

A Federated Information System for Quality Management Processes*

P. Peters, P. Szczurko, M. Jarke

Informatik V, RWTH Aachen,

Ahornstr. 55, 52056 Aachen, Germany

email: {peters,szczurko,jarke}@informatik.rwth-aachen.de

Phone/Fax: +49 241 8021521/ +49 241 8888321

M. Jeusfeld

University of Science and Technology, ISMT, Kowloon, Hong Kong

email: jeusfeld@usthk.ust.hk

Abstract

As the structure of modern organisations changes to networks of decentralised, autonomous units, the demands on information systems alter from simple sources of data to links that connect those units along the workflows and support groups of knowledge workers with timely and task appropriate information. Fulfilling this task depends heavily on the ability to change the information system according to the business process evolution. As our approach to deal with those problems, we present a task oriented, federated information system architecture. We show the impact of this approach in an example where quality management tasks and systems in the industrial product life cycle has been integrated. Starting from the integration of task and information modelling based on a repository architecture, we show how conceptual models support communication and information access in the heterogenous environment by directly relating tasks to the information they need or provide. Furthermore, we propose an information system change process that exploits the described relations between task and information modeling.

Keywords

Federated Information System Architecture, Quality Management Processes, Repositories, Information Access Mechanisms

* This work was supported by the German Ministry of Research and Technology under grant 02QF3004/1, by the DFG Graduate College 'Computer Science and Technology' at RWTH Aachen and by the European Community under Information Systems Interoperability Project No ECAUS003 and Basic Research Working Group No.8319 (ModelAge)

1 TASK ORIENTED INFORMATION SYSTEMS

In order to improve business processes, the organisation of industrial work is in the midst of transformation. Functionally divided, vertically integrated and hierarchically organised firms give way to process networks of islands integrated by flows of material and information ((Brynjolfsson and Mendelson, 1993), (Davenport *et al.*, Fall 1992), (Hammer and Champy, 1993), (Warnecke, 1992)). The interplay of ideas from Organisational Theory and the technological development in information science established the ideal of the 'flexible corporation' that meets customers' needs, makes effective use of assets and is able to adapt itself to new ways of working or changing business demands ((Hammer and Champy, 1993), (Markus and Robey, 1988), (Scott-Morton, 1994)).

Organisational consequences of this idea mainly deal with matters of distribution: How should information be distributed along the business processes? What responsibilities should be given to the agents performing the process steps? How much distribution is needed at all? What are the main aspects of integration in this distributed environment? ((Hammer, 1992), (Pels and Wortmann, 1992), (Veryard and Macdonald, 1994)) Based on these questions several proposals for both federated management concepts ((Huber and McDaniel, 1986), (Warnecke, 1992)) and information systems ((Davenport *et al.*, Fall 1992), (Sheth and Larson, 1990), (Sol, 1992)) argued that responsibilities for tasks as well as information resources should be distributed along the defined processes. From the information systems viewpoint, computer networks and access mechanisms to distributed information sources stored in databases are the technologically available backbone for the implementation of federated information systems ((Abramovici and Bickelmann, 1993), (Sol, 1992), (Arens *et al.*, 1993), (Brodie and Ceri, 1992)). Nevertheless, there are a multitude of organisational, conceptual, and information management problems to be solved in designing and implementing **process oriented information storage, access, or manipulation** of data.

Hierarchically structured departments have to be reorganised and integrated with respect to the business processes that should be supported. In addition to the spatial distance between working groups along the process chain there always exists a **conceptual distance**. For historical, technical or educational reasons, people use different language to describe their work and the information they need to perform it. Ideas proposed for the conceptual integration of distributed information sources range from information descriptions using standards or company-wide data dictionaries ((Scheer, 1994), (Shaw, 1991)) to fully federated systems where all sources provide interfaces to outside users (Menzel *et al.*, 1992). In recent years, the repository approach is becoming popular for the conceptual integration of systems in the business process reengineering context, because it allows modeling the relations between information and the environment in which it is used at various levels of abstraction (see, e.g., (Carlson and Ram, 1992), (Chen *et al.*, 1992), (Jarke *et al.*, 1993), (Woo and Lochovsky, 1992)).

Another problem is the **task-oriented access to and availability of information**. Information is just a means to support effective and efficient action of the responsible teams and persons ((Huber and McDaniel, 1986), (Gerrits, 1993)), and the access to information should fulfill the needs of the task performing agent. But it also has to be restricted to information

necessary to perform the task in order to avoid information overload phenomena. Concepts for task adequate information management have been developed (e.g. (Daft and Lengel, 1986) and (Hammer, 1992)). Though they have influenced data modeling, they are rarely implemented with respect to business process oriented information distribution and access today.

A third major problem is the **management of change** once a information system is installed. Implemented technical solutions tend to freeze the state of the organisation because they support given work structures. That's why process changes often are costly and lengthy ((Sol, 1992), (Warnecke, 1992)). But in an environment of steady changes the ability of fast adaptation to new requirements is a major competitive advantage. With business processes in the center of interest, the consequences of changing processes or products should lead directly to the identification and implementation of change with respect to the information system. This is possible only if process and information models are intertwined and is manageable with acceptable costs only if all parts of the system are clearly separable not only technologically but also conceptually.

The contribution of this paper is an information system design that is a trade off between high conceptual and system integration which facilitates fast business process execution, and clear system separation allowing better management of change. We will start with a description of the repository-based federated information system architecture developed during the WibQuS project. This project deals with the integration of quality management methods along the product life cycle in manufacturing. After reviewing the impact of conceptual modeling on the method integration task, we will show how communication and information access tasks are managed in the given architecture. Based on the experiences in the project, information access mechanisms at various levels of automation are presented. Finally, we will describe, how change processes and database evolution are facilitated by the knowledge stored in the repository and the developed communication mechanisms.

2 THE WIBQUS PROJECT

The goal of the Germany-wide WibQuS project is the integration of quality management methods based on information flows. In order to achieve high quality goals, those methods are embedded in all stages of the product life cycle from design tasks supported by Quality Function Deployment (QFD) to methods and tools for service engineers ((Jarke *et al.*, 1993), (Peters and Jeusfeld, 1994)). Quality management processes are of special interest in task oriented information management because information flows occur in various types (data, knowledge, strategy) between differing method executing agents (automated systems, engineers, teams). Additionally, the transfer of product and process information not only follows the product life cycle, but is crucial also along multiple control loops from the manufacturing floors back to work preparation and product planning. The implementation of those control flows is essential in order to achieve and maintain high level product and process quality (Pfeifer, 1993).

Information system development in the WibQuS project is based on a repository approach (ISO/IEC 10027, 1990). It started with a conceptual modeling phase, where all engineering teams, located at seven research institutes in four cities, had to develop a model of their

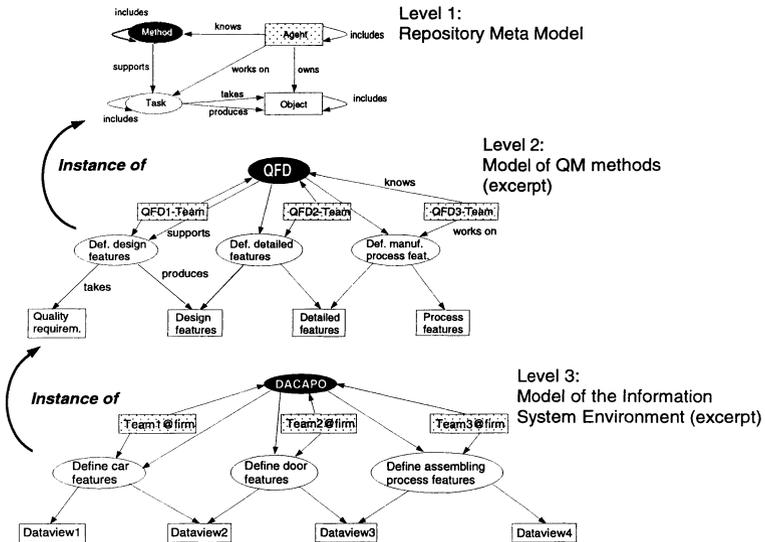


Figure 1 The levels of the WibQuS repository

method by instantiating the meta model depicted at Figure 1. This process, facilitated by the client-server architecture of the object base management system ConceptBase (Jarke *et al.*, 1995), was performed mainly for two reasons*:

1. To facilitate interoperability between the performing agents by establishing a common language (the conceptual model). In the model that forms this language, the methods have been related semantically by detecting information needs and sources that connect the various methods (represented by **objects** that connect **tasks** performed by different **methods** in Figure 1). These connections served as a starting point for information modeling on the data level.
2. To explicitly relate informations to the tasks that consume or produce them and to the agents that perform the tasks (see also (Mylopoulos *et al.*, 1993)). This way the needs, skills and responsibilities of agents in a specific decision process can be taken into consideration which is a first step to 'task and decision process oriented' information management.

The result of the modeling process was the integration of all models in one model of quality management methods containing 500 objects at level 2 of the repository. This model is a central element of the WibQuS information system architecture, because it describes conceptually the possible information flows with respect to the tasks to be solved (see Figure 1). Based on that model, the engineers in the next phase formulated the data models of their databases and the technical means for information exchange between the tasks at level 3 of the repository.

* A more detailed description of the distributed modeling process is given in (Peters and Szczurko, 1994).

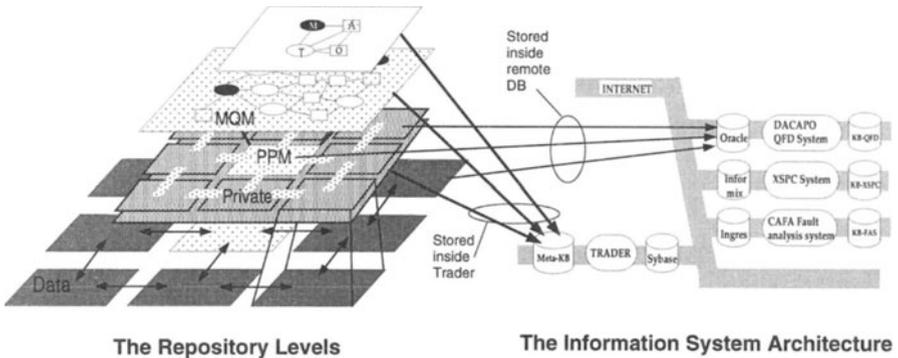


Figure 2 The WibQuS project

This was done by instantiation of the modeled **objects** at the method model level with relational schemas on the data level describing the contents of the **object**. The information needed for a specific **Task** form a view on these relational schemas (see Figure 5 for an example). **Agents** and **Tasks** are instantiated by an example organisational setting used in the WibQuS project and the **Methods** are instantiated by objects describing the tool configuration developed by the engineers for this specific method.

Technologically, standards like Internet as network communication protocol and SQL-databases as interfaces of the knowledge-based systems form a good framework for the federated architecture. This setting describes the typical industrial situation quite well. In order to keep the methods as autonomous as possible we needed a central communication manager that performs and controls information exchange. The *Quality Trader** stores and provides knowledge about the data and services of the whole information system (see Figure 2): It consists of an SQL-gateway that handles the technical connection to the federated databases, a number of communication processing mechanisms described in the following sections, and the repository stored in ConceptBase. In addition to the already described models at the higher levels, it contains all schema information of the SQL-databases that form level 3 of the repository with respect to the concept **Object**. This schema information consists of two parts :

1. The *Product and Process Model* (PPM) which is a globally available schema describing information that requires overall agreement because it is used in (almost) every method, e.g., product structure trees, or manufacturing process information.
2. *Private exchange schemas*, describing information of interest only for few agents like specific method results, e.g., matrix structures of QFD.

The distinction between those schema types prevents the company data schema from getting unmanageable, because changes in a special method might lead to changes in the relational schema of that method, but it does not influence other system schemas as long as the PPM is

* This approach is similar to the trader concept of Open Distributed Processing ((Raymond, 1995), (Bearman, 1995)) but restricted to database functionality.

not changed. Additionally, it emphasizes the autonomy of the participating groups: As long as information provided by a group is of interest only for one or two methods in the system, the group remains responsible for the schema, its evolution, or bilateral negotiations about information needed by other methods (c.f. sections 4,5). Only information of global interest should be represented in a globally identifiable and accessible way.

Nevertheless, all schemas, private or global, in WibQuS have to be represented in the Quality Trader, because it is responsible for the execution of all communication based on relational queries, especially if it refers to data distributed over several systems. The mechanisms underlying the communication in the given setting are described in the following section.

3 COMMUNICATION IN THE TRADER ARCHITECTURE

Queries, the basic communication technique of the Trader Architecture, are facilitated technically in two ways: First, all system configurations and their Internet addresses are represented at level 3 of the repository, instantiating the described **Methods**. So, in addition to the relational schemas, all routing informations are part of the Trader's knowledge base and the user need not care about where information is stored. Second, the SYBASE OmniSQL* server is able to connect to and communicate with all databases included in the system architecture if the routing informations are given. This simplifies the query formulation and data transfer between the four used database brands, because the user (or the knowledge-based system) must know only his own SQL-'dialect'.

Although these technical preconditions are helpful in implementing communication mechanisms, using SQL as a language is too restrictive to support efficient communication for several reasons:

1. It restricts system interfaces to relational databases.
2. The user has to learn the complex language to use the system. Even if he knows the language he still has to look up the schema information, find the information sources, and formulate the probably lengthy queries when he needs fast information access.
3. It restricts communication to the pure database access.

Because we wanted to embed other communication mechanisms like message broadcasting or assignment negotiation in the system, especially the last point caused the definition of a more abstract request language. It is based on relational concepts and predefined queries but is easily transferable to other environments. This communication exchange relation schema (see Figure 4). allows to take a relatively simple predictable view on information requests using predefined queries or by naming a **predicate**, filling in the **argument list** and putting the **order** into the systems order request list.

The whole process is exemplified by the search for features of a specific Product Structure Element (PSE) in Figure 3. The requesting agent puts a tuple with the predicate and its arguments into his order relation. The Quality Trader recognizes the requests, looks up the

* SYBASE OmniSQL is a trademark of Sybase Inc.

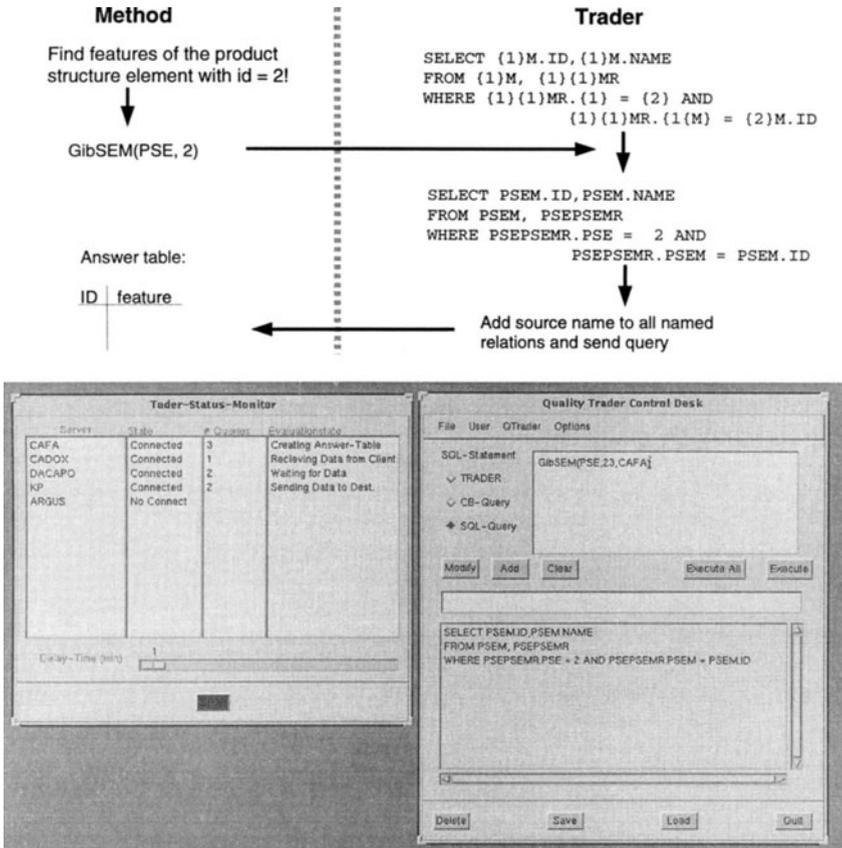


Figure 3 Requesting product element features using the order concept

query template related to the predicate, fills in the argument, and performs the query. He collects the answer tables and sends them back to the requesting system.

The Quality Trader windows depicted in Figure 3 are an application built for system engineering and control in the federated environment. The control desk allows manipulations on the repository using the ConceptBase Query Language and on all SQL-databases connected by using the facilities provided by OmniSQL. The status monitor gives an overview of the systems connected at the moment and the state of the queries currently under evaluation.

In detail, the process of order evaluation is the following (see also Figure 4): Placing the **predicate** and the **argument list** in the **order** relation and setting the value of the **comstate** relation to 1 initiates the order evaluation process by the Quality Trader based on triggers defined on the **comstate** relation:

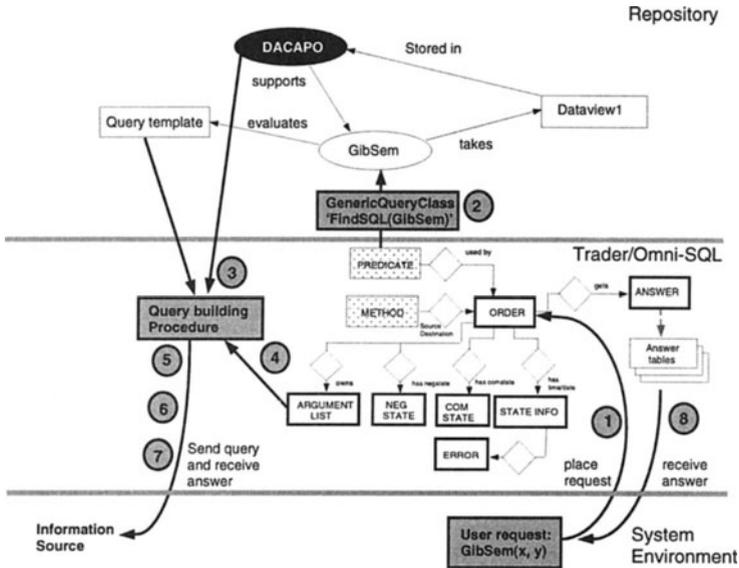


Figure 4 The order evaluation process.

1. **comstate** = 1: The Quality Trader checks the **comstate** relation of every system frequently and starts the query evaluation procedure.
2. The Quality Trader looks up the query represented by the **predicate** (Figure 4). This is done by evaluating the GenericQueryClass 'findSQL(**predicate**)' on the repository. Each predicate is represented as an instance of **Task** at repository level 3.
3. The relations which are involved into the query are related to the systems in which they are stored. The Quality Trader finds those information source(s) using the **stored in** attribute added to the structure of the repository.
4. The query variables in the predefined **query template** are instantiated with the **argument list** and the locations found in step 3.
5. SYBASE OmniSQL transfers the query into the database dialect(s) of the source(s). **Comstate** = 2
6. Execution of the (distributed) query.
7. The Quality Trader collects the answer(s) and transforms it into the dialect of the requesting system. **Comstate** = 3
8. The answer table is sent to the requesting system. **Comstate** = 4
9. The Quality Trader documents the querying process for statistical purposes.

Depending on the type of predicate referred to in the request, the **order** can be a query as in the example of Figure 3, a message to be broadcasted or a real assignment. This would lead to the initiation of a negotiation process between the involved methods supported by a mechanism based on the **negstate** relation that is currently under development.

Therefore, the implementation of the **order** concept solves the problems mentioned in the beginning of the section: It is at least more user-friendly than SQL, it allows other communication than pure data queries, and the language need not to be changed from the user's point of view if information sources other than SQL databases are added to the system. Because of the predefined queries the developed mechanism is restricted to a limited number of communication processes: *Highly standardised negotiation processes or information requests that can be detected and defined in advance*. According to the engineers using the system, there exist a lot of requests that are predictable or even automatable (embedded into the knowledge-based systems), but at least the same number of requests are *ad hoc* and needs different support.

4 GRAPHICAL INFORMATION ACCESS TO FEDERATED DATABASES

As a lot of the information needs are not predictable during quality management processes, the need for user-friendly ad hoc information access becomes obvious. But the classical problems of query languages (hard to learn, difficult to use, not adapted to the specific domain) are even more complicated in federated information systems. First, there are several query languages used in the system and second, there is no central database schema known by all system members but a number of loosely coupled schemas where structure and contents of the database are not easily understandable at once for someone looking for information. The advantages of graphical database browsing and access mechanisms for central databases have been pinpointed in various publications ((d'Atri, 1989), (Wong and Kuo, 1982), (Zhou, 1994)) and they are at least as striking for federated systems as we will show in this section.

The conceptual models of method interaction and product-/process structures were the starting point for a browsing and querying tool currently under development. Starting from the browsing facilities of ConceptBase and recent approaches in graphical query languages ((Consens *et al.*, 1992), (Zhou, 1994)), we are currently creating a *task and view-oriented information system browser* that allows direct access to information provided by the systems in the WibQuS environment.

As mentioned before, the model of quality management methods and all schema information are stored in the Quality Trader. This adds up to about thousand objects and searching the whole repository would be inefficient. Therefore, we started structuring it by first separating the method model from the database schema information and second dividing the schema information into PPM (general interest) and private schemas. This led to the browsing window layout shown in Figure 5. To adjust the search space to the tasks to be supported we have developed a view concept that restricts the information provided at the windows (Peters *et al.*, 1995a). The given instances of views were the result of various discussions with the quality management engineers:

1. The Method View:

This view puts the method and its tasks into the center of focus. All information flows (**objects**) taken by a specific **Task** and its subtasks are displayed at the MQM window. All schema information to a selected **Object** in the MQM window is displayed in the database schema window (These are the relations named 'dataview' in Figure 1 and 4).

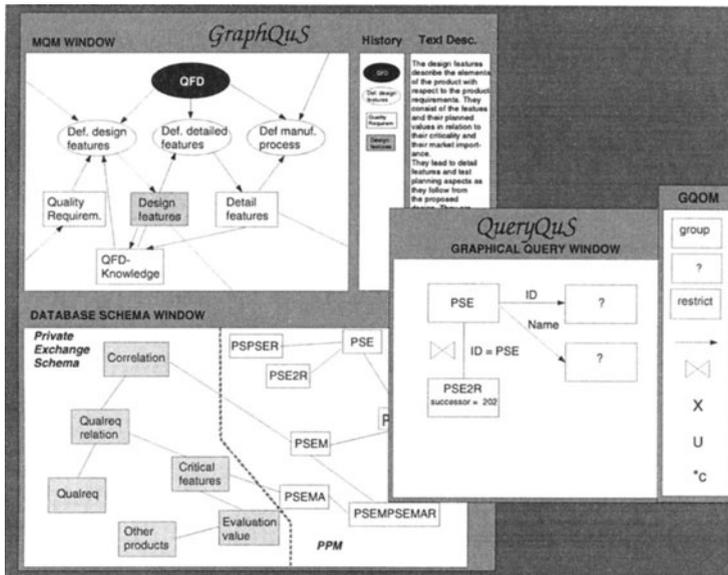


Figure 5 The information search and access tools

E.g., if an engineer working on a QFD uses the method view, all tasks related to QFD come up together with the connecting information flows (Figure 5). If he selects the 'Design features' object, the schema window displays relations representing the designed product structure (PSE, PSE2R) together with the product features and their values (other displayed PPM objects). Additionally, private relations are displayed describing relations between the features and the quality requirements to be fulfilled. Now the engineer can compose the query he wants to ask using the graphical query window.

2. The Information View:

While the first view groups around the method, the second focuses on the information related to a specific product. If information about a specific product structure is needed, selecting the PSE relation leads to the display of the method from which the PSE relation originates. This is the method that first fills the relation during the product life cycle or the one responsible for the administration of the relation (In WibQuS for PSE this would be QFD.). The schema window displays relations with distance one, meaning that the primary key of PSE is foreign key for the distant relation or the other way round. Further information could be provided by selection of any relation, causing that its connected relations will be displayed.

3. The Project View:

As engineers fill several roles in method performing teams* an overview about a project's specific information context inside the environment is often helpful, because it facilitates

* E.g., there is always a manufacturing plant engineer in a QFD team, who is also responsible for work preparation tasks and the manufacturing process.

the transfer of experiences from one task to another by displaying the specific interrelations among the project informations. The project view is the sum of those method views that show methods performed in the project. If, for instance, design and test data are needed to start the manufacturing process, the data search process is restricted to those methods already performed during the project and the data selection process is facilitated by restricting the available data to those related to the project.

4. The Service View:

As indicated by the name, service views provide services to people which are not in the group of **agents** for any method. If higher management needs reports about the performance of the methods, specific relations that store performance data (like tasks performed, information accesses,...) will be displayed at the schema window. If the contract between the enterprise and its customer include requirements traceability efforts or quality reports fulfilling ISO 900x, the way of the product through the product life cycle could be traced by following the information objects through the quality management tasks.

Although the tool is built mainly to satisfy information needs of users in the WibQuS environment according to the tasks they have to perform, the last view indicates that accelerated search for information in distributed databases is only one advantage of this view concept. Using this concept, it is possible not only to find information fast but also to present it in an task and user appropriate way. Every view could be embedded in a specific presentation tool that might generate quality reports from the SQL-databases or could display charts of the performance results. That way, the interpretation of data would be facilitated causing better and faster decisions (Daft and Lengel, 1986).

After finding the relations from which data should be retrieved, a graphical query can be formulated by shifting relations to the graphical query window and grouping them with operators from the graphical query operations menu (Figure 5). Two operations are performed together with the shift: The database where the relation is stored is attached to the relation (as described in the previous section) and after all needed relations have been shifted, the connections between the relations in the are computed and displayed in order to find possible join paths defined by foreign key dependencies.

The menu provides all operations that are necessary to ensure relational completeness plus a group operator and an operator for recursive queries which is useful especially in engineering environments because a lot of data are described in tree structures (products, machines, processes). The recursion is limited to simple one involving a maximum of two relations.

The formal query definition (Figure 6) follows an approach similar to (Angelaccio *et al.*, 1990). In a first step an SQL-independent representation of the queries is derived when the user formulates his graphical query. The representation consists of a triple (N, A, X) for every relation in the window where N denotes the relation name, A are the selected attributes, and X is a relational clause naming the operator and its arguments. Every unary operator causes a change in the related triplet (see projection and selection in figure 3) and every binary operator causes the definition of a new triplet that refers to its argument triplets (see join in figure 3). Because these triplets can be embedded recursively, relational completeness is ensured.

What are the products with the part that has ID = 202?

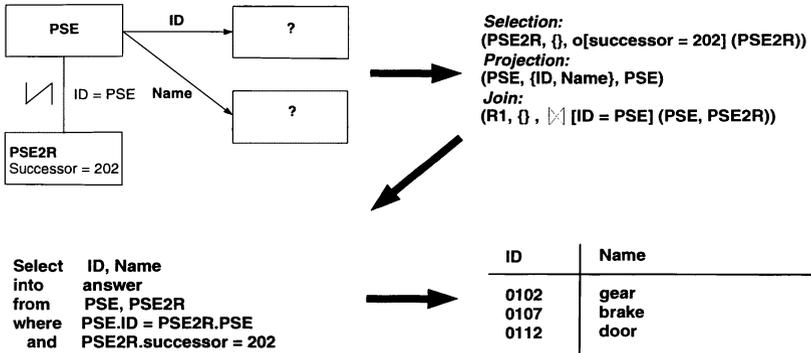


Figure 6 Definition and evaluation of a graphical query

In the next step, the generated triplets are translated into the specific SQL-dialect. This can be done by building a query tree (Elmasri and Navathe, 1989) where the whole ‘triplet tree’ is evaluated from the leaves upward by formulating a query for every triplet and using the results for the evaluation of the father query. This approach causes high network load because a lot of queries have to be sent. We used another approach by defining an interface for the translation of the whole query into OmniSQL, what might lead to higher costs for large join operations but keeps the network load relatively low. As recursive queries cannot be formulated in SQL, we implemented a ‘stored procedure’ that controls the recursive loop of the querying process performed by OmniSQL. Both the interface and the stored procedure can be transformed easily without changing the formal representation if another SQL-dialect is required.

As mentioned in section 3, the technical part of the querying process is performed by the Sybase OmniSQL gateway. The query answer appears in a separate window as table visualization. As we learned from the engineers, often a sequence of queries is performed, either because the information was not as complete as needed or additional information needs where triggered by the results. To support these step by step processes, we are currently working on query chaining mechanisms for a faster formulation of query sequences.

5 MANAGING CHANGE IN THE WIBQUS ENVIRONMENT

The models and information system structure implemented at the moment reflect the state of quality management processes and their support systems as perceived in the WibQuS environment. In this paper we will not reflect on technological changes of the latter, but on the agreement on the processes which are subject to continuous change: the composition of the expert groups may change, new methods are selected depending on management decisions, and the relevance of information is re-evaluated because of growing experience or changing market demands.

In a task oriented approach, changes of the federated information system reflect changes of the supported business processes. Therefore, based on the WibQuS experiences, we propose conceptual modeling of the changes as mediation between the agents perception of change and its representation in the database schemas (see also (Alonso and Barbara, 1989), (Jeusfeld and Szczurko,)),). The renewed process model facilitates implementing changes of information access, information needs or content of databases using the Quality Trader functionality.

As the described mechanisms support both communication and information access in a federated information system, they can assist change processes from the introduction of a new method to the schema change of an information object. The first step is the announcement of the possibility of, or the need for, a change in the whole system. In addition to frequent system engineering team meetings this can be done by sending messages using the Trader facilities. To support this change initialisation, we defined two special order types called request announcement and availability announcement. These messages will be sent to every method in the system and their arguments describe the subject of change. A comment gives further specifications about the needed/provided information.

The implementation of the two steps of change, conceptual modeling and database adjustment are currently possible only using the quality trader control desk described in section 3. Together with the browsing facilities of ConceptBase it forms a system engineering environment for the repository but changes cannot be propagated to the remote databases and working with this interface turned out quite complicated for the engineering groups. The development of the browsing and access tool described in the previous section provides a more easy-to-use front-end to the repository and even more, together with the OmniSQL-Gateway, to the whole system. We are currently working on an enhancement of the graphical query window to allow its usage as a system design window that supports tell/untell operations on the ConceptBase repository as well as create/update/delete operations on the relational databases.

This allows an engineering process according to the proposal above (Peters *et al.*, 1995b). In a first step the repository is changed using the access mechanisms to the repository. This step consists of changes of the model of quality management methods that induce changes of the relations of the private exchange schema or the PPM. These changes have to be translated into SQL commands using the graphical design mechanisms provided. Finally, order templates can be formulated using the graphical query language.

The information system change process is facilitated by the described tools mainly with respect to **system representation**, **operation formulation** and **enactment** in all steps. The processing of process changes into relational data model changes into database changes has to be performed by the system engineer together with the user groups. How adjustment in the latter steps depends on changes of the method model is subject of current research (Jeusfeld and Szczurko,).

6 CONCLUSIONS

The method integration approach presented here follows the tradition of heterogeneous database integration (e.g., CORBA (Murphy and Grimson, 1994), MRDSM (Litwin *et al.*, 1990), or CIM/Z (Norrie *et al.*, 1994)) and Distributed Object Management ((Manola *et al.*,

1992), (Saake *et al.*, 1993)). Most of the approaches presented in the field concentrate on information modelling and leave the supported tasks implicit. Influenced by the ideas of business process re-engineering, the explicit connection of the concepts of **task**, the **needed information**, and the **performing agent** has been considered recently as a necessary feature of task adequate information and database management (e.g., (Mylopoulos *et al.*, 1993)).

The evolution process during the WibQuS project showed that business process change management goes hand in hand with engineering information systems on the technical as well as on the conceptual level. Project results and various interviews with the participating engineers validated the feasibility of the implemented information system approach that is based on the idea of integrated task and information modeling. The modeling of quality management methods and their interactions proved to be a right-sized starting point for process integration, information system engineering in the given environment, and task oriented information and change management along the described quality management processes.

The repository based integration of conceptual models of the processes and the information flowing between them forms the cornerstone of the Quality Trader architecture. It is the starting point for the technological solution related to the presented management approach and fits typical business settings in both intra- and interorganisation cooperation, because it is based mainly on standard technology like SQL databases or Internet. Built on those standards, conceptual models and Trader architecture constitute the technical and conceptual environment for several tools currently under development, supporting information management at various levels of automation and enabling fast adaptation to changing information needs.

The advantage of the proposed approach — relying on standard technology — is its disadvantage at the same time: Support is restricted by the technological means integrated into the architecture. Though we are implementing several communication and cooperation mechanisms on top of this technology, one thing we've learned during WibQuS is that communication should be performed using various channels to optimise task performance and workflow. This includes a certain level of redundancy which is often needed as well as task oriented suitability of media (see also, e.g. (Daft and Lengel, 1986)). Certainly, database technology will become more powerful with respect to multimedia, but the integration of multiple communication channels is a main challenge for task oriented information system management.

REFERENCES

- M. Abramovici and S. Bickelmann. Engineering Data Management Systems – Requirements, State of the Art, and Potentials (in German. *CIM*, (5):20–28, 1993.
- R. Alonso and D. Barbara. Negotiating Data Access in Federated Database Systems. In *5th Intl. Conf. on Data Engineering*, pages 56–65, 1989.
- M. Angelaccio, T. Catarci, and G. Santucci. QBD*: A Graphical Query Language with Recursion. *IEEE Transactions on Software Engineering*, 16(10), 1990.

- Y. Arens, C.Y. Chee, C.-N. Hsu, and C.A. Knoblock. Retrieving and Integrating Data from Multiple Information Sources. *Int. Jour. of Intelligent and Cooperative Information Systems*, 2(2):127–158, 1993.
- M. Bearman. Trading in Open Distributed Environments (Tutorial). In *International Conference on Open Distributed Processing*, 1995.
- M.L. Brodie and S. Ceri. On Intelligent and Cooperative Information Systems: A workshop summary. *Int. Jour. on Intelligent and Cooperative Information Systems*, 1(2):249–290, 1992.
- E. Brynjolfsson and H. Mendelson. Information Systems and the Organization of Modern Enterprise. *Journal of Organizational Computing*, 3(3):245 – 256, 1993.
- D.A. Carlson and S. Ram. A Knowledge Representation for Modeling Organizational Productivity. *Journal of Organizational Computing*, 2(2):155 – 180, 1992.
- M. Chen, Y.I. Liou, and E.S. Weber. Developing Intelligent Organizations: A Context-Based Approach to Individual and Organizational Effectiveness. *Journal of Organizational Computing*, 2(2):181 – 202, 1992.
- M.P. Consens, I.F. Cruz, and A.O. Mendelzom. Visualizing Queries and Querying Visualizations. *ACM SIGMOD*, 16(3), 1992.
- R.L. Daft and R.H. Lengel. Organizational Information Requirements, Media Richness and Structural Design. *Management Science*, 32(5):554 – 571, 1986.
- A. d'Atri. From Browsing to Querying. *IEEE Database Engineering*, 8:110 –117, 1989.
- T.H. Davenport, R.G. Eccles, and L. Prusak. Information Politics. *Sloan Management Review*, pages 53 – 65, Fall 1992.
- R. Elmasri and S.B. Navathe. *Fundamentals of Database Systems*. B. Cummings Publishing Company, Inc., 1989.
- H. Gerrits. Business Process Redesign and Information Systems Design: A happy couple? In N. Prakash, C. Rolland, and B. Pernici, editors, *IFIP Transactions: Information Systems Development Process*. Elsevier Science Publishers B.V., 1993.
- M. Hammer and J. Champy. *Reengineering the Company*. Harper Business, New York, 1993.
- D. K. Hammer. Lean Information Management: The Integrating Power of Information. In H. J. Pels and J. C. Wortmann, editors, *IFIP Transactions: Integration in Production Management Systems*, pages 147–163. Elsevier Science Publishers B.V., 1992.
- G.P. Huber and R.R. McDaniel. The Decision-Making Paradigm of Organizational Design. *Management Science*, 32(5):572 – 589, 1986.
- ISO/IEC 10027. Information Technology – Information Resource Dictionary System (IRDS) - Standard. Technical report, 1990.
- M. Jarke, M. Jeusfeld, and P. Szczurko. Three Aspects of Intelligent Cooperation in the Quality Life Cycle. *Int. Jour. of Intelligent and Cooperative Information Systems*, 2(4):355–374, 1993.
- M. Jarke, R. Gallersdoerfer, M.A. Jeusfeld, M. Staudt, and S. Eherer. ConceptBase – A Deductive Object Base for Meta Data Management. *Journal of Intelligent Information Systems*, 4(2), 1995.

- M.A. Jeusfeld and P. Szczurko. Evolution of Federated Database Schemas by Trading Information Resources (*submitted for publication*).
- W. Litwin, L. Mark, and N. Roussopoulos. Interoperability of Multiple Autonomous Databases. *ACM Computing Surveys*, 22(3):267–293, 1990.
- F. Manola, S. Heiler, D. Georgakopoulos, M. Hornick, and M. Brodie. Distributed Object Management. *Int. Jour. of Intelligent and Cooperative Information Systems*, 1(1):5–42, 1992.
- M.L. Markus and D. Robey. Information Technology and Organizational Change: Causal Structure in Theory and Research. *Management Science*, 34(5):583 – 598, 1988.
- C. Menzel, R.J. Mayer, and L. K. Sanders. Representation, Information Flow, and Model Integration. In C. Petrie, editor, *Enterprise Integration Modeling*, pages 131 – 141, 1992.
- J.P. Murphy and J.B. Grimson. A CORBA based Architecture for Multidatabase Systems. In *Int. Conf. on Data and Knowledge Systems for Manufacturing and Engineering*, pages 630–635, Hong Kong, 1994.
- J. Mylopoulos, T. Rose, and C. Woo. Task-Oriented Development of Intelligent Information Systems. In M. Papazolou, editor, *Int. Conf. on Intelligent and Cooperative Information Systems*, pages 206 – 219, Rotterdam, The Netherlands, 1993.
- M.C. Norrie, W. Schaad, H.-J. Schek, and M. Wunderli. CIM through Database Coordination. In *Int. Conf. on Data and Knowledge Systems for Manufacturing and Engineering*, pages 258 – 263, Hong Kong, 1994.
- H. J. Pels and J. C. Wortmann. Integration in Production Management Systems: An Integrating Perspektive. In H. J. Pels and J. C. Wortmann, editors, *Integration in Production Management Systems*, pages 1–13. Elsevier Science Publishers B.V., 1992.
- P. Peters and M.A. Jeusfeld. Structuring Information Flow in Quality Management. In *Int. Conf. on Data and Knowledge Systems for Manufacturing and Engineering*, pages 258 – 263, Hong Kong, 1994.
- P. Peters and P. Szczurko. Integrating Models of Quality Management by an Object-oriented Repository. In *2nd Biennial European Joint Conf. on Engineering Systems Design and Analysis*, London, England, 1994.
- P. Peters, U. Loeb, and A. Rodriguez Pardo. A Task-Oriented Graphical Interface to Federated Databases. In *Proc. of 3rd Int. Conf. of the Int. Soc. for Decision Support Systems*, Hong Kong, 1995.
- P. Peters, P. Szczurko, M. Jarke, and M. Jeusfeld. Business Process Oriented Information Management: Conceptual Models at Work. In *Proc. of Int. Conf. on Organizational Computing Systems*, San Jose, Calif., 1995.
- T. Pfeifer. *Quality Management (in German)*. Carl Hanser Verlag, Muenchen, 1993.
- K. Raymond. Reference Model of Open Distributed Processing (RM-ODP): Introduction (Tutorial). In *International Conference on Open Distributed Processing*, 1995.
- G. Saake, R. Jungclaus, and T. Hartmann. Application Modelling in Heterogeneous Environments Using an Object Specification Language. In M. Papazolou, editor, *Int. Conf. on Intelligent and Cooperative Information Systems*, pages 309–318, Rotterdam, The Netherlands, 1993.

- A.-W. Scheer. *Wirtschaftsinformatik (Reference Models for Industrial Business Processes, in German)*. Spriner Verlag, Berlin, Heidelberg, ..., 1994.
- M.S. Scott-Morton. The 1990s research program: Implications for Management and the Emerging Organization. *Decision Support Systems*, 12(2):251–256, 1994.
- N. Shaw. STEP Part 1 – overview and fundamental principles. Technical report, 1991.
- A.P. Sheth and J.A. Larson. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, 22(3):183–237, 1990.
- E.-J. Sol. CIM – Communication and Information in Manufacturing. In H. J. Pels and J. C. Wortmann, editors, *IFIP Transactions: Integration in Production Management Systems*, pages 13–21. Elsevier Science Publishers B.V., 1992.
- R.A. Veryard and I.G. Macdonald. EMM/ODP: A Methodology for Federated and Distributed Systems. In A.A. Verriijn-Stuart and T.W. Olle, editors, *IFIP Transactions: Methods and Tools for the Information System Life Cycle*. Elsevier Science Publishers B.V., 1994.
- H.-J. Warnecke. *The Fractal Factory (in German)*. Springer-Verlag, 1992.
- H.K.T. Wong and I. Kuo. GUIDE: Graphical User Interface for Database Exploration. In *Proc. of the 8th Very Large Databases Conference*, 1982.
- C.C. Woo and F.H. Lochovsky. Knowledge Communication in Intelligent Information Systems. *Int. Journal of Intelligent and Cooperative Information Systems*, 1(1):203–228, 1992.
- J. Zhou. GUI to Databases – Survey and Practice. *GI Datenbank Rundbrief*, 13:49–51, 1994.

7 BIOGRAPHIES

Peter Peters studied computer science and medical informatics at the University of Dortmund, Germany. He is currently a doctoral student at the ‘Graduate College for Computer Science and Engineering’ and a member of the Information Systems group at the Technical University of Aachen. In his research he investigates the modeling, enactment, and analysis of information flow and communication in distributed organisations. During this work he has been participating in the interdisciplinary project WibQuS funded by the german ministry of research.

Peter Szczurko studied computer science and medical informatics at the Technical University of Aachen, Germany, where he is currently employed as a doctoral student and research assistant in the Information Systems group. In his research, he investigates the use of knowledge-based technology in quality management for distributed organisations, and has been participating in the interdisciplinary project WibQuS funded by the german ministry of research, as well as in projects concerning workflow management and quality control in medicine and chemical engineering. Since 1994, he is co-chair of the regional group Aachen within the German computer society, GI.

Matthias Jarke is professor of Information Systems and chairman of the computer science department at the Technical University of Aachen, Germany. After obtaining a doctorate from the University of Hamburg, Germany, in 1980, he held faculty positions at New York

University and the University of Passau prior to joining Aachen. His research interests lie in the development and usage of meta information systems for design applications. He has been coordinator of two European ESPRIT projects in this field, DAIDA (knowledge-based information system environments) and NATURE (requirements engineering environments), and was principal investigator in collaborative projects concerning IS applications in mechanical engineering (WibQuS) and medicine. He is editor-in-chief of the journal 'Information Systems'

Manfred Jeusfeld is currently a visiting professor at the Hong Kong University of Science and Technology. After obtaining his doctorate from the University of Aachen, he was employed as assistant professor at the Information Systems Group at the University of Aachen. During the ESPRIT projects DAIDA and COMPULOG (Computational Logic) he was a main developer of the ConceptBase system. He also was a main researcher in collaborative projects concerning IS applications in mechanical engineering (WibQuS) and medicine.