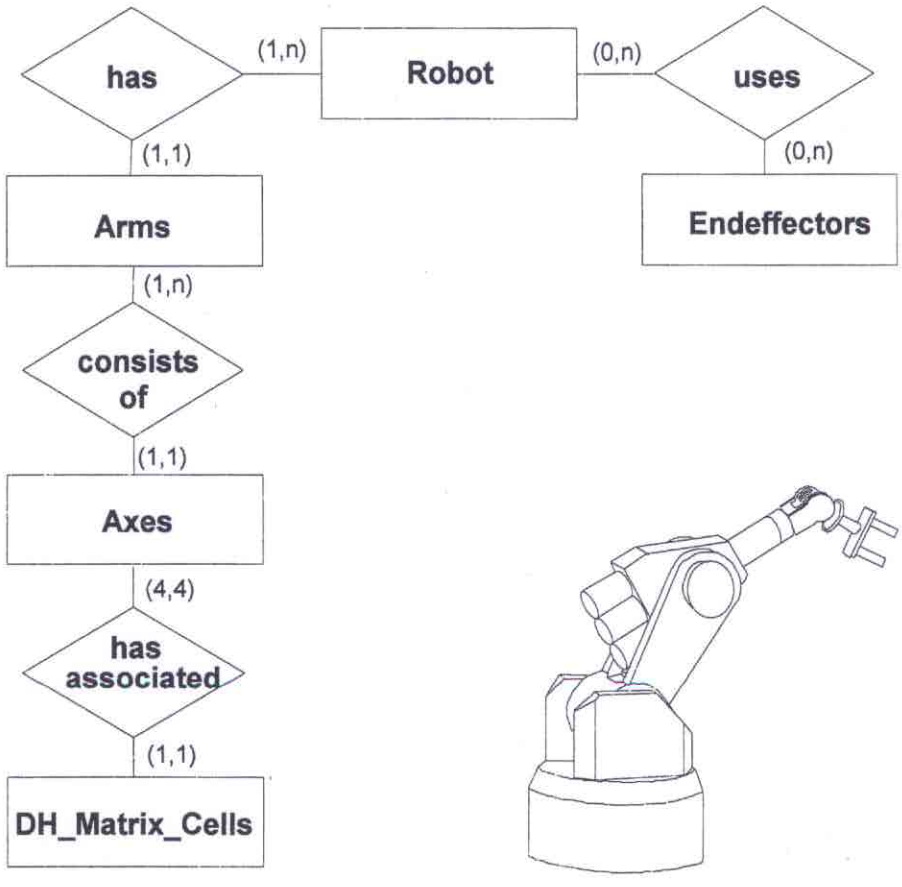
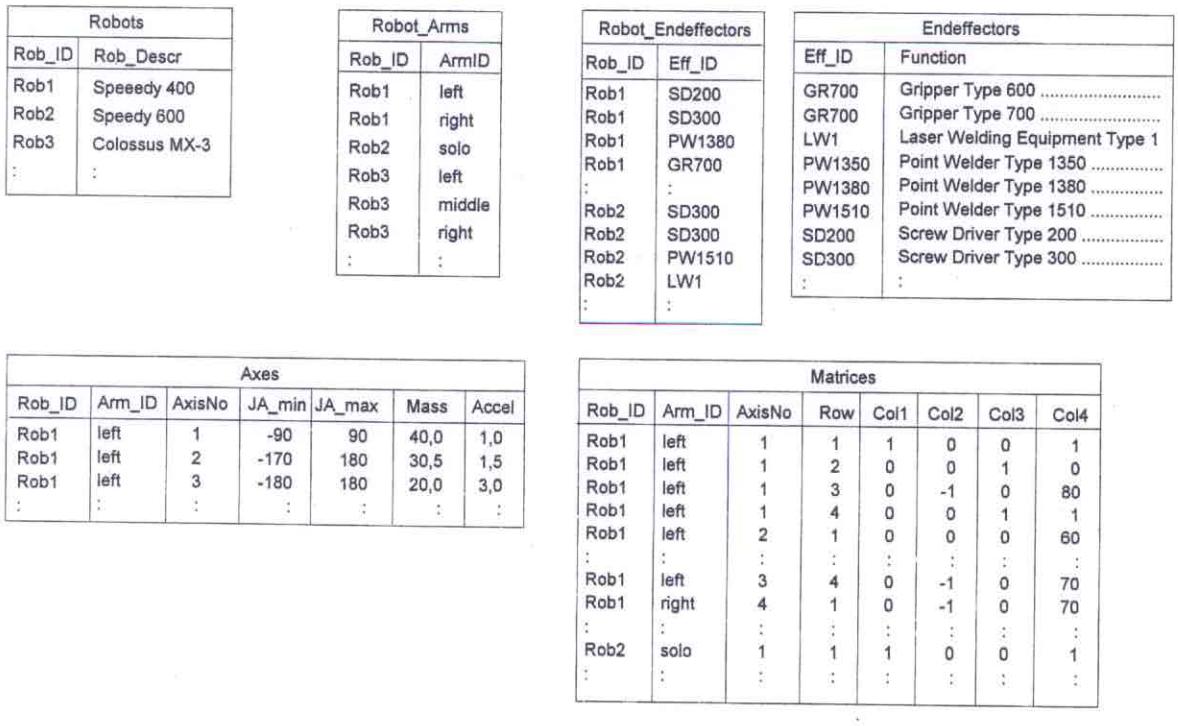



**Fig. 3. Robot (Simplified Example)**

The success of the relational database technology has also led to the increased use of relational database systems in areas for which they have not been originally concepted (office applications, engineering applications, geographical applications, etc.). For many of these application areas the current „flat" rela-

tional data model is too simple and causes an unnatural representation of data objects. For example, the data of a data object „robot" (see Fig. 3) has to be distributed to different relational tables (see Fig. 4). This is not only an unnatural representation of such an object, but leads – as a consequence – also to rather complex queries and inflated (by the relational join) result tables. But for these application areas not only the structure of the relational data model is too poor, also the simple operators and data types of the relational data model are not sufficient in this case. For these application areas, it is required that (analogous to programming languages) new application-specific data types and operations can bee added. For this reason, universities as well as industrial and federal research and development laboratories have been working for years on the development of a next generation of DBMS which offer a more richly structured data model as well as they allow the definition of new (possibly complex structured) data types and operations applicable on them (cf. [2, 6, 11, 15, 27, 29], as examples of this work). In the frame of the ISO standardization effort on SQL 3, since several years work is going on correspondent extensions of the relational data model [12, 30, 31, 35]. Up to now, however, no correspondingly extended relational data model has appeared on the market, that meets these requirements sketched below.



Robots

| Rob_ID | Rob_Descr |
|---|---|
| Rob1 | Speeedy 400 |
| Rob2 | Speedy 600 |
| Rob3 | Colossus MX-3 |
| : | : |

Robot_Arms

| Rob_ID | ArmID |
|---|---|
| Rob1 | left |
| Rob1 | right |
| Rob2 | solo |
| Rob3 | left |
| Rob3 | middle |
| Rob3 | right |
| : | : |

Robot_Endeffectors

| Rob_ID | Eff_ID |
|---|---|
| Rob1 | SD200 |
| Rob1 | SD300 |
| Rob1 | PW1380 |
| Rob1 | GR700 |
| : | : |
| Rob2 | SD300 |
| Rob2 | SD300 |
| Rob2 | PW1510 |
| Rob2 | LW1 |
| : | : |

Endeffectors

| Eff_ID | Function |
|---|---|
| GR700 | Gripper Type 600 |
| GR700 | Gripper Type 700 |
| LW1 | Laser Welding Equipment Type 1 |
| PW1350 | Point Welder Type 1350 |
| PW1380 | Point Welder Type 1380 |
| PW1510 | Point Welder Type 1510 |
| SD200 | Screw Driver Type 200 |
| SD300 | Screw Driver Type 300 |
| : | : |

Axes

| Rob_ID | Arm_ID | AxisNo | JA_min | JA_max | Mass | Accel |
|---|---|---|---|---|---|---|
| Rob1 | left | 1 | -90 | 90 | 40,0 | 1,0 |
| Rob1 | left | 2 | -170 | 180 | 30,5 | 1,5 |
| Rob1 | left | 3 | -180 | 180 | 20,0 | 3,0 |
| : | : | : | : | : | : | : |

Matrices

| Rob_ID | Arm_ID | AxisNo | Row | Col1 | Col2 | Col3 | Col4 |
|---|---|---|---|---|---|---|---|
| Rob1 | left | 1 | 1 | 1 | 0 | 0 | 1 |
| Rob1 | left | 1 | 2 | 0 | 0 | 1 | 0 |
| Rob1 | left | 1 | 3 | 0 | -1 | 0 | 80 |
| Rob1 | left | 1 | 4 | 0 | 0 | 1 | 1 |
| Rob1 | left | 2 | 1 | 0 | 0 | 0 | 60 |
| : | : | : | : | : | : | : | : |
| Rob1 | left | 3 | 4 | 0 | -1 | 0 | 70 |
| Rob1 | right | 4 | 1 | 0 | -1 | 0 | 70 |
| : | : | : | : | : | : | : | : |
| Rob2 | solo | 1 | 1 | 1 | 0 | 0 | 1 |
| : | : | : | : | : | : | : | : |

**Fig. 4: Relational Representation of Robot**

The major part of the so-called object-oriented database systems (OODBMS) currently on the market are still not much more than a C++ or Smalltalk language with the quality to manage persistent objects. In these systems, the logical and physical representation of objects is more or less the same. To access and manipulate objects, usually procedural (navigational) DBMS interface operations have to be used. Therefore, the OODBMS of today stand nearer to the database systems of the first generation (hierarchical data model, network data model) than to the relational databases – except to a certain degree for the system O2 [13, 14] whose query language is the target language for the OODBMS „standardization" of the ODMG [4, 27, 43]. Because of the small abstraction from the physical data structures and many other weaknesses these systems are usually not suitable for the support of large-scale, enterprise-wide applications. Of course, this is in all a very unsatisfactory situation for all enterprises which are faced with the problem to develop such applications.

For several years the interest in the integration of text, graphics, audio and video data into the business information systems has been increasing. For this, the database interface has to be extended by new functions as for example the search for substrings within a text, the play of audio and video data as well as the free positioning on such data streams. Most of the today's relational database systems offer so-called binary large objects (BLOBS, LOBS) as an attribute type which can integrate data in the mega byte or giga byte range. Many of these systems are still missing the capabilities for adding application specific functions to this binary large object. This gap is closed mainly by the so-called object-relational DBMS currently appearing on the market [7, 26, 36]. But they do not really extend the model of the flat relations and thus do not constitute a solution for the previously detailed problem of the missing complex object support. Therefore, these systems have to be considered as an intermediate step towards generalized relational or more powerful object-oriented DBMS.

## 2.2. Data Warehouse

To be able to quickly react on incoming customer inquiries, to quickly recognize changes in the customers' demand behavior, to have actual information on the use of capacity etc. a quick access to the relevant information within an enterprise is needed. Often it is needed to combine and relate different kinds of data, that is to perform complex correlation and regression analyses on these data. Most of the information needed is already available in computer-based information systems within the companies, in principle. As already mentioned, however, this information is often spread over many different information systems. Mostly it is available via application systems which are based on the first generation database technology or even file-systems.

To allow quick and convenient access to all this information it would be desirable, to store all data in a (logically) central database and migrate (re-

implement) all these applications to commonly work on this new common platform. However, the large amount of existing application programs make impossible a short-term change to such a common information system. Furthermore, not all of the applications could be adequately served by today's (relational) database technology, as already mentioned in Sect. 2.1. One solution to this problem would be, to implement additional application programs for data extraction and data analysis for every existing information system where this functionality should be supported. This, however, would result in a high implementation effort (old database technology, many individual application programs needed) and would also potentially disturb the performance of mission-critical applications.

**Fetch all information related to robot 'Rob1':**

```
SELECT  r.Rob_ID, r.Rob_Descr, ar.Arm_ID, ac.AxisNo, m.Row,
        m.Col1, m.Col2, m.Col3, m.Col4, ac.JA_min, ac.JA_max,
        ac.Mass, ac.Accel, re.Eff_ID, e.Function
FROM    Robot r, Robot_Arms ar, Axes ac, Matrices m,
        Robot_Endeffectors re, Endeffectors e
WHERE   r.Rob_ID = 'Rob1' AND r.Rob_ID = ar.Rob_ID AND
        ar.ROB_ID = ac.Rob_ID AND ar.Arm_ID = ac.Arm_ID AND
        ac.Arm_ID = m.Arm_ID AND ac.Rob_ID = m.Rob_ID AND
        ac.AxisNo = m.AxisNo AND r.Rob_ID = re.Rob_ID AND
        re.Eff_ID = e.Eff_ID
```

**Fig. 5. Query**

The currently preferred solution for this problem is „Data Warehousing". The term „Data Warehousing" resp. „Data Warehouse" (DW) shall suggest that the data resp. information of an enterprise are made accessible easily and conveniently like in a warehouse. From a technical point of view, a DWH is an autonomous dedicated DBMS with tools for data analyses and data extraction [21, 38, 44]. To not disturb the mission-critical applications by the execution of complex analysis queries, the DWH-system usually does not operate directly on the data of the operative systems, but only on a copy resp. a partial copy of them. This means that the DWH-system usually operates on not really current data but on a snapshot-copy which is being actualized in determined intervals by downloading data from the operational systems (see Fig. 6). This decoupling from the mission-critical application is very important in the case when very complex performance- and I/O-intensive queries are made like for example in the area of „data mining" [22, 40, 41, 42]. In most cases, the DWH-system is based on relational database technology, but also dedicated implementations are available (see [44] for a detailed discussion).

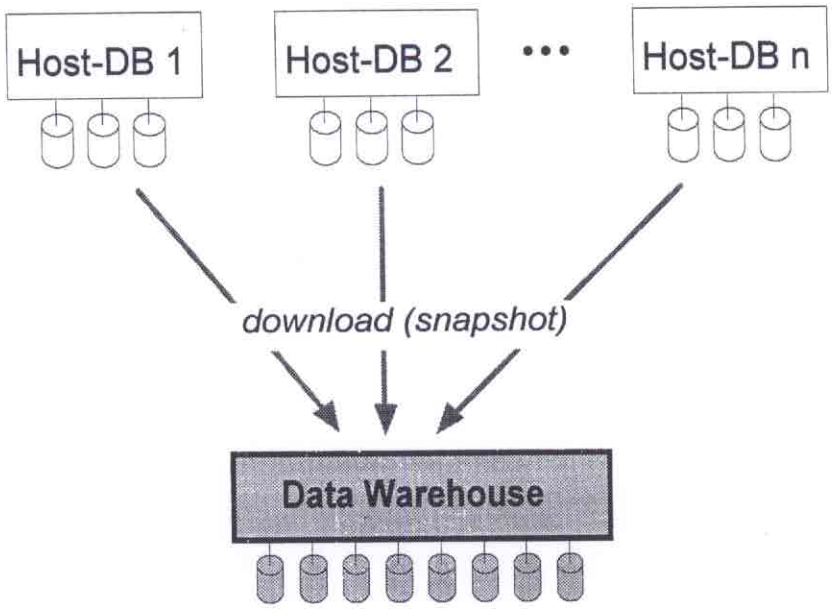



**Fig. 6. Data Warehouse**

In many companies the total amount of information stored in computer-based information systems is in the range of hundreds of giga byte or even tera byte. Thus it is not feasible to copy all the data to the DWH. Therefore, an „intelligent" selection (and also pre-processing) of this data is required. For the same reason, the „refresh" of data, that is the down-load of a newer version from the operational systems cannot be performed by down-loading the complete copy but must be restricted to the parts where changes have occurred since the last refresh. Under this aspect, maintaining a DWH has a close relationship to maintaining materialized views [5, 8, 20, 45] which is a non-trivial problem, if the existing operational systems shall not or cannot be modified and/or if no negative impact on their performance during normal operation can be tolerated. Therefore, installing a DWH is not the problem. The problem is its cost-adequate utilization and maintenance.

## 2.3. Decentralization and Client/Server-Systems

Today, a lot of companies have to face a hard competition and cost strain. Partially, they react with a restructuring of their enterprise into smaller operative units with their own cost responsibility (profit centers, independent business units). Cost strain and business decentralization also influence, of course, the central computing centers resp. the central information systems. The two catchwords mentioned most frequently are „downsizing" and „client/server". „Downsizing" generally means the (partial) substitution of the central computer (mainframe) by a number of small, interconnected systems, typically running under Unix or Windows NT. Cheap mass hardware is used here instead of expensive special hardware.

Although everybody is talking about „client/server" it is often not clear what is really meant. The definition „client/server" (C/S) just means that there is a passive part (server) and an active part (client) and does not say anything about the concrete realization or the separation of work between them. For a classification of the different C/S-versions it is helpful to distinguish between the presentation function, the application function, and the data management function of an application (see Fig. 7). The „cuts" in Fig. 8 indicate that the complete application can be separated in different ways between client and server. A section on point 4 for example describes a C/S-system where the server is a pure DB-server and the application runs completely on the client system. The client communicates with the server via the SQL interface in this case. A section on point 3, on the other hand, means that parts of the application run on the server, and some other parts run on the client. This form of realization can be found for example as stored procedures or triggers. This approach allows to reduce the number of required SQL-calls by implementing a part of the application logic at the server side using the functionality of stored procedures or triggers which is already offered by many SQL DBMS although not defined in the current SQL standard but only planned for SQL 3 [33].

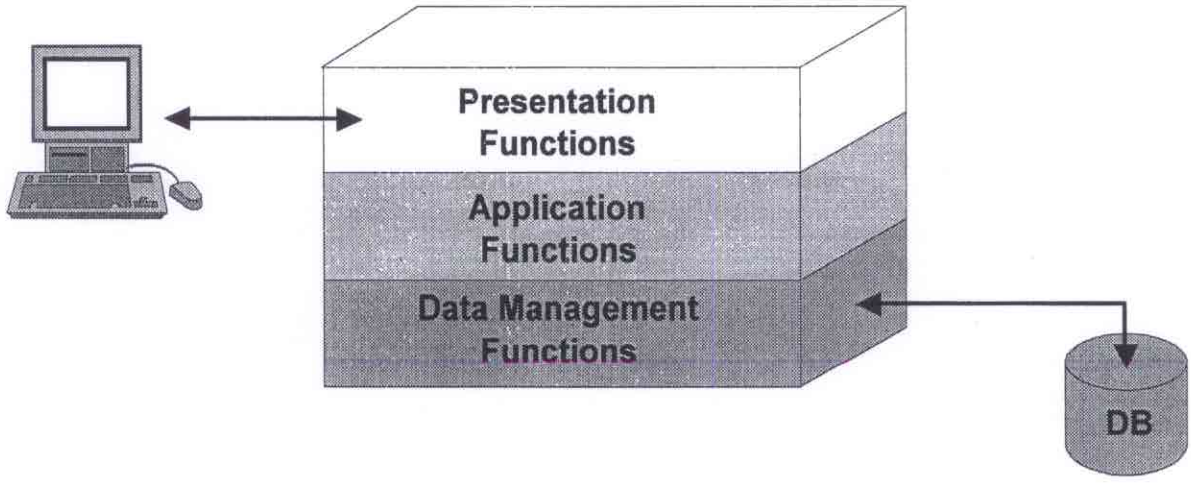



**Fig. 7. Functions of an Application Program**

The different variants for splitting the application into the client part and the server part lead to significant differences in the implementation of the systems, their maintenance and their performance. Nevertheless, all that is quite uncritical as long as there is only one server. Several servers are also uncritical as long as the complete application can be separated into independent partial applications, where each needs only its own database.

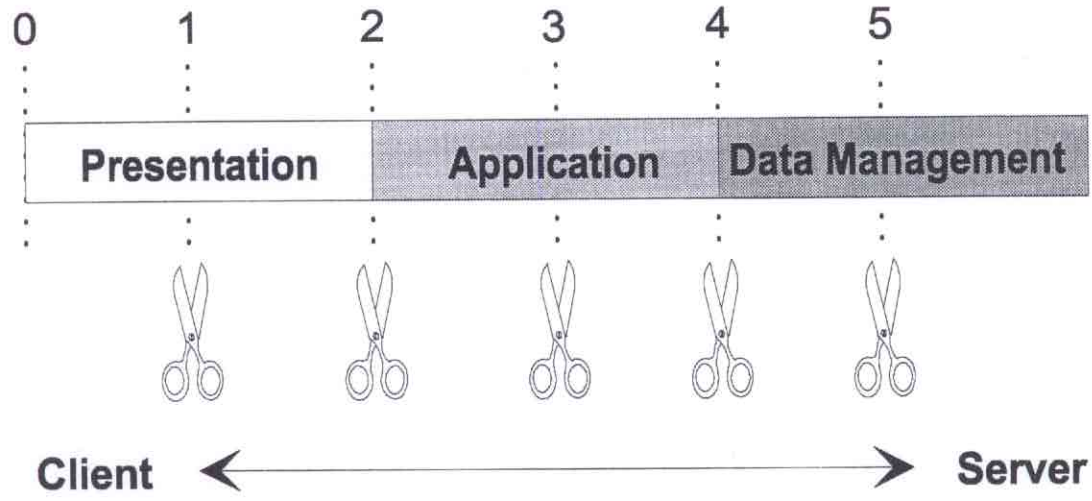



**Fig. 8: Possible Distribution of Functions**

Much care is necessary, if logically tightly integrated applications shall be separated on several servers. Basically, the complete range of problems, that one faces in distributed systems, can occur here. One can be confronted, for example, with global deadlocks that cannot be detected locally (see Fig. 9), by inconsistent databases because of non-compliance of the two-phase commit protocol, or by data loss due to not properly implemented data replication procedures (see [10] for a detailed treatment of these issues). These problems are often not seen or are completely underestimated when beginning such projects and consequently have often led to the unsuccessful termination of such projects resp. to remarkably higher costs in several cases.

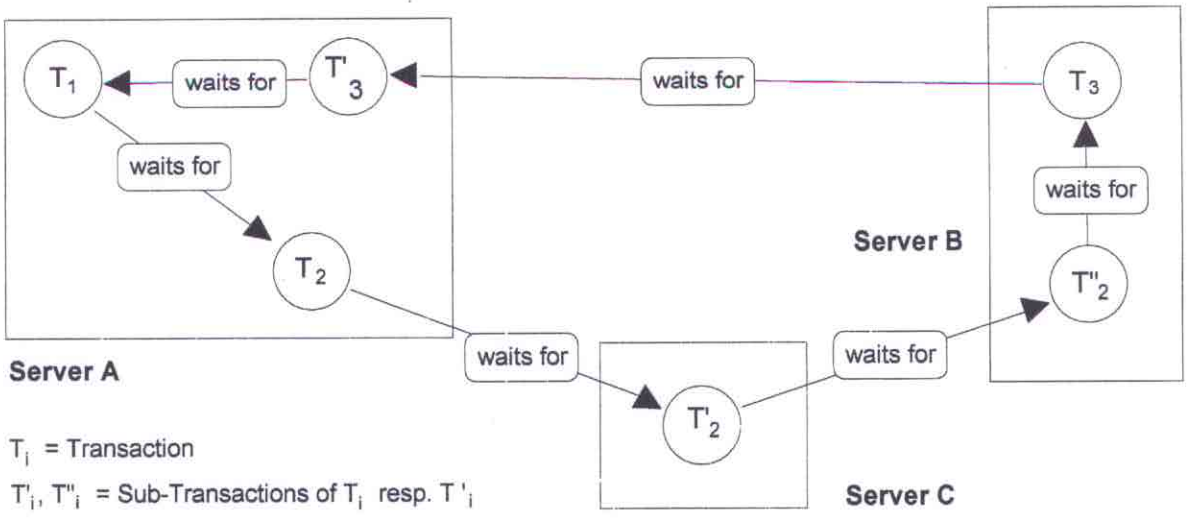



**Fig. 9. Global Deadlock**

## 2.4. Business Process Reengineering and Workflow Management

Today, the globalization of markets forces a lot of enterprises to face a very hard strain of competition. For many of these enterprises the leaning of processes and decision ways, a shortening of the manufacturing and development time with an equally good or improved quality, as well as a quicker reaction to the market fluctuations becomes a question of survival. The bundle of measures to achieve these aims is generally called business process reengineering (BPR). But the BPR can only be successful if afterwards the BIS can adequately support the new process resp. can be adjusted quickly to the new process. Furthermore, it would be desirable that the BIS does not only enable the new business process, but also supports it by offering actively the correct information in the correct granularity at the correct time to the correct person so that this person can make the required (partial) task as efficiently as possible. This means that our today's database-oriented BIS have to develop towards process-oriented BIS.

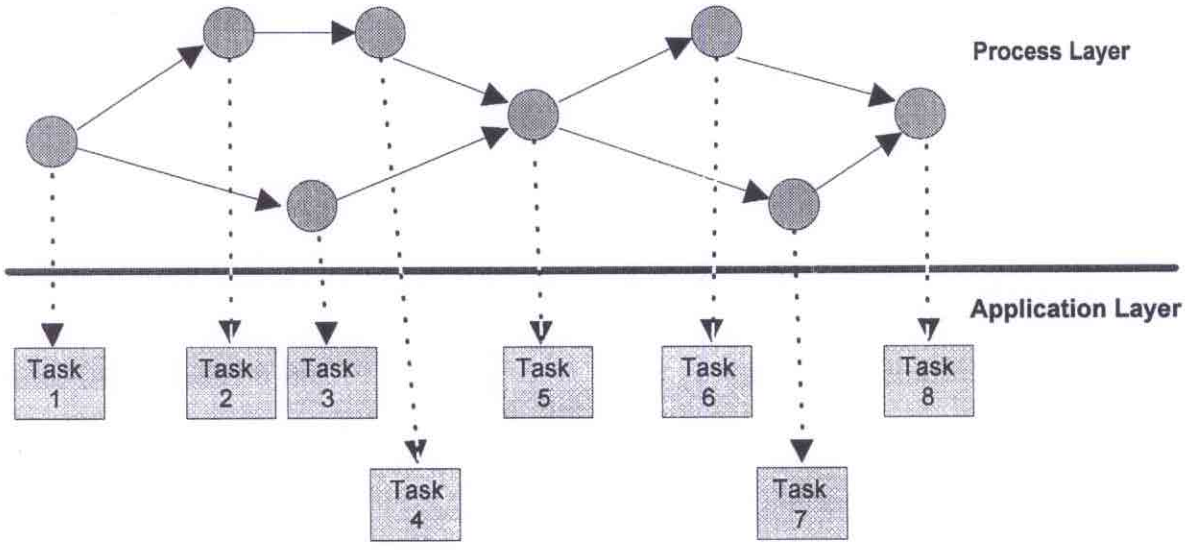



**Fig. 10. Separation of Process Description and Application Tasks**

As one can easily imagine such process-oriented BIS cannot be solved with conventional programming techniques, where the control flow is hard-wired in the program code. Each change of the process would cause high adjustment costs. The solution of this problem is the separation of the control flow and the application code like illustrated in Fig. 10. This is already realized (at least partially) in some process-oriented workflow management systems (WfMS), like FlowMark [23, 37] and WorkParty [39], for example. If the application components are properly implemented, in many cases a WfMS-based BIS can be adjusted to a change of the business process with a relatively low effort. The WfMS of today are still rather limited. Usually they enforce a strict execution of

the preplanned sequence of steps. Some systems allow the users to deviate for this sequence but at the risk, that this may lead to inconsistencies. In addition, little support is still available for the abortion of a running workflow instance and a system-supported „semantic rollback" of the changes and tasks performed so far. The on-going research in this area and related fields (see [3, 9, 17, 18, 19, 24, 25, 28] for example) will certainly positively influence the further development of these systems.

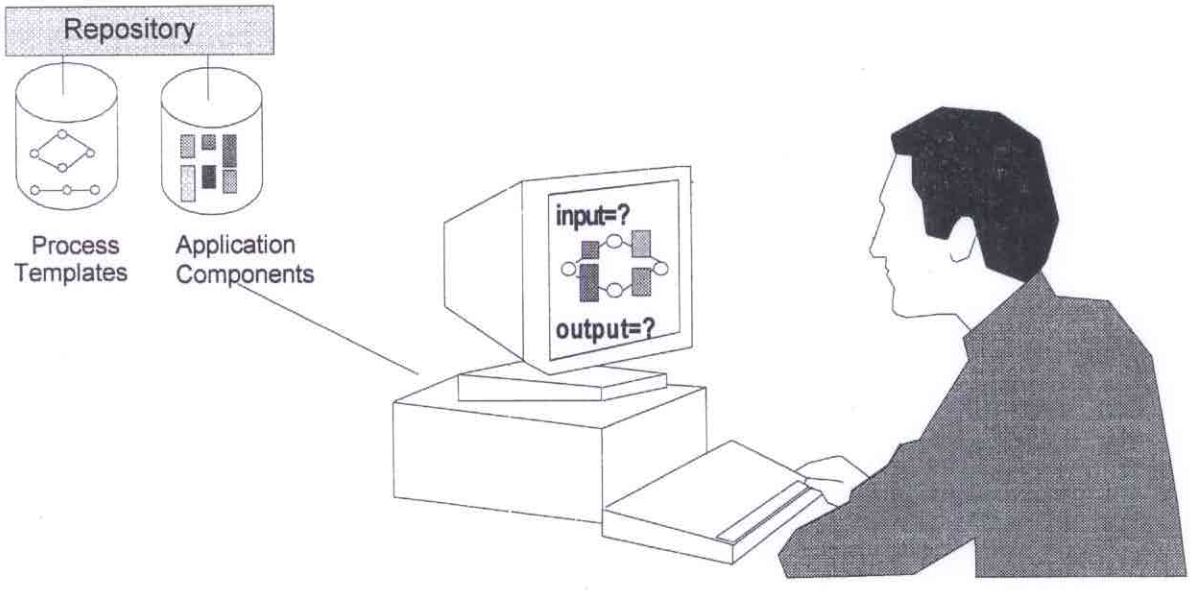



**Fig. 11. Future Application Development?**

For the future, this technology has the potential to lead to a completely different kind of application development. Like there are specific sample data models for specific branches, in the future there could be process templates for specific lines of business, too, that could be purchased. Furthermore, the current research and development efforts in the area of component ware [1, 32, 34] give hope that there will once be a market for application components.

Both together could mean that in future programming reduces to the selection of the correct process template, its configuration, and insertion of the appropriate application components in the style of „plug-and-play" (see Fig. 11).

## 3. Summary and Outlook

Today, all larger and medium-size companies are faced with a strong competition, with a high cost pressure, and with rapidly changing market situations. Thus, fast and up-to-date access to all relevant internal information on costs and capacities as well as on the status of ongoing business processes is a prerequisite

for quick and correct decisions. But fast access alone is not sufficient. The information must also be presented in the appropriate form, so that it often must be pre-processed, e.g. aggregated and correlated with other information, to become really useful. As an enterprise is usually not static but changes and adapts its business processes all the time, the kind of information needed, the context, and its adequate pre-processing is also changing. In addition, to really adequately and efficiently support the business processes, BIS should change from „passive" information systems to „active" or „cooperative" information systems. They should offer the right information at the right place to the right person so that this person can make the right (and necessary) decision.

Unfortunately, very often today's BIS do not offer this flexibility. Often implemented on software platforms which are 15 years and older, every modification to these old programs is very costly and must be made with great care to avoid malfunction of the program and resulting data inconsistencies. DWH technology can be used as a quick measure to gain access to relevant information without having to change the existing systems. It offers tools for data analysis and, by copying the data into a dedicated (DWH-)system, it does not disturb the execution of mission critical applications. But as soon as there is a large amount of data stored in the DWH systems, the maintenance of these replicas will become a problem, too. Thus the usage of this technology has to be planned with some care if it shall be become really useful.

On a longer term, however, the usage of DWH technology alone will not be sufficient. It can, at best, solve the information provision problem, but it does not solve the problem that the BIS can not be flexibly adapted to changes in the business processes and that they do not actively support these processes. Therefore, many of the BIS of today will have to be re-implemented. In many cases, re-implementation will go along with decentralization on C/S platforms for cost and other reasons. Decentralization can become very risky and costly, however, if logically integrated applications or data are distributed across several servers. The savings which can be achieved at first glance, may turn into the opposite if decentralization is not done with great care and competence.

For getting BIS more „active", (process-oriented) WfMS are a very promising technology. On a longer term, they may even lead to a new programming style or even programming paradigm by separating the (global) control flow from the pure application functions. By doing so, the resulting (BIS) programs can be adapted to process changes much easier than with conventional software technology. One must recognize, however, that the WfMS of today are the first generation of such systems. They still do not offer the flexibility, support for exception and error handling, and may also have problems to handle a large number of users. Thus, a careful selection of the right application to be supported by WfMS technology is mandatory.

Taking all concerns together, one could get the impression that the best decision is to do nothing and to just wait for the optimal solution appearing at the market. This, however, is also not a wise decision. One reason is that it is rather unlikely that ever an „optimal" solution for the application area under consideration will become available. Usually, there are too many conflicting goals which a software vendor has to strike for. Therefore one will always have to make a compromise. Another reason is that one has to build up the relevant experience to make the right decisions. How does one recognize, that it is the optimal system if one is not able to pose the right questions? The third reason is that the whole environment (inside and outside the company) is changing so quickly that staying with the old solutions can usually not be tolerated.

And the next challenge is already waiting: Within just a few years the internet has developed to an important medium for the enterprise representation as well as for the product marketing. Whether or not the presence in the internet becomes a business success, depends on the quality of the corresponding services as well as the corresponding costs. Offering services and goods via the internet does only make sense, if incoming inquiries can be processed automatically to a large extent. Otherwise the response time will be too long and handling of mass inquiries becomes too cost expensive.

Therefore the best advice one can give is: To look at and to try to understand the new technologies. Not to look at them in an isolated way but to look at the whole picture. If starting to use a new technology, to start with little steps. Not to use a new and unproved technology in projects which are critical to the success (or even survival) of the company. But start to use the new technologies in order to build up experience and to better understand their potential benefits and risks.

### *Acknowledgements*

*Many thanks to Christiane Köppl for her translation support and to Manfred Reichert for his valuable suggestions for improving the readability of this paper.*

## Literature

[1] Adler R. M., *Emerging Standards for Component Software*. IEEE–Computer, Vol. 28, No. 3, March 1995, pp. 68–77

[2] Abiteboul S., Fischer P.C., Schek H.J. (Eds.), *Nested Relations and Complex Objects in Databases*. Lecture Notes in Computer Science 361, Springer–Verlag, 1987

[3] Attie P.C., Singh M.P., Sheth A., Rusinkiewicz M., *Specifying and Enforcing Intertask Dependencies*. Proc. Int'l Conf. on Very Large Data Bases, Dublin, Ireland, Aug. 1993, pp. 134–145

[4] Bancilhon F., Ferran G., *ODMG–93: The Object Database Standard*. Data Engineering, IEEE Computer Society technical committee, Vol. 17, No. 4, December 1994, pp. 3–14

[5] Blakeley J. A., Coburn N., Larson P. A., *Updated Derived Relations: Detecting Irrelevant and Autonomously Computable Updates*. ACM Transactions on Database Systems, Vol. 14, No. 3, Sept. 1989, pp. 369–400

[6] Cattell R. G. G., *Object Data Management – Object-Oriented and Extended Relational Database Systems*. Addison–Wesley Publ. Comp., 1991

[7] Cheng J.M., Mattos N.M., Chamberlin D.D., DeMichel L.G., *Extending Relational Database Technology for New Applications*. IBM Systems Journal, Vol. 33, No. 2, 1994, pp. 264–279

[8] Colby L.S., Griffin T., Libkin L., Mumick I.S., Trickey H., *Algorithms for Deferred View Maintencance*. Proc. ACM–SIGMOD Conf., 1996, Montreal, Canada, June 1996, pp. 469–480

[9] Dadam P., Kuhn K., Reichert M., Beuter Th., Nathe M., *ADEPT: Ein integrierender Ansatz zur Entwicklung flexibler, zuverlässiger kooperierender Assistenzsysteme in klinischen Anwendungsumgebungen*. Proc. GI–Jahrestagung (GISI 95), Zürich, Sept. 1995, S. 677–686

[10] Dadam P., *Verteilte Datenbanken und Client/Server-Systeme*. Springer–Verlag, 1996

[11] Dadam P., Linnemann V., *Advanced Information Management (AIM): Advanced Database Technology for Integrated Applications*, IBM Systems Journal, Vol. 28, No. 4, 1989, pp. 661–681

[12] Demuth B., Demuth F., *Intergalaktische Kommunikation – SQL3 – die Datenbanksprache für das Jahr 2000*, iX-Magazin. Heft 3, 1994, S. 50–61

[13] Deux O., *The Story of $O_2$. IEEE Transactions on Knowledge and Data Engineering*, March 1990, pp. 91–108

[14] Deux O. et al., *The $O_2$ System*. Communications of the ACM, Vol. 34, No. 10, October 1991, pp. 35–48

[15] Dogac A., Özsu M.T., Biliris A., Sellis T. (Eds.), *Advances in Object-Oriented Database Systems*. NATO ASI Series, Series F: Computer and System Sciences, Vol. 130, Springer–Verlag, 1994

[16] Ellis C. A., Keddara K., Rozenberg G., *Dynamic Change Within Workflow Systems*. Proc. Conf. on Organisational Computing Systems (COOCS'95), Milpitas, CA, 1995, pp. 10–21

[17] Elmagarmid A. K. (Ed.), *Database Transaction Models for Advanced Applications*. Morgan Kaufmann Publ., 1992

[18] Forst A., Kühn E., Bukhres O., *General Purpose Workflow Languages*. Distributed and Parallel Databases, Vol. 3, 1995, pp. 187–218

[19] Georgakopoulos D., Hornick M., Sheth A., *An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure*. Distributed and Parallel Databases, Vol. 3, 1995, pp. 119–153

[20] Gupta A., Mumick I. S., *Maintenance of Materialized Views: Problems, Techniques, and Applications*. Data Engineering, IEEE Computer Society technical committee, Vol. 18, No.2, June 1995, pp. 3–18

[21] Hammer J., Garcia–Molina H., Widom J., Labio W., Zhuge Y., *The Stanford Data Warehousing Project*. Data Engineering, IEEE Computer Society technical committee, Vol. 18, No. 2, June 1995, pp. 41–48

[22] Houtsma M., Swami A., *Set-Oriented Mining for Association Rules in Relational Databases*. Proc. Int'. Conf. on Data Engineering, Taipei, Taiwan, March 1995, pp. 25–33

[23] IBM FlowMark – Managing Your Workflow, Version 2.1, IBM Corp., SH19–8234–00, 1994

[24] Special Issue on Workflow and Extended Transaction Systems. Data Engineering, IEEE Computer Society technical committee, Vol. 16, No. 2, June, 1993

[25] Special Issue on Workflow Systems. Data Engineering, IEEE Computer Society technical committee, Vol. 18, No. 1, March 1995

[26] Jungebluth V., *Objekt–relationale Datentypen: Illustra und Informix-Online verschmelzen zum Universalsystem.* Computerwoche 42/96, S. 33–34

[27] Kim W. (Ed.), *Modern Database Systems – The Object Model, Interoperability, and Beyond.* ACM Press / Addison–Wesley, 1995

[28] Leymann F., *Supporting Business Transactions via Partial Recovery in Workflow Management Systems.* Proc. BTW '95, GI–Fachtagung „Datenbanksysteme in Büro, Technik und Wissenschaft", Dresden, March 1995, S. 51–70

[29] Linnemann V., Küspert K., Dadam P., Pistor P., Erbe R., Kemper A., Südkamp N., Walch G., Wallrath M., *Design and Implementation of an Extensible Database Management System Supporting User Defined Data Types and Functions.* Proc. Int'l Conf. on Very Large Databases, Los Angeles, August/September 1988, pp. 294–305

[30] Mattos N., *Private communication*, Dresden, Germany, 1995

[31] Melton J., *Object Technology and SQL: Adding Objects to a Relational Language.* Data Engineering, IEEE Computer Society technical committee, Vol. 17, No. 4, December 1994, pp. 15–26

[32] de Mey V., Nierstrasz O., *The ITHACA Application Development Environment.* In: Tsichritzis D. (Ed.): Visual Objects, Centre Universitaire D'Informatique, University of Geneva, July 1993, pp. 267–280

[33] Melton J., Simon A. R., *Understanding the New SQL: A Complete Guide.* Morgan Kaufmann Publ., 1993

[34] Nierstrasz O., Gibbs S., Tsichritzis D., *Component-Oriented Software Development.* Communications of the ACM, Vol. 35, No. 9, Sept. 1992, pp. 160–164

[35] Pistor P., *Objektorientierung in SQL3: Stand und Entwicklungstendenzen*, Informatik–Spektrum, Band 16, Heft 2, April 1993, S. 89–97

[36] Reinwald B., Lehman T. J., Pirahesh H., Gottemukkala V., *Storing and Using Objects in a Relational Database.* IBM Systems Journal, Vol. 35, No. 2, 1996, pp. 172–191

[37] Schmidt M. T., *Notes – FlowMark Integration. Technical Report, IBM Software Solutions Division*, GSDL Böblingen, 1996

[38] Singleton J. P., Schwartz M. M., *Data Access Within the Information Warehouse Framework*, IBM Systems Journal, Vol. 33, No. 2, 1994, pp. 300–325

[39] Siemens Nixdorf Informationssysteme AG: WorkParty Benutzerhandbuch (Version 2.0), August 1995.

[40] Savasere A., Omiecinski E., Navathe S., *An Efficient Algorithm for Mining Association Rules in Large Databases.* Proc. Int'l Conf. on Very Large Databases, Zurich, Switzerland, Sept. 1995, pp. 432–444

[41] Srikant R., Agrawal R., *Mining Generalized Association Rules.* Proc. Int'l Conf. on Very Large Databases, Zurich, Switzerland, Sept. 1995, pp. 407–419

[42] Srikant R., Agrawal R., *Mining Quantitative Association Rules in Large Relational Tables.* Proc. ACM–SIGMOD Conference, 1996, Montreal, Canada, June 1996, pp. 1–12

[43] Wade A. E., *Object Query Standards.* ACM SIGMOD–Record, Vol. 25, No. 1, March 1996, pp. 87–92

[44] Wu M. C., Buchmann A. P., *Research Issues in Data Warehousing.* Proc. BTW '97, GI–Fachtagung „Datenbanksysteme in Büro, Technik und Wissenschaft", Ulm, 1997 (to appear)

[45] Zhuge Y., Garci–Molina H., Hammer J., Widom J., *View Maintenance in a Warehousing Environment.* Proc. ACM–SIGMOD Conf., San Jose, Calif., May 1995, pp. 316–327