

ENABLING SEMANTIC SEARCH IN STRUCTURED P2P NETWORKS VIA DISTRIBUTED DATABASES AND WEB SERVICES

Maria Teresa Andrade ^{§,†}

[§]*INESC Porto*

Rua Dr. Roberto Frias 378, 4200-465 Porto, Portugal

[†]*FEUP, Faculdade de Engenharia da Universidade do Porto*

Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal

Asdrúbal Costa [§]

[§]*INESC Porto*

Rua Dr. Roberto Frias 378, 4200-465 Porto, Portugal

ABSTRACT

First generation of overlay P2P networks had scalability problems although offering useful keyword-base searching functionality. The new generation of structured networks, based on Distributed Hash Tables (DHT), solved the scalability problem but at the same time eliminated the possibility of performing searches by proximity. This can be considered as an effective limitation as users of P2P networks will most often not have the complete information about the data they are looking for. Accordingly they tend to submit general queries to the network, hoping to receive back a list of results that match their needs. In this paper we describe a solution that is able to profit from the best of the two approaches. It relies on a two-layer architecture, offering on the top layer a set of Web Services to upload, search and retrieve content and on the bottom layer a fully structured P2P network (DHT-based) for distribution and downloading efficiency. Central to our solution, is the use of distributed databases, implemented at the top layer, holding descriptive metadata of the uploaded digital objects. The developed system also makes use of the JXTA project at the top layer and the Bit Torrent suite of protocols at the bottom layer.

KEYWORDS

Distributed, semantic, P2P, availability, search, repository.

1. INTRODUCTION

Distributed Hash Tables (DHT) is not the most efficient solution to perform semantic search of content in P2P networks. DHT allows to search for specific content identified by a hash key and to eventually perform boolean operations upon the results of searches that used different keys. It provides considerable fast search times in respect to unstructured solutions. However, it does not allow to perform semantic searches based on proximity. Different approaches are currently being investigated by researchers to overcome this problem [1], [2]. Studied approaches include Locality Sensitive Hashing (LSH) techniques or the use of hierarchical multiple indexes, allowing query-to-query mapping and thus the implementation of a recursively query process. In this process, an initial broad query is successively filtered in more focused queries, until the desired data is retrieved.

Although these approaches provide promising results in terms of efficiency for keyword-based searches, they still require rather complex solutions and consist on full P2P implementations, meaning that only P2P clients can use it. In this paper we propose a solution adopted within the European-funded project

MOSAICA¹, which is more flexible and ensures wider usage. In fact the MOSAICA project has broader goals, going even beyond offering semantic or keyword-based search functionality for native P2P clients. One of the objectives of MOSAICA is to have the broadest possible reach, acting as an enabler for the true access to rich cultural heritage content. This means that it should be possible for any person using a normal Web browser, to gain access to MOSAICA resources. Moreover, MOSAICA envisages the availability of multiple ontologies to enable to appropriately annotate different types of resources and of multiple ways of interacting with the annotated resources, among which using GIS/TIS (Geographical Information Systems / Temporal Information Systems) data. These requirements led to the necessity of implementing the distribution of a number of services, in addition to content search, and exposing these services to the end-user as transparently as possible. Finally, in MOSAICA we were interested in storing rich semantic descriptions and ontologies and offer high availability of this data.

For these reasons – accessibility via normal Web browser, wide availability and large number of services offered - it was decided to adopt a different approach for the search of content, relying on the use of distributed databases and Web Services, while still using DHT for the distribution of content in a P2P network. With this approach, it is possible to offer access to services and to content to any user while still providing efficient upload and download operations.

The developed system, the MOSAICA peer-to-peer content management system (p2p-cms) has thus adopted a 2-layer approach, implementing on the top layer, among other, functionality to perform the distribution of metadata using distributed data bases and accordingly to perform metadata-assisted distributed search for content. The top-layer is based on Web Services and JXTA [3] protocols and it implements functionality, adopting a Service Oriented Approach (SOA), to expose a number of MOSAICA services to the end-user and to other MOSAICA sub-systems. In particular it supports distributed metadata-assisted content searches, users and services management using distributed relational data bases, access to distributed ontologies and GIS/TIS information. The lower layer deals with the distributed storage of the MOSAICA digital objects, addressing aspects of content caching and replication, routing and location. The lower layer is implement via the Bit Torrent suite of protocols, which provide an efficient implementation of DHT networks [4]-[7]. Figure 1 provides an overview of the 2-layer approach for the MOSAICA p2p-cms.

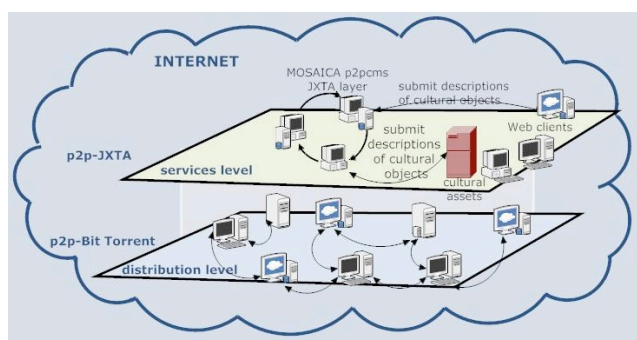


Figure 1 Two-layer approach of the MOSAICA p2p-cms

This paper is organized as follows. Section 2 provides a brief overview of the MOSAICA platform. Section 3 describes the functionality offered by the p2p-cms, presenting details of the formal specification of this sub-system. Section 4 illustrates the use of the p2p-cms within the complete MOSAICA platform. Finally section 5 draws the conclusions.

2. THE MOSAICA PLATFORM

MOSAICA is a European-funded project that aims to distribute cultural contents to promote and increase tolerance between people. To accomplish this goal, MOSAICA uses the P2P paradigm to connect users and provides a Web2.0-based toolbox dedicated to the preservation and presentation of cultural heritage. The

¹ IST-034984 MOSAICA is co-funded by the European Commission under the IST Programme <http://www.mosaica-project.eu/>

complete system provides a framework for any user to actively engage in preserving their heritage, by providing easy access to content and offering tools to navigate through this content, as well as to annotate it and build stories. Figure 2 illustrates the conceptual architecture of the MOSAICA system.

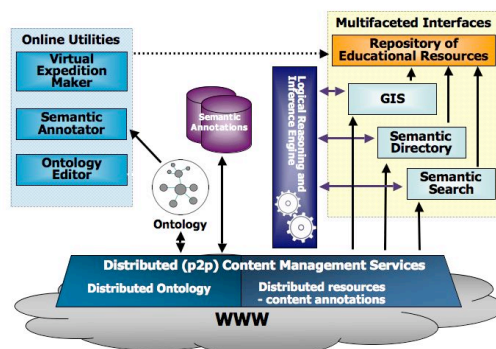


Figure 2 - Conceptual architecture of MOSAICA

MOSAICA outcomes can be utilized as a cognitive tool in three different ways:

- Explorative usage – Users will be able to visit places that evoke their interest and motivation, by merely zooming in on a particular area on MOSAICA's geographical, or by exploring MOSAICA semantic directory, or by submitting a query.
- Guided usage – Rather than starting exploration “from scratch”, users will be enabled to select ready-made, thematically-oriented Virtual Expeditions that will guide them through the virtual Worlds of MOSAICA. MOSAICA digital repository, distributed within the P2P network, thus comprises not only individual digital objects but also a variety of recommended Virtual Expeditions.
- Collaborative usage - users will be able to annotate digital cultural objects, either using free-text comments, or by semantically annotating them using the MOSAICA Ontology; these objects, owned by the user, can then be submitted and exposed to the public; moreover, users can design and record their own Virtual Expeditions, using digital resources already available in MOSAICA, and suggest them to other visitors by storing them in the MOSAICA P2P repository.

In MOSAICA, resources consist of content and ontologies that are stored in a purely distributed fashion. Access to them is possible via its peer-to-peer content management system (p2p-cms) together with its Distributed Ontology Alignment Service. The p2p-cms Service ensures high availability of data and satisfactory transfer performance, by embracing the P2P paradigm and practices such as the use of replication mechanisms and structured overlay networks. The Distributed Ontology Alignment Service, by providing explicit specifications of the existing conceptualizations and underlying semantics within the fields of cultural heritage, enables ontology alignment and thus the possibility to operate across domains.

3. THE MOSAICA P2P CONTENT MANAGEMENT SYSTEM

The p2p-cms is the MOSAICA subsystem responsible for enabling the access to the MOSAICA resources (contents and associated metadata), which are distributed throughout the Web. A semi-decentralized two-layer architecture was adopted, due to the reasons presented above in section 1. The top-layer is based on Web Services and JXTA protocols, which provide a suitable framework for establishing the bridge between higher-layer Web Services and lower-layer P2P implementations. The main function of the upper layer is to offer the end-user functionality to perform metadata-assisted content search, which is accomplished through the implementation of a set of services accessible via a Web browser and using distributed relational databases where semantic annotations are stored.

The principle behind this architecture is that the operations related with the submission and search of digital objects using metadata and the access to MOSAICA services are functionally/logically separated from the operations concerning the distributed storage of the MOSAICA digital objects.

The search is based on the distribution of metadata associated to MOSAICA digital objects and on the use of the ontology alignment service. It uses the concept of distributed databases and transparent replication and distribution of MOSAICA services. When new content is to be submitted to MOSAICA, the p2p-cms receives the metadata associated to the new digital object and stores it in the distributed database implemented in the JXTA network

The storage of digital objects and retrieval of selected content is based on P2P concepts using the Bit Torrent protocol. The choice of this protocol was motivated by the fact that it is presently one of the most popular P2P DHT-based implementations, offering good efficiency [4]-[7] and freely available client implementations for which it is possible to develop plug-ins to incorporate custom functionality, as required.

Digital objects being submitted to the MOSAICA system are partitioned into identically sized pieces (usually 256 Kbyte long), which are then distributed, stored and replicated across a network of peers. The location of the different pieces and consequent re-assembling of the complete digital object is possible through the generation and use of a file designated of “torrent”. This file contains information about the different pieces, and about the peer that coordinates the digital object’s distribution. A peer that wants to obtain a given digital object must first obtain a torrent of that object. This is accomplished as a result of a successful search and selection of content using the p2p-cms upper layer functionality. After being in possession of the torrent file, the peer connects to the tracker specified in the torrent. The tracker then provides a list of peers from where the various pieces of the file can be downloaded.

As illustrated in Figure 3, this architecture defines MOSAICA peers, which differ from MOSAICA users. MOSAICA users can be simple clients using the semantic layer applications of MOSAICA (semantic annotator, semantic browser, Virtual Expeditions browser and GIS browser) that, without the need for any of the p2p-cms modules installed, can access and consume the MOSAICA distributed services exposed by the MOSAICA peers using standard Web Services technologies. However, there is also the possibility that these MOSAICA users become part of the P2P network, offering part of their own storage capacity to the overall MOSAICA distributed storage. To achieve this, these peers need only to install a simplified version of the p2p-cms. This version contains basically only the Bit Torrent client and interface modules with the MOSAICA application. A specific package was developed for this purpose with a self-installation procedure (Windows-like). On the other hand, MOSAICA peers are those that have installed all the modules of the p2p-cms, and that thus offer the full functionality of this subsystem.

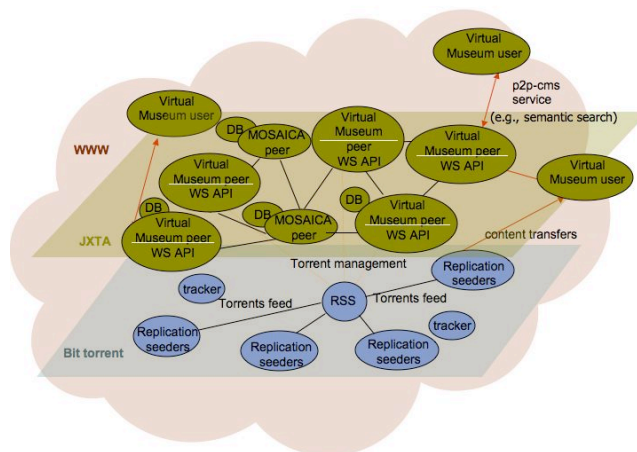


Figure 3 - Peers in the MOSAICA p2p-cms

3.1 High-level componential design of the p2p-cms

The componential design adopted for the MOSAICA p2p-cms follows the layered peer architecture represented in figure 4 below.

In each layer, functionality associated with the use of the JXTA platform and that with the use of the BitTorrent network are grouped and implemented by different subsystems. Figure 5 presents this architecture in a more formal way. It is thus possible to see that the MOSAICA p2p-cms is composed of 4 main subsystems. The sub-systems JXTA-WS (JXTA Web Services) and BT-WS (Bit Torrent Web Services) operate at the interface level, whereas the sub-systems MOSAICA-JXTA and MOSAICA-BT operate at the core level, implementing the core functionality of the p2p-cms for the management of content, metadata and users. They directly use the functionality of the JXTA platform and that of the BitTorrent network. The former sub-systems expose the core functionality of the p2p-cms to the MOSAICA upper layer systems, namely the Semantic Browser and the Semantic Annotator, through a Web Services based interface. Internal interfaces are implemented using JAVA RMI.

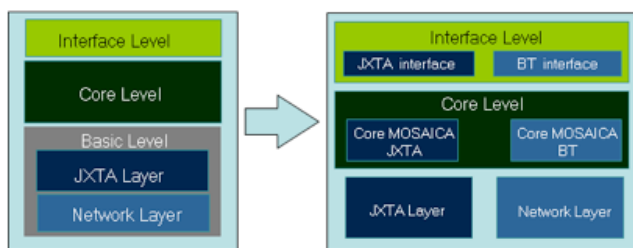


Figure 4 - Layered peer architecture

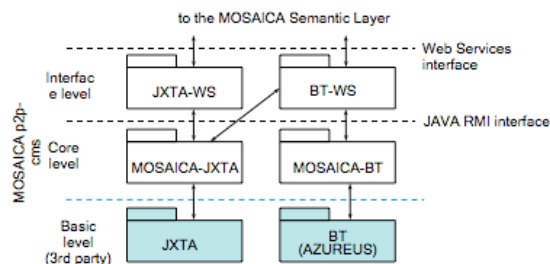


Figure 5 - Formal representation of the p2p-cms components

3.2 Use case model of the p2p-cms

Three main different actors have been identified as the entities establishing interaction with the p2p-cms:

- MOSAICA advanced user
- Sharing user
- Semantic browser

Administrative-like use cases, such as the validation of a resource being submitted to the MOSAICA p2p-cms, are performed at the semantic layer. Likewise, user authentication and assignment of users to groups, is left to the MOSAICA application layer. It is therefore assumed at all the requests that arrive at the p2p-cms have already been validated by the MOSAICA application. If a request to upload content to the MOSAICA platform arrives at the p2p-cms, this MOSAICA sub-system assumes that all user validation tasks have already been successfully conducted and that the user has the right to perform the operation he/she requested. Moreover, that the content being submitted has already been validated. The p2p-cms still implements a login operation, assigning an identification to each user and thus being able to create sessions assigned to the user and cross-match results with each session-user pair.

The actor *MOSAICA advanced user* can use the value delivered by the p2p-cms to perform operations such as register a new digital object or initiate a semantic metadata-assisted search for content, among others. Table 1 indicates all the use cases where this actor is involved.

Table 1 - Description of use cases initiated by the actor MOSAICA Advanced user

Use case name	Overview
Register content	The user invokes this use case providing information concerning the torrent and metadata associated to the new resource to be added to the MOSAICA platform. Torrent binary and metadata are sent in XML. The semantic metadata and the torrent (if it already exists) is uploaded to the distributed data base in the JXTA network. The newly registered content is given a unique content identifier.

Upload content	The user sends a request to submit new content to the system. The resource binary file must be placed in a special local shared folder (mosaica_shared). If the content hasn't yet been registered, then the registration in the JXTA layer is performed (using the value delivered by the use case "Register content") and a content identifier is assigned to that resource. The torrent file is created with the name <contentID>.torrent and the Bit Torrent peer starts seeding it. Finally, torrent base64 data is added to content data in JXTA network.
Search content	The user sends a query composed by keywords and Boolean operators. An XML list of content identifiers that matched the query is sent back to the user.
Request content	The user requests additional information about a specific resource, identified by its unique content identifier. The p2p-cms retrieves from the distributed database the available information for that content_id. It sends back to the user an XML string containing metadata and the torrent associated to the requested resource. This information can then be used to actually retrieve/download the resource from the Bit Torrent network.
Download content	The user requests the download of a selected resource. Using the content identifier associated to that resource, the system obtains from the JXTA layer, the corresponding torrent (a binary base64 data file). The torrent is saved to the shared folder (mosaica_shared) and the AZUREUS client starts the download. Once the resource has been completely downloaded
Remove content	The user requests the deletion of existing registered content. The information associated to the indicated content is removed from the distributed database in the JXTA layer.
Add metadata	Adds a metadata entry (indextype, property, value) in the distributed database, to an existing resource
Remove metadata	Removes a metadata entry (indextype, property, value) from an existing resource
User Management	User management functionalities.

The actor *Sharing user* represents the simple P2P application that was developed to be easily installed in any machine running the MOSAICA tools. This application comprises an Azureus Bit Torrent client and some extra modules that were developed to provide the interface between the MOSAICA application and this client. When installed together with the MOSAICA application, it allows the peer to contribute in a seamless manner with storage space to the overall distributed storage capacity of the MOSAICA p2p-cms. Accordingly, this actor only performs operations concerning the reading of torrent feeds and the automatic download of corresponding files. Functionality has also been implemented to allow defining the sharing storage space, indicating to the Bit Torrent client the maximum amount of available storage in the peer for altruistic downloads. The use cases where this actor is involved are described in table 2.

The actor *Semantic Browser* intervenes only in those cases where the interactions are made in an automatic way, without the actual intervention of the MOSAICA end-user. This happens in the use case "Is downloaded", which offers functionality to allow the Semantic Browser to know when the download is completed. The content being downloaded from the P2P network is temporarily stored in a local folder, shared between the Semantic Layer and the p2p-cms. When the resource is completely downloaded, the Semantic Browser can access this folder and present the resource to the user. The use cases where this actor is involved in are described in table 3.

Table 2 - Description of use cases initiated by the actor Sharing user

Use case name	Overview
Add torrent feed	A new torrent is added to the RSS server. This use case is automatically performed by the p2p-cms when the "Upload Content" use case is invoked and a new torrent is created.
Remove torrent feed	The content torrent, identified by the content identifier, is removed from the RSS server. This use case is automatically performed when the use case "Remove Content" is invoked.
Read torrent feeds	This use case is automatically implemented by the system in a periodic way, in those peers that have been set up to act as seeders of the content (those peers that have been set up to donate a given amount of their own storage capacity to the P2P network). When this use case is invoked, a list of RSS feeds is sent back to the calling peer, with information about the link to download the torrent and the title of the corresponding resource.

Table 3 - Description of the use case initiated by the actor Semantic Browser

Use case name	Overview
Is downloaded	After the MOSAICA advanced user has requested the download of a given resource (through the use case “Download Content”), the Semantic Browser periodically invokes the “Is downloaded” use case, to know when the download is complete .

4. USING THE P2P CONTENT MANAGEMENT SYSTEM

The p2p-cms offers a Web Services based interface to the MOSAICA upper layers. This means that the value delivered by the use cases implemented at the core level of the p2p-cms can be directly accessed through this interface. Accordingly, this WS interface enables the Semantic Layer, on behalf of *the MOSAICA Advanced user*, to invoke specific methods that allow to submit, locate, retrieve and remove resources. The previous sections of this deliverable have already provided a description of this interface.

During the execution of these use cases, metadata resources are directly exchanged between the two sub-systems in the form of XML strings, as arguments of the invoked methods and consequent replies (in the body of the SOAP messages). However, this approach is not suitable to exchange content resources, which typically have larger sizes. Accordingly, it was devised a different approach to transfer these resources between the application layer and the P2P layer. This approach maintains the transparency between the Semantic Layer and the Bit Torrent P2P modules, which are the ones that effectively manipulate the resources across the MOSAICA distributed storage platform. Figure 6 presents the implemented solution.

The submission of a new digital object to the MOSAICA p2p-cms, requires that the object has been previously annotated using the MOSAICA Semantic annotator. The content to be submitted must be registered in the JXTA layer and the previously created semantic metadata is inserted in the distributed database implemented at this layer. If the user is uploading the actual binary file of the resource, then an associated torrent must be created and inserted together with the metadata in the distributed database. However, the user may as well submit only the link to the resource in the form of a URI (for example, a URL). In that case only the URI is inserted in the database together with the semantic metadata and the submission process is completed with the “Register Content” use case.

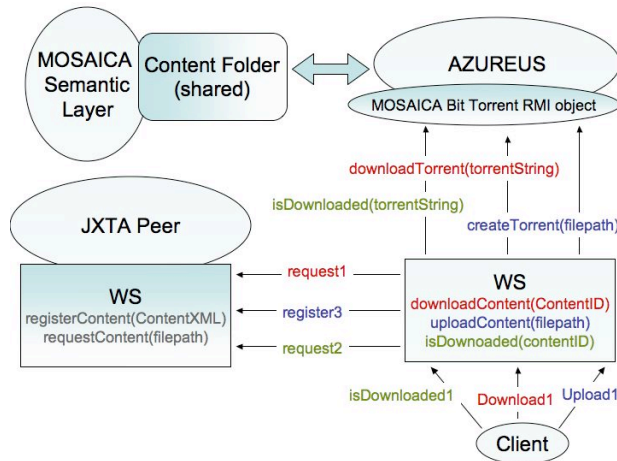


Figure 6 – Integration approach between the p2p-cms and the MOSAICA semantic layer

When the user is submitting the actual resource file, it is necessary to 1) create the associated torrent; 2) announce the new content in the Bit Torrent network; and 3) initiate the seeding of it. This is accomplished with the “Upload Content” use case. The torrent file is automatically created during the “Upload Content” use case, if a binary file is placed in the local folder designated as “mosaica_shared” (represented in figure 6). This is a local folder used both by the Semantic tools and the p2p-cms. The p2p-cms locates the binary file in that folder and automatically invokes the AZUREUS Bit Torrent client to generate a corresponding torrent.

This torrent is stored in the distributed database and a corresponding feed is added to the RSS server. This content starts then to be spread across the MOSAICA P2P network in two possible ways: 1) a download request specifically for that content is issued by a MOSAICA advanced user; 2) automatically by MOSAICA peers which are set to periodically check the existence of new feeds in the RSS server (using the “Read torrent feeds” use case).

The use case “Search content” requires that the peer where this use case has been invoked (the initiating peer) propagates the query in the JXTA layer, performing at the same time a search into its own local data base (in case this local data base exists). It should be however noted that, whereas the metadata database of the p2p-cms is implemented in a distributed fashion across the MOSAICA JXTA peers, some peers may not have a local database. Still, nothing prevents from having all MOSAICA peers with a local database as a part of the overall MOSAICA distributed database.

The information contained in each local database may not be complete or updated. When a peer receives a query, it performs a local search into its own database, if present, and forwards the query to other peers in the JXTA network to get additional information. The initiating peer re-assembles all the partial results and constructs the complete list of results to be returned to the user, matching his/her query. When different peers return overlapping results, a timestamp is used to verify the most recent version of the associated information. The local database, if present, should then be updated with the most recent information received.

Normally, two periods are considered during the execution of a “Search Content” use case. The initiating peer waits for a given period before presenting an initial list of processed results to the user. The duration of this period can be configured manually. When this first period ends, the initiating peer automatically launches a thread to continue to receive results from the network. The lifetime of this thread constitutes the second period of the search use case and can also be configured manually. During this second period the data base of the initiating peer, or its current list of results, continues to be updated with incoming results. When this second period ends, the final (updated) list of results is presented to the user.

As mentioned above, it is not necessary to have a database in all peers. Only peers explicitly configured to store and answer queries (normally rendezvous peers) should have a local database. Peers that do not have a database, upon receiving a request to search for content, send (propagate) the query to the JXTA network and wait for answers, as explained above. Operations to insert, remove and update information on the database are propagated over the JXTA network using specific routing protocols.

The value delivered by the use cases concerning the manipulation of torrents and briefly described in table 2, is implemented using an RSS server, which is also seen as part of the MOSAICA p2p-cms, more specifically within the module “MOSAICA-BT” of the p2p-cms core layer described in section 3. It is well known that the Bit Torrent network delivers better performance, in terms of downloading times, when there is high demand for a given resource [8]. This means that the more a resource is replicated across the network, the more efficient will be its retrieval. Accordingly, as soon as a new resource is available within the MOSAICA platform, the corresponding torrent feed is added to the RSS server (through the use case “Add torrent feed”). Peers within the MOSAICA network, which have been accordingly configured, may periodically check the RSS server announcements (through the use case “Read torrent feeds”) and start automatically downloading the new available resource. As explained above, the sharing user can be seen as the simple application that was developed specifically to allow any MOSAICA peer to contribute in an altruistic way to the overall distributed storage capacity of the MOSAICA P2P network and, consequently, to its better performance in terms of download time.

Although it is the actor Semantic Browser that initiates the use case “Is downloaded” as described in table 3, this actor is not represented in the use case diagram of figure 4 for simplistic reasons. Additionally, the execution of this use case can only occur if a “Download Content” use case has been performed. Nevertheless, it was felt worth indicating and describing a little further this use case, to allow a better understanding of the complete downloading process and the way the resource actually arrives at the user, given that the user does not directly interact with the P2P modules. The idea is precisely the one of hiding all the eventual complexity associated to the P2P mechanisms from the end-user as well as avoiding the necessity of exposing different interfaces to the user. The main aim of MOSAICA is to provide the user with easy to use tools that allow him/her to semantically annotate and search for digital multimedia resources, as well as to build and navigate through appealing virtual expeditions. In order to use these tools, the user needs not to be aware of any of the specificities of the storage mechanisms of MOSAICA. Accordingly it was decided that resources being submitted to or being retrieved from the MOSAICA storage infrastructure, would be exchanged between the Semantic layer and the p2p-cms layer through a common local folder. The

use case “Is downloaded” enables the Semantic Browser to know when the requested download of a given resource has finished, so that it may retrieve it and present it to the end user through the MOSAICA GUI.

5. CONCLUSIONS

The analysis performed on the current state of the art in P2P technologies and the investigation conducted on emergent research work on semantic-based search in P2P systems, have guided the approach taken and the concepts adopted for the specification of the MOSAICA p2p-cms. Accordingly it was decided to support the semantic-based search using distributed databases. Moreover, it was decided that the system should be implemented as a 2-layer architecture where the high-level functionality would be based on the open-source JXTA platform and the lower layer on the Bit Torrent network. The devised layered and structured architecture of the p2p-cms, offering its value to external actors via a Web Services interface, is very flexible. It enables different implementations of the same application or even different applications, to use the value delivered by the p2p-cms. The fact that all P2P specificities are hidden to the application layers is also an added value. Together with the fact that the p2p-cms has a well-defined structured and modular architecture, enables to easily replace modules, maintaining the same interface but using different P2P technologies to effectively implement the distributed storage of the MOSAICA content.

Moreover the use of the two-layered approach, by which metadata is manipulated in one layer and content in the other layer, enables to obtain an optimal solution: on one side, it enables to take advantage of the great efficiency of DHT (Distributed Hash Tables) for the distribution of and access to binary resources; on the other hand, through the implementation of distributed databases for managing metadata, it enables to overcome the limitation of DHT-based P2P networks for performing semantic-based searches.

ACKNOWLEDGEMENT

The work here described has been developed within the framework of the MOSAICA project. The authors would like to thank their colleagues in the project for the fruitful collaboration.

REFERENCES

- [1] Jie Wu, Handbook On Theoretical And Algorithmic Aspects Of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, Auerbach Publications, Boston, MA, 2005.
- [2] Yingwu Zhu and Yiming Hub, “Efficient semantic search on DHT overlays”, Science direct, Journal of Parallel Distributed Computing, no. 67, pp.604-616, February 2007
- [3] JXTA v2.0 Protocols Specification. <http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.pdf>; accessed in February 21, 2007.
- [4] Dongyu Qiu and R. Srikant, “Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks”, in Proceedings of the ACM SIGCOMM’04, International Conference, Aug. 30–Sept. 3, 2004, Portland, Oregon, USA.
- [5] J. Pouwelse, P.Garbacki, D. Epema, H. Sips, “The Bittorrent P2P File-sharing System: Measurements and Analysis”, in Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS 2005), February 24-25, 2005, Ithaca, NY, USA.
- [6] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, “Measurements, Analysis, and Modeling of BitTorrent-like Systems”, in Proceedings of the ACM Internet Measurement Conference (IMC 2005), October 19-21 2005, Berkeley, USA.
- [7] A. Bharambe, C. Herley and V Padmanabhan, “Analyzing and Improving BitTorrent Performance”, in Proceedings of the International Conference INFOCOM 2006, April 23-29 2006, Barcelona, Spain.
- [8] P. Felber, E. Biersack, L. Garces-Erce, K.W. Ross, G. Urvoy-Keller, “Data Indexing and Querying in P2P DHT Networks”, in Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS 2004), 23-26 March 2004, Tokyo, Japan.