Asynchronously Replicated Shared Workspaces for a Multi-Media Annotation Service over Internet

Hartmut Benz[1], Maria Eva Lijding[2]*

[1] University of Stuttgart, Institute of Parallel and Distributed High-Performance Systems (IPVR), Breitwiesenstraße 20–22, 70565 Stuttgart, Germany,
benzht@informatik.uni-stuttgart.de

[2] Technical University of Catalonia, Department of Computer Architecture (DAC), Campus Nord D6-008, Jordi Girona 1–3, 08034 Barcelona, Spain, mariaeva@ac.upc.es

Abstract. This paper describes a world wide collaboration system through multimedia Post-its (user generated annotations). DIANE is a service to create multimedia annotations to every application output on the computer, as well as to existing multimedia annotations. Users collaborate by registering multimedia documents and user generated annotation in shared workspaces. However, DIANE only allows effective participation in a shared workspace over a high performance network (ATM, fast Ethernet) since it deals with large multimedia object. When only slow or unreliable connections are available between a DIANE terminal and server, useful work becomes impossible. To overcome these restrictions we need to replicate DIANE servers so that users do not suffer degradation in the quality of service. We use the asynchronous replication service ODIN to replicate the shared workspaces to every interested site in a transparent way to users. ODIN provides a cost-effective object replication by building a dynamic virtual network over Internet. The topology of this virtual network optimizes the use of network resources while it satisfies the changing requirements of the users.

1 Introduction

DIANE (Design, Implementation and Operation of a Distributed Annotation Environment) is a service to create multimedia annotations to every application output on the computer, as well as existing multimedia annotations. DIANE is suitable for every field of application since it is not restricted to be used only with certain programs or application areas. In contrast to normal multimedia authoring tools, DIANE provides authoring on the fly capability to users. It thereby satisfies the users need to efficiently and effectively create short, precise multimedia annotations to every object of day to day use (on the computer). Field tests of DIANE indicate that multimedia annotations are an ideal service to support collaborative work in a wide variety of application areas.

DIANE is a European Union funded ACTS project near its completion. During the project, the service has been extensively tested by users in the areas of medical diagnosis, computer based training, and industrial research collaboration. During the project,

* Working on a grant of Fundación Estenssoro, Argentina.

T. Plagemann and V. Goebel (Eds.): IDMS'98, LNCS 1483 pp. 260-271, 1998.

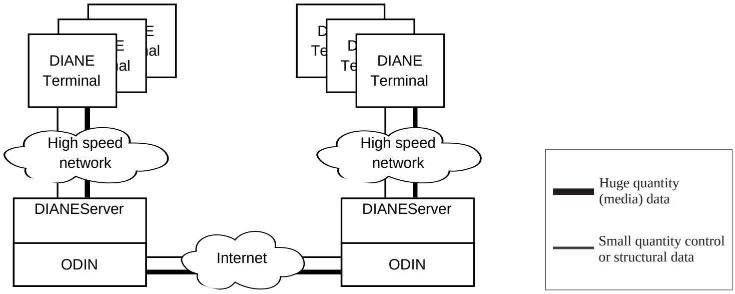

Fig. 1. Principle architecture of connected DIANE sites.

doctors from the pathology department of a hospital evaluated DIANE doing diagnoses, asking colleagues for second opinions, and training students.

However, DIANE requires a high performance network (ATM, fast - dedicated - Ethernet) between the DIANE server and its DIANE clients since it deals with large multimedia object. When only slow or unreliable connections are available between a DIANE terminal and server, useful work becomes impossible. Accessing a DIANE server via Internet on the other side of the globe is possible, but results in poor performance and low quality multimedia presentations.

The current task described in this paper is to overcome this restriction and enable DIANE to be usable between distant sites only connected by low-quality networks (low bandwidth, high latency, high packet loss). Following the current trend and using the available technology, we connect DIANE sites via Internet. This gives us a cheap, simple and world wide reach, but limits strongly the quality of the connection.

We achieve this by introducing *asynchronously replicated shared workspaces (ARSW)* into DIANE using the distributed replication service ODIN over the Internet. An ARSW can be seen as a special type of directory very similar to a newsgroup. It enables users worldwide to see each others documents and create annotations to them. In this paper we use the term *shared workspace* synonymous for the ARSW.

Figure 1 shows the principle architecture of two DIANE sites connected via the replication service ODIN. Each site consists of a set of DIANE client terminals which are connected via a high speed network to a DIANE server. Each connection consists of two channels, an asynchronous control channel to exchange document information and presentation schedules, and a high bandwidth channel for continuous media streams and other high volume data [1]. Two DIANE servers are connected over a low quality Internet connection.

Both DIANE and ODIN have been implemented almost completely in Java making extensive use of standard packages like RMI, SSL, JMF. Only some minor function currently not available via platform independent API like grabbing screen contents and reading the audio device have been realized in Java classes wrapping native code.

The remainder of this section presents two usage scenarios in *Medical Consultations* and *Computer Based Distributed Learning* to illustrate the application of DIANE/ODIN and replicated shared workspaces. Section 2 and 3 give short overviews of DIANE and ODIN. Section 4 describes the semantics of the replicated shared workspaces. Section 5 gives an overview of the architecture and the changes introduced by combining DIANE and ODIN. Section 6 describes the building of the communication topology and the protocols used to replicate objects. Finally, Section 7 concludes the paper and highlights some future work.

1.1 Usage Scenarios

Asynchronous medical consultations between practitioners, specialist, and students of medicine has proven to be an effective means of communication during our test experiences with DIANE. With a network infrastructure being installed in developing areas we wish to provide distant medical institutions access to the experience and technology of important medical centers and universities.

Consider a provincial hospital using software to manage the data of their patients which include images from X-ray, computer tomography, etc. A doctor who is reviewing a diagnosis with this system likes to get third-party opinions on his diagnosis. Using the annotation service DIANE, he records the X-ray image and the most recent physiological data of his patient displayed by several tools of the management software. He adds his diagnosis talking into a microphone and pointing out relevant regions on the image with the mouse. He links the resulting annotation into a shared workspace he uses to confer with a specialized colleague on the other side of the continent or the world.

The colleague may view this document at a convenient time and annotate it in a similar fashion to support or contradict the original diagnosis. These annotations may consist of parts of the original document and other media objects, e.g. images, results from current research, or references to comparable cases from the literature.

Another example is computer based distributed learning which is a cost-effective way to allow individuals, teams, and organizations to manage, share, and develop their knowledge assets and learning capabilities.

Consider a multinational company where instructors prepare a simple multimedia tutorial about the use of a graphical user interface. Using DIANE, the tutorial can be created directly by recording the correct and efficient interaction with the program. The instructor enhances this recording with spoken comments and links to related material. This tutorial is asynchronously distributed by ODIN to all sites of the company where users of the program are located. Now available locally, users can conveniently review the tutorial. Additionally, they can add their own annotations, either for private use, or as a contribution to the multimedia discussion forum of the users of the program.

The participants in each scenario may be distributed all over the world. Since the communication with multimedia annotations is asynchronous, discussions do not suffer from time differences between different parts of the world.

2 Overview of DIANE

DIANE supports various ways of making annotations. Several users can produce annotations on top of the same document. Also, users can create annotations to other annotations, reflecting in this way evolving discussions. Annotations are attachable (can be created on top of) both as discrete and continuous media objects such as images, texts, audio, and mouse pointer movement. Therefore, the annotation itself and its relation to the annotated document have a temporal dimension.

The distinction between document and annotation is primarily semantical. Both are identically structured multimedia objects internally represented by the same class. The distinction is made to highlight the special semantics assigned to the annotation by the user during creation and the fact, that an annotation always refers to the document it annotates, whereas the more general term document does not imply this.

Users of DIANE can freely organize their multimedia documents in directories. This organizational structure is independent of the implicit structure given by the annotation process itself, e.g. document A is an annotation of document B. A user interacts with DIANE by navigating through the hypermedia structure (supported by several types of visualization), selection of documents for replay, recording new documents, and linking them into the workspace document structure.

DIANE implements security with authentication, role based authorization, and encryption of communication channels. This ensures that only authenticated users can interact with DIANE and that they only access documents appropriate to the users defined rights. Normally, only the owner of a document or directory has access to its contents.

DIANE does not allow documents to be modified after creation. This very severe restriction is necessary because of the *annotates* relation. An annotation A to a document B incorporates a presentation of B. Since A is likely to refer to the contents of B (e.g. spatially by pointing to elements of B or temporally by repeatedly pressing pause, resume, fast-forward, or fast-backward on the presentation of B). Modifying the contents of B will in almost all cases render the annotation A completely useless. The problems of keeping temporally and spatially anchored annotations consistent over changing objects has been judged too complex to solve in the DIANE project.

The objects managed in DIANE (documents, annotations, and directories) are versatile large multimedia objects with a complex internal structure. The DIANE document, for example, aggregates several organizational attributes (creator, creation date, access rights, etc.) and a hierarchically aggregated multimedia content. For structural and efficiency reasons the objects, the hyperlink graph structure, and the large multimedia raw data are stored in separate specialized databases.

Figure 2 shows an example of a document structure. It shows a directory (*Dir*) containing a document (*Doc1*) annotated by another document (*Doc2*). Each document consists of several aggregated media objects. A special media object (*VCR*) is attached to the link (*Link*) which includes all manipulations made to the annotated document during recording of the annotation (e.g. pause, resume, ff, fb, scroll).

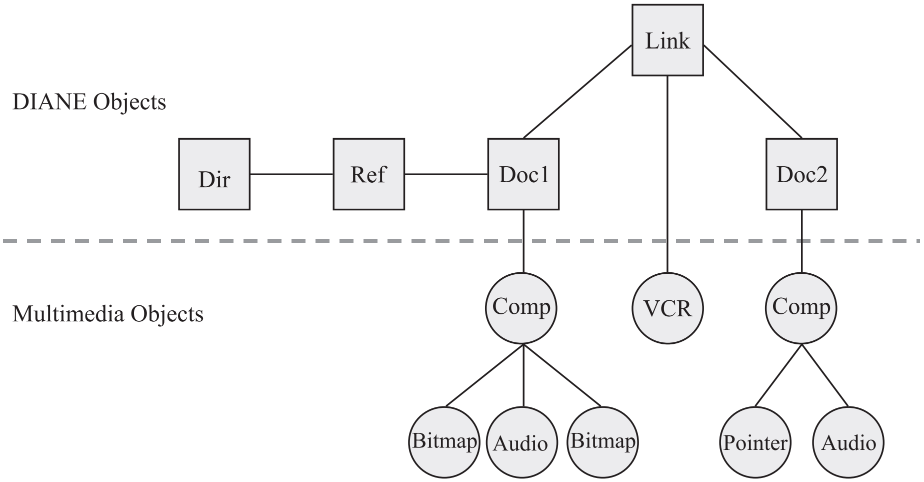

Fig. 2. The Document Model of DIANE.

3 Overview of ODIN

ODIN (Object Distribution Intelligent Network) is a service to replicate arbitrary objects over the Internet in a cost effective way. ODIN, thereby, provides a virtual network over Internet. Many virtual networks can be set up and managed independently from each other. The membership of sites to a network can be controlled to keep it private if desired. Once a site belongs to a network it can access all objects distributed in that network. ODIN is a refinement of ODS presented in a previous paper [4].

Each virtual network can contain several disjunct distribution chains. A distribution chain provides an optimal path for the replication of objects in a shared workspace. It minimizes the use of network resources based on the subscription matrix between users and shared workspaces, the preferred replication schedule of each distribution chain and site. It also adapts to the state of the underlying network. Distribution chains ensure that every object generated in a shared workspace is replicated to all sites subscribing to it. The replication process is transparent to the users which gain a service with high availability of the replicated objects and - except from replication delay - short response times, because objects can be accessed at their local servers.

ODIN follows the replication model from USENET, providing replication of objects with high read/write ratio. As in USENET objects are read-only, but in ODIN they can be updated (creating a new version) on the site the object has been created at. However, the main difference between ODIN and USENET is that USENET uses a static distribution topology where each link must be configured by a system administrator. Therefore, this topology is primarily governed by administrative forces. Mailing list, on the other hand, do not use distribution topologies at all but use random flooding when distributing objects.

The replication process in ODIN is asynchronous and can therefore easily cope with a wide range quality of network links, varying load situations, and interrupted transport connections. Replication is performed by a connection-oriented application layer protocol named *Data Forwarding Protocol (DFP)*. DFP uses UDP for its control channel and TCP to distribute the data objects itself. The TCP connection is enhanced with transaction control allowing to detect communication failures and interrupted connection and restart transmission from the last byte transfered correctly whenever the network allows. DFP is described in Sect. 6.2 in more detail.

4 Asynchronously Replicated Shared Workspaces

In DIANE, a *shared workspace* is a set of documents and directories which are known to and accessible by more than one person. Each shared workspace has a base directory which is a regular DIANE directory to which documents, annotations, and subdirectories are added explicitly by the users. All objects in the base directory and its subdirectories belong to the shared workspace (transitive closure of the *part of* relation). Additionally, all documents annotated by objects of the shared workspace also belong to it (transitive closure over the *annotates* relation).

Compared to existing CSCW systems like [3, 5, 2] we use a very simple model for shared workspaces that does not realize any of the advanced features like event recording and notification services. It was considered to be sufficient within the DIANE project following the requirements analysis of our users. Nevertheless, system design and implementation are prepared to easily incorporate these features.

A directory becomes a shared workspace when its owner sets the access rights of the directory and its recursive contents to allow access to other users and makes the directory known to the future participants (e.g. by mailing a reference to it or publishing it in an agreed upon public directory).

Figure 3 shows an example shared workspace with its base directory (*WS-Base*) and some documents (*D1, D2, D3*) shared by two users A and B. Objects accessible by both users are shaded. The left side shows the integration of the workspace in user As private directory structure. The right side shows this view for user B. Note, that document *D2* is not a member of the base directory but is part of the shared workspace since it is annotated by *D3* and, therefore, is necessary for the presentation of *D3*. Documents *D4* and *D6* are private to the users A and B respectively, since they are not part of (the presentation of) any shared document.

Participation in a shared workspace described so far (and implemented in the DIANE project) is restricted to the DIANE server the workspace has been created on. In order to access it, a user has to connect to the correct server. Effective participation in a shared workspace, though, is only possible using a high performance network (ATM, fast Ethernet) since DIANE deals with large multimedia objects and continuous stream based presentations. When only slow or unreliable connections are available between a user' s DIANE terminal and server, useful work becomes impossible.

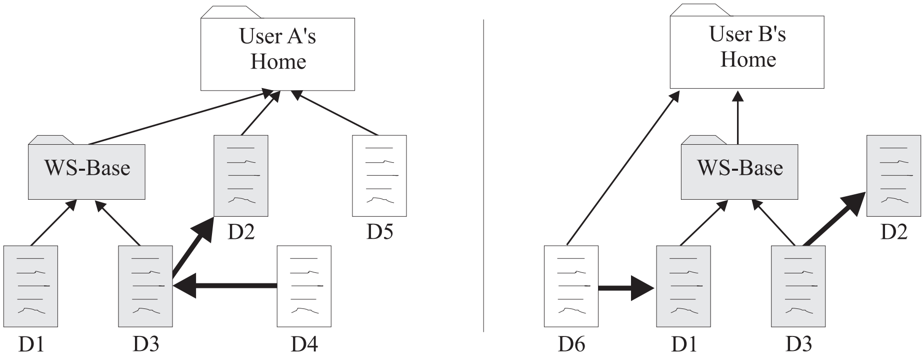

Fig. 3. Example (asynchronously) shared workspace of users A and B. Grey objects are part of the shared workspace and are accessible by both users, white objects are private. Thick arrows represent the *annotates* relation, thin arrows the *part of* relation. Left and Right side shown may be on a single DIANE server or replicated on several DIANE/ODIN connected sites.

To overcome this restriction *asynchronously replicated shared workspaces (ARSW)* are introduced which allow the objects of a shared workspace created on one server to be accessible on other servers. A user may now use or install a DIANE server in a local, sufficiently fast environment. Since the replicated documents are available locally, the access quality a user experiences is now the same as using a non-replicated DIANE service over a high bandwidth connection.

Asynchronous replication between sites is transparent to the user except for the replication delay. This is similar to the semantics used in USENET where each user just sees the documents at the local site. Therefore, the two user directories shown in Fig. 3 can as well be on different DIANE sites. In contrast to USENET, a DIANE user will never encounter a (temporal) inconsistency in the available documents, as regularly happens in USENET when a reply becomes available at a site prior to its source. The replication service ensures that an annotation only becomes available simultaneously to or after the annotated document is available.

The users ability to remove objects from a shared workspace or later restrict its access rights follows the newspaper paradigm: once you have shared an object and someone actually is interested in it (e.g. annotates it, links it to a private directory), you cannot delete it any more. A user can attempt to remove an object from a shared workspace at any time but this will fail as long as other people still reference it. Only if no more references exist the object is actually removed from the shared workspace. Restricting the access rights of a shared object is not allowed at all.

5 Architecture

The architecture of DIANE can roughly be divided into four quadrants (Fig. 4). The vertical separation follows the classical client/server paradigm. The horizontal separa-

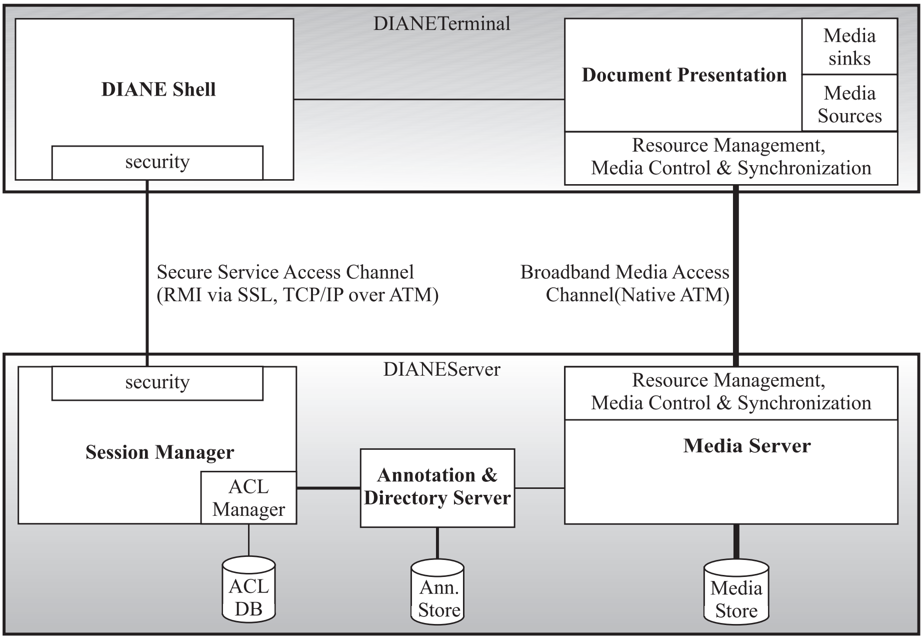
DIANETerminal
DIANE Shell
security
Document Presentation
Media sinks
Media Sources
Resource Management, Media Control & Synchronization
Secure Service Access Channel (RMI via SSL, TCP/IP over ATM)
Broadband Media Access Channel(Native ATM)
DIANEServer
security
Session Manager
ACL Manager
Annotation & Directory Server
Resource Management, Media Control & Synchronization
Media Server
ACL DB
Ann. Store
Media Store

Fig. 4. DIANE annotation system architectural blocks (without ARSW extension).

tion distinguishes between broadband multimedia transport using streams (multimedia access channel) and narrowband object transport using RMI (service access channel). This distinction reflects the documents model which also separates between DIANE object structure (documents, annotations, and directories) and its multimedia contents.

The raw media data are stored on the MediaServer via direct broadband stream connections between the media sources and sinks (right half of Fig. 4). The document, link and directory objects and their hypermedia graph structure are stored in the Annotation&DirectoryServer. The login, session management, navigation, security, object relocation and loading functionality are realized in the DIANEShell in the client and the SessionManager as its counterpart on the server.

The architectural changes resulting from the introduction of ARSWs are restricted to the DIANEServer (Fig. 5) which is extended by two components of the ODIN replication system: NetworkRouter and ObjectReplicator. The existing DIANE components remain unchanged because the new components can use their interfaces as they are. The replicated objects (documents, directories, media data) remain unchanged since ODIN treats them as unqualified objects (binary large objects, BLOB).

On each site, two special system directories `/Available` and `/Subscribed` list the available ARSW in the network and those the site has subscribed to. The operations to create, subscribe, unsubscribe, and remove shared workspaces have been mapped to existing operations of manipulating directories. To subscribe to an available ARSW a

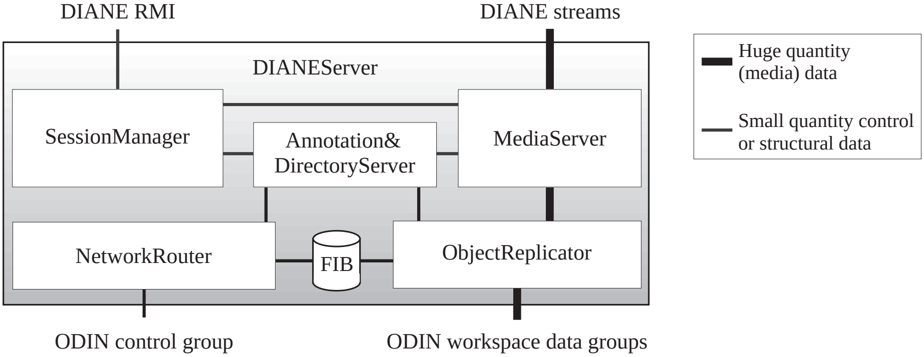

Fig. 5. Principle architecture of the extended DIANEServer.

user links it into `/Subscribed`. To create a new ARSW a user creates a new directory in `/Subscribed` which automatically becomes visible in `/Available`. A site administrator can easily restrict usage of ARSWs by setting the access rights of these system directories appropriately. To selectively prevent the import of undesired ARSWs the system administrator can adjust the read access rights of the unwanted ARSWs in the `/Available`.

The NetworkRouter monitors operations on shared workspaces by observing the contents of the directory `/Subscribed`. In this way it knows which shared workspaces the site participates in and forwards this information to other NetworkRouters (cf. Sect. 6.1). It, furthermore, learns about new shared workspaces created at its site, shares this information with its peers and publishes the complete list in `/Available`. It also builds the *Forwarding Information Database (FIB)* which is used by the ObjectReplicator and contains the sites it must exchange objects with for each of the shared workspaces the site participates in.

The ObjectReplicator collects DIANE objects, raw media data, and relevant hypermedia graph structure and aggregates them into a uniform ODIN objects. These are distributed according to the information provided in the FIB. Likewise, the ObjectReplicator inserts the objects received from other servers into the respective databases. The complexity of DIANE's objects is hidden from the replication service by the ObjectReplicator. Raw media data collected from DIANE already is compressed according to data type. In addition, the ObjectReplicator allows the transparent incorporation of compression schemes.

6 Inter-site Communication

The communication model used between sites is *group multicast*. Each shared workspace is associated to a process group called *workspace data group* which acts as the logical destination for multicast messages. ObjectReplicators belong to the process groups corresponding to the shared workspaces indicated in `/Subscribed`. There is also a special process group called *control group* to distribute control information over the network. All NetworkRouters communicate through this process group.

We implement group multicast in each process group by store-and-forward of objects between neighbouring sites in the distribution chain of the group. The next section describes the building of the distribution chains, the following subsection describes the object distribution inside a workspace data group.

6.1 Distribution Chains

The NetworkRouters build a distribution chain for each workspace data group in order to implement group multicast communication with optimal communication cost. A distribution chain guarantees that every object generated in the respective ARSW reaches all sites in the workspace data group.

To build distribution chains we use routing mechanisms that can adapt to the traffic's service requirements and the network's service restrictions. Network routing, as a basis, ensures that packages take the best path between a given source-destination pair. In addition, our routing decides on the optimal source-destination pairs to build a distribution chain. This routing mechanism uses the same principles as network routing. All communication networks share a core of three basic routing functionalities [7]: assemble and distribute network and user traffic state information, generate and select routes, and forward user traffic along the routes selected.

To aggregate the network state information and obtain the communication topology of the control group we selected topology-d [6]. Topology-d is a service for applications that require knowledge of the underlying communication and computing infrastructure in order to deliver adequate performance. It estimates the state of the network and networked resources for a group of given sites. To join a group, a machine sends a request to a master. With these estimates it computes a fault tolerant, high bandwidth, low delay topology connecting participating sites. This topology is to be used by the control group for assembling user state information.

The functionality of generating and selecting routes is replicated in each site. Given the network and user state information each NetworkRouter computes the distribution chains independently for the shared workspaces the site belongs to. As a mechanism to provide stronger consistency of these computations, only important changes in the network state information must be taken into account.

The algorithm to compute a distribution chain is very simple. A minimum spanning tree is generated with all the sites in the workspace data group. Since sites have unique identifiers that can be ordered with lexicographic order, whenever there are two nodes that may be incorporated to the minimum spanning tree, we choose the one with lower identifier.

The last routing functionality, forwarding of user traffic, is carried out by the ObjectReplicators using DFP in each site. This protocol is described in the next subsection.

6.2 Data Forwarding Protocol

The *Data Forwarding Protocol (DFP)* is the connection-oriented application layer protocol used between adjacent sites in a distribution chain. DFP is optimized to transport large objects over unreliable low bandwidth connections.

The protocol consists of the notification of available objects and the actual transfer of objects. A site with a new object - either from the distribution chain or created locally - immediately notifies its neighbours in the distribution chain using an acknowledged UDP message. Neighbours later request the objects to be transferred at a convenient time (e.g. low traffic hours, low workload, fixed time, more than N objects to transfer) When an object is announced by several neighbours simultaneously the site chooses the one with the best connection. Objects successfully received are treated as new objects.

The object transfer is realized as a transaction over a TCP connection that deals with communication failures, interrupted connection, and restarts transmission from the last byte transfered correctly whenever the network allows.

Each site keeps a database with information about the protocol-state of its neighbours which consists of the objects the neighbour has and the announcements made to it. When a site is assigned a new neighbour in a distribution chain, it starts a synchronization phase by sending it announcements for every object in the workspace data group.

7 Conclusions and Future Work

The paper describes a simple and effective way to extend the distributed annotation environment DIANE with asynchronous replicated shared workspaces to enable collaborative work with DIANE over low quality networks.

DIANE has proved to be an ideal service to provide support for collaborative work through the use of multimedia post-its. Due to its network requirements effective collaboration is limited to high performance networks, even though the collaboration paradigm is asynchronous. Recently, users extended their requirements in order to be able to collaborate beyond the local networks using DIANE over Internet.

ODIN, also based on the paradigm of asynchronous collaboration proved to be the ideal extension to DIANE to satisfy the new user requirements. Another very important benefit is that both systems are almost completely implemented in Java which eases integration very much.

In the future, we will analyze the introduction of semi-public shared workspaces, where subscription is restricted to a subset of sites, and the security issues this will raise. It will, furthermore, deal with managing uniform access rights over multiple DIANE sites.

Work continues on improving the procedure for building distribution chains for better scaling capability. We are looking into other publishing paradigms and their consequences which, for example, allow users to 'take back' (delete) or modify documents and annotations they made.

References

1. H. Benz, S. Fischer, and R. Mecklenburg. Architecture and implementation of a distributed multimedia annotation environment: Practical experiences using java. In H. König, K. Geihs, and T. Preuß, editors, *Distributed Applications and Interoperable Systems*, pages 49–59. Chapman & Hall, Oct. 1997.
2. L. Fuchs, U. Pankoke-Babatz, and W. Prinz. Supporting cooperative awareness with local event mechanisms: The groupdesk syste. In H. Marmolin, Y. Sundblad, and K. Schmidt, editors, *Proceedings of the 4th European Conference on Computer-Supported Work, ECSCW'95, Stockholm, Sweden*, pages 247–262. Kluwer Academic Publishers, Sept. 10–14 1995.
3. A. Gisberg and S. Ahuja. Automatic envisionment of virtual meetin room histories. In *Proceedings ACM Multimedia '95*, pages 65–75, San Francisco, CA, USA, Nov. 1995. ACM Press, ACM Press.
4. M. E. Lijding, L. Navarro Moldes, and C. Righetti. A new large scale distributed system: Object distribution. In *Proceedings of the Thirteenth International Conference on Computer Communications - ICCC'97, Cannes, France*. International Council for Computer Communication, Institut National de Recherche en Informatique et en Automatique, INRIA, Nov. 1997. Annex.
5. S. Minneman, S. Harrison, B. Janssen, G. Kurtenbach, T. Moran, I. Smith, and B. van Melle. A confederation of tools for capturing and accessing collaborative activity. In *Proceedings ACM Multimedia '95*, pages 523–534. ACM, ACM Press, Nov. 1995.
6. K. Obraczka and G. Gheorgiu. The performance of a service for network-aware applications. Technical Report 97-660, Computer Science Department - University of Southern California, Oct. 1997.
7. M. Steenstrup, editor. *Routing in Communication Networks*. Addison-Wesley, 1995.