AN INTERNET-BASED INFORMATION SERVICE TO SUPPORT THE PRODUCT DEVELOPMENT PROCESS IN DESIGN OFFICES

Brigitte Koch, Marianne Koch

Fraunhofer Institute for Computer Graphics
Rundeturmstr. 6
64283 Darmstadt
Germany

ABSTRACT: This paper describes a Teleservice Centre for Design. The System is considered to support an Industrial designer starting at the very early stages of the design process up to the production of the prototype. Special emphasis is placed on the logical architecture of the system. An overview of the provided services and the components of the user interface and an outlook on further work complete this paper.

INTRODUCTION

The development of innovative products stems from a company's ability to create new product related knowledge, whereby human resources and computers are used in an integrated manner. A precondition for this is the meaningful use of efficient technologies to co-ordinate knowledge. However, new knowledge is only converted into concepts through the interaction of human creativity and experts who are masters of their professional domain. Coaching in knowledge on design promotes the dissemination of new knowledge for the subsequent development projects. At 'Sony', there is an internal 'Think-Tank' with the mission of generating knowledge on the future of the media sector and presenting and publishing it once a year internally.

Innovation underlies the creation of knowledge. Here, the information is the necessary raw material: intention, interpretation and dedication can transform this raw material into knowledge. The theory of knowledge creation describes the development of knowledge along a spiral process of interaction between two types knowledge contents. The subjective, implicit knowledge is hard to verbalise or put into form. Objective explicit knowledge takes a shape of words or numbers. The interaction between these forms of knowledge result in the creation of new knowledge. Pioneering design agencies, in turn, are strengthening their knowledge bases (computers, staff, and networks) in order to combine explicit knowledge, to offer companies sound and specific knowledge.

Design management is highly qualified to act as a co-ordinator for the combination of new knowledge with the submerged knowledge and knowledge of the future. However, various different sources of relevance to design can be coordinated if the design management maintains a network of knowledge resources.

The development of new products is the potential core of dynamic knowledge creation in a company. However, in reality the results of the design process often lag behind these expectations, with the result that the creation of knowledge for the strategic development of design, image and identity is neglected. The predominant tactics are still market focus, or a focus on competitors or styles. Here, a fundamental link is often misunderstood. Each product is a materialisation of technical and social knowledge, and each innovation is a shape given to newly created knowledge. As a consequence, what gets overlooked is the resource, which makes continuous innovation possible.

ANALYSIS OF TARGET GROUPS

For the European project TRIDENT and the German project TeCeD (Teleservice Centre Design), which is supported by the ADAPT Initiative, we have developed a prototype of an Internet based information and knowledge management system with a design oriented user interface. Therefore we had to analyze the target group of Industrial and Graphic designers and their requirements on future information tools. Another goal of the project was to validate the current technology, which is used by industrial designers during the process of information acquisition for product development. For the evaluation of usability aspects and user friendliness of our demonstrator we held several workshops with industrial and graphic designers.

Evaluation

For the evaluation of the system requirements we chose small and medium sized enterprises. We have created a questionnaire for the target group of designers to find out the state of the art of their technical equipment and of the currently used software tools to support the product development process. During the several workshops we have prepared presentations to demonstrate the current state of the art of World Wide Web and Internet application developments. Together with group discussions, eye-to-eye talks and protocols of the different workshops we have developed a simulation of an application scenario. This means, that the users (designers) had to describe a task where it is necessary to collect specific information during a product development step. Besides the results for the application specific requirements we also got clues for the analysis of the user interface and also for the integrated mechanism of interaction.

Results

The questionnaire and the group discussions together with designers were very important to find out what focus to the content the user group will have on the system. Therefore in one of the first step we have collected together a wide variety of search terms. Because these terms had no hierarchy at this stage of the analyse we had to find out the different importance of the terms. After that we have divided the terms into logical classes, and find a first hierarchy of the search trees for our system. This is the starting point to built up the logical structure of our database.

SYSTEM ARCHITECTURE

The TeCeD information service has the objective to provide the user with an internet-based information system that has access to several information sources like e.g. an Oracle database that is especially developed for TeCeD to meet the requirements of designers. One concept of TeCeD is platform independence concerning the connection with other databases. Thus the functionality of the TeCeD-system can be enlarged step by step.
Furthermore, an important element is the implementation of agent technology which enables the user to apply intelligent, user-friendly and advanced search strategies.
This paragraph gives an introduction to the fundamental design issues the system is based on. Afterwards the user interface is presented, realized in a java environment. Finally the logical architecture and the agent technology are described.

Design issues

Java environment
The environment of the system is based on java. Java has become the programming language of the internet because it is portable, architecturally neutral and has built-in functionalities.

Java can be used to create two types of programs: standalone applications and applets. The most fundamental difference between applications and applets is their runtime environment. Applets are only valid within a browser such as Netscape Navigator, etc. Applications can be run directly from the command prompt with the use of the java interpreter.
The heart of java is the Java Virtual Machine (JVM). The JVM is a virtual computer that resides in memory only. Java source code compiles into portable bytecodes that require an interpreter to execute them. The JVM enables Java programs to be executed on a variety of platforms as opposed to only the one platform for which the code is compiled. In order for Java programs to run on a particular platform the JVM must be implemented for that platform. The JVM is the reason for the portability of java. It provides a layer of abstraction between the compiled Java program and the underlying hardware platform and operating system as is shown in figure 1. Here the platform independent parts are indicated in white whereas the boxes marked in grey show the proprietary parts.

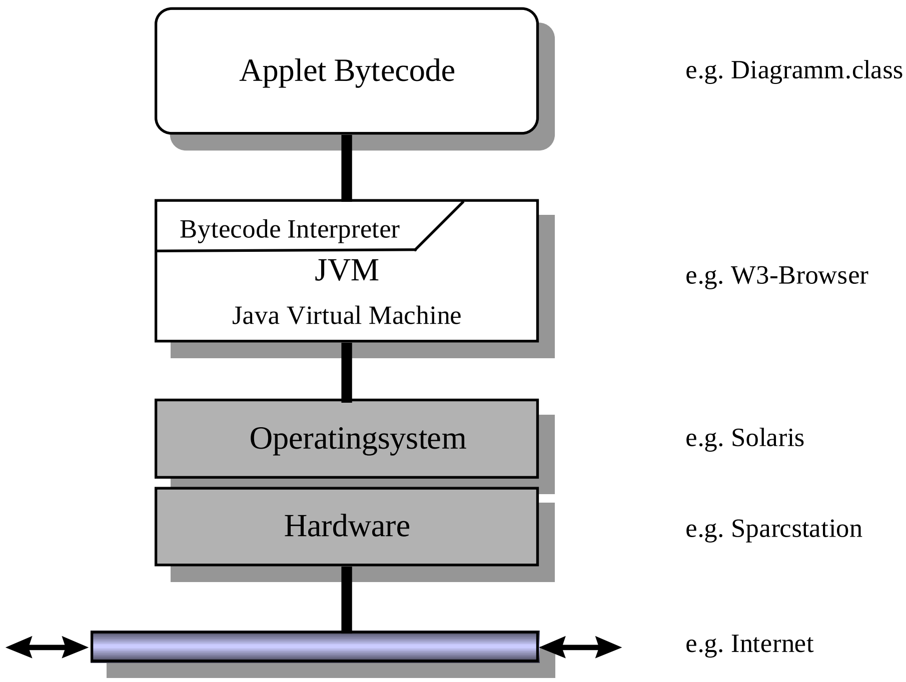

Figure 1. Structure of Java Applications

The user interface for the internet-based TeCeD-system is realized with a Java applet.
The browser loads an HTML-document containing a reference to an applet. Then the applet is loaded and executed in the browser of the user. In contrast to a simple HTML-page it is possible to perform transactions such as setting requests to different underlying databases or the input of various kinds of information which is an important feature of the TeCeD information service.
HTML follows the page-oriented concept of HTTP: the client loads page after page but the WWW-server is not able to store a users' request.
An applet is able to support the transaction concept of databases. It is possible to perform longer transactions in terms of the database server is able to save the current context. The user has the possibility to edit and refine his requests.

Logical architecture
The system is based on a client/server concept. As mentioned before the realization takes place step by step. The first step is a traditional 2-tier architecture: the client-server application consists of two layers. The client side contains the logic of presentation, the business logic and the logic to manipulate data. This is a so called 'fat client'. The server could be a relational database management system (RDBMS).

The browser orders the applet from the WWW server that sends the applet to the client. Here it is executed in the browser of the user. The communication between the client and the relational database is enabled with a JDBC interface (see figure 2). A special feature is shown in figure 2a where an agent is implemented in an application server. A software agent is a computing entity that performs user delegated tasks autonomously [2]. With the login of the user the agent loads the user preferences stored in the database and will be configured accordingly. This personalized agent is now able to generate requests for the JDBC-interface with individual user preferences. Furthermore, the agent is able to process the results according to the users' needs.
The advantage of this architecture is that with the use of the agent the tasks are distributed and the result could lead to a better load balance.

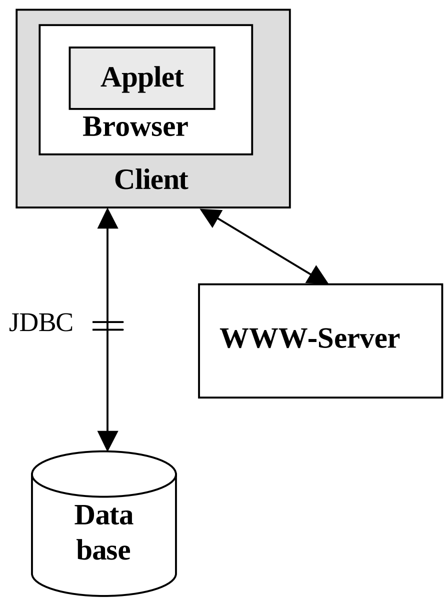

Figure 2. 2-tier architecture

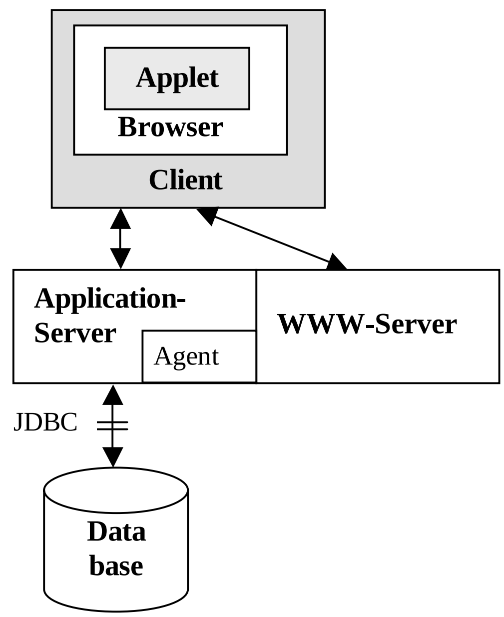

Figure 2a. 2-tier architecture with agent technology

An increasing number of users require a scalable architecture to solve the resulting increasing number of requests stated by the users. As the scalability of 2-tier client/server architectures is limited it is necessary to introduce another layer between client and server. To overcome these disadvantages like load-balancing a middleware layer has to be implemented into the architecture where the distribution logic of the system is realized. In multi-tier architectures the client implements the presentation logic. The business logic and the access to data is realized with several software components that are distributed on one or more servers.
Figure 3 shows a three-tier system, the most typical architecture of multi-tier systems. In this system the functional client/server components are logically divided into three layers:
1) The user interface, where on client side the GUI (Graphical user interface) is implemented.
2) The middleware, where the business logic is realized.
3) The server layer, where the actual data are stored.
The communication between the different layers is implemented by the middleware.
One advantage of splitting up the different functions is better performance because of reduced network traffic when a pool of requests is collected in the middleware and performed by a connection to the database server.
Another advantage is high scalability and efficient download of the client software (e.g. java applets): with an increasing number of users the system can be enlarged easily.

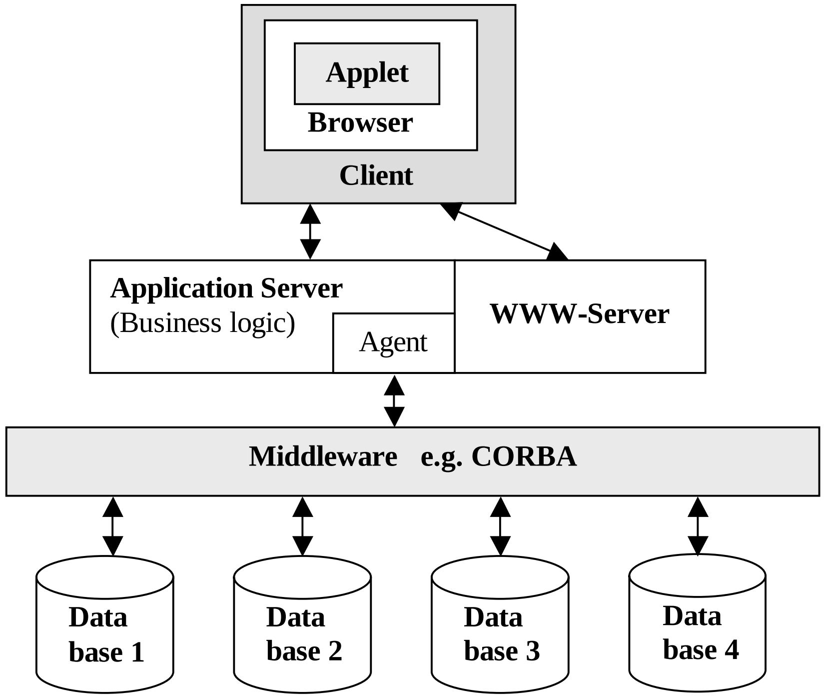

Figure 3. 3-tier architecture

A typical 3-tier system is shown in figure 3. The components for communication with the underlying databases could be CORBA (Common Object Request Broker Architecture). Within this architecture a CORBA object bus is used as a transport mechanism for the protocol of TeCeD. The CORBA object bus defines the interface of the components, the location and how to activate the components and the interoperability between components [9]. A special feature (see also figure 2a) is again the agent in the application server. The advantage is once more the scalability: when more users get connected to the system the application server could be enlarged. The agents become communication agents delegating tasks to other agents who perform the 'real' calculations. Only the communication agent can access to the applet.

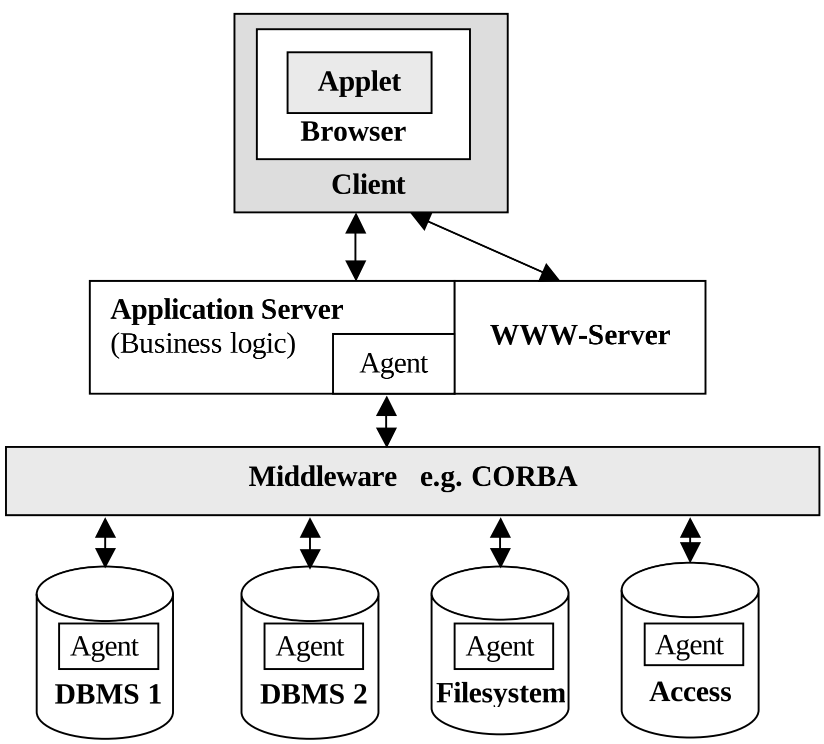

Figure 4: Extended architecture of the system

Figure 4 shows another step of extension. One extension to the previous architectures are the defined protocols inbetween the agents on the different layers. This gives the possibility to have access to different types of databases, e.g. DBMS, Filesystems, other kinds of databases such as Access, etc. The user performs a request to the system. The request is given to the agent that can communicate with the different components of the systems because of defined protocols. Furthermore, this agent can activate an internet agent (see also paragraph 'Agents').
Several advantages of the architecture are: any kind of new database can be added to the system without changing the architecture or even shutdown the system. With CORBA the agent in the application server does not have to know where the databases are. The database specification concerning interfaces and functionality are encapsulated by the database agents. The agent in the application server just needs knowledge about the agent language.

Realization architecture

The architecture of the TeCeD-system consists of various components. The graphical user interface (GUI) on client side is implemented as a java applet and runs within an HTML-page. With this GUI the user gets connected to the system and communicates with the underlying components of the system, e.g. the TeCeD-database.
The TeCeD-database is realized with Oracle 8i. The scheme of the database is designed especially according to the needs of designers.
Oracle 8i is a database system with an object-relational approach. This means, it consists of a relational database system that manages data in tables but it also contains object-oriented concepts like object types (a user-defined data type that encapsulates a data structure together with the belonging executable methods) and object tables (tables are able to contain objects instead of tuples).
As an integral feature, Oracle 8i provides an Internet platform. Java, as the language of the Internet, is integrated with several Java development and deployment environments:
With the Oracle JDeveloper tool Java applications are programmed.
Moreover, Oracle 8i delivers the server-side Java Virtual Machine (JVM) that executes Java applications and can be ported to any type of system.
Net8 enables client-server and server-server communications across any network. In a distributed computer web there can exist several databases each with an Oracle-instance. With Net8 access to each of these remote databases is possible. Furthermore, access to many distributed databases at one time is enabled.
To access the TeCeD database from the java applet there is an interface for communication: the java database connectivity (JDBC).
The way java applets work is that they constantly read data from a data storage source, process the data and afterwards write them back to the data store so that the data can be processes by other applications. When access to a database by a java program or applet is required an open standard for data access is necessary because of an applets modularity and portability. Java provides the java database connectivity the so called JDBC interface. With JDBC it is possible to enable a java application or applet to have access e.g. to an Oracle database.
Java Database Connectivity is a set of relational database objects and methods for interacting with SQL data sources. The JDBC API's are part of the Enterprise API's of Java 1.1 and thus are part of all Java Virtual Machine (JVM) implementations.
JDBC defines a set of API (Application programming interface) objects and methods to interact with the TeCeD-database. The java applet first opens a connection to the database, makes a Statement object, passes SQL statements to the underlying TeCeD-DBMS (Database Management System) through the Statement object, and retrieves the results as well as information about the result sets.
As mentioned before, the user communicates with the TeCeD-system by means of the java applet. Here, the designer is retrieving information in terms of sending requests or giving in information and providing the system with new data. For the TeCeD-system it is to make database-queries or transactions with SQL-statements. The SQL (Structured Query Language) is a universal standardized database language to perform actions on data. SQL has grown into a mainstream

database language that has constructs for data manipulation, data definition, data management and, most importantly, transaction processing. The concept of transactions is an integral part of the TeCeD-system. A transaction is the succession of SQL-commands which present a logical procession unit. In other words a transaction is a group of SQL statements that update, add and delete rows and fields in a database. Transactions have an all or nothing property - either they are committed if all statements are successful or the whole transaction is rolled back if any of the statements cannot be executed successfully. With a restart the transaction is repeated and the data is recovered. Transaction processing ensures the data integrity and data consistency in a database. JDBC supports transaction processing with commit and rollback methods.

Agents

Apart from the agents described in the logical architecture (see paragraph 'Logical architecture') the concept of the TeCeD-system supports agents with advanced features: the support of internet-retrieval.

To retrieve information in the internet it is common to use search engines. The ability to retrieve relevant information is very different and the user has to make a request with free search terms. One disadvantage of search engines is e.g. what the user is retrieving is not indexed or he gets too many (not relevant) hits. World-wide there are more than one thousand search engines. The user has to decide which one is the best and which one covers the part of the internet he needs. Each search engine indexes only a very small part of all documents in the internet. The explosion of information in the internet goes on. To locate relevant information is time-consuming and it becomes more and more difficult. There is a need for high-quality search tools that get information for the user from the various information sources and that make information retrieval as easy-to-use, efficient and powerful as possible.

Agent-based applications can be used to overcome these problems. Information agents support the user in his retrieval for information in e.g. in the internet and make retrieval more automatic. Agents are components that are able to perform tasks independently according to the users needs [4]. The information agents serve as information brokers between information suppliers (e.g. Websites, online databases, etc.) and Web users or other agents. Internet agents match the information needs of web users against the attributes of information suppliers, type of information and information content. [2]. An agent represents the user and finds and extracts information sources, filters and presents information according to the users' needs. The benefit for the user is to save time by not having to make a time-consuming uncomfortable retrieval. User benefits are more efficiency, lower costs and reduced process time with useful retrieval techniques.

The GUI is divided into two frames: in the left frame the user finds the navigation toolbar. The right frame has two representations: a field of information to specify the request or the presentation of results. In the left frame the user can decide whether he wants to retrieve information or provide the system with information. For both ways the navigation is the same. The left frame has basic areas. The main area is the navigation tree. The structure of the tree consists of three layers. The first layer consists of seven fields of information. Each of them has subfields that represent the second layer. Finally there is the third layer. This layer is a substructure of the second layer. The representation of this layer is on the right frame of the GUI. Other areas of the left frame are icons for getting current helping items, for subscribing to mailing lists, for giving in user preferences or for going back to the homepage from any place in the system. By scrolling over an icon the meaning of it is explained, e.g. input of user preferences, etc. Underneath the icon-area there is a bar where the user has the possibility to make a request with free search terms. On the bottom of the left frame the user has a button for leaving the system at any time.

The right frame presents the more detailed sub-information fields. With a search formula the user is guided and supported in specifying his request. As the TeCeD system is tailored to the needs of designers, the terms are especially adapted according a designers' needs.

DEVELOPMENT OF THE PROTOTYPE

User interface

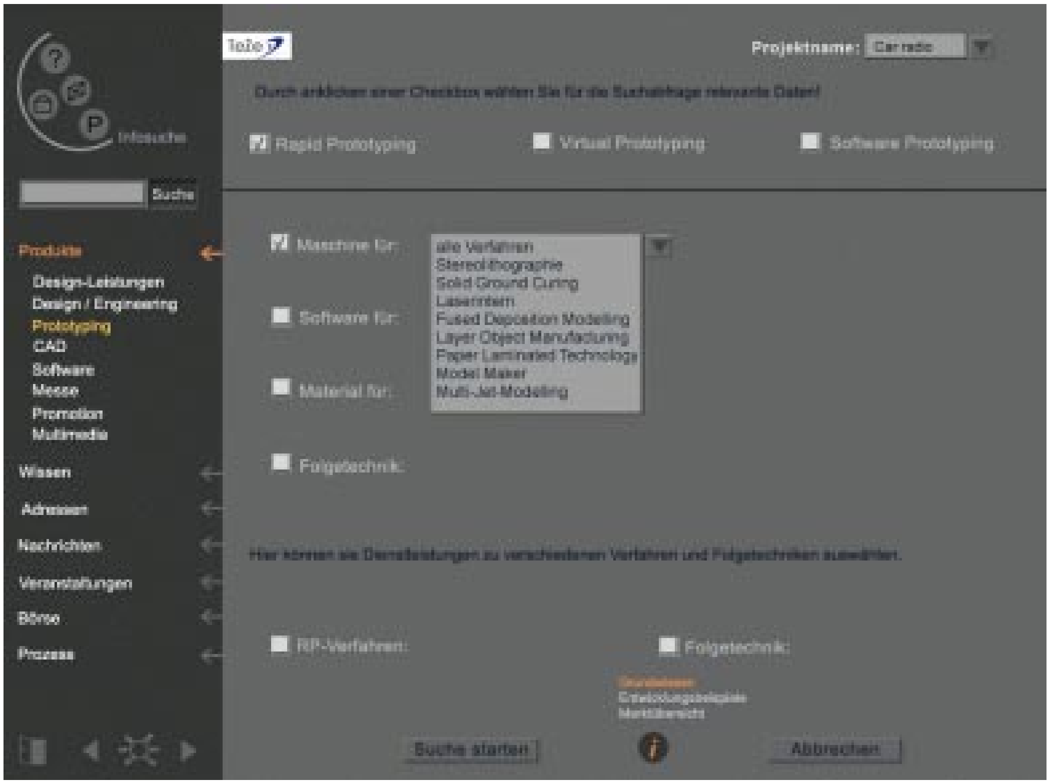

Figure 5. The graphical user interface of TeCeD: The user is retrieving information about machines in the field prototyping.

Search results are also presented in the right frame of the GUI. The presentation can be textual or graphical or in other multimedia ways, like e.g. a video. On the bottom of the frame the user finds buttons to activate a search, to store input data such as user preferences or giving in new information. The user always has the possibility to finish the current session or to quit the system. Furthermore, there is an information icon for getting basic knowledge about the field of information he is currently retrieving.

A characteristic of the graphical user interface (GUI) of TeCeD is the clear and user friendly structure. The navigation-tree serves as a help for orientation. The structure of the tree has a flat hierarchy. Thus the user does not get lost in the system. Highlighted arrows and headings show the user where he currently is. The left frame stays while the user interacts with the system. This helping features support the user in clear orientation and user friendly navigation. This refers to a collection of rules, which determines the form the interaction will take in terms of the software design. Essentially, it establishes which steps a user has to take to perform a certain action and the alternatives that are available to him as he does so. Such an interaction grid allows certain sequences to be transferred analogously to other situations, thus making learning easier.

As shown in figure 6 the user is enabled to give in his preferences to the system. He can do this for different projects. Apart from his personal data like name, address, profession etc. he can give in information about the subject he is interested in and can give in keywords and the project name. If he activates the search or lets it activate by an agent he has no need to give in the terms every time. Moreover he can specify a search for different projects that means one search preference for one project. All these information are stored in the underlying TeCeD-database. The user benefit is a personalized request with the support of agent technology (see also paragraph 'Logical architecture').

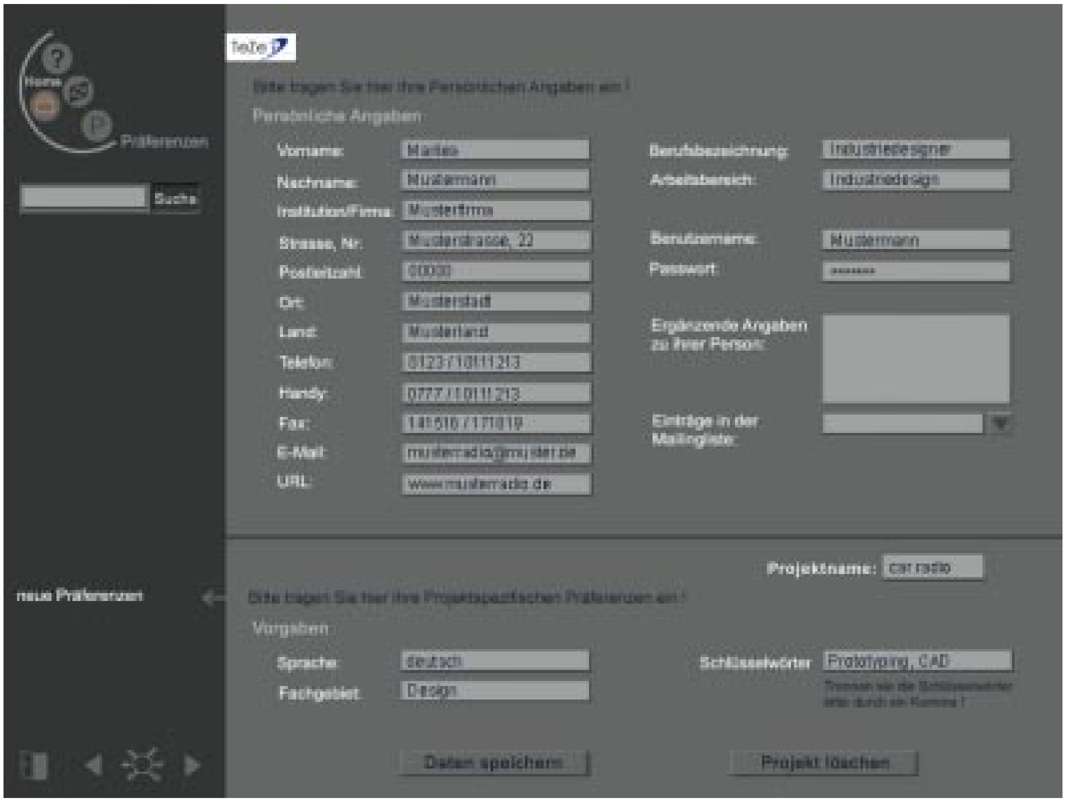

Figure 6. The graphical user interface of TeCeD: The user provides TeCeD with user preferences.

Menu scheme

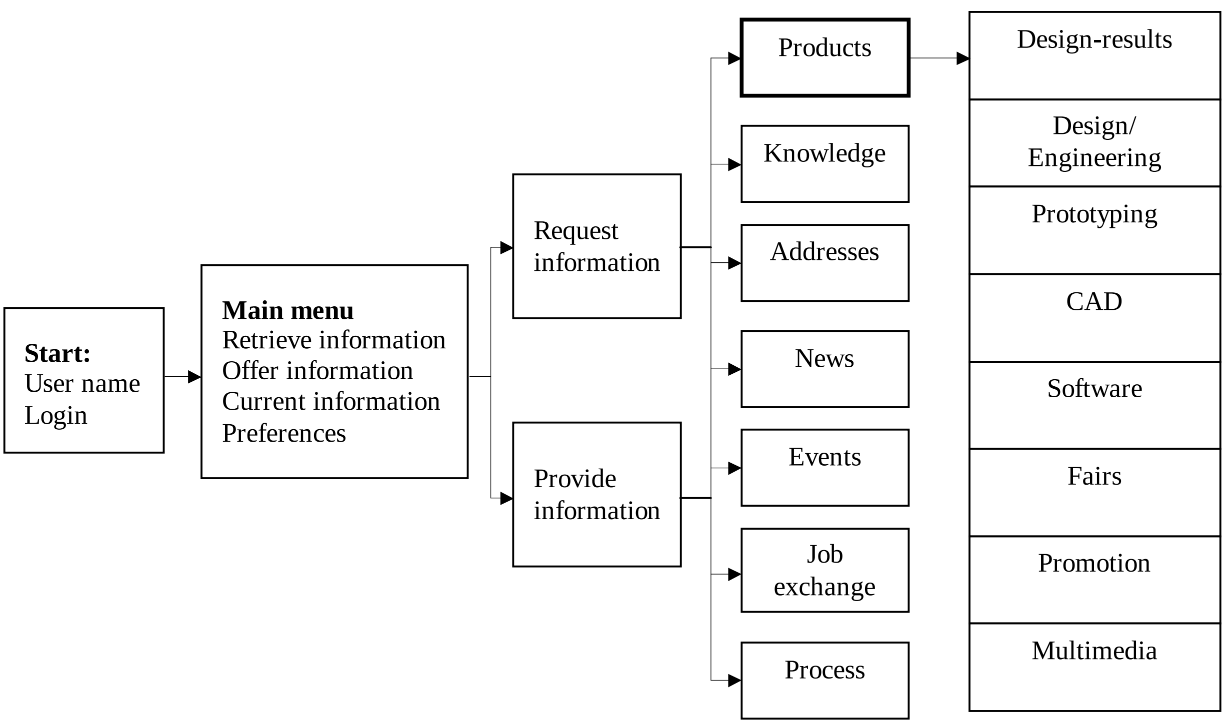

Figure 7: The menu scheme of TeCeD

The structure of TeCeD is as figure 7 indicates as follows: to start communication with the TeCeD-system the user has to login to the system by giving in his account. After registering the user gets to the main menu where he can basically choose between retrieving information and providing information. These are the two main ways for the user to communicate with the system. In both ways the user is leaded to the seven main fields of information: Products, Knowledge, Addresses, News, Events, Job exchange and Process. Each of them is structured like a tree into two more layers of detailed sub-information fields. So the hierarchy of each of the seven main fields of information consists of three layers. The number of layers is limited to three to prevent that the user gets lost within the navigation. Figure 7 shows the main navigation of a user through the system. The information field 'products' is shown as one example for the substructure of one field. Of course each of the other six fields has its refined structure as well.

CONCLUSION

The TeCeD system described in this paper will support designers during the different stages of their personal designs tasks. It focuses on the information and knowledge collecting process of the design tasks during a product development process. Our main goal is to implement a prototype of a design oriented internet based information and knowledge management system.

The paper presented a prototype which combines different techniques for getting new ways of access to information. Following the incremental software development approach the TeCeD-system supports the concept of a scalable architecture The system can be enlarged without necessarily modifying the conceptual architecture. Modern techniques like agent technology enable the user to have fast and user friendly access to information sources. Agents enable the user to apply intelligent and advanced search strategies.

REFERENCES

1. **Belz, C., Jung, H.,** Aspekte zur Gestaltung von Benutzungsoberflächen mobiler Multimedia Online Systeme. *In: "Technik für den Menschen. Vorträge der ITG-Fachtagung,* Eichstätt, Germany, June 26-27, 1998, 79-85

2. **Caglayan A., Harrison C.,** Agent sourcebook, *John Wiley & Sons, Inc.* USA, 1997

3. **Fiedler, J.,** Mobile Agenten im Umfeld personalisierter Nachrichtendienste, *Diplomarbeit Brandenburgische Technische Universität Cottbus,* Germany, 1997

4. **Frotscher, T.,** Entwicklung von Methoden und Konzepten für den Einsatz von skalierbaren und erweiterbaren Middleware-Technologien für verteilte Geo-Metadaten-Informationssysteme im WWW, *Diplomarbeit Technische Universität Darmstadt*, Germany, 1999

5. **Hohenstein U., Pleßer V.,** Oracle 8: Effiziente Anwendungsentwicklung mit objektrationalen Konzepten, *dpunkt Heidelberg,* 1998

6. http://www.oracle.com

7. **Jasnoch U., Koch M.,** An environment and an information service for global manufacturing, *Proceedings FAIM 1998, Portland, USA,* 1998

8. **Nonaka, I., Takeuchi, H.,** The Knowledge Creating Company. New York, *Oxford University Press,* 1995

9. **Orfali R., Harkey D.,** Client/server programming with Java and CORBA, *John Wiley & Sons, Inc. USA*, 1998

10. **Sander-Beuermann W.,** Schatzsucher Die Internet-Suchmaschinen der Zukunft, c't Heft 13, Verlag Heinz Heise Hannover, 1998, 178-184

11. **Urban, B., Kreutschmann, R.,** Information Management Tool, *IuK-Tage MV,* Schwerin, Germany, June 25-26, 1997

12. **Weber J.,** Special edition using Java 2 Platform, *Que USA*, 1999

13. **Zabel A., Deisinger J., Weller F., Hamisch T.,** A multimodal design environment, *Proceedings ESS European simulation symposium and exhibition 1999*, Erlangen-Nuremberg, Germany, 1999