We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



3

The Use of Multi-Agents' Systems in e-Learning Platforms

Tomasz Marcin Orzechowski
AGH University of Science and Technology
Poland

1. Introduction

E-learning systems become more and more popular in many areas of education. They enable distance learning as well as allow digital content creators to establish specialized training centres. Such systems are commonly used by numerous governmental organizations as well as commercial content vendors. All these systems share a common purpose: they provide access to on-line learning resources for education. These resources are used by teachers or instructors to create courses in a given subject or topic. As browsing such repositories is inconvenient, inefficient and sometimes impossible, various search engines have been developed to help users retrieve specific data. However, often it is not possible to formulate the search queries alone so that they can pick out the elements, which are of value to a user. The problem of ordering search results by predicted level of interest for the user pertains both to general-purpose Internet search engines and to browsing assistants in specialized repositories. It gets even more severe if elements are very similar as far as their keywords or descriptions are concerned.
The increase in the amount of resources available in e-learning systems will require the need for designing new architecture of such systems. The most important is to provide the ways for sharing resources in accordance with copyrights and to offer possibilities to rearrange course contents consisting of elementary resources.
The main purpose of the presented system was to design and implement the advanced search system, which will allow teachers and pupils to obtain results according to their profiles and interests.
This work was partially supported by the European Commission under the Information Society Technologies (IST) program of the 6th FP for RTD as part of the CALIBRATE project, contract IST-28025.

2. State of Art

2.1 E-learning Systems

The term "e-learning" is commonly defined as this kind of educational approach, where ICT (Information and Communication Technology) is used.

Computers are the basic equipment used in the e-learning process. Different kinds of educational software created and verified by domain specialists and educators allows the implementation of the principles of education, which are the individualization of the trainings and teaching by examples.
Technical capabilities make possible a visualisation of knowledge as well as the alignment of both the pace and structure of the education content to the individual recipient's perception.
For example, existing applications, already used in preschool teaching, successfully support the learning of colours, shapes, letters, and perceptivity. Popularity and fast access to the Internet extend learning opportunities both for access to new resources and allow for different kinds of contact such as voice calls, chats, and video transmission among users of the system.
The authors distinguish 2-4 main types of training using e-learning techniques. In this example, Schulmeister (2003) distinguishes between individual – type A and group training – type B, closely related to the form of materials (Fig. 1).

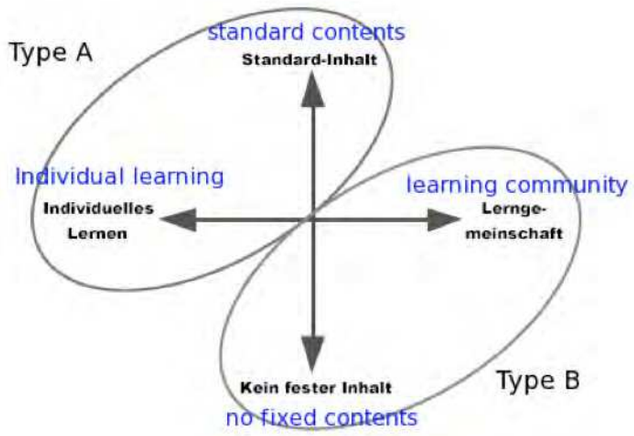


Fig. 1. E-learning types *(orig. "Typisierungvon Lernumgebungen" Schulmeister, 2003, p.16)*

According to Schulmeister (2003), individual trainings – type A – is person-to-standardised content interaction, whilst type B – person-to-person interaction – is the acquisition of knowledge in interaction with members of specific communities of practise (Schlager & Fusco, 2004).
This categorisation is similar to Michael Graham Moore's (1989), considered as an educational theory classic, who suggested three kinds of interactions:

- *learner – content*
- *learner – instructor*
- *learner – learner*

Moore acknowledges as the crucial interaction between student and instructor, who is a domain specialist. This specialist has impact on increasing student's motivation to the course, and acceleration of interest to the presented contents. This specialist is responsible for combining theoretical knowledge with practice to make the knowledge be better assimilated. The amount of taught knowledge is constantly verified during the teaching process.

Hillman, Willis and Gunawerdena (1994) have expanded the list of interaction by adding the learner – interface interaction. They indicate the importance of electronic equipments used in the education process.
Another author also divided e-learning processes into the following three groups:

- *self-study*
- *synchronous education*
- *asynchronous education*

We can also distinguish blended-learning, which is actually the classical learning process extended by some e-learning elements.
The basic unit of electronic content is called Learning Object – LO, which may be either elementary content (Learning Asset) such as: a single image, video, or text file, or may consist of several other LOs respectively associated with each other.
Complex LO could form a single unit such as classrooms, and even the entire course.
A well-described LO by IEEE LOM Standard (IEEE 1484 Learning Objects Metadata, 2005) can be shared not only by members of the given e-learning platform but accessible from various platforms. Currently, there are many standards that describe requirements for various e-learning platforms that can share resources.
The most popular are SCORM (Sharable Content Object Reference Model), specifications provided by IMS GLC, or PENS (Package Exchange Notification Services), or the oldest recommendations defined by AICC (Aviation Industry CBT Committee).

E-learning courses are created, managed, and used facilitating the existing e-learning management systems belonging to following main categories:

- LMS – Learning Management System (Fig. 2);
- CMS – Content Management System (Fig. 3);
- LCMS –Learning Content Management System (Fig. 4).

Sometimes, the following category is also distinguished:

- VCS –Virtual Classroom System.

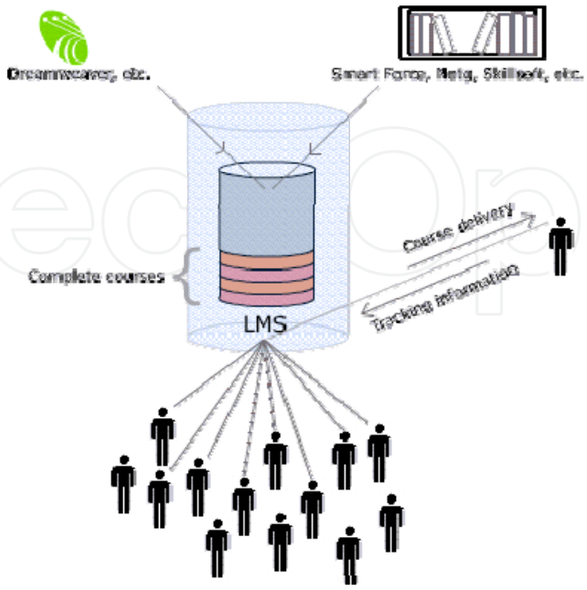


Fig. 2. General Structure of LMS (Nichani, 2001)

LMS is a family of systems, which enables management of all training activities as well as provides presentation of content associated with courses. In particular, LMS systems categorize the users, give them certain permissions to training modules and assign users to specific training groups. In addition, LMS systems offer the ability to monitor the progress of knowledge and skills learned by individual students (Nichani, 2001; Rengarajan, 2001).

CMS is a family of e-learning systems dedicated to creating, storing, managing and presenting content. Significant in this model is to provide opportunities for reusability of Learning Objects, called RLO (Reusable LO) (Nichani, 2001) or content components.
Another feature of CMS is the separation of content from layout (Nichani, 2001).

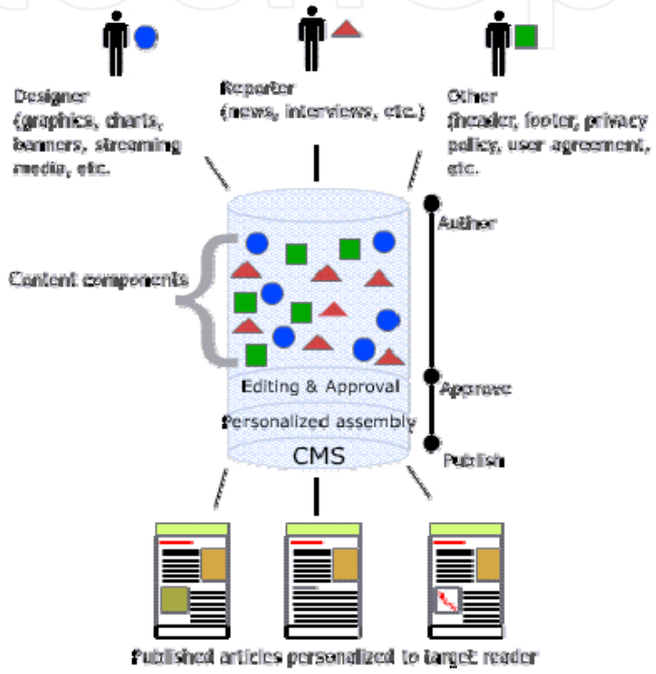


Fig. 3. General Structure of CMS (Nichani, 2001)

It is noteworthy that the LMS can be used only for the teaching process – presenting content and management of students, whilst the CMS assists users to create the content from available elementary components. Development of CMS systems is an example of B2C services (business-to-consumer). Resources collected by these systems may be offered to external institutions.
LCMS is a family of the most technologically advanced e-learning systems. LCMS systems integrate features of LMS and CMS (Nichani, 2001), ensuring both the creation and management of educational content. They provide opportunity for the creation of WBT courses (Web-Based Training) as well as the management and the management of the learning process, taking into account the evaluation of knowledge assimilated by the users (Nichani, 2001; Rengarajan, 2001).

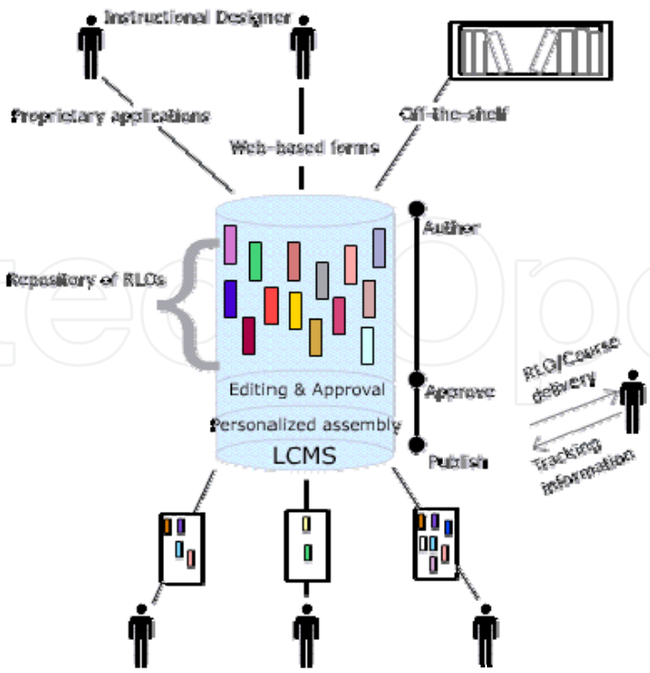


Fig. 4. General Structure of LCMS (Nichani, 2001)

The impact of e-learning trainings on education processes will be expanded. This is caused by fast technology development and the efficiency of this form of training.
Marschall McLuhan Herbert, media specialist, who lived in the period 1911-1980, noted that the media are a key factor in any changes in society. His theory that "the medium is the message" indicates a significant impact on the nature of the medium both on the transmitted content and the quality of the content reception. Media, according to him, may "exclude" (hot media) or "engage" the recipient (cold media).
E-learning systems are able to use the described effect of the transfer of content involving the user's perception almost entirely. Users in constant interaction with content can assimilate the knowledge contained in the received content faster and more efficiently.
The correctness of this prediction was confirmed by the analysis carried out by us at IBM, which pointed out two main advantages of e-learning solution: the effectiveness of training and a real reduction in the cost of training. E-learning trainings allowed this company to reduce training costs by 50-70% and shorten the training time from 11 to 9 weeks, while the productivity of educated workers grew to 100-170%.
It is therefore not surprising that e-learning techniques are becoming popular and are promoted throughout the world.
European Union promotes e-learning techniques under the EU Structural Funds (2007-2013), and within the framework of the "Lifelong Learning Program (LLP).
There are two main problems to be solved when we decide to make a semi-global open e-learning system. The first is: "how to describe and register learning objects (LOs)". The second is: "how to find the LOs that the given user is really interested in".
Independently in both of these two situations we can consider a centralised or decentralised model. The centralised systems are better managed but they are not safe in the case of

scalability. On the other hand, the decentralised systems can solve the problem of scalability but they cannot be managed simply.
A solution is to create a hierarchical structure (semi-centralised) and use the multi-agents' platform to manage this structure and arrange all offered services (Orzechowski & Dziech, 2001a; Orzechowski, 2001b; Orzechowski, 2002).

2.2 Learning Objects in the federation of repositories

The need to provide educational exchange of data has been known for years. The best-known initiatives include:

- ARIADNE Foundation in Europe (www.ariadne-eu.org);
- Multimedia Educational Resources for Learning and Online Teaching (MERLOT) in USA (www.merlot.org);
- LORNET in Canada (www.lornet.org);
- education.au in Australia (www.educationau.edu.au/jahia);
- National Institute of Multimedia Education (NIME) in Japan (www.nime.ac.jp)

These organizations have set up a joint group: "The Global Learning Objects Brokered Exchange" (GLOBE), intended to integrate all standardization work to create and develop open standards for educational data exchange in one integrated brokerage environment.
GLOBE initiative has been met with great interest and now consists of many new members. The most influential institutions are: COSL, EUN, III, KERIS and LACLO.

Two basic standards enabling search in repositories are:

- SQI – Simple Query Interface defines how to access and exchange data among repositories (Simon et al., 2005);
- OAI-PMH – The Open Archives Initiative Protocol for Metadata Harvesting creates independent interoperability framework based on metadata harvesting *(www.openarchives.org/OAI/2.0/openarchivesprotocol.htm)*.

SQI interface is used to transmit queries and results received during the session between the client and repository of metadata. It is fully independent from both query language and format of returned results. Although it is possible to implement SQI to work with any data representation, XML is the most popular. Usually, SOAP, RPC, or RMI are used as exchange protocol between clients and repositories. Session management of SQI is equipped with optional authentication mechanisms. It also supports working in synchronous and asynchronous mode for LOM retrieval, which is important when federation search is used to search the whole federation of repositories at once.
Schema of SQI-oriented communication among clients and repositories is shown in Figure 5. (Simon et al., 2005)

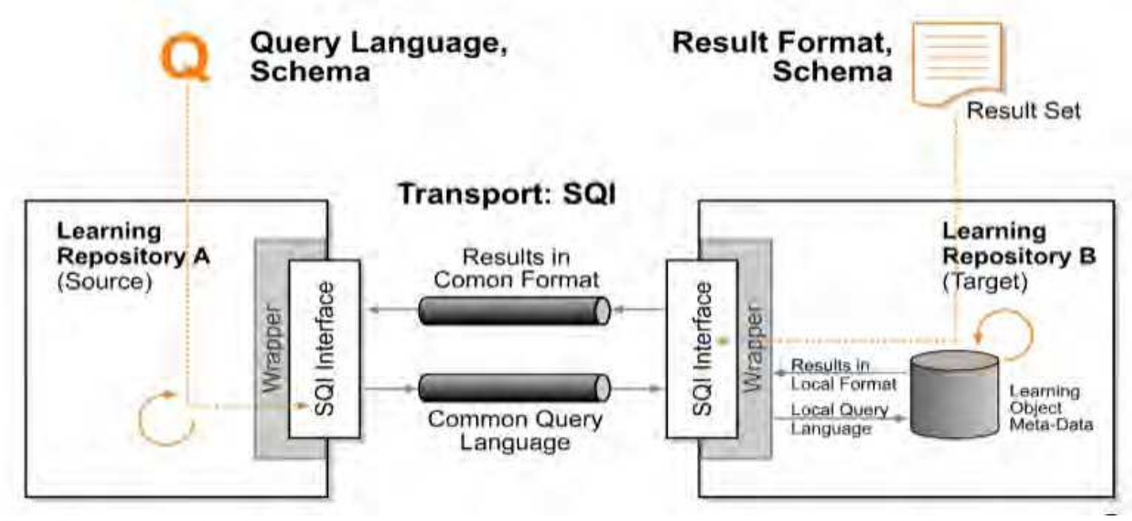


Fig. 5. Communication between components of the system based on the SQI

SQI specification was officially approved during the CEN/ISSS workshop on Learning Technologies in July 2005 by the group of experts working in various EU projects concerning federations of learning resources topics, most specifically: Celebrate, Elena, Ariadne, and Edutella.

One of the languages used to create queries sent via SQI interface is S2QL. S2QL (Simple School Query Language) is a simple query language designed specifically for finding learning resources in different types of repositories. It is abstract language, referring to some abstract pattern, which does not reflect the actual data. Queries in this language cannot be directly processed by the repository. They must be converted to the structure, which is supported by the given database engine of the repository.

S2QL queries are limited only to three possibilities:

- keywords;
- age range; and
- language of learning object.

S2QL schema and namespaces are available in the following locations: http://fire.eun.org/xsd/s2ql-1.0.xsd, and http://fire.eun.org/xsd/s2ql-1.0.xsd.

OAI-PMH is a protocol for the selective gathering of metadata describing learning objects. It allows the automatic exchange of information between systems gathering metadata and their clients. Unlike SQI, OAI-PMH doesn't limit the number of found LOM transferred to the client. Moreover, it is especially dedicated to situations where we actually do not want "to search" by keywords but "to gather" all LOMs by their repository ids, or by the date of deletion, modification or storing in the repository. Communication in OAI-PMH is based on client-server structure and uses HTTP protocol where results are delivered as an XML file. This protocol was used in the ABSS presented in section 3 of this chapter.

The Ariadne Network is a good example of a distributed network of repositories. It was developed by the ARIADNE Foundation that was formed to continue the work and further development of EU projects: ARIADNE and ARIADNE II.

There are also other interesting initiatives such as MACE Project (Metadata for Architectural Contents in Europe 2006-2009). Its aim is to establish the integrated access to repositories

containing any data concerning architecture domain, which were the results of other EU research projects such as: DYNAMO, INCOM and WINDS. Actually, MACE project doesn't target to a network of distributed repositories, but to create a single coherent virtual repository of data (Prause et al., 2007; Stefaner et al., 2007). This approach is similar to the one presented in section 3.

The basic element of network structure, Ariadne is a repository that can be searched using SQI. Ariadne Network users can both browse and publish their contents. The structure of the Ariadne network is shown in Figure 6.

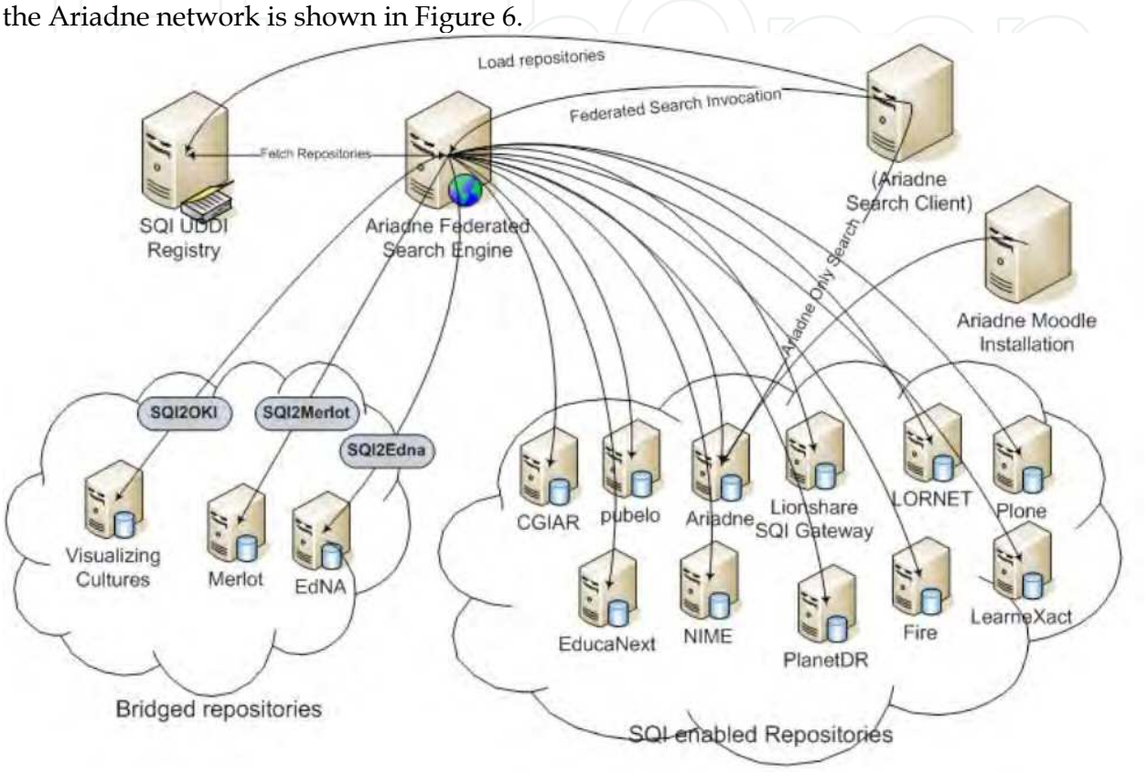


Fig. 6. The structure of the Ariadne Network (http://www.ariadne-eu.org)

Ariadne offers many additional tools, for example Web-Based Learning Environment (WBLE) for the creation and management of educational courses. A person who is responsible for creating a course may use WBLE to create it on the base of LOs collected in the KPS (Knowledge Pool System), which is a distributed database of teaching materials, and then distribute it to students (Van Durm et al., 2001).

KPS consists of resources called Local Knowledge Pool Systems (LKPs) localised in different countries (Najjar et al., 2004). The following APIs were defined to establish the functionality described above: 1. AMI (Ariadne Management Interface) allows teachers access to courses they created, and 2. ALI (Ariadne Learner Interface), which allows access to approved courses (Fig. 7)

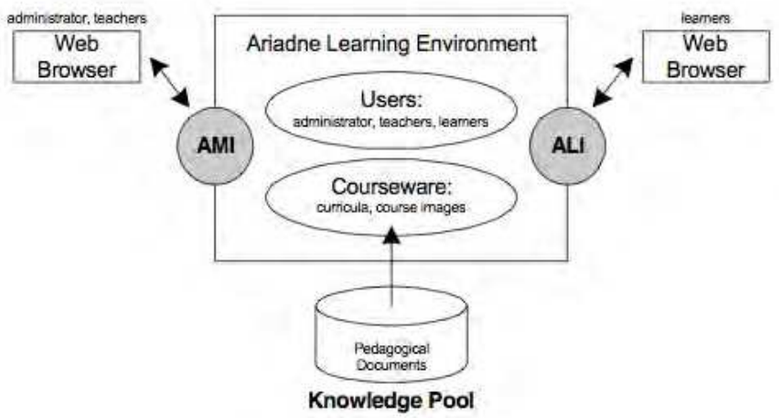


Fig. 7. The Ariadne Web-Based Learning Environment (Van Durm et al., 2001)

An important feature of the Ariadne Network is that it is possible to combine the various repositories in the network by connecting them to the broker. The Ariadne Network is hidden from the end-users' sight under a brokerage layer (Fig. 8).

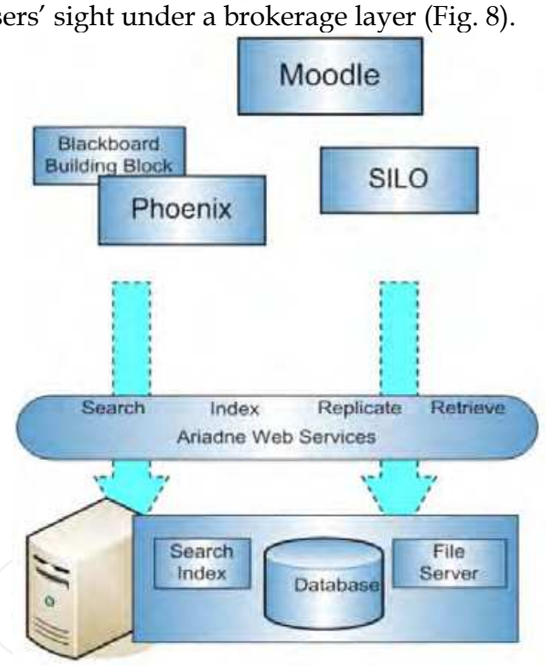


Fig. 8. The schema of communication between clients and the Ariadne Network (http://www.ariadne-eu.org)

Three-layered structure, described above, consists of:
1. Clients – client applications such as tools for queries and indexing, search portals such as PROLEARN, and VLE systems through appropriate plug-ins;
2. Middleware – a layer, which enables Federated Search functionality. It is responsible for sending queries to the repositories, merging received results, sorting them, and then transferring to end-users. Information about repositories connected to this middleware is stored in a UDDI SQI Registry. This register contains information about the active

repositories supporting SQI, and some other additional information such as: - the URL needed to access the Service, - supported query languages, - the type of communication that is supported (synchronous or asynchronous), - contact person responsible for each repository, - a description of the repository, etc.
3. Repositories – Learning Objects Repositories, which are components of the federation and support being searched by SQI.
Federation Search was proposed for the fast search done in real time. It is performed by sending requests at the same time to all repositories connected to the Federation (Simon et al., 2005). All steps of this process are presented in Figure 9.

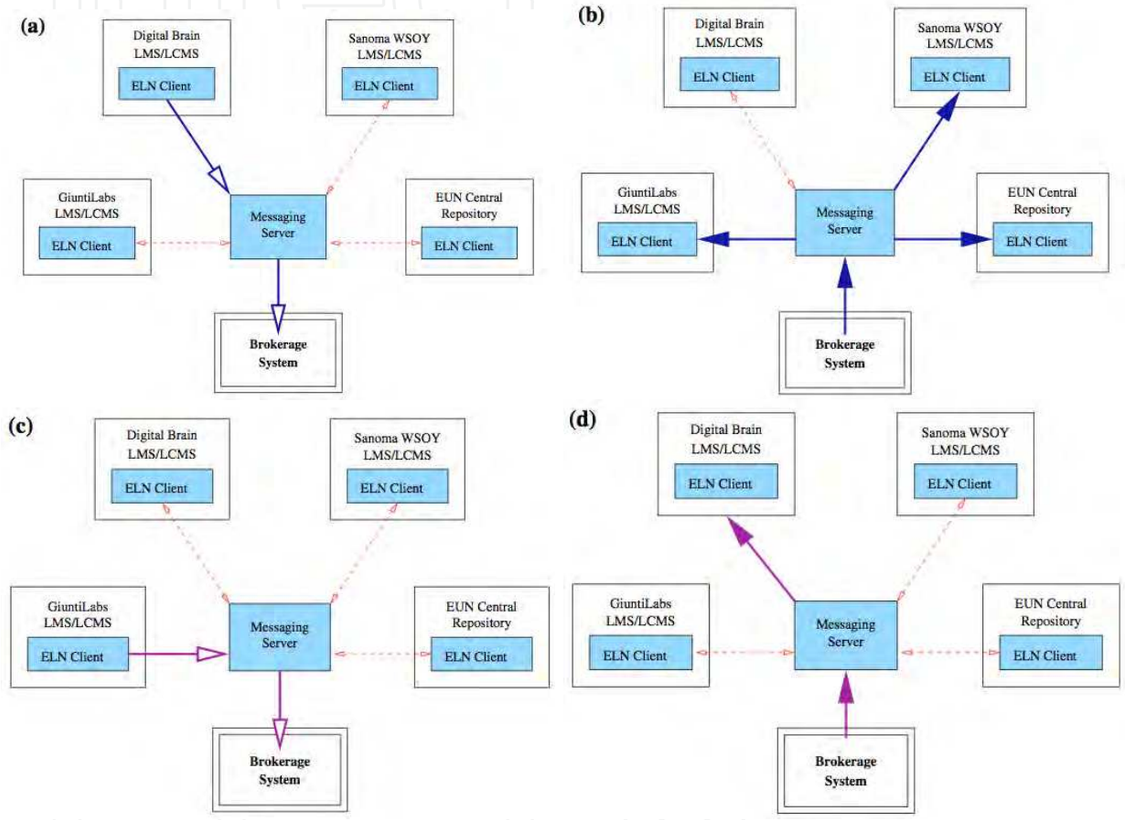


Fig. 9. The schema of Federated Search process (a) sending query by client, (b) distributing query to all repositories, (c) receiving and processing results by broker, (d) sending processed results to client (Van Assche & Massart, 2004).

A well-known disadvantage of this approach is the simple lack of possibility for the introduction of new algorithms into the Federation of Repositories. Implementation of advance search algorithms needs to provide changes done directly at repositories' sites, whilst presented in this chapter, the agents' system approach does not have this kind of limitation.

2.3 Multi-Agents' Platforms

An agent, which is a specific programming entity, is defined as "one who acts" or "one who acts in place of" (Franklin & Graesser, 1997). An agent could be described by the following features: autonomous, reactive, communicative, learning, cooperative, or mobile.

The most important feature of agents is autonomy. According to Franklin & Graesser (1997), "an autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future". Another definition provided by Jennings (1996) is, "the term is usually applied to describe self-contained programs which can control their own actions based on their perceptions of their operating environment".

Multi-agents' platforms are used for efficient agent managing. The platforms are based on an organizational model of multi-agents' platforms described within a standard that consists of the group of specification documents concerning system architecture, agent communication language, etc. Each platform implements a set of specifications of chosen standard.

Nowadays there are two standards for multi-agents' systems available: MASIF (www.omg.org) and FIPA standard (www.fipa.org). MASIF was dedicated to enable agents travelling among agent systems via CORBA interfaces and does not address inter-agent communication, whilst the FIPA defines specifications for agent communication, agent transport, agent management, abstract architecture and applications.

The core category of the FIPA multi-agents' system is agent communication. FIPA's models do not imply any physical configuration and it is not a scope of FIPA standards to deliver strict guidelines for multi-agents' platform implementation. FIPA standard contains Agent Management Reference Model, being a base for developers who want to design and implement own multi-agents' platform. Architecture of the system contains FIPA's specification of mandatory components for platform managing:

- ACC – Agent Communication Channel;
- AMS – Agent Management System;
- DF – Directory Facilitator.

All of these three components are automatically activated at the agent platform start-up.

There are several FIPA-compliant agent platforms. Very popular, commercial and well described is Tryllian Agent Development Kit. There are also some Open Source Agent Platforms, such as in the examples: April Agent Platform (agents must be written in April programming language), Comtec Agent Platform (it is not developed now), FIPA-OS (not developed since 2003), Grasshopper (official web page is no longer accessible), JADE and JADE-LEAP.

Java Agent DEvelopment Framework *(jade.tilab.com)* is written in Java Open Source and very advanced agent platform, which is fully compliant with FIPA 2000 standard, implemented in pure Java (Bellifemmine et al., 1999). It offers agents mobility, and a highly efficient communication interface based on several different protocols (Chmiel et al., 2005; Shakshuki & Jun, 2004).

The JADE-LEAP extension to JADE allows a wide variety of devices, like mobile phones, to connect to the platform. It also improves communication by adding different protocols including SSL secured ones.

The JADE platform is under continuous development. It was successfully used by many research and commercial organizations including INRIA, Nice-Sophia-Antipolis, ACACIA

research team, ATOS Sophia Antipolis agency within the European CoMMA project, CSELT, KPN and Starlab within DICEMAN project, and many more, so the Agent Based Search System presented in the next section was established on the basis of this Platform as well.

3. Agent Based Search System

3.1 Overview

The main purpose of this work was to establish a System, which could improve search quality. Metadata is harvested from remote repositories to gather all needed information in the central repository for further advanced processing (compare with: Chmiel et al., 2005; Chkoliar, 2002, Curry et al., 2003).

The proposed System for use within the e-learning platform contains one multi-agent's layer that is physically divided into two parts: local main core, and remote parts integrated with LOs repositories (Fig. 10).

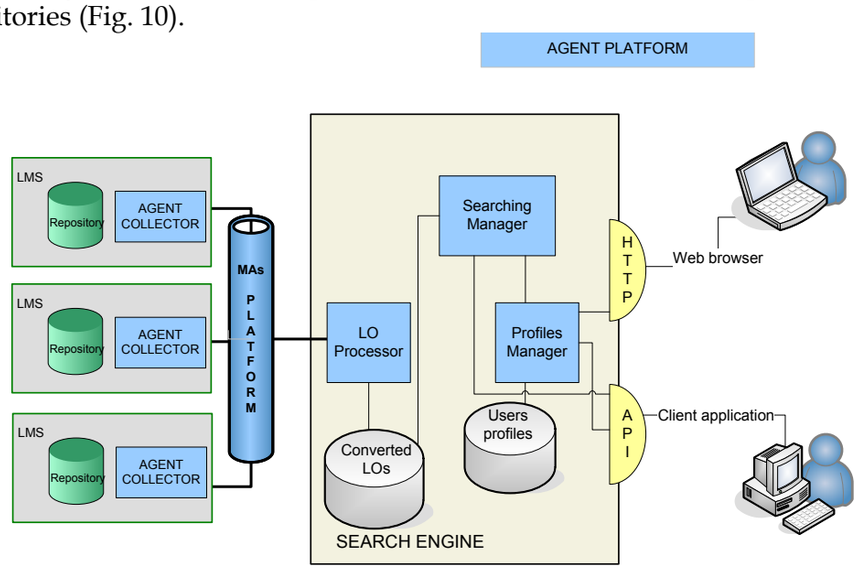


Fig. 10. General System Architecture (Orzechowski, 2007b)

3.2 General Structure

The Agent Based Search System (ABSS) was created on the basis of the Open Source JADE Platform. It consists of two main modules:

- Data Collection Module (Baran et al., 2007), and
- User Management and Resource Search Module (Orzechowski et al., 2007a).

According to functionality the second module could be divided into two functional modules:

- Search Module
- User Profile Module

The structure of the ABSS divided into three modules is presented in Figure 11.

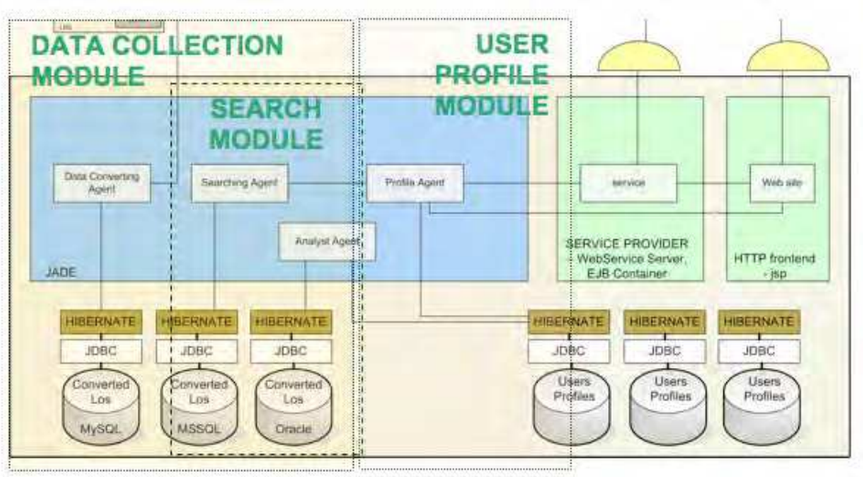


Fig. 11. General System Architecture divided into three modules (Orzechowski, 2007b)

A web interface is also available and it offers accessibility to main features the ABSS. It was built with the intention to calibrate some of the parameters of the Agent Based Search System (ABSS), and to display the most important functionalities of ABSS.

3.2.1 Data Collection Module

The Data Collection Module is responsible for gathering metadata from Repositories (LMS), performing necessary operations on it, and storing it in local databases. The following types of agents were implemented to perform the necessary operations:

- Content Manager Agent (CMA) with Collector Manager Agent (C2MA);
- Collector Agent (CA);
- Converter Agent (CV).

These agents cooperate with agents which are the part of Agent Platform, such as:

- AMS (Agent Management System); It manages agents and containers. CMA uses AMS to trace agent containers and to verify the presence of CA agents in these containers.
- DF (Directory Facilitator). It provides a Yellow Pages service in the Agent Platform so agents can find one another. This service is used by CMA to trace availability of Converter Agents registered in the DF.

The schema of internal interaction among components within DCM is presented in Figure 12.

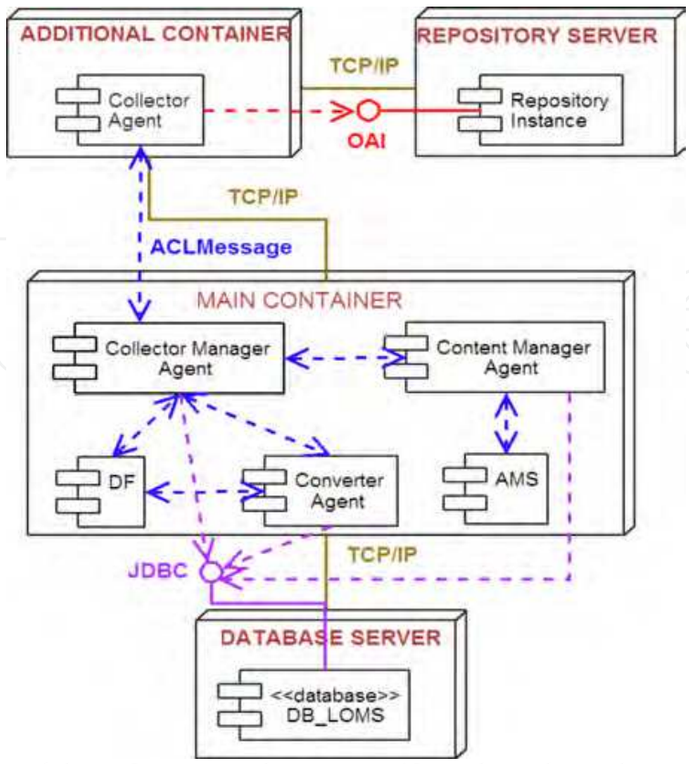


Fig. 12. The schema of the DCM internal structure (Orzechowski et al., 2009)

The Content Manager Agent is the most important agent, which is responsible for communication among all remote parts of the agent platform. There is only one instance of it in the whole system. It maintains an up-to-date list of all connected repositories. After detecting new LMS, the Content Manager creates a new Collector Agent and initializes it. The Collector Agent must be capable of detecting changes in the metadata repository, therefore it needs information about every LO that was previously (last time, when the LMS was connected) present in its database. After the initialization is finished, the Collector Agent moves to the repository and starts observations. When a change of metadata is detected (new LO added, LO deleted or metadata modified), the appropriate information is sent to the Content Manager.

Each Converter Agent, installed in the platform, is associated with a different search algorithm. It must analyse and store received information about LOs, so that it is available in a format that supports fast search. One Converter Agent is usually capable of processing metadata in one language. Each Converter Agent stores results of metadata processing in a separated table or database.

After a new Converter Agent is installed in the system, an initial converting action must be performed. It is very demanding, because every LOM in the adequate language, that has been stored in the local database, must be processed. After that, the Converter Agent publishes its capabilities in the Yellow Pages Service. When the Content Manager receives information about changes in metadata, it informs every appropriate Converter Agent, so that the change is introduced immediately.

Metadata Repositories (e.g. LMS) are accessed via the Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH).

The OAI-PMH is a lightweight harvesting protocol for sharing metadata between services. The OAI-PMH protocol is based on HTTP. Request arguments are issued as GET or POST parameters. The OAI-PMH supports six request types such as: Identify, ListMetadataFormats, ListSets, ListIdentifiers, ListRecords and GetRecord.
The following scenarios of data harvesting were implemented (Fig. 13):
1. Collector Agent (CA) is sent to a remote repository to perform all actions locally at repository's site. It controls contents all the time and notifies CMA if any changes occur.
2. Collector Agent (CA) is staying and operating locally. In this situation all metadata analysis is performed locally at the ABSS Core host. This approach is designed to the systems, which do not support DCM directly.

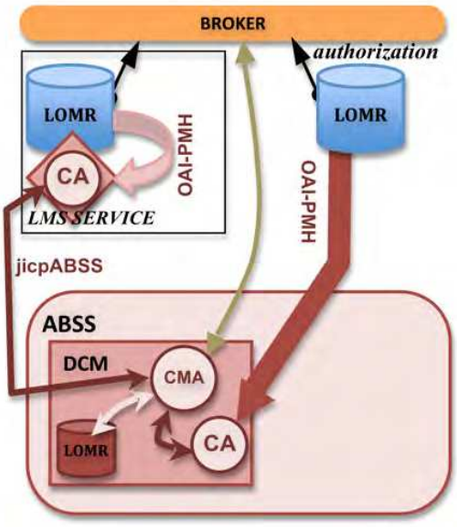


Fig. 13. DCM data harvesting approaches (Orzechowski et al., 2009)

ABSS-LMS is the name of modified JADE-LEAP components and some specialized software intended to be used by administrators of remote repositories to start up DCM components. Services provided by the ABSS-LMS guarantee that communication and registration processes are secure and stable. Security mechanisms include data encryption, achieved by using the SSL/TLS Java protocol and mutual authentication, realized by using digital certificates.
Specialized thread controls guard all platform and communication parameters and correct them automatically if something is out of order. They are responsible e.g. for re-initialization & reconnection of peripheral containers if connection between them and the main container has been lost. They can also kill an agent container, create a new one and register it if connection to the main container cannot be established again.
If the LMS host is part of a private-area network, the proposed and presented solutions (Orzechowski et al., 2009) can omit all of these problems. LMS ABSS-Component creates and runs an agent peripheral container and maintains its registration to the main container. Connections between peripheral and the main containers and communication between CA and CMA agents use specially designed protocol as the extension to existing JADE-LEAP JICP.

The main advantages of the proposed and implemented solutions can be pointed out as:
- Enabling secure communication among agents;
- Introduction of secure data transmission including mobile agents' transmission;
- Introduction of the thread control on each peripheral container;
- Secure registration of new peripheral container connected to DCM;
- Protection of main agent container against fake remote containers;
- Possibilities to connect to remote repository even if it is hidden by NAT;
- Decreasing the number of used ports on peripheral to only two ports.

3.2.2 Search Module

One of the main functionalities of the Multi-Agents' Search System is situated in the Search Module, which is responsible for selecting LOs that are satisfying a given request. The process of selection is performed by algorithms called "search methods" (Baran et al., 2007, Chmiel et al., 2005; Chkoliar, 2002).

The Search components as elements of the Search Module are intended to return the learning object identifiers that match the received query. Communication with other parts of the system is provided by a search agent object, which uses ACL messages to send all necessary information included (Orzechowski, 2007b).

Each search method is implemented by several Search Agents. Usually one Search Agent can perform search of LOMs in one language, as in the case of Converter Agents. The language-based division of local repositories and processing solves several important problems:

- Search algorithms operate on smaller data sets;
- Metadata repository can be distributed;
- Search process can be distributed;
- More popular languages can be supported by larger groups of agents.

The following Search Methods were implemented:

Simple Search Method

After a request is received from a personal agent, all queries are processed to obtain single words set in each category. Query processing eliminates redundant and meaningless words. There is also special character processing. For each language supported in the system a stop words list can be made, which includes words that are omitted in the search process. In English, this list contains e.g.: "for", "on", and "the." Next, a search query is launched for each word from the query in each set (keywords, title, author, description). It's based on comparing keywords and an index describing learning objects (in each – keyword, title, author, and description category). When there is a match – identificator of LOM (lom id) is returned. After a search for all words in a set is finished, results processing is launched in each category. It's based on the local merge mode value, which indicates if results are considered as a union or intersection. Based on results of this merge, there is a rate value assigned to each learning object.

The global merge mode is used as the parameter value to determine how to process results from each category (keyword, description, author, title). Based on this parameter, the next result processing is made.

The next step is to determine if other conditions are sent in the request (age, size, technical role, etc.). If so, actual results are restricted only to those which fulfill all conditions.

Inverted Index Search Method

The Inverted Index method is based on typical strategy of text search. Each word in every category (keyword, title, author, description) is associated to every object it refers to. This search algorithm is based on this relation. After the query is singled out and processed to a set of words (query processing similar as in Simple Search method), the search method finds all learning objects that are associated to those words. Based on the local merge mode ("and" or "or") there are unions or intersections taken and returned. It's done for each category, and after, based on the global merge mode, the next result processing is made.
Sensitive search is a variation of the regular Inverted Index method, which takes into account the multiplicity of repetition of words in a keyword, title or description set. When the learning object is described by one word more than once in one category, and this word is included in the query, the returned results will increase. It also works internally in each category as well as externally, comparing results from all categories (Joung et al., 2005; Su & Widom, 2005).

3.3 UPM – Personalization and offered end-user assistance

UPM is the essential part of the Management and Resource Search Module, which supports: end-users' accounts management including personalized search, and additional functionalities offered for a end-users' recommender system with different implementation of collaborative filtering approach (Orzechowski et al., 2007a; Sarwar et al., 2000) and several internal and external interfaces to the services.
Using a profiling mechanism is essential in e-learning Platforms. Only in this way is it possible to present search results in such a way to suit the end-users' (learners / teachers) expectations. It is necessary to build a local database, which will store information about registered users, including both their interest and the history of their work.
A personal Agent is an agent that will represent a single user. Each registered user can have only one Personal Agent. An unregistered user is also represented by a Personal Agent, but with much reduced functionality.
A Personal Agent is created when a user starts his session in the system. It remains alive, while the user is working. If a longer period of inactivity occurs, the agent stores its state in the database (only if the user is registered), and will die.
The Agent can perform operations on the account, can create, modify or destroy it. Some of these operations are performed automatically as a result of analysis of other user's operations, and some of them can be directly invoked by the user.
The most important function of the Personal Agent is connected with the execution of search requests. The Agent stores a list of recent requests, each associated with a list of results. It must continuously gather the results, valuate and sort them, and return them to the user when asked. In order to perform the search, it must use services offered by the Search Module.
Ranking and Collaborative Filtering component
Most search methods used in Internet repositories match words in the search query and keywords in the known available elements to produce a list of results. As query conformity often isn't enough to order the search results properly, various schemes are used to prioritise the results of the most predicted value for the user. However, people evaluate things on a very subjective basis. Thus, marks on a ranking scale provided by different users may have completely separate meanings. It is impossible to provide a uniform ranking scale that's universal – applicable for all users of a system. The user will always face the question:

"Do the ranks assigned to that element reflect my preferences?"
Collaborative Filtering (CF) is commonly used to organize recommender system on the basis of end-users' behaviour. Introduction of CF into e-learning systems requires adaptation of these general Collaborative Filtering methods in order to improve efficiency (Orzechowski et al., 2007a).
The first step for creation of a CF-based system is selection of data used to predict users' preferences. Analysis of e-Learning object repositories and search systems resulted in establishment of a set of explicit and implicit data collection methods.
The explicit methods include:
•ranking: when the user applies a rank within a given scale to an object,
•labelling: when the user attaches a label to an object.
The implicit methods are:
•selection: when the user clicks a link on the result list; a page with an extended item description is displayed,
•downloading: when the user clicks a link on an item description page; this either displays the Learning Object itself in the browser or downloads the object to the user's computer.
Behaviour vs. profile-based user clustering
Most collaborative filtering systems rely on user behaviour and user actions to determine user preference similarity. The underlying idea is that contents of the objects and characteristics of user profiles are not taken into account (Orzechowski et al., 2007a).
However, characteristics of e-Learning applications require a slightly different approach, for a few reasons. They are connected to the characteristics of e-Learning systems themselves as well as the target audience. Firstly, the most dependable user behaviour data collected by the system – the item rates – are extremely sparse. Secondly, the implicitly collected data has low dependability, as the user may decide at any point that the item is not what they are looking for after all. Because some users of the system are young children (users of type learner), rates may not be reliable, and their action records are prone to be chaotic.
Therefore, it was decided that another approach to user similarity computation was to be designed and implemented. This approach features a new way of grouping users – based on their profiles rather than their behaviour. The following profile elements are taken into account: age, gender, country of education and points of interest.
The procedure is similar to the procedure used in "classic" collaborative filtering. A matrix of size m times n is created. However, columns of the matrix are composed of all possible values for the four profile fields, instead of object ranks/events. In the current implementation, enumerated fields get a '1' if a given value applies, and a '0' if it does not. For other fields, the value is numeric.
To provide balance between the four sections, weights have to be defined for each one of them. This allows the administrator to fine-tune the distribution of users after feature extraction in the n'¬-dimensional space.
Types of user clusters
There is a strict division of users in e-Learning systems into two separate groups: teachers and students (called learners). Therefore, the system should maintain separate clustering schemes for these types of users (Orzechowski et al., 2007a).
Moreover, as presented earlier, the suggested architecture of a ranking system for e-learning consists of two separate approaches towards user similarity assessment: behaviour-based (classic) and profile-based.

Therefore, six different types of user classification are present in the current implementation:
– all users / behaviour-based, – learners / behaviour-based,
– teachers / behaviour-based, – all users / profile-based,
– learners / profile-based, – teachers / profile-based
Calculations are performed independently and separately for each of the cluster types presented above.

Positioning of results using search criteria and profile conformance data

While developing the methods for discovering users' preferences, another problem has been investigated as well. It concerns methods of presenting search results using three different indicators computed by the system (see below) within a single set of results. Output from each of the classification schemes is a set of percentage values, representing the three aspects of conformity of a given LO (search result):

- Search query conformity specifies how the given LO (described in compliance with the IEEE LOM standard) matches a given search method. The system delivers separate sets of results for each of the used search methods.
- User profile conformity specifies how the given LO matches the user's profile, provided upon registration. The system applies values to LOs found previously by the search methods.
- User cluster conformity delivers separate sets of results representing the predicted attractiveness of given LOs within clusters to which the user belongs. Detailed information regarding cluster types and clustering methods have been described in previous sections.

The simplified architecture of the entire result generation system is shown in Figure 14.

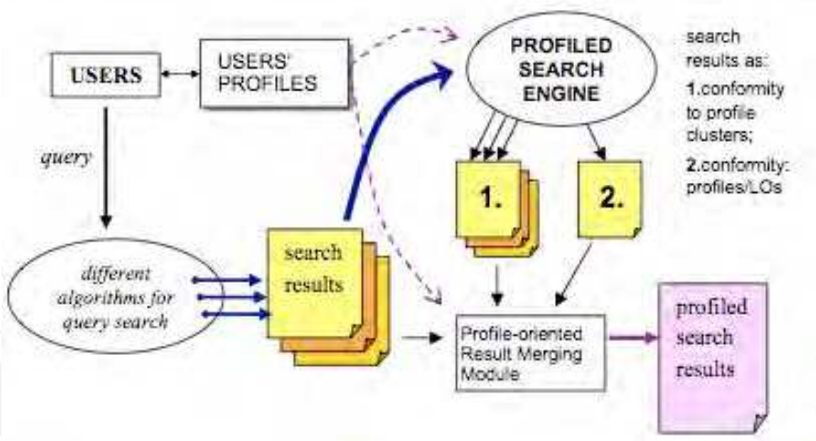


Fig. 14. Simplified schema of merging search results process (Orzechowski et al., 2007a)

A Web GUI was implemented to allow users of the system to choose their preferred way of merging results. Furthermore, an API based on the Web Services technology was implemented to enable utilisation of results generated in our system by external systems (Orzechowski et al., 2007a). As each of the criteria is separate from others, we let users choose which features are used to create the end list of results. These features were divided into two separate parts: search methods relevance (query conformity) and profile relevance (user profile/cluster conformity).
Even if the user decides to utilize all of the offered features, the system prepares two

separate lists of results for each part.

Search method/query conformity

Two different solutions were suggested:

- weighted average – end user can choose the influence of each search method on the end results' list;
- maximum – the maximum value of LO conformity is taken.

Popularity and conformity to profiles and clusters

This part was divided into the following elements:

- conformity to user profiles,
- conformity to clusters determined by user behaviour similarity,
- conformity to clusters determined by multi-dimensional user profiled analysis.
- general popularity (not taking individual users' preferences into account).

For all four elements presented above, the weighted average is used to create a single list of results. Additionally, as ranks for each element can be based on data collected using explicit or implicit methods, the user can choose how these methods influence the computed value for the relevant elements, also using the weighted average

Merging profiles and search methods

The methods presented above generate two lists of results – one sorted by search method conformity, and one sorted by profile conformity. Determination of methods for merging these two lists is a significant problem. While profiling methods are very important in search engines, they should not diminish the importance of conformity returned by the search methods (Orzechowski et al., 2007a).

While research is being performed, the problem has been temporarily solved by using geometric means. Ultimate conclusions in this aspect will require extensive real-life testing and fine-tuning of the production system.

4. Conclusion

The implemented Agent Search System offers new, important features not available in currently used e-learning systems. Simultaneous theoretical research and practical development work open the unique opportunity of testing how new ideas perform in real life and lets us acquire valuable experimental data.

Moreover, analysis of the history of the users' activities, made possible by our system, in connection to analysis of LOM fields, is a significant step towards creation of an intelligent Learning Object search engine, which will present search results in a way as close to the user's expectations as possible.

The Agent Based Search System is stable and easy for management and development thanks to the integration of multi-agent architecture with the stable and scalable EJB JBOSS application Server.

The searcher of Brokerage System, based on the Simple Query Interface (SQI), was optimised.

The proposed Agent Module is the alternative for the existing search module. It has the open character so its work can be easily distributed on all levels, such as (Orzechowski, 2007b):

• FLEXIBILITY

Each new search method can be added to our working system really fast.

• HOT SWITCH-ON
This System provides the easiness of administration of working agents.
• EASY TO DEVELOP & FEEDBACK OPEN
This solution has really spread for adding as many search algorithms and general search functionality as we need.
• ENABLING PROFILED SEARCH AND RECOMMENDATIONS
This System analyses' explicit and implicit knowledge of end-users works with connection to their profiles, improving the positioning for found results and offering different kinds of recommendations.
• OFFLINE SEARCH
Our System allows users to store queries and be informed, i.e. by e-mail, if new LOs, matching user's criteria, will be added to the System.
Some of these tasks could be implemented without an Agent Layer but multiprocessing, distributed data harvesting and inter-components communications makes this agent approach well-founded.

5. References

Baran, R., Zeja, A., Orzechowski, T. M., Dziech, A. & Lutwin, M. (2007). The data collection module of the agent based search system, *Proceedings of WEBIST'2007: "Web Information Systems And Technologies"*, Barcelona, Spain, March 2007

Bellifemmine, F., Poggi, A., & Rimassa, G. (1999). Jade - A FIPA compliant Agent Framework., *Proceedings of 4th International Conference on Practical Applications of Intelligent Agents and Multi-Agent Systems*, 1999

Chkoliar, N. (2002). E-Russia: Programme for the Innovation in Government, *Proceedings of the 10th NISPAcee Annual Conference: DELIVERING PUBLIC SERVICES IN CEE countries: Trends and Developments*, April 25-27 2002, Cracow, Poland

Chmiel, K., Gawinecki, M., Kaczmarek, P., Szymczak, M. & Paprzycki, M. (2005). Efficiency of JADE agent platform, *Proceedings of International Symposium of Parallel and Distributed Computing & International Workshop on Algorithms*, vol.13(2), pp. 159-172, Amsterdam, The Netherlands, April 2005, IOS Press

Curry, E., Chambers, D. & Lyons, G. (2003). A JMS Message Transport Protocol for the JADE Platform, *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology, IAT'2003*, Washington, DC, USA, 2003, IEEE Computer Society

Franklin, S. & Graesser, A. (1997). Is it an Agent, Or Just a Program?: A Taxonomy for Autonomous Agents, *Lecture Notes in Computer Science*, 1997, pp. 21-35, Springer, ISBN: 978-3-540-62507-0

Hillman, D. C. A., Willis, D. J. and Gunawerdena, C. N. (1994). Learner-Interface Interaction in Distance Education: An Extension of Contemporary Models and Strategies for Practitioners., *The American Journal of Distance Education*, 8(2) 1994, pp. 30–42.

Jennings, N.R. (1996). Software agents. IEEE Review, pp. 17-20, 1996

Joung, Y.J., Fang, C.T. & Yang, L.W. (2005). Keyword Search in DHT-based Peer-to-Peer Networks, *Proceedings ICSCS 2005*

Moore, M. G. (1989), Three Types of Interaction, *The American Journal of Distance Education*, 3(2) 1989, pp. 1–6.

Najjar, J., Ternier, S. & Duval, E. (2004). User Behavior in Learning Object Repositories: An Empirical Analysis, *Proceedings of The World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004*, pp. 4373-4379, Lugano, Switzerland, AACE, 2004

Nichani, M. (2001). LCMS = LMS + CMS [RLOs], *http://www.elearningpost.com/articles/archives/lcms_lms_cms_rlos*

Orzechowski, T. & Dziech, A. (2001a). Communication between end-users and databases using mobile agent technology, *Proceedings of the IASTED IMSA'2001 Conference: "Internet And Multimedia Systems and Applications"*, pp. 433-436, Hawaii, USA, August 13-16 2001, ACTA Press

Orzechowski, T. (2001b). Concepts of Systems for Information Retrieval, *Proceedings of the IEEE SIBCOM'2001 Conference: "MODERN COMMUNICATION TECHNOLOGIES"*, pp. 39-44, Tomsk, Russia, November 28-29 2001

Orzechowski T. (2002). Meta-data distribution in intelligent databases system, *Proceedings of the IEEE EDM'2002 Conference*, vol.II, pp. 21-24, Erlagol, Russia, July 1-5 2002

Orzechowski, T., Ernst, S. & Dziech, A. (2007a). Profiled Search Methods for E-Learning Systems, *Proceedings of the EC-TEL-LODE Workshop'2007: "2nd European Conference on Technology Enhanced Learning*, Sissi – Crete, Grece, September 2007

Orzechowski T. M. (2007b). The use of multi-agents' systems in e-learning platforms, *Proceedings of the IEEE SIBCON'2007 Conference: "Control and Communication"*, pp. 64-71, Tomsk, Russia, April 2007

Orzechowski, T. M., Zeja, A. & Dziech, A. (2009). Secure Data Harvesting in the Federation of Repositories Managed by Multi Agent Systems, *Proceedings of the IEEE SIBCON'2009*, Tomsk, Russia, March 2009

Prause, C., Ternier, S., de Jong, T., Apelt, S., Scholten, M., Wolpers, M., Eisenhauer, M., Vandeputte, B., Specht, M. & Duval, E. (2007). Unifying Learning Object Repositories in MACE, *Proceedings of The First International Workshop on Learning Object Discovery & Exchange (LODE'07, EC-TEL'07)*, Crete, Greece, September 2007

Rengarajan R. (2001), LCMS and LMS. Taking Advantage of Tight Integration, Click2learn Inc, *http://www.e-learn.cz/soubory/lcms_and_lms.pdf*

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. (2000). Analysis of Recommendation Algorithms for E-Commerce, *Proceedings of the ACM E-Commerce Conference*, pp. 158-167, 2000

Schlager, M. S., & Fusco, J. (2004). Teacher Professional Development, Technology, and Communities of Practice: Are We Putting the Cart before the Horse?, *Designing for Virtual Communities in the Service of Learning*, Cambridge University Press, 2004

Schulmeister, R. (2003). *Modellversuch Lehrqualifikation für Wissenschaft und Weiterbildung*, (in German), Abschlussbericht, Universität Hamburg, 2003

Shakshuki, E. & Jun, Y. (2004). Multi-Agent Development Toolkits: An Evaluation, *Lecture Notes in Computer Science (Innovations in Applied Artificial Intelligence)*, 3029/2004, pp. 209-218, Springer Berlin / Heidelberg, ISBN: 978-3-540-22007-7

Simon, B., Massart, D., van Assche, F., Ternier, S., Duval, E., Brantner, S., Olmedilla, D., and Miklós, Z. (2005). A Simple Query Interface for Interoperable Learning Repositories, *Proceedings of the 1st Workshop on Interoperability of Web-based Educational Systems*, pp. 11-18, Chiba, Japan, May 2005

Stefaner, M., Vecchia, E. D., Condotta, M., Wolpers, M., Specht, M., Apelt, S. & Duval, E. (2007). Mace – Enriching Architectural Learning Objects for Experience Multiplication, *Lecture Notes in Computer Science*, 4753, pp. 322-336, Springer Berlin, 2007, ISBN: 978-3-540-75194-6

Su, Q. & Widom, J. (2005). Indexing Relational Database Content Offline for Efficient Keyword-Based Search, *Proceedings IDEAS 2005*

Van Assche, F. & Massart, D. (2004). Federation and Brokerage of Learning Objects and Their Metadata, *Proceedings of the IEEE International Conference on Advanced Learning Technologies, ICALT'2004*, pp. 316-320, Washington, USA, 2004

Van Durm, R., Duval, E., Verhoeven, B., Cardinaels, K. & Olivié, H. (2001). Extending The ARIADNE Web-Based Learning Environment,. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pp. 1932-1937, AACE, Norfolk

IntechOpen

IntechOpen

IntechOpen



E-learning Experiences and Future

Edited by Safeeullah Soomro

ISBN 978-953-307-092-6
Hard cover, 452 pages
Publisher InTech
Published online 01, April, 2010
Published in print edition April, 2010

This book is consisting of 24 chapters which are focusing on the basic and applied research regarding e-learning systems. Authors made efforts to provide theoretical as well as practical approaches to solve open problems through their elite research work. This book increases knowledge in the following topics such as e-learning, e-Government, Data mining in e-learning based systems, LMS systems, security in e-learning based systems, surveys regarding teachers to use e-learning systems, analysis of intelligent agents using e-learning, assessment methods for e-learning and barriers to use of effective e-learning systems in education. Basically this book is an open platform for creative discussion for future e-learning based systems which are essential to understand for the students, researchers, academic personals and industry related people to enhance their capabilities to capture new ideas and provides valuable solution to an international community.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tomasz Marcin Orzechowski (2010). The Use of Multi-Agents\' Systems in e-Learning Platforms, E-learning Experiences and Future, Safeeullah Soomro (Ed.), ISBN: 978-953-307-092-6, InTech, Available from: http://www.intechopen.com/books/e-learning-experiences-and-future/the-use-of-multi-agents-systems-in-e-learning-platforms

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen