# A Peer-to-Peer-based Distributed Link Service Architecture

Beat Signer[1], Alexandre de Spindler[2], and Moira C. Norrie[2]

[1] Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussels, Belgium
bsigner@vub.ac.be

[2] Institute for Information Systems, ETH Zurich
CH-8092 Zurich, Switzerland
{despindler,norrie}@inf.ethz.ch

**Abstract.** Due to the ever increasing number of different digital media types that we use in our daily work, it is no longer sufficient to manage them in an isolated way but desirable to define associations across the media boundaries. While cross-media information systems enable associations between digital and physical information, there is often a lack of support for flexible authoring and sharing of these associations (links) as, for example, required in meeting situations. We present a solution for this kind of ad-hoc information exchange and collaborative authoring in cross-media environments which is based on a link service in combination with peer-to-peer technologies. Our goal was not to replicate hypermedia documents based on peer-to-peer technology, but rather to use a distributed link service architecture to enable the sharing of link metadata in collaborative information spaces. To avoid information overload and guarantee a certain link quality in these highly dynamic information environments, we adopt some general mechanisms for user and link rating.

## 1 Introduction

While hypermedia offers a non-linear information environment that allows users to easily browse back and forth between different forms of digital media, its generalisation to cross-media information spaces enables interaction services that can be based on arbitrary resources including physical objects. The concept of being able to freely link arbitrary resources is now also being adopted in fields such as ubiquitous computing to span physical and digital information spaces.

In terms of information services, a cross-media solution effectively enables the model of information publishing and access known from the Web to be extended to any type of information resource and service and possibly to also link these to physical entities. Thus a geographical location could be linked to an audio file, a product to a query for inventory information and a figure within a book to a video or a specific application control command. The use of unique physical identifiers, for example RFID tags, provides the basis for linking physical objects to digital services. However, to support the development of complex information services involving context-awareness, hypermedia style browsing and dynamic authoring, general cross-media frameworks are required. We have developed such a framework, called iServer, based on a link metamodel, object database technologies and a resource plug-in mechanism to enable linking between different types of digital and physical media.

Several applications have been developed based on the iServer platform and, so far, most of these have been based on a strict publisher and consumer model, where users can browse pre-authored information and access services based on a predefined information space. However, we also want to support cooperative, community-based models of cross-media information environments, where users can publish new personal associations between existing resources or annotations over existing media. We have therefore developed a peer-to-peer (P2P) version of iServer that enables users to share their link spaces. In contrast to many existing P2P systems, we do not aim to replicate resources based on P2P technology, but rather to use a distributed link service architecture for sharing metadata about links between existing resources.

In our distributed iServer solution, link services are discovered in a completely decentralised manner and we do not rely on a dedicated server. Users can dynamically join the community and start annotating or adding links to arbitrary third-party resources as well as access link metadata from other users. To avoid issues of information overload and link fraud, we introduce a collaborative filtering mechanism based on a combined ranking of users and link resources.

We start in Sect. 2 with a more detailed look at the motivation for collaborative cross-media information spaces and a discussion of related work. In Sect. 3, we briefly introduce the existing iServer platform and its underlying cross-media link model before going on to describe the functionality and operation of the cooperative iServer version. In this section, we also discuss how our notion of collaborative information spaces relates to existing distributed information platforms that are based on P2P technologies. Our P2P-based distributed architecture is presented in Sect. 4. In Sect. 5, we introduce different approaches for user and link rating. The implementation of the motivational scenario is discussed in Sect. 6. Concluding remarks and a discussion of future work are given in Sect. 7.

## 2 Motivation

The original hypertext and hypermedia information models are based on the concept of connected document spaces in which additional meaning is usually associated with the links between documents. Early models have been extended over the years, both to broaden the scope of the model and to improve the functionality and maintenance of systems through more powerful and flexible link management. Two key innovations in the case of link management were the possibility to address parts of documents in defining link anchors and the management of link metadata in separate linkbases rather than embedding links in documents [1]. With the emergence of ubiquitous and pervasive computing, physical hypermedia systems have been proposed that enable real-world objects to be linked to digital media, and vice versa, by allowing physical resources to also be included as nodes in the connected information space [2].

Underlying all of these developments, the basic information model remains the same. In all cases, an information space is a connected graph where the nodes are resources and the links are represented by edges. The anchor and target of a link can either be an entire resource or an element within a resource. In contrast to database systems, links are defined at the instance level rather than at the type level. Our goal was to develop a general framework to support a range of hypermedia tools and services. We achieved this by adopting a database approach to the problem that first involved developing a generic link metamodel and then implementing it using an object database management framework.

One of the benefits of managing links separately from resources is that new links can easily be created, not only by the publisher of a resource, but also by arbitrary users. An advantage of such an *open link authoring* system is that the information space evolves over time based on the users' current interest. For example, in a teaching and learning environment, students could not only consume the material prepared and published by a teacher, but also add their own links between different resources: These could be links from copies of course slides to web pages, from examples in a printed tutorial document to applications or from physical objects in a lab to digital manuals. Of course, the students could not only create new associations between existing resources but also define links to content that has actually been created during a lecture.

An example of such a tool for capturing notes during a lecture or meeting and sharing them online is the Pulse Smartpen application recently released by Livescribe[3]. Based on the digital pen and paper functionality developed by Anoto[4], handwritten information is captured, synchronised with a concurrent voice recording and can later be published on a web portal. While the sharing of information in this case forms part of a specific application, we envision a more open approach where associations can be defined across different applications as well as different types of media. In a lecture setting, we could imagine the use of other tools for the capturing and linking of information. For example, the interactive paper-based PaperPoint [3] presentation tool could, not only help the lecturer in controlling and annotating presentations, but also support students in annotating content on slide handouts with their own comments and supplemental information.

The important thing about such an open link authoring approach is that we have a combination of the traditional publisher and consumer model with a democratic authoring process based on an open link platform that potentially supports any form of existing and upcoming types of resources. Note that the dynamic authoring of new links could be carried out by software agents by

[3] http://www.livescribe.com
[4] http://www.anoto.com

deriving new associative links based on the analysis of user access patterns as well as being performed manually by individual users.

If users can create their own links, it makes sense that they can share the link metadata with community members. Thus, in our teaching scenario, the students could not only benefit from the links authored by the teacher, but also from new links introduced by other students. Further, students and teachers in one university could share links with those in other universities—possibly also leading them to new resources. However, the idea is not to rely on a client-server architecture where information is created and accessed by the client and then stored on a dedicated server for later sharing. Rather, the students should be able to share their information with other peers in the system in an ad-hoc manner. This could even mean that for some forms of peer-to-peer information sharing we could rely on some kind of ad-hoc connectivity offered by Bluetooth networking and other technologies, where information can only be shared with users in the near vicinity.

When it comes to sharing links, students will no doubt find that certain students provide more useful links than others, especially when linking resources provided by teachers to arbitrary external resources. The course will also evolve over time and so it may be the case that the information of their colleagues may be more reliable than information that was added by former students. However, the exception might be previous students who obtained excellent grades. This implies that we do not simply want to share new link metadata, but also introduce a notion of link quality based on user trust.

The teaching environment is only one application scenario where people want to organise and share information with other local or remote users. The iServer platform is not limited to a specific application, but rather provides a general platform for cross-media information management that can be applied to a large variety of application domains. Two important aspects of the P2P iServer framework are access controls to ensure user privacy as required and the filtering of links to avoid information overload and possible system abuse. The former is handled by the fact that the iServer framework has a user model integrated into its core link model and provides the required user management functionality. The latter is supported by mechanisms for user and link ranking that are used to evaluate the quality of shared link information.

The idea of community-based link sharing, as provided by the P2P iServer framework, is somehow related to proposals in distributed hypermedia [4]. The benefit of our approach lies mainly in the means to realise the goal. By building on advanced database technologies and exploiting practices of metamodel-driven system engineering to the full, we have succeeded in designing a flexible link server platform that supports digital as well as physical resources and is open to new communication paradigms including peer-to-peer technology. In the following sections, we first provide a short overview of the iServer framework and then go on to describe the architecture and operation of our new decentralised P2P iServer framework.

## 3 Cooperative Cross-Media Link Server

iServer is a general cross-media link server capable of supporting an extensible set of digital and physical media. The framework is based on the resource-selector-link (RSL) metamodel and supports the integration of new media types based on a resource plug-in mechanism. It is beyond the scope of this paper to describe all the details of the RSL model and we only introduce the main features relevant to the new cooperative iServer extension. Details of the complete RSL model can be found in [5].

### 3.1 RSL Metamodel

The RSL model consists of three core *entity* types represented by *resources*, *selectors* and *links* as shown in Fig. 1. A link is always associated with one or more source entities and one or more target entities where each of these entities might be a resource, a selector or even another link. While a resource represents an entire information unit or service, a selector allows us to address parts of a resource in a similar way to the reference objects introduced in the FOHM [6] model. Note that a selector always has to be associated with a single resource but a resource may have multiple selectors. By modelling links as an entity subtype, we gain the flexibility to define links with other links as source or target entities, allowing us to annotate links with any form of supplementary information.

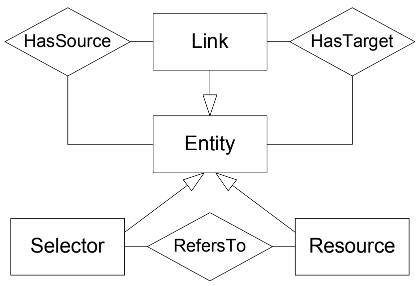


**Fig. 1.** Resource, selector and link

The user model defines the access rights associated with an entity and is essential for information sharing issues discussed later in this paper. Users may be either *individuals* or *groups* and each entity has exactly one individual creator who defines the user access rights. When it comes to sharing links and resources in the peer-to-peer version, it is up to the creator of an entity to specify whether the information is private or should be publicly available.

On the RSL model level, we have the abstract concepts of resources and selectors and the system can be extended to support new media types by providing concrete implementations (plug-ins) for the resource and selector types. In the past, various iServer plug-ins have been developed to support different media types. This includes, for example, the iWeb plug-in for HTML resources where iServer acts as an external link repository for web pages in a similar way to existing linkbases. The definition of an iWeb selector is straightforward since we can build on the XML Pointer Language (XPointer) that is used in the context of the XML Linking Language (XLink) [7]. Note that we obtain some additional features not available in the XLink language such as the cross-media linking to an extensible set of resources. In addition to the iWeb plug-in, we have developed iServer resource plug-ins for images, sound and movies, as well as an RFID plug-in for tagging physical objects. Furthermore, a plug-in for linking paper documents (iPaper) has been realised based on interactive paper [8, 9] technologies where position information is encoded on paper as part of the document printing process using, for example, the digital pen and paper functionality offered by Anoto.

A number of applications have been implemented based on the iServer framework, including general browsers for cross-media information spaces, a paper-based mobile tourist information system [10], educational applications and information services as well as an art installation for collaboratively writing and accessing interactive cross-media narratives [11]. Recent work includes the development of advanced authoring tools and, related to this, the cooperative version of iServer for open link authoring.

### 3.2 Open Link Authoring

The authoring of links in highly-connected cross-media information spaces can be both time-consuming and tedious. Authoring tools have been developed to enable publishers to easily create links across the rich mix of resources and selectors offered by the iServer plug-ins. Making these tools available to users presents a shift from the publisher and consumer model to an open authoring system where every user also becomes a potential publisher and can create their own links between existing resources. Thus, in our student and teacher scenario introduced earlier, individual students can augment the resources and links published by the instructor to create their own, personalised information space. Further, they can integrate external resources through the creation of links to these resources, thereby expanding the cross-media information environment. A next logical step is to allow users to share their links, thereby promoting knowledge sharing and reducing the individual authoring effort. In this way, what might originate as a minimally connected set of resources offered by the instructor can rapidly evolve into a rich, highly-connected information space that reflects the interests and experience of the students.

To support such forms of collaborative authoring, we have developed a cooperative iServer framework based on a P2P architecture. Before describing the implementation of the system in more detail, we would like to compare and contrast our solution with existing P2P information sharing architectures.

P2P technology is well-known from file sharing systems such as Napster[5] and Gnutella. Within the database community, P2P technologies have been proposed for information sharing and distributed services. In the case of information sharing, the familiar problems of detecting semantic equivalences exist where there is no agreed common schema and one approach is to use some form of mediator to assist in the interpretation of shared data [12].

Metadata and ontologies have been used by the hypertext community to help users search for information based on a semantic proximity relationship between documents and also for automatic link generation between documents as described by Dolog et al. [13]. Common to many of these systems is the analysis of metadata for selecting objects to be returned to the requesting node based on an attempt to match the semantics of the objects to some form of search query. In contrast, our approach is not based on finding resources which are similar to the resource at hand, but rather on simple associations between resources that users themselves have found to be meaningful. Thus our view is closely related to the proposal for associative linking originally presented by Bush [14].

Of course, we are not alone in this view since distributed link services [15], where link information is stored separately from the resources, have investigated associative linking of resources. More recently, P2P technologies have been considered for building new forms of open hypermedia systems where central link databases are no longer required [16]. Results in information retrieval research further show that link knowledge plays a crucial role in classifying unstructured information spaces such as web resources [17] and can improve information access.

Our approach differs in several aspects from other proposals for P2P-based open hypermedia systems. First, we do not use the peer-to-peer network functionality to replicate any hypermedia documents and perform searches in the distributed network of peers as proposed, for example, by Larsen and Bouvin [18]. In the cooperative iServer solution, the original hypermedia documents are clearly separated from any annotations or external links and it is only the link or annotation metadata that is shared in an ad-hoc manner over the distributed peer-to-peer architecture.

The clear separation between resources available from specific locations, for example defined by a unique URI, and the distributed link metadata implies that we do not have to care about replication-relevant issues such as identifying identical resources stored by different peers or algorithms for providing optimal distribution and accessibility. A client will always access a resource from its original location and, only in a second step, is additional link metadata acquired over the dynamic peer-to-peer iServer network as illustrated in Fig. 2. The separation of content and metadata further implies that a resource should always be available, provided that the server on which the resource is hosted is up and running, whereas any additional link information may change dynamically over time based on the set of iServer peers currently available in the network.

[5] http://www.napster.com

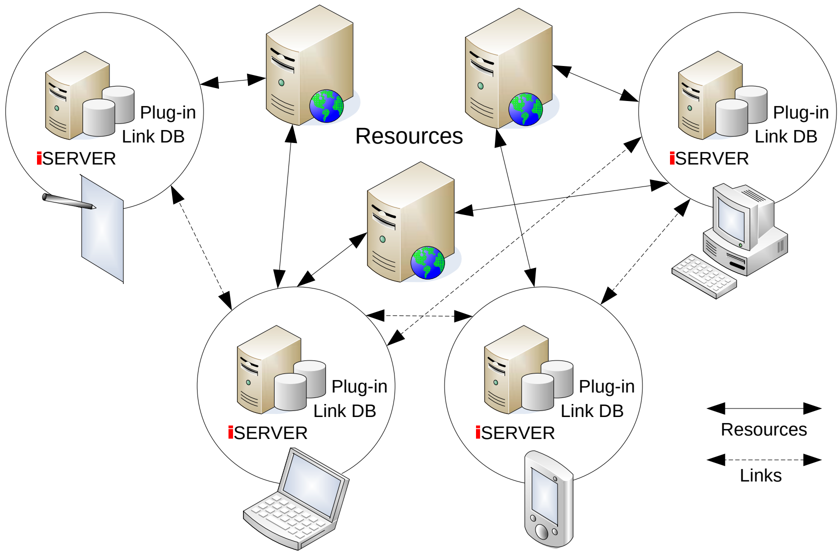


**Fig. 2.** Cooperative iServer architecture

To exemplify the main idea behind dynamically distributed link metadata let us compare our approach for link sharing to Annotea [19], a project for collaborative web annotations. While browsing the Web with Amaya[6] or any other Annotea-enabled web browser, a user can annotate arbitrary third-party web pages. The original web page is not modified at all and only new annotation metadata is stored on a server that has to be specified explicitly by the user. Any user or group of users who want to share their annotation metadata have to connect to the same annotation server which is equivalent to having a single iServer instance within our cooperative iServer framework. If our approach for sharing link metadata is applied to the Annotea scenario, this means that the metadata no longer comes from a single annotation server but from a whole network of distributed annotation peers. A single user or a group of users may still have a personal annotation server but, at the same time, it becomes possible to get access to annotations from other groups or communities. What we achieve with the cooperative iServer solution is that a user can access information from their personal iServer instance as well as retrieve link metadata from other iServer peers.

We have to distinguish between *persistent link metadata* in the form of link information that is stored in a personal iServer instance and the *transient link metadata* received from the set of remote iServer peers. While the quality of the persistent link metadata can be ensured by controlling the users who have access to a personal iServer instance, we do not have any direct control mechanism to guarantee the integrity of information provided by remote peers. Therefore, an

[6] http://www.w3.org/Amaya/

important part of the cooperative iServer framework is a collaborative filtering mechanism based on the rating of remote users and links as described later in Sect. 5. Note that users do not have control over all linked resources that may be changed or even deleted on remote sites. However, in this case, the link metadata may be updated accordingly or, in the case that a resource has been deleted, the link may be removed too.

The availability of additional link information over the P2P network can be seen as an optional extension to information stored in a single iServer instance. Users are free to use the new cooperative iServer functionality or to work solely with their personal iServer instance. The transient supplemental link metadata provided by the set of iServer peers represents optional suggestions by the members of the collaborative information environment. However, if a user finds some of the suggested link information relevant, they can store it persistently in their personal iServer instance.

## 4 Distribution Architecture

In designing a distributed architecture for iServer, it was one of our goals to ensure that the framework kept to its principle of being as general as possible, with a clean separation of concerns. Even if iServer is primarily an extension of an object database system, it provides well-defined Java, XML and Web Service APIs for accessing and updating information. The general interaction between peers consists of sending single API calls from one peer to another, the execution of that request on the remote site and the transfer of the result back to the initiating peer. The information returned by the remote peers has to be combined and integrated with information that is available from the local iServer instance. The functionality of a remote iServer system is offered to a local iServer instance via a peer service. This service is implemented within a peer service platform that separates the aspects of *interaction* and *connection*.

The interaction architecture is shown in Fig. 3. For the sake of simplicity, we reduce service interaction to the transfer of request and response data between a requesting local system shown on the left-hand side and a responding remote system on the right-hand side. One-to-many and many-to-one interactions which typically occur in distributed iServer scenarios can always be represented by multiple one-to-one data transfers. We now describe the main steps of interaction labeled ⓐ to ⓗ in the figure.

ⓐ In the local system, a request handler is used to post a local request. Depending on the concrete peer service, data may have to be provided by the requesting client. ⓑ The handler creates a message object containing the request data and sends its XML representation to a remote request handler. ⓒ On the remote site, the message is reconstructed from the string value received and the request data is extracted. ⓓ The remote request handler then processes the request which includes access to the iServer API yielding a response data object. ⓔ This response data object is wrapped with a message object and ⓕ its XML representation is sent back to the local response handler. ⓖ The response

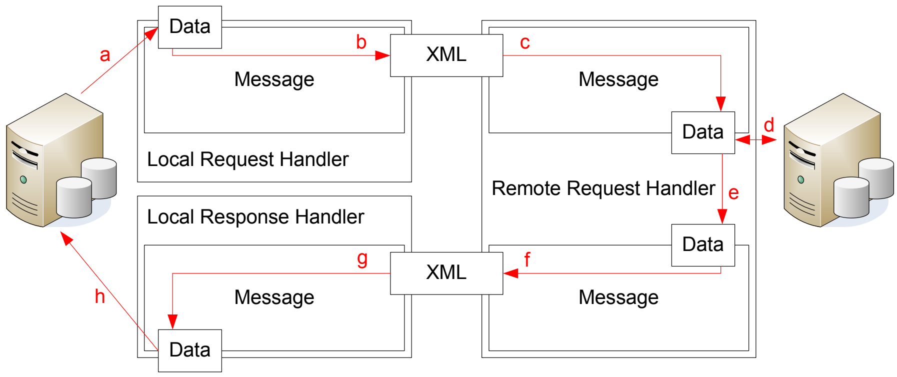


**Fig. 3.** Interaction architecture for peer services

handler reconstructs the message and extracts the response data. ⓗ Typically, the response is then integrated into the local iServer link structure.

We can identify three components each responsible for a particular aspect of service interaction: *handlers*, *data* and *message* objects. A `Data` class encapsulates the iServer API call while the `Message` class enables content-independent implementation of message formats such as plain XML, compressed or encrypted representations. Handlers implement the data object processing which consists of executing API calls and integrating results.

A request handler relies on a connection service providing the means to address a particular or multiple peers and to send and receive messages. This functionality is encapsulated in a connection service component. It consists of a peer abstraction that can be identified by other peers, a class representing the group of peers sharing links as well as resources and a connection handler creating and maintaining physical channels to remote peers.

The peer abstraction is responsible for starting and stopping a connection handler and it maintains a collection of running connection handlers. In order to deploy a particular peer service, the corresponding request and response handlers must be registered with the peer. The registration causes the peer to create a new connection handler object and associate it with the handlers. Note that we can deploy multiple services using the same class of connection handlers, in which case, each of them forms a logical peer-to-peer channel while they actually all use the same connection. However, we can also have multiple connection handler classes, each implemented for a different connection technology. In this case, different technologies such as JXTA[7], Web Services and ad-hoc networking can be used in parallel.

[7] https://jxta.dev.java.net/

## 5 User and Link Rating

In a cooperative community of publishers and consumers who are equally responsible for the available content, a variety of issues must be addressed. Brookshier et al. [20] name three typical problems. First, a *leecher* is a participant who consumes without contributing. Communities with too many leechers produce little content and all network traffic is focused on a few publishers which decreases their availability. Second, any community supporting democratic publishing can be a target for *spamming*, namely the distribution of unsolicited content. Last, *malicious content* can be published for various reasons with the goal of reducing the community's worth.

Some systems try to avoid the first problem by keeping designated publishers to ensure the information supply, and of course it is always possible to support this model in the iServer P2P framework. The content in our information spaces consists of resources and links and we primarily envisage shifting the authoring of links between resources to the user community rather than the authoring of the resources themselves. This means that the effort required from the user is minimal and therefore should not discourage them from cooperating.

The rapid and considerable emergence of technologies enabling democratic content publishing has led to the development of rating and filtering systems addressing the issues of unsolicited or malicious content. Existing collaborative filtering (CF) techniques are frequently classified as being either *user-based* or *item-based*. In *user-based* CF [21–23], the ratings from users similar to the requesting user are aggregated for the target item. The assumption is that similar users share similar opinions. User similarity is based on the comparison of user profiles representing a user in terms of features relevant in the scope of the application domain. In contrast, an *item-based* CF approach [24, 25] first retrieves items similar to the one to be rated and then aggregates the ratings from all users about these items. In this case, the assumption is that similar items tend to be rated similarly. Both of these approaches have in common the fact that the rating inference is based on some notion of similarity. However, the assessment of relevant features as well as the computation of similarities still form a bottleneck in current rating systems.

The main interaction of iServer peers consists of exchanging link metadata. A common request is a query for in- and outgoing links from a particular resource. As a result, the user receives a set of unknown links. Filtering can be achieved by deciding, for each item received from a particular user, whether to accept or reject it. In real world situations such decisions are often made based on the trustworthiness of the sender in addition to the relevance and quality of the item. Electronic communities such as eBay [26] have successfully implemented trust and reputation techniques supporting users in taking decisions. Research in trust and reputation systems has given rise to a variety of techniques for rating users and interpreting rating values [27].

Our framework supports individual users in deciding on their own levels of trust in certain users, thereby accounting for any lack of social agreement about the trustworthiness of individual publishers. A user who knows and is therefore able to rate a relatively small number of other users can infer ratings from a vast number of other users by exploiting transitive propagation of trust.

We provide a rating implementation where a rating value is interpreted as the amount of trust flowing from a rating user to a rated user. The transitive propagation of trust consists of searching the minimum trust along a path connecting the two users. If multiple such paths exist, all minimum trusts are summed up. Given a graph where the nodes represent users and edges connect those users that have explicitly stated their trust in each other, this can be achieved by running a max flow algorithm with the rating user as source and the rated user as sink.

Since a user receives responses from multiple users, the information on which a filtering decision is based consists of the sending user and the currently received item as well as the set of previously received items. In order to exploit all information available, we also take into account the frequency with which a particular item has been received. Therefore, we combine user ratings with response ratings to filter responses returned from remote peers. This may, not only help to improve the quality of information presented to the user, but also reduce the quantity, thereby preventing the information overload that could result from a large, highly-connected information space.

We encode user ratings as tuples containing the rating user, the rated user and the rating value. Such a tuple is created by the rating user and propagated to all other members of the peer group. A user rating manager ensures that all peers have the same set of tuples stored locally. This tuple synchronisation is achieved as follows:

1. On startup, the peer tries to read a file containing tuples stored in previous sessions. If this file does not exist, a new empty one is created.
2. The peer creates a tuple set $S_{local}$ containing all tuples in the file. Whenever $S_{local}$ changes, the file is updated.
3. When a peer joins the group, it requests the tuple set $S^i_{remote}$ from all other members $i$ of the group.
4. Every incoming tuple set $S^i_{remote}$ is compared with the local set $S_{local}$ as follows:
   - If $S_{local}$ does not contain all tuples in $S^i_{remote}$ then the local set is updated.
   - If $S^i_{remote}$ does not contain all tuples in $S_{local}$ then the local set is broadcast to all other members.
5. When no more sets are broadcast, all tuple sets contain the same tuples.
6. Whenever a new rating is set locally, the local set of tuples is broadcast to all other members of the group.

We have implemented a user rating manager which encapsulates the process of synchronising rating tuple sets. It maintains a graph containing all user ratings and provides the interface for rating inference. A rating manager is owned by a peer and is used by one of its response raters as described next.

Since every iServer P2P API request is broadcast to all members of a group, a peer possibly receives multiple responses. A response rating manager filters incoming responses before they are made accessible to the user. The selection is based on rating values that are computed by response raters for every response. The rating values of multiple response raters are combined by an interchangeable aggregator function. We implemented two response raters, one returning the rating of the responding user as described above and the other returning the frequency of the response within the collection of previous responses.

A response is uniquely identified by the initiating request and the responding user. For every request, the rating manager maintains a set of responses, each paired with the responding user. It also manages a set of registered response raters that are used to compute an overall rating value. Each rater is associated with an individual weight which is multiplied with the rating value computed before being aggregated with the overall rating value. The local response handler adds incoming responses to the request specific set. There are two ways to access the resources and links received: Either by explicitly requesting the set of filtered responses to a particular request or upon notification about a received response that passed the filtering.

## 6 Cooperative iServer Application

After presenting the iServer P2P architecture and the related rating scheme, we now explain how applications supporting the type of scenarios described in Sect. 2 can be realised based on the outlined architecture. The iServer API enables the creation of new entities as well as the definition of links between source and target entities. Each iServer peer manages a local graph structure containing the linked entities and can further transiently integrate links received from other iServer peers.

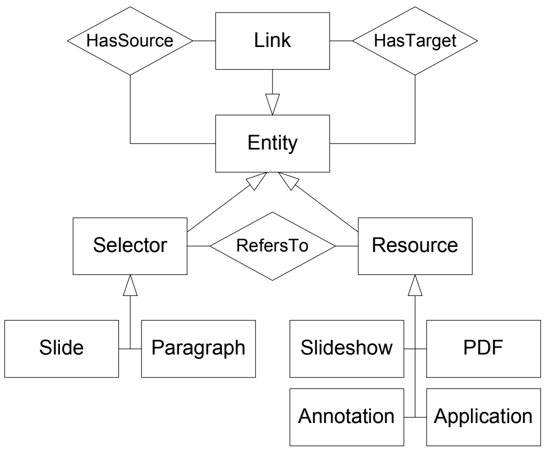


**Fig. 4.** iServer plug-ins

The functionality to create entities and links between them is provided by specific applications. An application is defined in terms of some application logic in combination with a set of iServer plug-ins. The implementation of a new iServer plug-in includes the definition of the resource types such as slides, web pages and annotations, as well as the specification of the corresponding selectors.

In Fig. 4, we show two plug-ins providing some of the functionality motivated in Sect. 2. The slideshow annotation is defined in terms of the `Slideshow` and `Annotation` resources. In this case, the selector is a particular `Slide` within a slideshow which can be annotated. The association of a programming tutorial with example applications is declared by the `PDF` resource from which selected `Paragraphs` can be linked to `Application` resources.

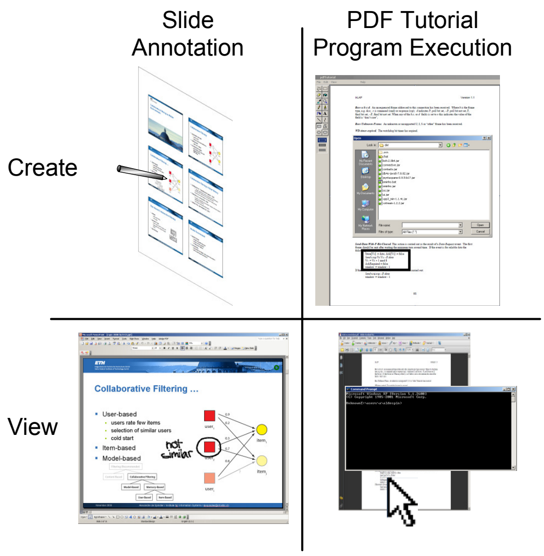


**Fig. 5.** Create and view links

The application user interface must provide the means for users to create and view links as shown in Fig. 5. In order to annotate a slideshow, a note is written on a printed paper handout with a digital pen and interpreted by the PaperPoint application as shown on the top left in Fig. 5. Based on our interactive paper iServer plug-in, the PaperPoint application can not only capture the handwritten note but also assign it to a specific slide based on where the user writes within the handouts. The annotation is stored as a new resource instance containing the handwritten text and is linked to the corresponding slide selector. Note that since we only share links and not resources using the P2P iServer architecture, the application has to make sure that the annotation resource is made available, for example on a dedicated server, if it is to be accessible later to other users. Any link with a non-available resource will just not be visualised. In the case where a paragraph of a PDF tutorial should be associated with some executable code, the authoring tool on the top right is used to create a link between a selected paragraph and the corresponding application executable.

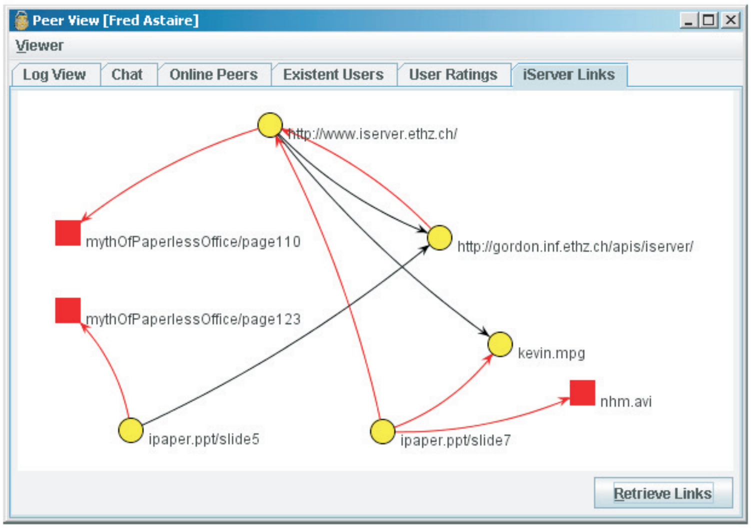


**Fig. 6.** Visualisation of local and remote links

The P2P iServer framework provides some built-in functionality to visualise the collaborative information spaces as shown in Fig. 6. In this simple visualisation tool, which is mainly used by developers of new services for checking link sharing at a fairly low level, resources of any media type are just represented by simple text labels. Furthermore, nodes and edges are represented in different colours to highlight the original information sources. Yellow circular nodes and black edges represent entities originating from the local iServer, whereas red rectangular nodes and the lighter red edges represent remote entities. This distinction has been implemented for visualisation purposes only and it can be omitted if remote entities are to be integrated transparently into the local system. The snapshot of the graphical user interface (GUI) also indicates the auxiliary functionality offered by the tool. Users can also view information about user ratings and specify their own explicit ratings if required.

## 7 Conclusion and Future Work

We have presented a notion of collaborative cross-media information environments based on community-based link authoring between arbitrary resources and elements within these resources. The underlying information model is similar to those familiar from hypertext and hypermedia systems, but allows links between arbitrary resources, including physical objects, paper documents and application services. Cooperation is actually supported on two levels—the user management integrated into the core RSL model and a P2P architecture for distributed information sharing.

While many of the features of the iServer framework and its cooperative version can be found in other projects and systems developed within the hypertext community, the main contribution of our work is to combine concepts in a single, extensible framework. Our cooperative iServer framework applies P2P technology for metadata sharing whereas the actual resources are not replicated.

In order to avoid information overload and ensure link quality, we introduced a collaborative filtering mechanism that is based on a graph model. Response raters and aggregating functions allow arbitrary filtering techniques to be adopted. The use of a graph opens the rich variety of existing graph algorithms. A rater has been presented that implements filtering using a max flow algorithm. Through the development of a flexible framework, we have obtained a platform allowing for future investigation with new combinations of interaction procedures, connection technologies and collaborative filtering techniques.

We have further highlighted how the cooperative iServer framework supports the application developer in realising solutions for sharing link information across different applications as well as different types of resources. The presented architecture ensures that developers do not have to deal with link sharing and link rating details. At the same time, the iServer framework guarantees that the resulting solution for dealing with cross-media information spaces is extensible in the sense that new resource types can be easily integrated via a plug-in mechanism.

Our original implementation implies that all collaborating peers must be connected through a single network. We are investigating another approach which does not require such a permanent network. It supports ad-hoc connections between peers within reach of their Bluetooth or Wi-Fi facilities. A system including such handlers allows opportunistic data sharing between any peers that happen to meet in physical space. Mobile users greatly benefit from delay tolerant multi-hop data dissemination in the absence of a network infrastructure. It also provides the notion of user similarity resulting from the nature of ad-hoc networks. We are currently evaluating this similarity measure in the filtering process described in this paper which involves the implementation of a third response rater and its deployment within our iServer P2P framework.

## References


1. Davis, H., Hall, W., Heath, I., Hill, G., Wilkins, R.: Towards an Integrated Information Environment with Open Hypermedia Systems. In: Proc. of ECHT '92, ACM European Conference on Hypertext, Milan, Italy (December 1992)
2. Hansen, F.A., Bouvin, N.O., Christensen, B.G., Grønbæk, K., Pedersen, T.B., Gagach, J.: Integrating the Web and the World: Contextual Trails on the Move. In: Proc. of Hypertext 2004, 15th ACM Conference on Hypertext and Hypermedia, Santa Cruz, USA (August 2004)

3. Signer, B., Norrie, M.C.: PaperPoint: A Paper-Based Presentation and Interactive Paper Prototyping Tool. In: Proc. of TEI 2007, 1st International Conference on Tangible and Embedded Interaction, Baton Rouge, USA (February 2007)
4. DeRoure, D., Carr, L., Hall, W., Hill, G.: A Distributed Hypermedia Link Service. In: Proc. of SDNE '96, 3rd Workshop on Services in Distributed and Networked Environments, Washington, DC, USA (1996)
5. Signer, B., Norrie, M.C.: As We May Link: A General Metamodel for Hypermedia Systems. In: Proc. of ER 2007, 26th Intl. Conference on Conceptual Modeling, Auckland, New Zealand (November 2007)
6. Millard, D., Moreau, L., Davis, H., Reich, S.: FOHM: A Fundamental Open Hypertext Model for Investigating Interoperability between Hypertext Domains. In: Proc. of Hypertext 2000, 11th ACM Conference on Hypertext and Hypermedia, San Antonio, USA (May 2000)
7. Wilde, E., Lowe, D.: XPath, XLink, XPointer, and XML: A Practical Guide to Web Hyperlinking and Transclusion. Addison Wesley Professional (July 2002)
8. Signer, B.: Fundamental Concepts for Interactive Paper and Cross-Media Information Spaces. PhD thesis, ETH Zurich (May 2006) Dissertation ETH No. 16218.
9. Weibel, N., Norrie, M.C., Signer, B.: A Model for Mapping between Printed and Digital Document Instances. In: Proc. of DocEng 2007, ACM Symposium on Document Engineering, Winnipeg, Canada (August 2007)
10. Belotti, R., Decurtins, C., Norrie, M.C., Signer, B., Vukelja, L.: Experimental Platform for Mobile Information Systems. In: Proc. of MobiCom 2005, 11th Annual International Conference on Mobile Computing and Networking, Cologne, Germany (August 2005)
11. Vogelsang, A., Signer, B.: The Lost Cosmonaut: An Interactive Narrative Environment on Basis of Digitally Enhanced Paper. In: Proc. of VS 2005, 3rd Intl. Conference on Virtual Storytelling, Strasbourg, France (December 2005)
12. Ooi, B.C., Tan, K.L., Zhou, A., Goh, C.H., Li, Y., Liau, C.Y., Ling, B., Ng, W.S., Shu, Y., Wang, X., Zhang, M.: PeerDB: Peering into Personal Databases. In: Proc. of SIGMOD '03, the ACM SIGMOD Intl. Conference on Management of Data 2003, San Diego, USA (June 2003)
13. Dolog, P., Gavriloaie, R., Nejdl, W., Brase, J.: Integrating Adaptive Hypermedia Techniques and Open RDF-Based Environments. In: Proc. of WWW 2003, 12th Intl. World Wide Web Conference, Budapest, Hungary (May 2003)
14. Bush, V.: As We May Think. Atlantic Monthly **176**(1) (July 1945)
15. Carr, L., Roure, D.D., Hall, W., Hill, G.: The Distributed Link Service: A Tool for Publishers, Authors and Readers. In: Proc. of WWW4, 4th Intl. World Wide Web Conference, Boston, USA (December 1994)
16. Bouvin, N.O.: Open Hypermedia in a Peer-to-Peer Context. In: Proc. of Hypertext 2002, 13th ACM Conference on Hypertext and Hypermedia, College Park, USA (June 2002)
17. Calado, P., Cristo, M., Moura, E., Ziviani, N., Ribeiro-Neto, B., Gonçalves, M.A.: Combining Link-Based and Content-Based Methods for Web Document Classification. In: Proc. of CIKM 2003, 12th Intl. Conference on Information and Knowledge Management, New Orleans, USA (November 2003)
18. Larsen, R.D., Bouvin, N.O.: HyperPeer: Searching for Resemblance in a P2P Network. In: Proc. of Hypertext 2004, 15th ACM Conference on Hypertext and Hypermedia, Santa Cruz, USA (August 2004)
19. Koivunen, M.R.: Semantic Authoring by Tagging with Annotea Social Bookmarks and Topics. In: Proc. of SAAW2006, 1st Semantic Authoring and Annotation Workshop, Athens, Greece (November 2006)

20. Brookshier, D., Govoni, D., Krishnan, N., Soto, J.C.: JXTA: Java P2P Programming. Sams (March 2002)
21. Breese, J., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proc. of UAI '98, 14th Conference on Uncertainty in Artificial Intelligence, Madison, USA (July 1998)
22. Claypool, M., Gokhale, A., Miranda, T., Mumikov, P., Netes, D., Sartin, M.: Combining Content-Based and Collaborative Filters in an Online Newspaper. In: Proc. of the ACM SIGIR Workshop on Recommender Systems, Berkeley, California (August 1999)
23. Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B., Riedl, J.: Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. In: Proc. of CSCW 1998, 10th ACM Conference on Computer Supported Cooperative Work, Seattle, Washington (November 1998)
24. Robu, V., Poutré, J.L.: Constructing the Structure of Utility Graphs Used in Multi-Item Negotiation through Collaborative Filtering of Aggregate Buyer Preferences. In: Proc. of RSS'06, 2nd Intl. Workshop on Rational, Robust and Secure Negotiations in Multi-Agent Systems, Hakodate, Japan (2006)
25. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.T.: Item-based Collaborative Filtering Recommendation Algorithms. In: Proc. of WWW10, 10th Intl. World Wide Web Conference, Hong Kong (May 2001)
26. Resnick, P., Zeckhauser, R.: Trust Among Strangers in Internet Transactions: Empirical Analysis of Ebay's Reputation System. The Economics of the Internet and E-Commerce **11** (2002)
27. Jøsang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems (2005)