A Task Oriented Graphical Interface to Federated Databases[1]

P. Peters, U. Löb, A. Rodriguez-Pardo
Informatik V, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Germany
email: {peters,loeb,almudena}@informatik.rwth-aachen.de
Phone/Fax: +49 241 8021521/ +49 241 8888321

Abstract:
Because of the trend to business reorganization the concepts of organizational and distributed decision support have recently received a lot of attention. Computer networks and database technology are the technological backbone of Distributed Decision Support Systems (DDSS) that integrate decision processes along the product life cycle, but still the task oriented design and use of information sources are major problems in DDSS design. In this paper we will describe a federated architecture that integrates knowledge-based DSS for quality management tasks from product design to service. The baselines of the presented approach are first the conceptual integration of these systems by an object oriented repository that stores a model of the tasks and data models of the information to be communicated between those tasks and second the technological integration of the systems by Internet and SQL-databases.
Based on this approach we will show how embedding the models into the architecture by installing a central communication unit (the Quality Trader) will not only facilitate the automated communication between the systems but also the user oriented information search and access in the given environment. The typical problems of information access (even more in heterogenous, distributed environments!) are complex access mechanisms and information overload because of improper representation or information chunks that do not fit the tasks they are needed for. In order to accelerate information access and to avoid information overload, the main ideas of the task oriented information access to information provided in the DDSS are as follows:

1. *Integrate business process models and data models conceptually by adjusting the second to the first and coupling both models inside of a repository.*
2. *Create views on the given models that refer to the tasks and information needs of agents in the given environment.*
3. *Facilitate information access based on the given views by abstraction from the underlying technical restrictions (language, tools) and the use of graphical interfaces.*

The tool design features resulting from these ideas and their current implementation in the graphical information browsing and query formulation tool are described in the last section of the paper.

[1] This work was supported by the German Ministry of Research and Technology under grant 02QF3004/1, by the German Research Society's Graduate College 'Computer Science and Technology' at RWTH Aachen and by the European Community under Information Systems Interoperability Project No ECAUS003.

Section 1 Supporting Decision Making in Business Process Oriented Environments

In order to accelerate and improve decision making in business processes, the organization of industrial work is in the midst of transformation. Functionally divided, vertically integrated and hierarchically organized firms give way to process networks of islands integrated by flows of material and information ([2], [9], [30]). The interplay of ideas from Organizational Theory on one hand and the technological development in information science on the other established the ideal of the 'flexible corporation' that meets customers' needs, makes effective use of assets and is able to adapt to new ways of working or changing business demands ([14], [26], [27]).

This trend of enterprise reorganization led to DSS research effort in Distributed Decision Support Systems (DDSS) and Organizational Decision Support Systems ([10], [13], [18], [31]). Computer networks and access to distributed information are the technological backbone of these systems ([1], [21], [28]) and distributed databases become an accepted means to integrate DSS and to facilitate information storage and access. Nevertheless, the design and use of data sources with respect to the tasks referring to those sources are still major problems in the process of DDSS design. Several authors stated that there should exist mechanisms for extracting data from the (distributed) corporate database along with a suitable user-system dialog component ([18],[19],[24]). In designing those mechanisms one has to face several problems:

1. *How to integrate distributed information sources conceptually?*
   The ideas proposed in the literature range from an integrated information descriptions using standards or company-wide data dictionaries to fully federated systems where all sources provide interfaces to outside users [17]. In recent years, the repository approach for the conceptual integration of distributed applications becomes more popular, because it allows to model the relations between information and the environment in which it is used at various levels of abstraction (see, e.g., [3], [4], [15], [32]).
2. *How to get the right information at the right time to the right place?*
   Information is just a means to support effective and efficient decision making along the process chain [11]. The automation and facilitation of data production and the higher availability of information often lead to information overload causing just the opposite. Concepts of lean and adequate information management have been developed in recent years (see, e.g.,[6], [8]). Though they have influenced data modeling, with respect to decision oriented information access these concepts are rarely implemented today.

In the following we will show that conceptual models of business processes and information needs are means to support information access in distributed environments in various ways. Beyond being used for the specification of data and systems, we describe how they help solving problems related to the above questions if they are embedded into the developed DDSS architecture. We will start with a description of the repository-based

federated information system architecture developed during the WibQuS-Project[2]. This project which with the integration of DSS for quality management methods along the product life cycle in manufacturing. After reviewing the impact of conceptual modeling on the method integration task, we will describe a task and view oriented information access mechanism for the DDSS architecture developed during the WibQuS-Project.

Section 2 The Trader Architecture

The goal of the Germany-wide WibQuS-project is to develop and integrate knowledge-based DSS for quality management methods at all stages of the product life cycle from design tasks supported by Quality Function Deployment (QFD) to support methods and tools for service engineers ([15], [22]). Quality management processes are of special interest for DDSS because information flows and method interaction occur regarding various information types (data, knowledge, strategy) between differing method executing agents (automated systems, engineers, teams) . The transfer of product and process information not only follows the product life cycle (c.f. fig. 1), but it's crucial also along multiple control cycles from the manufacturing floors back to work preparation and product planning. The usage of this kind of organizational knowledge is a necessary precondition in any effort for better product and process quality and for fast adaptation to changing demands, laws, and customer's wishes [25], but the transfer of experiences with respect to decision support has been analyzed and implemented only to a limited extend (see, e.g., [10]).

The information system development in the WibQuS project is based on a repository approach. It was 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 method by instantiating the meta model depicted at figure 1 (the topmost level of the repository). This time consuming process, facilitated by the client-server architecture of the conceptual modeling tool ConceptBase [12] and the Internet networking facilities (ftp, e-mail), was performed mainly for two reasons, related to the questions in the introduction[3]:

1. To first facilitate interoperation between the locally as well as conceptually isolated tasks and methods by a common language (the conceptual model) and then relate them semantically by detecting information needs and sources that connect the methods and serve as a starting point for information modeling on a more detailed level.
2. To explicitly relate information to the task where it is needed and to the agent that performs the task (see also [20]). This way the needs, skills and responsibilities of agents in a specific decision process can be taken into consideration explicitly which is a first step to 'task and decision process oriented' information management.

---

[2] WibQuS is the German abbreviation for 'Knowledge-Based Systems in Quality Management'.

[3] A more detailed description of the distributed modeling process, which is quiet interesting with respect to system integration aspects is given in [22] and [23].

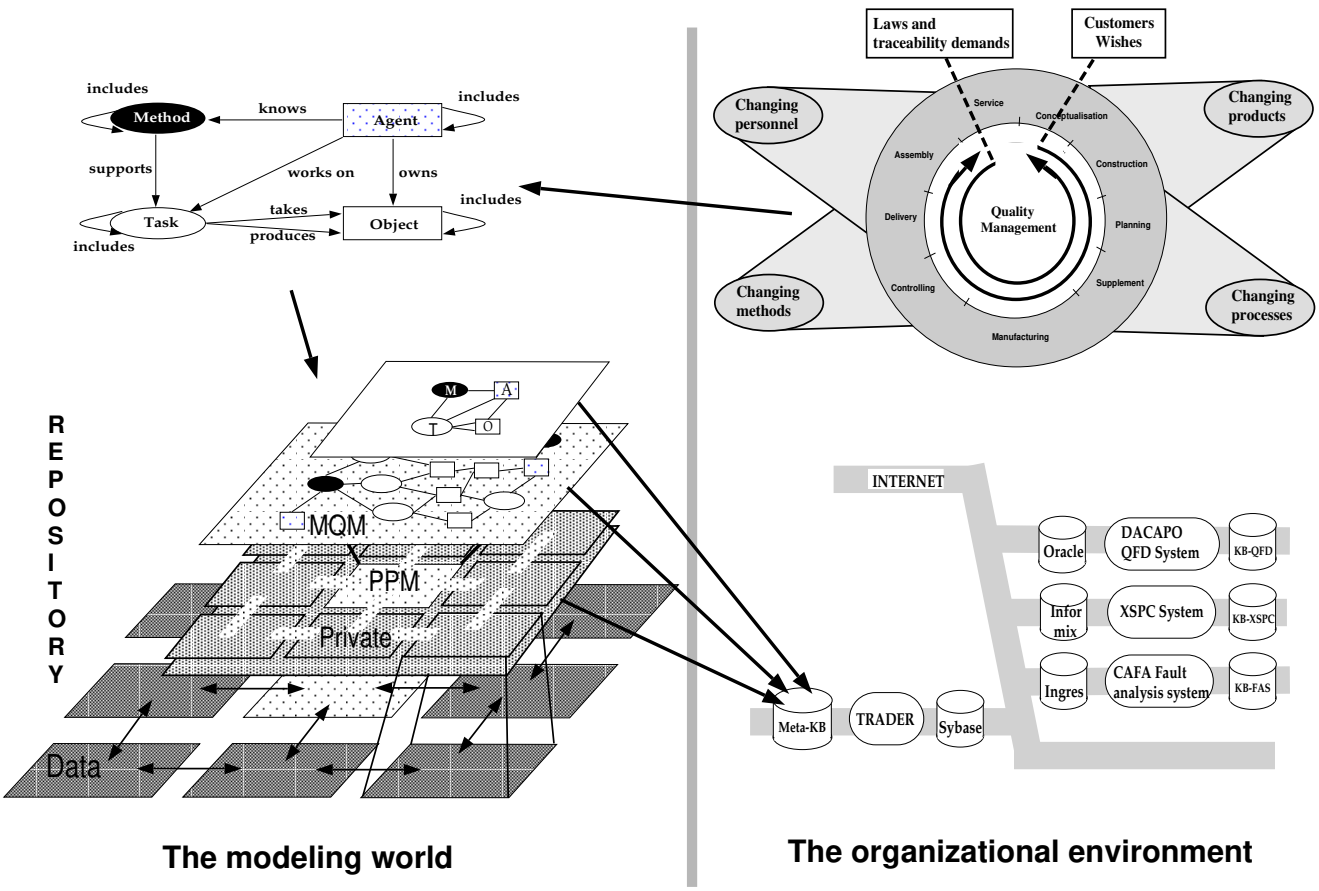

Figure 1 The WibQuS project

The modeling process ended in an 500 object model of quality management methods (MQM) at level 2 of the repository shown in figure 1. This model is a central element of the WibQuS information system architecture. Technologically, we decided to rely on standards like Internet as network communication protocol and SQL-databases as system interfaces, because this setting describes the typical industrial situation quite well. Because we wanted to leave the methods and their DSS as autonomous as possible and also keep there information management effort small, we decided to install a communication organization system, the so-called *Quality Trader* that, alike the trader concept used in open distributed processing in wide area communication networks, stores and provides knowledge about the whole information system: It contains the MQM (repository level 2), and all schema information of the SQL-databases that together form level 3 of the repository with respect to the concept `Object`, consisting of two parts :

1. The *Product and Process Model* (PPM) which is a global schema where information is stored 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 or experiences gained from

previous products or processes as well as statistics describing the method performance for controlling aspects.

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 the SQL-interfaces, especially if they refer to data distributed over several systems. This communication handling is facilitated in two ways: First all DSS configurations and Internet addresses are represented at level 3 of the repository, instantiating the described `Methods`. So all routing informations are part of the Trader's knowledge base. Second, the SYBASE OmniSQL[4] server is able to connect to and communicate with all databases included in the system architecture, what simplifies the technical query adaptation and data transfer between the four used database brands. This architecture forms a basis for a graphical information access mechanism that allows ad hoc task oriented search for information.

Section 3 Searching for Information in the WibQuS Environment

Based on the WibQuS architecture we are currently developing several communication and information access mechanisms [24]. We started with the development of predefined query templates that can be hardcoded into the DSSs' or could be filled with arguments and sent by the performing agents. The evaluation of these instantiated queries is performed by the Quality Trader [29]. But the idea of predefined, variable queries proved its relevance in the WibQuS-Environment only for a limited set of information requests. The quality management engineers were not able to formulate their information needs as queries in advance unless the information was highly standardized, e.g., product structure trees or attributes of process steps. Furthermore, they agreed that information needs especially in the design process often arise unpredictable and search mechanisms to explore the distributed environment were considered very helpful.

The conceptual models of method interaction and product-/process structures were the starting point for a browsing and querying tool currently under development (c.f. figure 2). Starting from the browsing facilities of ConceptBase and recent approaches in graphical query languages ([5], [33]), we are currently creating a *task and view-oriented information system browser* based on a client-server architecture that allows direct access to every information from every DSS in the WibQuS environment.

View and Task Oriented Access to Distributed Information

As mentioned before, the MQM and all schema information is stored in the Quality Trader using the Object Base Manager ConceptBase. This adds up to about thousand objects and searching the whole object base would be ineffective because of information overload. Therefore, we started structuring the search space by firstly separating the MQM from the database schema information and secondly dividing the schema information into PPM (general interest) and private schemas what led to the browsing window

[4] SYBASE OmniSQL is a trademark of Sybase Inc.

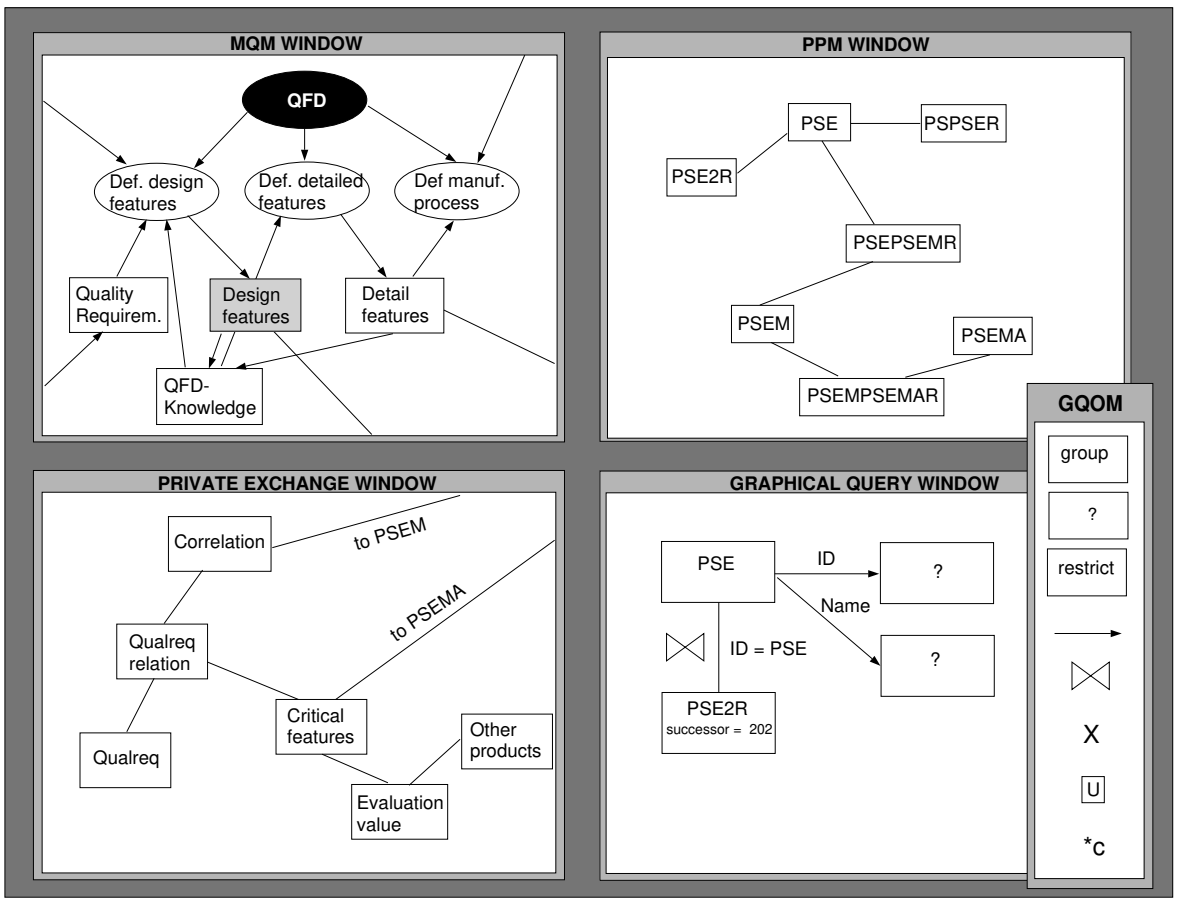

Figure 2 The information search and access tool

layout shown in figure 2. To adjust the search space to the information access needs we have developed a view concept that restricts the information presented at the interface with respect to the following views[5]:

1. **The Method View:**
   This view puts the method (and with that all the tasks to be solved by that method and the involved DSS) 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 (private or PPM) that is an instance of a selected `object` in the MQM window is displayed in the schema windows. E.g., if an engineer working on a QFD selects the task view, all tasks related to QFD come up together with the connecting information flows (figure 2). If he clicks at the 'Design features' object, the PPM schema window displays relations representing the designed product structure (PSE, PSE2R) together with the product features and their values (other displayed PPM objects). The private schema window displays objects 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 as we will described in the next section.

[5] These views were the result of various discussions with the quality management engineers.

2. **The Information View:**
   While the first view groups around the method, the second focuses on the information related to a specific schema element of the PPM. If, for instance, 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 (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 windows display relations with connecting 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 displayed by selection of any relation, causing that its connected relations will be displayed.
3. **The Personal View:**
   As engineers fill several roles in method performing teams (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.) an overview about this person's specific information context inside the environment is often helpful, because it facilitates the transfer of experiences from one task to another by displaying the specific interrelations among the tasks to be performed. The personal view is the sum of those task views that content tasks were the engineer is involved as `agent`. If, for instance, manufacturing data from a previous product are needed to justify a design decision during QFD and this information is not explicitly defined in the MQM (otherwise it would be included into the QFD task view), the manufacturing engineer can access the needed information fast by using his personal view.
4. **The Service View:**
   As indicated by the name, this views provide services to people which are not in the group of `agents` for any method in the DDSS. If the higher management needs reports about the performance of the embedded methods, specific relations storing performance data (like tasks performed, information accesses,...) will be displayed at the schema windows. If the contract between the enterprise and its customer include requirements traceability efforts or quality reports fulfilling ISO 9000, 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 DSS and their 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 the information fast but also to *present it in an task and user appropriate way*. Every view could be embedded into a specific DSS or presentation tool that might generate quality reports from the SQL-databases or could display charts of the performance results. In this way, the interpretation of data would be facilitated causing better and faster decisions [6].

The Graphical Query Language

After finding the schemas from which data should be retrieved a graphical query can be formulated by selecting the relations, shifting them to the graphical query window (GQW) and grouping them with operators from the GQOM (graphical query operations menu, fig. 2). With the shift, two operations must be performed: First, the database where the relation is stored is (internally) attatched to the relation[6], so the user don't have to care about the distribution of data over the environment. Second, after all needed relations have been shifted, the connections between the relations in the GQW are computed and displayed if necessary in order to find all possible join paths defined by foreign key dependencies.

The GQOM provides all operations that are necessary to fulfill relational completeness plus a group operator and an operator for recursive queries which we found necessary for 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 follows an approach similar to [16]. It is described by the example in figure 3. In a first step an SQL-independent representation of the queries will be derived as the user formulates his graphical query. The representation consists of a triple *(N, A, X)* for every triplet in the GQW where *N* denotes the relation name, *A* is a number of selected attributes, and *X* is a relational clause that describes the operator and its arguments. Now 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.

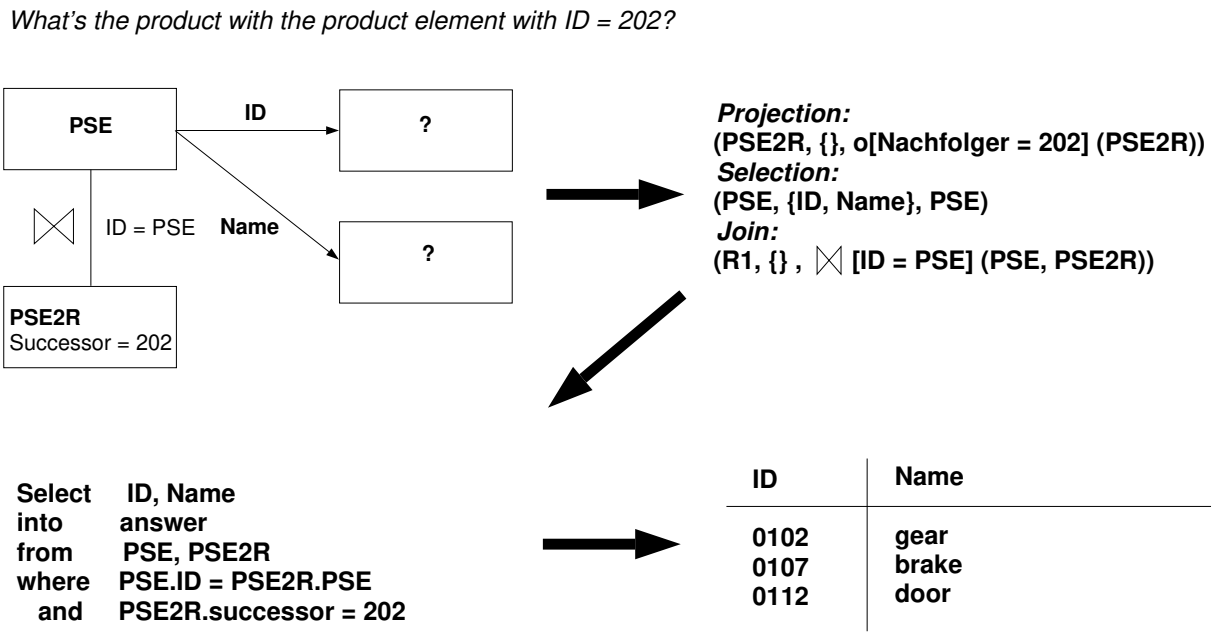

Figure 3 Definition and evaluation of a graphical query

[6] This is necessary because the Sybase Omni-SQL has to 'find' the databases in the distributed environment. The database names are omitted in figure 3 for clarity reasons.

In a next step, the generated triplets are translated into the specific SQL-dialect. This can be done by building a query tree [7] where the whole 'triplet tree' defined in the previous step 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. Therefore we defined an interface for the direct translation into Omni-SQL, 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 Omni-SQL. 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 2, the technical part of the querying process is performed by the Sybase Omni-SQL gateway. The answer to the query appears in an separate window in a table visualization. As we learned from the engineers, often a sequence of queries is performed, either because the information was not as complete as they though or additional information needs where triggered by the results. To support these step by step processes, we are currently working on the reuse of queries or query results for a faster formulation of query sequences, by temporarily embedding the formal queries into the repository representation of the relational schemas.

Section 4 Conclusion

This paper has emphasized the need for a conceptual integration of distributed decision support systems not only for purposes of systems design but also in order to facilitate information access in a distributed environment. We showed that conceptual models of business processes coupled with the data models of the information to be transferred between the supported tasks are the starting point for task oriented information access inside the trader architecture developed for the WibQuS project.

On top of the implementation of those models in an object oriented repository we have developed a graphical information browsing and query formulation tool that allows view and task oriented search for information provided by the systems included into the trader architecture. The access to data is possible in a user friendly way without the necessity to learn all SQL dialects supported in the heterogenous environment and without even knowing where the needed data are stored.

In the current stage of research we are using the prototype of the developed tool only for accessing data. But in a highly heterogenous environment of mainly autonomous DSS the management of change is another major task. This includes technological changes as well as the establishing of new information flows between the DSS or the embedding of new DSS. As Omni-SQL provides the technological means to solve this tasks and the repository provides the conceptual means, we are currently thinking of a functional extension of the browsing tool that enables the engineering of the conceptual models and the relational schemas of the remote databases.

Bibliography

[1] M. Abramovici and S. Bickelmann. Engineering Data Management Systems – Requirements, State of the Art, and Potentials (in German. *CIM*, (5):20–28, 1993.

[2] E. Brynjolfsson and H. Mendelson. Information Systems and the Organization of Modern Enterprise. *Journal of Organizational Computing*, 3(3):245 – 256, 1993.

[3] D.A. Carlson and S. Ram. A Knowledge Representation for Modeling Organizational Productivity. *Journal of Organizational Computing*, 2(2):155 – 180, 1992.

[4] 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.

[5] M.P. Consens, I.F. Cruz, and A.O. Mendelzom. Visualizing Queries and Querying Visualizations. *ACM SIGMOD*, 16(3), 1992.

[6] R.L. Daft and R.H. Lengel. Organizational Information Requirements, Media Richness and Structural Design. *Management Science*, 32(5):554 – 571, 1986.

[7] R. Elmasri and S.B. Navathe. *Fundamentals of Database Systems*. B. Cummings Publishing Company, Inc., 1989.

[8] D. K. Hammer. Lean Information Management: The Integrating Power of Information. In H. J. Pels and J. C. Wortmann, editors, *Integration in Production Management Systems*, pages 147–163. Elsevier Science Publishers B.V., 1992.

[9] U. Harms. Generals and Car Manufacturers take Advantage of CALS (in german). *Computer Zeitung*, 26(3):21, 1995.

[10] T.H. Hofbauer, C.C. Woo, and C.D. Martens. Decision Support through Facilitating the Exchange of Experiences in a Distributed Environment. *Int. Jour. of Intelligent and Cooperative Information Systems*, 3(3):255–278, 1994.

[11] G.P. Huber and R.R. McDaniel. The Decision-Making Paradigm of Organizational Design. *Management Science*, 32(5):572 – 589, 1986.

[12] M. Jarke, S. Eherer, R. Gallersdoerfer, M.A. Jeusfeld, and M. Staudt. ConceptBase – A Deductive Object Base Manager. *Journal of Intelligent Information Systems*, 3(3):1–37, 1994.

[13] K.E. Kendall and B.A. Schuldt. Decentralizing Decision Support Systems: A field Experiment with Drug and Criminal Investigators. *Decision Support Systems*, 9(3):259–268, 1993.

[14] M. Lynne Markus and D. Robey. Information Technology and Organizational Change: Causal Structure in Theory and Research. *Management Science*, 34(5):583 – 598, 1988.

[15] Jarke M., Jeusfeld M., and Szczurko P. Three Aspects of Intelligent Cooperation in the Quality Life Cycle. *Int. Jour. of Intelligent and Cooperative Information Systems*, 2(4):355–374, 1993.

[16] T. Catarci M. Angelaccio and G. Santucci. QBD*: A Graphical Query Language with Recursion. *IEEE Transactions on Sodtware Engineering*, 16(10), 1990.

[17] 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.

[18] L.L. Miller and S. Nilakanta. Organizational Decision Support Systems: The Design and Implementation of a Data Extraction Scheme to Facilitate Model-Database Communication. *Decision Support Systems*, 9(2):201–216, 1993.

[19] J. Mylopoulos. Expert Systems and Databases. In *Expert Systems: Integration with Databases and Real-time Systems*, 1990.

[20] 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.

[21] 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.

[22] 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.

[23] P. Peters and P. Szczurko. Integrating Models of Quality Management by an Object-oriented Repository. In *2nd Biennal European Joint Conf. on Engineering Systems Design and Analysis*, London, England, 1994.

[24] P. Peters, P. Szczurko, M. Jarke, and M. Jeusfeld. Business Process Oriented Information Management: Conceptual Models at Work. In *submitted to Int. Conf. on Organizational Computing Systems*, San Jose, Calif., 1995.

[25] T. Pfeifer. *Quality Management (in German)*. Carl Hanser Verlag, Muenchen, 1993.

[26] F. Roithmayr and J. Wendner. Results of an Empirical Study on Relations between Organization Strategy and Information System Strategy (in German. *Wirtschaftsinformatik*, 34(5):472 – 480, 1992.

[27] M.S. Scott Morton. The 1990s research program: Implications for Management and the Emerging Organization. *Decision Support Systems*, 12(2):251–256, 1994.

[28] E.-J. Sol. CIM – Communication and Information in Manufacturing. In H. J. Pels and J. C. Wortmann, editors, *Integration in Production Management Systems*, pages 13–21. Elsevier Science Publishers B.V., 1992.

[29] P. Szczurko. Integration of Distributed and Heterogenous Databases in a Federated Quality Management Architecture (in German). *submitted to GI Jahrestagung*, 1995.

[30] H.-J. Warnecke. *The Fractal Firm (in German)*. Springer-Verlag, 1992.

[31] R.T. Watson. A Design for an Infrastructure to Support Organizational Decision-Making. In *Proc. of the 23rd Hawaiian International Conference on System sciences*, pages 111–119, 1990.

[32] 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.

[33] J. Zhou. GUI to Databases – Survey and Practice. *GI Datenbank Rundbrief*, 13:49–51, 1994.