Abstract. Knowledge management is often viewed as a structured process of eliciting knowledge, storing knowledge, and later retrieval by individuals. In this paper we argue that knowledge management should be seen as a dynamic process—an interaction between experts. Therefore, environments should support the cooperative management of information in workgroups or online communities. We start with a motivation for this cooperative perspective of supporting knowledge management through support for the creation and exchange of knowledge in communities. We introduce the CYCLADES environment—an open cooperative virtual archive environment based on open archives. We report in detail on the requirements analysis and functional design, on the specification, on the implementation, as well as on the evaluation of the environment. Finally, we draw conclusions for the design and implementation of cooperative knowledge management.

Subject Categories and descriptors:

H.1 [MODELS AND PRINCIPLES]: E.4 [Formal models of communication]; H.3.3 [Information Search and Retrieval]; H.2 [Database Management]

General Terms: Knowledge management

Keywords: Information Management, Database Management

Received 30 Oct. 2004; Reviewed and accepted 30 Jan. 2005

1. Introduction

Knowledge management is often viewed as a structured process of eliciting knowledge, storing knowledge, and later retrieval by individuals. In this paper we argue that knowledge management should be seen as a dynamic process—an interaction between experts. We would like to begin with some epistemological considerations. A thorough discussion of epistemology would go beyond the scope of this position paper; nevertheless, we need to clarify some basic notions: in the context of this paper data is seen as raw; information is data with meaning; and knowledge is verified expertise. Therefore, environments should support the cooperative management of information in workgroups or online communities. We start with a motivation for this cooperative perspective of supporting knowledge management through support for the creation and exchange of knowledge in communities. We introduce the CYCLADES environment—an open cooperative virtual archive environment based on open archives. We report in detail on the requirements analysis and functional design, on the specification, on the implementation, as well as on the evaluation of the environment. Finally, we draw conclusions for the design and implementation of cooperative knowledge management.

1 The research work underlying this paper was done while the authors were working at Fraunhofer FIT.

human. It is important to also see the ‘knowing’ and not only the ‘knowledge’ (Cook & Brown, 1999). Therefore, sharing and exchanging knowledge and expertise cannot be automated totally. Twidale and associates (1997) identified several tactics of knowledge seeking. Information seekers can consult colleagues and ask them for information and references; they can wander around to search for information and hope to meet others coincidentally; they can do brainstorming to generate numerous ideas; or they can bibble—that is, they can look for existing bibliographies to the same or a similar topic (this strategy is often used on the WWW, when users use the bookmark list of other users with similar interests).

The results of the above studies have clear implications for the design of systems that aim at supporting the flexible sharing and exchanging of knowledge:

- First, technology has to extend the reach of the single knowledge worker; so, she can easily and flexibly contact other persons and share and exchange knowledge when needed. Flexible switching between solitary work and information seeking and spontaneous contacts and dialogues with others are a core requirement for flexible sharing and exchange of knowledge. Kuhlthau (1991) developed the notion of zone of intervention as an area in which users are very efficient in gathering the information needed. She defines a zone of intervention as: ‘that area in which an information user can do with advice and assistance what he or she could not do alone’.

- Secondly, technology needs to empower communities of knowledge workers to establish a common knowledge base over time—often called community memory. Marshall and associates (1994) argue that ‘when people work together—whether in designing a product, or creating training materials from video-based documentation, or writing a coherent analysis of a complex situation in the world—they require, and put effort into constructing and maintaining, shared understandings of what they are doing’. They define a community memory as an ‘open-ended set of shared interpretations and understandings developed and maintained by the group’.

Existing systems typically only provide a part of the functionality needed. For instance, some systems support the information retrieval of online literature such as the digital libraries of the ACM and of IEEE for computer science, Math-Net for mathematics (Groetschel, 2004), or arXiv for physics (Webmaster, 2002). Other systems such as Wikis (Wiki, 2002) or BSCW (FIT, 2004) allow users to share and exchange documents in groups or communities, but do not provide functionality for information retrieval of online literature. In general, most of these systems are designed and implemented as stand-alone, closed applications.

Subsequently, we introduce the CYCLADES environment, which is based on this cooperative approach to sharing and exchanging
knowledge. The CYCLADES environment combines functionality for information retrieval and for sharing and exchanging information in groups and communities. The CYCLADES environment is an open collaborative virtual archive environment that supports knowledge workers with functionality for searching and browsing in pre-print (e-print) archives as well as for sharing and exchanging their findings in virtual communities. It is compliant to the Open Archive Initiatives (OAI) standard—a technical and organisational framework facilitating the discovery of content stored in distributed e-print archives (OAI, 2004)—and can connect to any e-print archive. We report on the process and results of the requirements analysis and functional design, and on the process and results of the specification. We then describe the implementation of the CYCLADES environment. Finally, we report on a thorough user evaluation of the system and draw some conclusions.

2. Requirements Analysis and Functional Design

In this section we describe the approach for gathering requirements for the CYCLADES environment and the resulting functional design. We report on the requirement analysis approach, and give a general overview of the CYCLADES functionality.

2.1 Requirements for Cooperative Information Management

The initial requirements for the CYCLADES environment were specified by a multi-disciplinary and distributed team with backgrounds in digital libraries, databases, information retrieval, Web-based systems, as well as computer-supported cooperative work and virtual communities. These initial requirements together with some questions for additional requirements were put into a Web-based user requirements questionnaire. The CYCLADES requirements were presented to prospective users worldwide, and their feedback was captured. The results of the questionnaire were input for the final specification of the CYCLADES environment. Some examples of questions from the questionnaire are: ‘Would you rate documents so that other users with similar interests can profit from your ratings?’, ‘What kind of information about other users would you like to receive?’, ‘What kind of information would you be willing to share with other users?’, or ‘What information would be suitable in describing a new community that is recommended to you by the system?’. Answer options for each question were given, also if the user was expected to select only one or multiple options. Based on the results of this survey the functionality of the CYCLADES environment was specified. The most important information could be gained regarding details of the functions, where the developers saw different options and were unsure which one to implement. For example rating facilities were included; annotations were indicated by icons and only fully displayed on request according to the wishes of the users.

2.2 Functionality

The CYCLADES system offers a broad range of functionality for both individual scholars, who want to search and browse in open archives as well as for communities of scholars who want to share and exchange search results. The CYCLADES system integrates a set of functionalities that support the user when accessing very large virtual e-print archives with: functionality for efficient and effective retrieval of relevant information from many large, distributed and multi-disciplinary digital archives; feedback on the degree of relevance of the retrieved information; regular information about new publications in the archive environment which are relevant to the users’ interests; automatic retrieval of users’ long-term information needs (user profiling); as well as rapid dissemination of the search results world-wide. A special set of features provides communities of scholars with functionality for: the dissemination of relevant information to community members in the form of recommendations, which are based on collective profiles and behaviour; very quick on-line annotations on research results published by members of the community; carrying out community services such as peer review, which requires the annotation of online manuscripts by reviewers, and the sharing of these annotations among editors, authors, and others; as well as enabling community members to learn from, contribute to, and collectively build upon the community’s knowledge.

The functionality of the CYCLADES environment is provided by various services (Gross, 2003):

- The Mediator Service (MS) plays a central role among the services. Every other service that is integrated in the system registers with the MS. Whenever a service communicates to another, the MS is queried for the appropriate service and address. This provides flexibility and transparency to the distributed system. The MS also manages the registration and login of users to the CYCLADES system.
- The service responsible for accessing all connected archives is the Access Service (AS). It harvests, indexes and retrieves metadata records from the underlying archives. It builds the bridge to the archive environment and makes a huge amount of distributed knowledge stored in many digital libraries accessible to the CYCLADES users.
- The Collaborative Work Service (CWS) provides a folder-based environment for managing metadata records, queries, collections, external documents, and annotations. The CWS supports users with functionality for the management of folders and their contents and for cooperation among community members, including special awareness features reflecting the groups’ activities in the shared information space.
- The Rating Management Service (RMS) handles ratings of items stored in the CWS and provides that information to the users as well as via a query interface to other services of CYCLADES.
- The Search and Browse Service (SBS) supports searching for records from the various collections, formulating search queries, and browsing the resulting metadata records.
- The Filtering and Recommendation Service (FRS) provides personalised filtering of queries and query results as well as recommendation of relevant records, collections, users and communities. The FRS analyses a user’s behaviour and data, generates a user profile describing the user’s interests, and provides information according to this profile.
- The Collection Service (CS) supports the creating and editing of collections. Collections partition the information space according to the users’ interests and make the individual archives transparent for the user. The CS allows each user to create and maintain a personal list of collections consisting of a sub-set of all the collections available.

Subsequently we describe the Collaborative Work Service in some more detail, since it is the core service of the CYCLADES environment for the community support.

In order to use the CYCLADES environment and its functionality users have to register with the system. As the CYCLADES user interface is Web-based users simply enter a valid email address in the CYCLADES registration Web page. A registration email is sent to this address containing regis-tration information. After the user has carried out these steps, the user is registered to CYCLADES and has a login name, password, and (empty) home folder.

Users can then log in in order to access their home folder or add some additional user information such as full name, affiliation, postal address, phone and fax numbers, and so forth. Furthermore, users can start working with folders and folder contents. They can create private folders, community folders, as well as project folders and add subfolders and documents to them. Private folders and community folders can contain metadata records and queries (metadata records are data sets that can be retrieved from the open archives and typically contain information on the publications such as title, author names, publication date, etc.); project folders additionally can contain any type of document (e.g., PDF-files, slides, text files). The contents of private folders can be seen, accessed, and changed only by its owner. The contents of community and project folders can be seen, accessed, and changed by the community and project members respectively. Users can add documents in various ways: they can store records they found in a search and browse activity, they can upload documents from the local hard disks, they can add links, and so forth.

Figure 1 shows the user interface of the CYCLADES CWS; in particular, the home directory of the user ‘tom.gross’ is shown. Several folders can be seen: ‘My Private Folder’ is a private folder, which is not shared with other users. The folders ‘New Users’ and ‘Physics – Astrophysics’ are community folders, where the user ‘tom.gross’ is a member, but which were created by other users (‘wirsam’ and ‘silvio.caizel’ respectively). The open menu shows
how easy it is to create new folders by simply selecting the respective entry in the File-menu.

In order to become member of an existing community users have two options: either they browse a list of names and descriptions of open communities and subscribe to them according to their interests or they get invited by community managers of closed communities. After a user has become a member, the community folder is visible and accessible from the user’s home folder. If a user leaves a community, the community folder is removed from her home folder, but is still available for the other community members. Only when the last community member leaves the community, the community folder is actually deleted.

For each user the Filtering And Recommendation Service (FRS) generates recommendations about other users, communities, records, and collections, which are stored into special recommendation folders. Users can see these recommendations in the CWS, and can contact recommended users, join recommended communities, store recommended records, and add recommended collections to their personal set of collections.

All documents can be rated on a simple scale. The median of the ratings is then attached to the document and visible for any user who has access to the respective document. Additionally, annotations containing free text can be added to documents. Annotations can also be seen by any user who has access to the respective document.

The CWS provides two features for informing users about activities in the CYCLADES system: event icons and activity reports. Event icons are attached to the individual documents and show recent actions that were performed on the documents (e.g., creation, change, read). Activity reports are sent out daily via email; they contain information about changes to documents since the last report was sent out.

### 3. Specification

The formal specification of the overall CYCLADES system was done with a light-weight variation of a UML approach (Berners-Lee et al., 1999; Fowler & Scott, 2000). That is, several modelling techniques such as use cases, interaction diagrams, and class specifications were used, but the syntax was simplified. This simplification was possible, because of several face-to-face meetings of the distributed design and implementation team and was chosen in order to reduce the effort for the specification. We will present some examples of use cases, interaction diagrams, and class specifications below.

#### 3.1 Use Cases

A use case can be defined as: ‘a set of scenarios tied together by a common user goal’, while a scenario is: ‘a sequence of steps describing an interaction between a user and a system’ (Fowler & Scott, 2000, p. 389). Since the CYCLADES environment offers a broad range of functionality for single-users and for communities, a considerable number of use cases had to be developed. Therefore, verbal descriptions of the use cases were written and, in some cases, combined with activity diagrams.

The use cases were grouped into use cases for the single user, use cases for a community, use cases for a project, as well as use cases for collection management, and use cases for archive some special use cases related to management activities such as

---

**Register as new user** (the user gives her email address and in turn gets a user name and password with which she can login)

**Login** (the user enters her user name and password. After login, she is presented with her home folder)

**View and edit user information**

---

For the first step—Register as new user—an activity diagram was drawn; the latter two steps were only described verbally. Figure 3 shows the corresponding activity diagram, which shows further details for the first step.

---

**Create a new community folder** (in her home folder, given the appropriate rights, the user can create a new community by creating a community folder; she enters a description for the new community and defines if other users can subscribe without or only with invitations; as creator of a community, she automatically gets the rights to administrate the community and thus becomes an administrator of the community)

**Invite and kick out members of the community** (the administrator of a community can invite other users, kick out other users, and define access rights for members)

---

For this use case and its two steps no activity diagrams were produced, but further verbal specifications were done. Figure 5 shows an example of the step invite new members.

---

**Invite new members**

The administrator hits the ‘Add member’ function in the home folder of the community. She is presented with a form where to specify the invitees and the message with which the invitees are invited.
There are two alternatives for specifying a new member to invite; either by user name for already registered users or by email address for a future member that still has to register. The invitation of an existing user has the effect that the community home folder appears as a subfolder of the invited user’s home folder. The invitation of a new user results in a similar procedure as if this person had tried to register herself (cf. use case CO 3-1 ‘Register after invitation’). There are also two alternatives for the message to be sent to the invitees: it may be an automated message for which the administrator has to select the appropriate language, or it may be a specific message to be specified by the administrator herself. In the latter case she may also have this message only be sent to the invitees who are no registered users or to all invitees.

By submitting this form, all invitees become members of the community. Non-registered invitees will receive an invitation to register. Invitees who are already registered users will have the community home folder with all its subfolders in their personal home folder.

![Figure 5 Invite new members step](image)

### 3.2 Interaction Diagrams

Interaction diagrams are: ‘models that describe how groups of objects collaborate in some behaviour’; they typically ‘capture the behaviour of a single use case’ and ‘show a number of example objects and the messages that are passed between these objects within the use case’ (Fowler & Scott, 2000, p. 67). As the CYCLADES services have been developed at different sites in different countries, we decided to use interaction diagrams mainly for modelling the communication that occurs in a use case, in which different services are involved. Basically interaction diagrams can be drawn as sequence diagrams or as collaboration diagrams; in sequence diagrams: ‘an object is shown as a box at the top of a dashed vertical line’, and in collaboration diagrams: ‘the example objects are shown as icons’ (Fowler & Scott, 2000, p. 67f).

As an example we show the interaction diagram for the invitation of a user to a community or project folder. The invitation involves the mediator service, and the collaborative work service. This interaction diagram is particularly interesting, because it shows a cooperative process with two users involved—the inviter, and the invitee. Figure 6 shows the corresponding interaction diagram.

![Figure 6 Invitation interaction diagram](image)

### 3.3 Class Specifications

The specification of the classes had to be very flexible, since the individual services were developed at different partner sites, and each partner was using their preferred programming languages and environments. We, therefore, decided to have simple verbal descriptions of the signatures of the classes of the individual services. Figure 7 shows the beginning of the description of the signatures of the collaborative work service.

![Figure 7 Signatures of the collaborative work service (excerpt)](image)

### 4 Implementation

The archive environment is composed of a large number of heterogeneous, multidisciplinary archives and supports interoperability between them (Gross, 2004). It is based on the Open Archive initiative (OAI, 2004). As has been said above, the OAI standard is a technical and organisational framework facilitating the discovery of content stored in distributed e-print archives. The different services communicate over a Web services interface that is based on standardised Internet communication technology (XML-RPC) (UserLand Software Inc., 2003). This architecture allows the different servers to be optimised to achieve best performance in the specific task for which it is responsible. Highly time-consuming operations (e.g., re-indexation of archives) can be performed in the background without decreasing the overall system performance. This specification also gives excellent flexibility in the adaptation, updating or optimisation of services or the integration of a new functionality.

The different parts of software (CYCLADES services) are running on four different servers located at the different locations (cf. Figure 8). The services are very heterogeneous concerning the base technology used—this is due to the fact that for each service the ideal based technology is used. The Access service, for example, is operated on a Java based platform (Sun, 2005) and uses a PostgreSQL database (PostgreSQL Global Development Group, 2005) as storage backend and indexing infrastructure. This specific service regularly rebuilds its internal indexing database according to new articles found in the underlying OAI compliant archives. Because this task is very complex and resource consuming, it is performed while the system is running on low capacity. The Collaborative Work Service is operated on a UNIX platform and is programmed in Python (Python, 2005).

The interfaces of the different services were implemented as an API that can be accessed by all other services. The method names and the parameters were encoded using XML-RPC. This allows providing all functionality to any other service independent of the underlying
programming language. For instance, the Collaborative Work Service (CWS) notifies the Filtering and Recommendation Service (FRS) when community folders are created or deleted, or when recommendation preferences are changed via the FRS API methods addFolder(), deleteFolder(), setRecommendationPreferences(). This kind of remotely invoking activities form one server to another via a standardised (HTTP compliant) transportation channel is a variant of Web services. Although SOAP (He, 2003) is more frequently used to transmit remote procedure calls, XML-RPC has the advantage to produce less data traffic overhead. Web services components like WSDL and UDDI, which are part of Web services, but are missing in XML-RPC, are not needed in the CYCLADES environment, because the localisations and purposes of the different services are known at design time and communication is monitored by the Mediator Service.

Figure 9 shows as an example the remote procedure call of the method saveRecommendedRecords(). If the user or community enables this feature the Filtering and Recommendation Service (FRS) regularly recommends items that match the users’ interests using this method. The FRS calls this method that is provided by the CWS to store records in a folder. The transmitted XML data contains the encoded method name and two arguments: the folder ID indicating where the record has to be stored and an array of record identifiers.

```xml
<?xml version="1.0"?>
<methodCall>
  <methodName>service.saveRecommendedRecords</methodName>
  <params>
    <param>
      <value>CW665_12706</value>
    </param>
    <param>
      <value>
        AC143_oai_dc_oai:CompSciPreprints:Conferences/0202171
      </value>
    </param>
  </params>
</methodCall>
```

Figure 9 XML Remote Procedure Call

Dividing the software into several independent services that fulfil a specific task has several advantages, but it also requires some special treatment in the software development process. The advantages of a distributed software system are:

- Increase the overall performance because every service can use the available system resources in an optimal way.
- A grand flexibility of the systems functionality because the communication among the services is performed over a well specified interface. Any service can be exchanged by another implementation that is compatible to the interface specifications. New services may be integrated with little effort.
- The system environment of every service can be optimised to achieve the best results for the specific task. The different services of the CYCLADES system use specific programming languages (Java (Sun, 2005) and Python (Python, 2005)) as well as different Database technologies (MySQL (MySQL, 2005), PostgreSQL (PostgreSQL Global Development Group, 2005), Berkeley DB (Sleepycat, 2005), and Oracle (Oracle, 2005)).

The following measures had to be taken in the development process of the distributed CYCLADES environment:

- A reasonable diversion of the software’s functionality into pieces: so the services can be operated mostly independent of each other and require only little well-defined data exchange.
- A simple yet powerful communication protocol: XML-RPC engages remote procedure calls coded in XML and transmitted as HTTP requests.
- Definition of mechanisms to handle data inconsistencies: because data redundancy can’t be avoided.
- User interface guidelines: since individual services provide their own user interfaces, clear user interface guidelines are needed to present the system as a whole to the user.

5 Evaluation

The user evaluation aimed at gathering information about the users’ satisfaction with the usability and usefulness of CYCLADES. The CYCLADES environment was available on the Internet. Anyone was allowed to register and test the system during the eight-week test phase. We spread the information in Usenet newsgroups, on mailing lists, on Websites and in journals and gave demonstrations for potential users. We prepared a ‘QuickStart to CYCLADES’ as a fast introduction into the system. After reading the QuickStart the user could easily explore the system.

5.1 Methods

We created a Web-based questionnaire. The questions were implemented using four point Likert scales (Likert, 1932). These are apt to elicit subjective data about how the users perceive the system. But since the data are ordinal scaled, further statistical calculation could not be performed with them. Free text fields were offered for further explanations, suggestions for improvement or remarks.

A pre-test was conducted to gain information about the suitability and comprehensibility of this questionnaire. Considering the user comments the questionnaire was redesigned and released for the public. The final questionnaire contained questions regarding the responding user group as well as some general information about the system use (e.g. the browser environment used, the time spent using the system). The principal part consisted of six sections, each covering a set of services and features that users typically use together (Communities and Folder Handling; Searching Archives; Collections; Advanced Search Features; Community Management and Recommendations). At the beginning of each section the user stated if she had used this function. If a function had not been used, the user specified her reasons for not using it, and then proceeded to the next section. For those who used a function, some specific questions about the according function and (including appraisal) followed. After assessing the single functions the user rated the usefulness and usability of CYCLADES as a whole.

Technically the questionnaire was implemented as Web pages using PHP scripts to fill the values entered by the user in a MySQL database. For the data acquisition SQL queries were performed to extract the specific information needed.

In addition to the questionnaire we recorded all actions the users performed during the system use, to get an impression about the frequency and area of use of the particular functions.
5.2 Results
Altogether 238 new users created an account that was needed to use the CYCLADES system. The high number of registrations shows the substantial interest in CYCLADES. About one third of all registered users started to fill in the questionnaire, but only 35 completed most of it and were included in the analysis. We assume that this was due to the complexity of CYCLADES; some experience with the system was mandatory to reasonably answer the questions. Still there were some blank fields; these are referred to as "not specified".

The age of the respondents spread from 21 to 60 years, with an average age of 29 years. The CYCLADES system and the QuickStart were accessed from countries all over the world. The majority of the respondents worked in computer science, but there were also among others respondents from physics, business administration, and librarianship.

Firstly, the respondents gave a self-assessment how good they could handle the system (cf. Figure 10). Every respondent categorised herself as beginner, intermediate, advanced or expert in handling the according function. The function ‘Recommendations’ is an exception, since the user is passive in the actual process of creating the recommendations. She only chooses if she wants to get recommendations and which kind of recommendations when creating a new folder or community. The recommendations are then automatically created by the CYCLADES system. Therefore the recommendation service is not accounted for in this analysis. The figure shows that only few users reached an advanced or expert level. This is supposedly due to the relatively short time of exploration.

Finally, the respondents reconceived each function under the aspects of usefulness and usability. For the overall appraisals we combined closely related function sets to larger categories. The resulting sets were Community Support, Search Functions, Collections and Recommendations. Additionally the CYCLADES system as a whole was rated. The evaluation of the usefulness per function set is mapped in Figure 11.

A lot of users thought functions sets to be useful for their work, even though they did not use it in the exploration. Reasons for this low use were among others the lack of time, but also that the users did not to know how to use the function or they could not find the function. Some users stated that certain activities within CYCLADES caused trouble with their computer.

The usability of CYCLADES (cf. Figure 12) was assessed on a scale ranging from ‘self-explaining’ to ‘difficult’. Since the usability can only be judged by those who know a function the extra option “not used” was offered. Regarding the usability the Community support received the best rating. The collections function was the most difficult to use. The usability of the whole CYCLADES system received a good to intermediate grate.

5.3 Discussion
On a whole the evaluation clearly showed that users used the community support and the search functionality. The functions related to these two sets also received the best results concerning usefulness. We think that this is the case for two reasons: first, these functions are basic for any cooperative knowledge management system—that is, they are real prerequisites for successful sharing, exchanging, and learning on demand—and, secondly, they were easy to use. While some functions related to collections might be interesting for long-term and/or experienced users, these functions are either too difficult to use for non-regular users and the benefit of these functions is not clear if the system is used for a short- to medium-term duration only. With the recommendations it is similar: for novel users the system, the information, as well as the communities are new, and therefore novel and interesting. It is only after a longer time (around six months) that the users have a very good overview of the contents offered and the communities around. Only then, the recommendation of other users, communities, and so forth that are still not known is a positive surprise. Furthermore, recommendations often require a thorough understanding of users about their genesis; otherwise, users do not accept them.

All in all we are satisfied with the evaluation: on the one hand it gave us some positive feedback on designs and implementations that are accepted by users, and on the other hand we got valuable feedback concerning possible improvements of some functions.

6 Conclusions
In this paper we motivated the need for cooperative knowledge management. While for some purely technical systems the strict focus on technology is sufficient, this is not the case for socio-technical systems. Supporting the flexible exchange among knowledge workers has to address both the user perspective and the technical perspective. From the user perspective the approach taken for CYCLADES early involved users into the analysis of the requirements and specification of the functionality. And, later real users used the environment and provided feedback on its usefulness and usability.

From a technical perspective the CYCLADES environment is an example of a successful lightweight, yet powerful implementation of a distributed real-time environment. The CYCLADES environment provides utmost flexibility by simple and clear regulations—concerning the technical communication among the components via XML-RPC. It also shows that—although SOAP is widely propagated as the best solution for service-oriented architectures (He, 2003)—XML-RPC also has its strengths. In fact, for service-oriented architectures in which the components provide standardised descriptions of their service, the WSDL is not essential. Likewise, if the environment
provides its own registration service for components, UDDI is not essential. However, the concept and implementation of the CYCLADES is not without its limitations. For instance, the regulations of the communication among the components are static and cannot be changed during runtime. And, adding a new service component requires some—although small—manual adaptations in the Mediator Service. Finally, concerning the user interaction the loose coupling of the components can in some occasions be a challenge. For instance, when a user of the Collaborative Work Service component creates a new folder and then wants to add some collections, the Collaborative Work Service queries the Collection Service for an up-to-date list of all collections at runtime. If for some reason the Collection Service is down or the network connection is slow, the whole process of creating a new folder can fail.

Acknowledgements
The research presented here was carried out and financed by the IST-2000-25456 project CYCLADES, partly funded by the EC. We would like to thank our colleagues from the CYCLADES team at Fraunhofer FIT—and particularly Thomas Kreifelts—as well as at ERCIM, CNR, FORTH, and UNIDO. We would also like to thank the anonymous reviewers for valuable comments.

References