A System Architecture for the Extension of Structured Information Spaces by Coordinated CSCW Services

Peter Manhart
Research and Technology
Process Engineering

DaimlerChrysler AG
P.O. Box 2360, 89013 Ulm, Germany
Peter.Manhart@DaimlerChrysler.COM

ABSTRACT

The World Wide Web is an emerging platform for information systems; however established system architectures for web systems focus mainly on the creation and storage of consistent hypermedia information structures and on the efficient distribution of the resulting documents. The interaction between the information users is seldom supported.

As many application scenarios profit greatly from human interaction, the paper presents a platform- and application-independent generic system architecture designed to extend existing web-based information systems by coordinated services for human interaction. One prototype implementation of the architecture supports user awareness and human interaction on corporate web sites.

Keywords

Group interaction, system architecture, CSCW services, web-based human interaction

INTRODUCTION

Web technologies basically support the *platform-independent* representation, storage, distribution and visualization of hypermedia information. This is a significant innovation, compared to former technologies, and it explains the exponential growth of WWW-based solutions in the last years. But even if thousands of web users browse a common web site, every user sees himself as an individual performing human-computer interaction.

The guiding vision of the presented approach includes web systems which are not only a medium for the distribution of information, but also a medium for interaction. Improving the web so that it no longer acts merely as an "Info Space" but instead becomes a "Meeting Place" might change the nature of the medium fundamentally.

However, current internet protocols do not directly support the integration of human interaction and web browsing: on the one hand, there are the hypermedia representation and distribution protocols, HTML and HTTP, and, on the other hand, we have the well-established human interaction-oriented protocols such as IRC or NNTP. An integration of the web technologies with systems to support human interaction is an extra task.

What has been stated about web browsers can be seamlessly extended to other business-level tools relying on structured information spaces, e.g. EDM[1] systems. Again, they mainly operate independent of systems to support group interaction, and the integration of the two worlds can become a costly and complex task.

Resulting from this argumentation, the proposed architecture has three major goals:

1. Provide and integrate new basic functionalities that are prerequisites of group interaction in information spaces to stimulate group interaction.
2. Flexibly integrate the use of hypermedia and CSCW systems to make system engineering easier.
3. Coordinate the use of hypermedia and CSCW systems to provide an increased use of the technologies.

In the first section of the paper "Application Scenarios" the motivation for increased group interaction on hypermedia information spaces is presented. The section "Requirements" analyzes the scenarios for demands on a system architecture. "System Architecture" describes our architectural approach for meeting the scenario's requirements. Then a "GIA[2] Prototype" in the field of customer support is presented. The rest of the paper shows perspectives, related work and references.

APPLICATION SCENARIOS

Web-based group interaction is, in principle, not a domain-specific approach. Nevertheless, concrete application scenarios impose substantial requirements and constraints on general approaches such as architectures. In the following, the intended contexts of the proposed architecture are described: interaction-intensive

GROUP 99 Phoenix Arizona USA

[1] EDM: Engineering Data Management

[2] GIA: Group InterAction

applications of information systems that are based on structured information spaces within large industrial companies.

Cooperative Learning and Knowledge Management

Cooperative learning is an interesting application area for web-based group interaction since an increasing number of learning systems are implemented based on WWW technologies. The following learning scenarios are considered to be a part set - but an interesting one - of organizational learning forms. They have been selected as the motivation for the advanced interaction services that are not implemented in conventional WBT[3] systems.

In *asynchronous learning courses*, human interaction on demand can break learning barriers and, as a result, speed up the learning process. Regular communications additionally increase the learners' motivation and discipline. *Video-/audio-based lectures* enhance asynchronous learning courses by means of synchronous multimedia communication and interactive presentation forms. This can result in an increased learning performance as learners can ask questions on the spot. Although the required network bandwidths are comparatively high, this learning form is well established in corporate distance training (e.g. [15]). *Individual training* and *training on the job* are comparatively efficient (yet expensive) individual forms of learning adequate for many situation. The interaction of learners and tutors is inherently high and is also a central point in virtual settings. *Project-oriented learning and distributed task forces* focus on supporting the learner in his or her current working situation. The interaction of learners and tutors is a key success factor and should, consequently, be supported as far as possible. *Content-oriented learning groups* sets out a learning situation where learners share a common interest, learn and create new knowledge in parallel. Intensive interaction is a main fundament of this learning form.

As the above examples show, cooperative learning forms can be deployed as one of the most interesting application areas of web-based group interaction.

Electronic Commerce

Web commerce systems are an increasing segment of today's corporate web-based information systems and are, therefore, also an interesting application area for the presented architecture. The existing functions of available electronic commerce systems concentrate mainly on product visualization, shopping baskets, accounting and electronic payment. Advanced systems store the users' preferences regarding the products [9]. Our approach adds communication and awareness to the pure exchange of goods [5]. The guiding vision here is the "busy and crowded market-place" instead of the "cheap ordering system". In this market-place, customers and suppliers can interact to break down uncertainties and other buying barriers.

Virtual Communities

Virtual communities have the potential to become a significant market in the future [10]. Some key success factors of communities on the web supported by group interaction are: attractiveness of the site, possibilities to build personal relationships and partnerships, and customer-oriented web services. In addition to the global community of a brand's owners, special communities for product segments can be implemented, e.g. the community of truck drivers.

Web Events

The additional interaction services of GIA can be used for communication between customers and well-known or famous persons related to a company. Marketing events are quite common on the internet. GIA affords the opportunity of combining web content with interesting online communication - which makes a web system more attractive.

Engineering and Service

The access to design documents and the exchange of product-relevant information plays a significant role in engineering and service. The enhancement by human interaction services increases the effectiveness, as well as the efficiency, of web-based engineering systems as problems, uncertainties and incompleteness of design information can be solved at once and possible implications of intended design changes can be discussed in advance.

Customer Support

The area of customer support is the domain of our current prototype implementation of the GIA architecture. In the following, some concrete scenarios are sketched.

Just-in-time user support: Customers who prefer to use corporate web services rather than browse information need support in uncertain situations. Group interaction makes it possible to provide customers with on-the-spot support right when they need it, e.g. when they are not sure which consequences the sending of a web form to a corporate organization would have. Assistance encourages more customer interaction in this case. Today's customer advisory services are often inadequate for the specific needs of customers. Through the provision of different kinds and levels of customer services, individuals can get exactly the level of attention they need or prefer. In general, we could distinguish between visitor services, customer services and premium services. Web-based customer services include customer information, consulting, servicing, the sale or rental of goods, after-sales services and the sale of accessories. Services that depend on customer agreement will work better, if the customer can discuss open questions about the underlying business process. GIA can be a platform for advice by corporate staff.

[3] WBT: Web-based Training

REQUIREMENTS TO AN ARCHITECTURE

Functional and non-functional requirements may be derived from the scenarios described in the previous section:

Functional requirements

The cooperation functionalities required in the intended application areas cover a broad range. Whereas scenarios such as virtual communities may rely more on synchronous functionalities like chat, cooperative learning also needs asynchronous forms of interaction such as discussion fora. As group-building mechanisms play a significant role in virtual communities, customer interaction and learning, the support of user awareness is a useful function. In scenarios such as engineering, dependencies between the nodes of the information space and the results of changes are a critical point. Here, notification services serve to stimulate and support user interaction. Both learning and engineering applications need the integration of domain-level tools with CSCW functionalities. All the scenarios build on structured information spaces (by definition). A system architecture can use this fact to provide additional services for the users.

In summary, the scenarios would profit from the integration of CSCW tools for different modes of interaction and from the integration of domain-level tools with group interaction systems.

Non-Functional Requirements

The different organizational settings of the scenarios imply a wide variation of non-functional requirements for a supporting architecture:

- *Number of users*: The scenarios differ widely. Whereas, in cooperative learning, five users per week seems a realistic figure, applications in the area of virtual communities or customer support must handle 10,000 clients in parallel.
- *Network structure*: In the sketched application contexts, different network structures appear. Easy-use cases come with clients that connect from intranets or LANs. Here, normally all types of connections work. More complex are the mixed internet and intranet scenarios such as virtual communities as certain kinds of connections can be blocked by firewalls or routers.
- *Client equipment*: This is strongly correlated with the preceding argument. Within an organizational closed scenarios like cooperative learning, the client software equipment can be defined according to the needs of a special application. In an open scenario such as electronic commerce, the architecture has to support different client settings.
- *Performance*: The same arguments apply to the performance of the clients and the network. Some scenarios enable the building of high-end solutions, whereas others must get along with a mixture of high- and low-performance infrastructures.

In addition to the specific requirements of the scenarios, there are also general requirements. For example, it should be possible to integrate new software components generically and efficiently.

The following section describes the GIA system architecture that was designed to meet the requirements of the application scenarios set out.

GIA SYSTEM ARCHITECTURE

The following description of the system architecture starts with a brief illustration of the architecture's key concepts. This is followed by an overview of the architecture. Then the components of the approach are described in more detail. After that the types of connections between the components are classified.

Key Concepts of GIA

Before describing the architecture's structure and components, the main concepts of the approach are presented.

Structured information space: Central to our group interaction approach is the concept of a "structured information space". This term is an abstraction of all kinds of (digital) information representations that come with an explicit and coarse-grained inner structure. Examples include hypermedia information (nodes and links), directory structures (directories and files) or database representations (tables and records). As the WWW is currently the most successful structured information space, our implementations concentrate on HTML-encoded hypermedia information. The architecture, however, is not restricted solely to web information.

Coordination of domain and CSCW tools: When the users of a computer system access the information via supporting tools, they normally work in a single-user mode. We could call this domain-level computing. An example would be an engineer accessing a three-dimensional product model, using a web browser. We separate this "*domain-level computing*" from the necessary human interaction that takes place during cooperation in a business process or in coordinating the domain work.

When the users need to cooperate although they are distributed in space or time, CSCW systems support the communication, cooperation and coordination required. Consequently, we come to a computing model where the users have domain tools to drive their primary business processes and CSCW tools to interact while doing so.

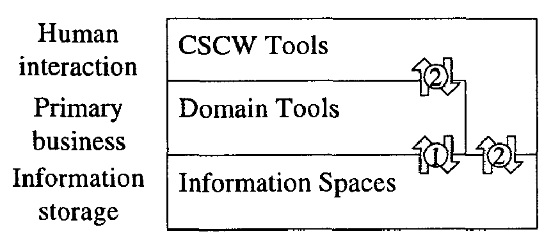

Figure 1: Layered Computing Model

In many existing information systems, the tools in the layers of human interaction and primary business are not integrated but work independent of each other. Only the domain tools access the information space (see Figure 1, flow 1). An engineer in a distributed team, for example, would work with his or her CAD[4] system until a need for interaction with the responsible engineer of an involved part arose. Then he or she might search for the telephone number of the engineer involved and try to reach them. If that failed, a search for an email address might be the next step in the efforts to establish contact: an email client is started and a message sent. Here, another central concept of the GIA system architecture comes into play: to better support the distributed interaction of the information system users, we coordinate the CSCW services with the domain tools, the access to the information space and between each other (see Figure 1, positions 2). If, for example, an engineer accesses a specific part, he should be aware of who else is involved in the corresponding information node, and the interaction tools (awareness view, telephone, chat, email, etc.) should be automatically configured for interaction with the persons involved.

Integration mechanisms for existing components: this coordination of tools can be achieved in a variety of ways. Often a tight integration of CSCW systems is synonymous with re-implementing both the client and server sides. The reason is that fine-grained control over existing CSCW components or systems usually cannot be achieved adequately or requires complex solutions as generic approaches are missing. However re-implementation has a number of disadvantages. Integrating existing tools is more efficient and reliable, and generic tool integration would enable us to "plug in" new functions more easily. So, another main concept of the proposed architecture comprises elements called "adapters" that allow us to bridge the gap between native GIA components and applications from the market.

In conclusion, the basic concepts of the GIA architecture are as follows:

- The explicit representation of structured information spaces,
- the coordinated integration of CSCW systems in the primary business processes on the basis of the structured information spaces, and
- the generic integration of domain-level and CSCW client/server systems.

In the following, the main building blocks of the proposed architecture are introduced.

Overview of the Basic Structure

Being a client/server architecture, GIA is structured in client components, servers and connections (see Figure 2).

[4] CAD: Computer Aided Design

There are two categories of connections between components of the GIA system architecture. On the one hand, we use standard internet protocols, such as HTTP or NNTP, to Internet protocol clients with Internet protocol servers. Additionally, we deploy a Distributed Event System (DES) to reliably and efficiently connect GIA components in such a way that the distribution is transparent.

The GIA components can be divided into internal applications, external applications, and adapters. *Internal applications* are available in source code and are, therefore, subject to fine-grain integration and control. *External applications* are components which are regarded as "black boxes": their source code is either not available or too complex to be adapted. As a result, the interaction of GIA architecture with external applications is poor compared to the possibilities offered by internal applications. Adapters are bridges to external applications that cannot be connected via the DES.

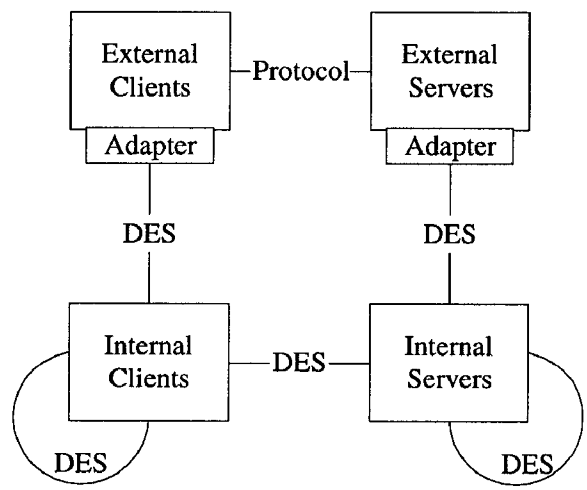

Figure 2: GIA Architecture Overview

The *internal server* applications are available in source code and implement all the system functions necessary for the group interaction services which are either inadequately or not implemented within commercially available CSCW servers. The *external server* applications support the GIA functions in the form of commercially available systems. They are divided into Internet protocol servers and domain-level servers.

Internal client applications implement all those user interfaces to group support functions which are not available as external applications or cannot be integrated as desired. *External client* applications are either domain-level applications integrated as black boxes or Internet protocol clients implementing interfaces to CSCW services based on internet protocol servers.

Detailed Description of the Architecture's Components

In this section, the components of the GIA architecture are described in further detail (see Figure 3).

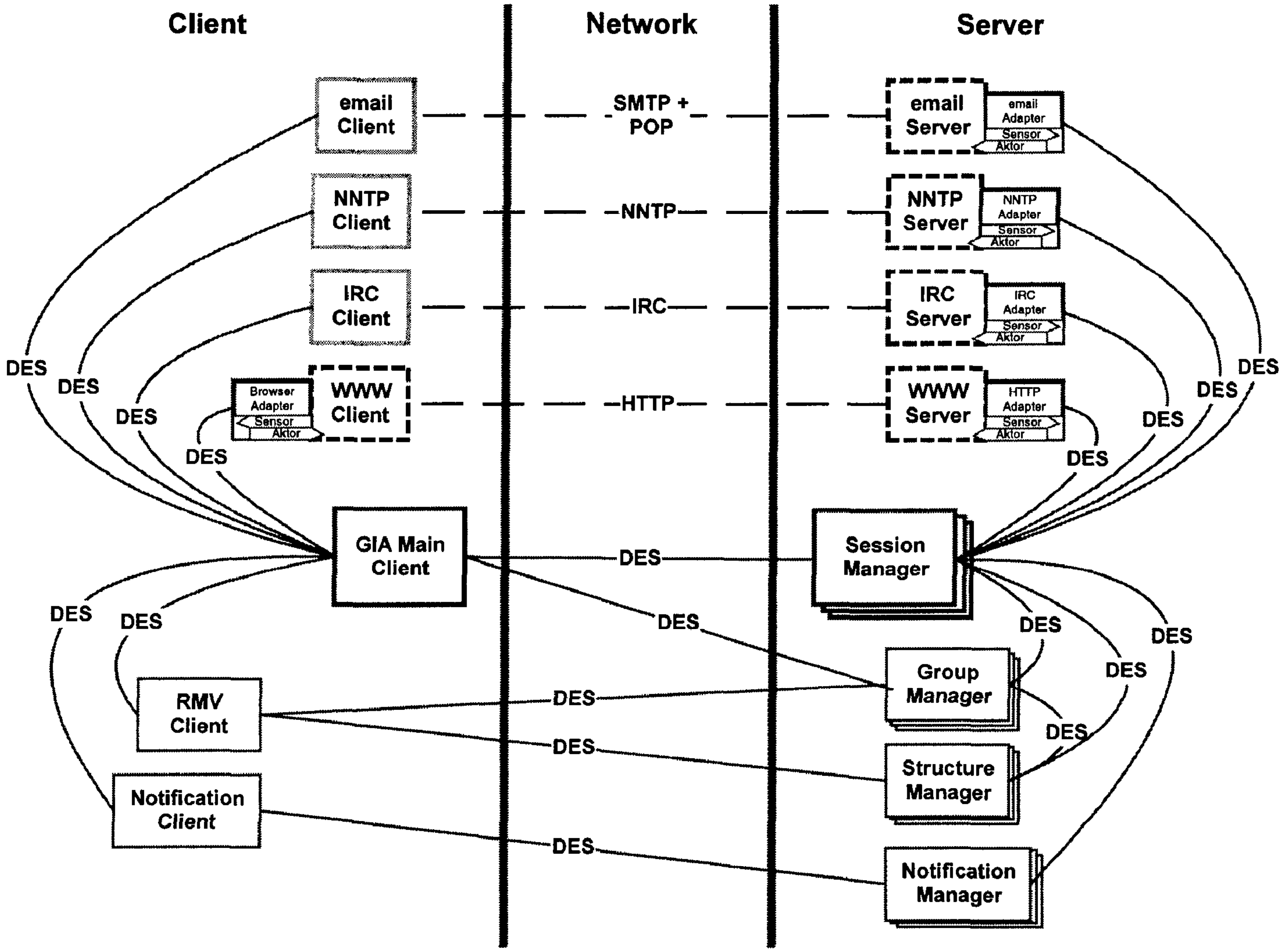

Figure 3: Components of the GIA Architecture

Internal Server Applications

The Session Manager (SM) plays a significant role in the architecture. The services provided by this manager are primarily logon and logoff procedures for GIA clients. When a client logs in the system, the Session Manager creates a corresponding session object. The session object consists of session specific data and a client object that holds all the necessary information about the GIA client. As there is no further client manager, the Session Manager is also the server for queries about a client's state, e.g. its current location in the information space.

The Group Manager (GM) provides another central service in the architecture: the dynamic grouping of GIA clients according to their state and history. Therefore, the Group Manager creates and processes a list of group objects. A group object holds a set of client objects that are members of the group. Whenever a logged-in client acts in the GIA system, the Group Manager reacts to the corresponding interactions by manipulating the client's group membership according to the implemented grouping policies. One simple grouping policy is to make every client a member of its current Location Group. This means the client is grouped with all of the other clients that share the same location in the information space.

The Structure Manager (SM) implements the functions required for structuring the information space related to the GIA system. If, for example, a GIA component requires information about the logical structure of a hypermedia information space, the Structure Manager has to partition the web and relate the partitions in a tree structure. A GIA component could then query the location of a piece of information or the relationship between two information nodes in the space. The complexity of partitioning and relating a information space is quite different, depending on the circumstances. Although it is fairly easy to automatically structure a standard file system hierarchically, it is quite difficult to do this with an unknown html web.

The Notification Manager (NM) implements those functions linked with reacting to changes in the information space or the GIA system. He has a set of sensors to query for changes and manages a list of notification requests of GIA components. A notification request specifies the type of change and the corresponding service quality parameters, e.g. the polling period. When a requested change in the system occurs, the Notification Manager sends a corresponding change event to all of those GIA components registered for the change.

Adapters (A) are internal server components that serve as a bridge between external and internal applications. Internal applications can be connected via mechanisms such as the distributed event system (DES) or RMI[5]. As external applications are generally not available in source code, the interaction with them is not only limited but also cannot be arranged by the flexible mechanisms that are used for the interaction of internal components. The Adapters are GIA components that implement application-specific access functions, as well as, a DES interface. Consequently, external applications can be accessed more like (but not as fine grained as) internal components. An adapter is divided into a sensor and an actor. The sensor reads information from an external application, whereas the actor writes data to it or controls its operation. A typical adapter would be an NNTP adapter that would enable a GIA component to send messages to a NEWS server without implementing the NNTP protocol.

External Server Applications

External server applications are ready-to-run applications of the server; they are integrated via the previously introduced Adapters. In the following, some examples (NNTP, EMAIL, IRC and HTTP) are described.

NNTP servers manage asynchronous message structures that are sorted with respect to similar content in so-called newsgroups. Additionally, statement- and comment-like message structures are supported, as messages can be posted to an NNTP server in the form of a new message or a response to a message received. The related RFC[6] is the Network News Transfer Protocol (NNTP).

EMAIL servers manage the asynchronous flow of messages between individuals. The corresponding reading and writing RFC protocols are POP and SMTP.

HTTP servers are hypermedia servers which store and distribute information written in HTML. HTTP is a request/response protocol that describes the exchange of information between an HTTP client (a browser) and an HTTP server (a so-called web server).

IRC[7] servers provide services for the synchronous exchange of text messages in so-called channels. When an IRC client enters a channel and posts a text message, the text is immediately transmitted to all of the clients logged in the same channel.

Internal Client Applications

Internal Client components are available in source code and can, as a result, be controlled precisely.

The GIA Main Client (GMC) carries out two main functions. The first is to initiate and hold the primary connection to the GIA Session Manager. In centralized implementations as are sometimes necessary to build client server systems over firewalls, the entire flow of information is routed over the GIA Main Client. The second task of the Main Client is the coordination of the other client components.

The room map view (RMV) Client visualizes a hierarchical view of the information space from the current position of a client. We call the underlying visualization metaphor a room map metaphor, as the client, together with other clients residing in the same location, is drawn in the form of an icon in a corridor. The logically sub-ordered places are visualized as rooms which may be reached via the corridor, together with the icons of the clients residing in those places.

The Notification Client (NC) is a user interface to the Notification Manager, enabling the user to specify rules consisting of (info space) change events and subsequent actions. One example would be to send a specific email, once a new contribution in a NEWS group occurs.

An EMAIL Client allows a user to send messages to other users of the GIA system. As this client is an internal component, this application can be coordinated with the other components, e.g. in a specific help context, the email recipient can be provided by the GIA system.

An NNTP Client enables the GIA user to post messages to the NNTP server. The advantage of an internal client is that it can be controlled by the GIA System, e.g. the current newsgroups can be set according to the groups related to the actual sub-web.

The IRC Client is an interface to the IRC chat server of the GIA system. As it is available in source, it can be coordinated with the user's actions. For example, the IRC channel can be switched according to the user's actual focus of interest.

On the client side, the Adapters basically play the same role as they do on the server side: they are a bridge between the internal components and the external applications. One example is an RMI-based HTTP Adapter that implements a sensor for navigation actions (clicks on links) within the browser and an actor that can navigate the user's browser (load a new url).

External Client Applications

These are available client-side applications used as executables.

[5] RMI: Remote Method Invocation

[6] RFC: Request for Comment

[7] IRC: Internet relay chat: standard for the synchronous exchange of text messages.

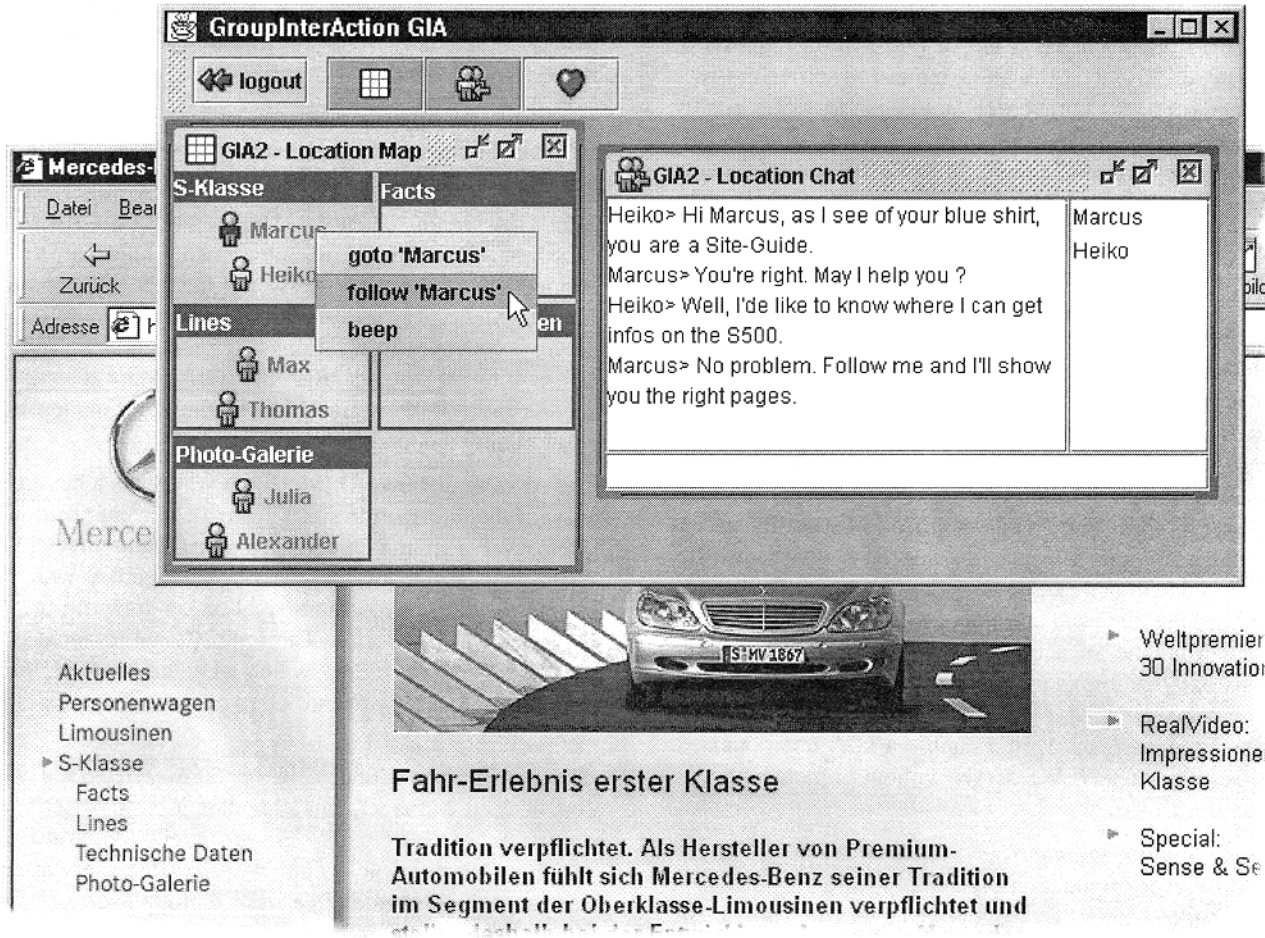

Figure 4: A Screen Shot of the Current GIA Proto

HTTP browsers such as Netscape or Explorer are an example for comparatively complex applications which allow the loading, posting and visualization of html data according to the http standard. Other examples could include office programs or database clients.

Types of Connections Between the Components

Basically there are three different types of connections between the components of the architecture:

1. interactions within one process address room,
2. interactions amongst components in the same local network environment (LAN, no filtering or blocking of connections),
3. interactions amongst components over an internet (WAN, firewalls and routers).

The used DES was developed to support a transparent exchange of events between components in all three cases. However, as shown, the internet protocol-based components do not interact via DES events but directly via the protocols. A problem can arise when, for example, firewalls block connections to specific ports. In this case, tunneling mechanisms must be implemented to enable the communication between client and server.

A PROTOTYPE IMPLEMENTATION OF THE ARCHITECTURE

The current prototypes of the GIA architecture are embedded in the scenarios of customer interaction on a company's corporate web-site and of cooperative learning within companies.

Implemented Functionalities

Currently four group interaction functionalities are implemented in the GIA prototype:

1. **User Awareness**: While the customers browse the corporate web site, they can recognize other customers and corporate staff that share their location or subordinate places. The web users are visualized in a room map metaphor (see Figure 1, upper left). User awareness servers here for making a web site busier and more attractive. It is also a prerequisite for meeting people in the context of virtual communities.
2. **Location Chat**: Users are grouped together with others who share the same location on the web (see Figure 1, right-hand corner). A group can communicate via chat. If a customer navigates to a

different location, his chat group changes to his current location.

3. **Cooperative Navigation**: Customers can link their browsers to another user and follow him or her (see Figure 1, pop up menu in the middle). Individual advice or interactive guided tours can be implemented in this way.
4. **Acquaintance Chat**: Chat groups which are independent of their users' current locations may be invoked. In the case of individual care, a customer can build a communication group with a staff member and discuss open questions. Customers can also be distributed over the web and still remain in contact via an acquaintance chat group.
5. **Discussion Forum**: A GIA user browsing special locations of the web site can invoke a NEWS-like asynchronous discussion forum which allows asynchronous interaction amongst the users of a web-based system. As a result, a user can annotate questions or comments to the page he or she is currently viewing. The creator of the web page or another user of the system can reply to these annotations. When the user navigates in the information space, the new groups are automatically replaced according to the users' current locations.

Implemented Components of the Architecture

Since not all of the concepts of the architecture are used in current prototypes, we describe the components implemented so far and their most important interactions.

Internal Server Applications

The current *Session Manager* is the main internal component of the GIA prototype. In order to make it easier to bridge firewalls, all client / server communication is currently routed via the Session Manager.

The *Group Manager* currently supports dynamic creation of groups and dynamic inserting and deleting of clients. Clients that rely on dynamic group building, for example the IRC client, can register for member changes.

The *Structure Manager* is able to build hierarchical location trees of web content in two different ways. First, manually-edited configuration files can be provided that define a mapping between HTML pages and locations as well as between locations. Second, a directory parser is available that is able to automatically build up location structures if the site has been stored in specific, pre-defined fashion. This parsing functionality of the Structure Server enables uploading of new content and efficient adaptation of the server to the new content.

The *News Adapter* is a proprietary adapter to the news server product that we are currently using. It allows us to configure and manipulate the server beyond the possibilities of the NNTP protocol.

External Server Applications

We are using a standard *WWW Server* to distribute the hypermedia information space which is the common basis of our current prototypes.

Instead of re-implementing the chat functionality, we are using a available *IRC Server* to service the chat clients.

A commercial *NNTP Server* serves as a store for the asynchronous messages of the discussion forums. The implemented NNTP adapter provides additional ways of accessing this product.

Internal Client Applications

The *GIA Main Client* initiates and holds the connection to the session server. All communication between client and server is currently routed over the GIA Main Client. In addition to being an event forwarder, the session server coordinates and configures the other client components.

The *RMV Client* implements user awareness using a room map metaphor. It visualizes all the visitors to a user's current location and its logically sub-ordered places.

The current *IRC Client* implements the IRC protocol and interacts directly with the external IRC server. Whenever a user changes his or her location, the chat channel is switched to the new location. As available IRC clients could not be controlled adequately, this component was re-built as an internal application.

The same holds true for the *NNTP Client*, which updates the news group list to the set of groups related to the current location of the user every the time the user navigates in the information space.

The *Browser Adapter* was implemented as an applet which is embedded in every page of the information space to be considered. The sensor launches an event every time a new page is loaded in the browser. The actor makes the browser load a new page.

External Client Applications

Currently the *WWW Browser* is the only external client application. It is used for loading and viewing the hypermedia information and runs the Browser Adapter, which generates the main events to configure the CSCW functions.

Connections between the Components

Currently the only components that are not connected via the DES are the IRC client and server as well as the NNTP client and server. All the other components communicate via events of the distributed event system. Consequently, firewalls may block the exchange of news and chat messages at present.

Non-Functional Qualities of the Prototype

In addition to the functionality of a prototype, its performance, its reliability and evaluation issues are of interest.

Performance and Reliability

Earlier prototypes of the GIA system were restricted to about fifty users. However, in the customer interaction scenario, it is likely that thousands of parallel users would browse a corporate web site. As a consequence, the architecture had to be able to cope with a situation where, most of the time, many of the clients are not accessible as the network is too busy or the clients' computers are not functioning properly. By switching to a asynchronously-coupled event architecture, we were able to deal more effectively with large numbers of users and unreliable clients. Performance measurements have shown that the server components of the current implementation were able to deal with more than a thousand parallel users. The reason why these numbers would nevertheless fail to work in practice would lie with the too low network bandwidths.

Evaluations

So far, the prototype has been tested in two cases. It was used as a distributed annotation environment for project documents within our research groups, which is similar to an engineering context. Although it was considered to be a helpful approach, problems were documented: It proved to be too great an effort to update the web from standard document formats of word processors, and the synchronous interaction facilities were not used, as the users were available next door. The second evaluation was in the same context, but between real distributed groups. Here also, although asynchronous interaction was deployed synchronous interaction was not investigated as, in small groups, the meeting probability is too low.

RELATED WORK

Related work on cooperative use of the web comes from two sources: CSCW-oriented publications and work on specific application scenarios for cooperative systems.

CSCW-oriented Publications

User awareness is a main criterion for the support of group building and user interaction in groups [3]. WAP [7] is a protocol for user awareness in the web, but it fails to address the group interaction aspect. [2] developed the Internet foyer, which provides awareness exclusively via the definition of an unified entry point. BSCW is an "work space"-oriented web-based collaboration system. The strict restriction to pure HTTP based communication makes it more platform independent than the GIA approach. Its main focus is on asynchronous and document oriented interaction but in the form of the monitor tool it also integrates synchronous awareness. The groupware toolkit GroupKit [13] is a tcl/tk-based approach for creating real-time, distributed computer-based conferencing systems in an easy script-like code. It concentrates on synchronous CSCW. Worlds [3] is a CSCW framework that is based on a meta language specifying dynamic changes that should occur during runtime. *Mushroom* [11] is an Internet-based framework for coordinated, collaborative work on shared resources. Piazza [14] supports coincident communication in virtual communities in addition to planned interaction.

Specific Areas of Application

Electronic commerce is an interesting field for the support of human interaction. The architecture presented is derived from experiences with former prototypes of web-based group interaction systems [5] . Broadvision [9] is an approach that builds on user preferences. Direct interaction on a virtual market-place, however, is not supported.

With the increasing amount of new knowledge, there is a growing demand for organizational learning systems. In the following, a few synchronous and asynchronous *learning systems* are mentioned: "Answer Garden 2" [1] consists of a system architecture for organizational memory and collaborative help support in an asynchronous fashion. Examples for synchronous learning are: "GestureCam" [4], an approach that supports spatial workspace collaboration via a video-mediated communication system. In [6], several advantages of constructing an interactive educational system based on the web are collected.

PERSPECTIVES

Our future work on web-based group interaction will concentrate on the following topics:

As not all of the possible or useful components are currently available, the development and integration of new components is an ongoing task. Examples of currently considered enhancements are components for synchronous multimedia communication (audio or video).

We expect a great deal of improvement potential in the field of the component integration mechanisms. Enhancing the DES to bridge firewalls can become one of the most important tasks for using the GIA system in certain application areas. Another topic would be the development of more sophisticated and generic methods for the integration of external applications than we have now. These approaches should make it possible to integrate new components without having to know or to access the DES mechanisms.

Last but not least, the application and evaluation of our current prototype systems will give major impulses to an improvement of the approach.

CONCLUSIONS

We have presented an architecture for the generic and flexible enhancement of web-based information systems with CSCW functionalities. The GIA architecture integrates domain applications with CSCW systems and presents the cooperation functionalities in a coordinated and, thus more useful, way.

The current GIA prototype was developed within the context of commercial scenarios such as electronic commerce, customer interaction and cooperative learning and will be evaluated, tested and improved in our corporation as well as in external institutions.

REFERENCES

1. Ackerman, M.S., & McDonald, D.W. Merging Organizational Memory with Collaborative Help. ACM Conf. On Computer Supported Cooperative Work (CSCW). Boston, MA, USA, 1995
2. Benford, S., & Brown, C., & Reynard, G., & Greenhalgh, C. Shared Spaces: Transportation, Artificiality and Spatiality. ACM Conf. On CSCW. Boston, MA, USA, 1996
3. Fitzpatrick, G., & Kaplan, S., & Mansfield, T. Physical Spaces, Virtual Places and Social Worlds: A study of work in the virtual. ACM Conf. On CSCW. Boston, MA, USA, 1996
4. Kuzuoka, H., & Kosuge, T., & Tanaka, M. GestureCam: A Video Communication System for Sympathetic Remote Collaboration. CSCW'94. ACM Conf. On CSCW. Chapel Hill, NC, USA, 1994
5. Manhart, P., Schmidt, K., & Ziegler, H. Group Interaction in Web-based Multimedia Market Places, Vol. VI, pp. 645-654, Annual Hawaii Int. Conf. On System Sciences. Hawaii, USA. 1998
6. Nakabayashi, K., & Koike, Y., & Maruyama, M. (1995). An intelligent tutoring system on the world wide web: Towards an integrated learning environment on a distributed hypermedia. ED-Media &ED-Telecom 95 Conference. 1995
7. Palfreyman, K., & Rodden, T. A Protocol for User Awareness on the World Wide Web. ACM Conf. On CSCW. Boston, MA, USA. 1998
8. Wan, D., & Johnson, P.M. Computer Supported Collaborative Learning Using CLARE: the Approach and Experimental Findings. ACM Conf. On CSCW. Chapel Hill, NC, USA.
9. Peppers D., & Rogers M., The One to One future, Doubleday Publishing Group, NY, 1993
10. Hagel J., Net Gain, Harvard Business School Press, 1997
11. Kindberg, T.; G. Couloris; J. Dollimore; and J. Heikkinen. 1996. "Sharing Objects over the Internet: the Mushroom Approach." *Proceedings of the Global Internet*, London, England.
12. Bentley, Appelt, Busbach, Hinrichs, Kerr, Sikkel, Trevor, Woetzel, Basic Support for Cooperative Work on the World Wide Web, International Journal of Human Computer Studies: Special Issue on Novel Applications of the WWW, Academic Press, Cambridge, Spring 1997
13. Greenberg, S., Collaborative Interfaces for the Web. In C. Forsythe, E. Grose and J. Ratner (editors), Human Factors and Web Development, Chapter 18, p241-254, LEA Press, ISBN 0-8058-2823-0, 1997
14. Isaacs, E.A.; J.C. Tang; and T. Morris. 1996. "Piazza: A Desktop Environment Supporting Impromptu and Planned Interactions." *Proceedings of Computer Supported Cooperative Work* (CSCW'96), Cambridge, MA, USA.
15. Eberhardt R. 1998. An Interactive Multimedia Distance Learning System Using Distributed-Object Computing Technology. Proceedings of the IEEE Globecom '98, Sydney, Australia