

# An Agent-based Federated Learning Object Search Service

**Carla Fillmann Barcelos and João Carlos Gluz**  
*Interdisciplinary Program in Applied Computer Science (PIPICA) –  
Vale do Rio dos Sinos University (UNISINOS), São Leopoldo, RS, Brasil*

[carlafillmann@yahoo.com.br](mailto:carlafillmann@yahoo.com.br); [jcgluz@unisinobr](mailto:jcgluz@unisinobr)

**Rosa Maria Vicari**  
*Informatics Institute, Federal University of Rio Grande do Sul  
(UFRGS), Porto Alegre, RS, Brasil*

[rosa@inf.ufrgs.br](mailto:rosa@inf.ufrgs.br)

## Abstract

The cataloging process represents one of the greatest issues in respect to the use of learning objects because it is through this process that appropriate objects can be found through search engines. Incorrect cataloging causes inefficacy in the search process, and this situation is aggravated when the objects are distributed and maintained in several distinct repositories. The present work proposes the creation of an agent-based federated catalog of learning objects: the AgCAT system. This system is part of the MILOS infrastructure, which will provide the computational support to the OBAA metadata proposal, a Brazilian initiative to create a new learning object metadata standard, able to support the requirements of multi-platform adaptability, compatibility with current standards, special needs accessibility, and technological independence of hardware and software platforms. The paper presents the functional structure and organization of the AgCAT system, showing its architecture, the main aspects of its prototype, and main results obtained till now.

**Keywords:** Search Service, Learning Objects, Software Agents.

## Introduction

The Brazilian Ministry of Education provides free digital pedagogical content by means of the Virtual and Interactive Net for Education program (RIVED, 2009), distributing these objects through the International Base of Educational Objects repository (BIOE, 2010). The main goal of these programs is to aid in the development and distribution of electronic educational material by using Learning Objects (LO) as the foremost technology to publish and disseminate such material.

---

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org) to request redistribution permission.

The material is formed by educational activities, which may contain multimedia resources, animations, and simulations. To locate a particular object in a repository is a difficult problem depending on the rightful indexation and cataloging of its material. This process corresponds to the fulfilling of the LO *metadata* with correct information. Metadata is information that describes the characteristics of certain documents,

material, or LO. The main purpose of metadata is still to be understood and used by people or software agents in cataloging, searching, and similar tasks (Taylor, 2003).

The cataloging and indexation process represents one of the greatest issues to locating educational contents, such as learning objects, because it is through this process that these objects can be found through search engines. Incorrect LO cataloging or indexation causes inefficacy in search processes. This situation is aggravated when LO are distributed and maintained in several distinct repositories. The increase of LO production in Brazil (and around the world) by several different institutions has shown the risk that the material remains unused by the general community, or at least with very restricted use, limited only to the members of the institution in case a unified search mechanism exists capable of finding LO in repositories of most anyone in the institution. Currently there is no standard infrastructure that gives support to a unified search and retrieval of educational resources such as LO (CORDRA Management Group, 2009).

To assist in this situation, the present work proposes the creation of an agent-based federated catalog of learning objects (AgCAT). The general objective of this system is to provide an infrastructure of federated LO catalogs that are able to help in the search and retrieval of these educational resources. The system will make intensive use of technologies from Distributed Artificial Intelligence (DAI) and Multi-Agent Systems (MAS) research fields (Weiss, 1999; Wooldridge, 2002), seeking to optimize the LO search process. The system will use several protocols and technologies to harvest metadata from LO repositories and digital libraries. Several AgCAT systems can also be *federated*, forming a *federation* of LO catalogs. The search for LO in the federation is transparent for its users. A query made in any federated AgCAT system is transparently propagated to all other AgCAT systems in the federation. Therefore, apart from communication delay, a query in any AgCAT system is equivalent to the same query in any other federated system. Only the search propagation protocol must be supported by each federated AgCAT system. The administration and management of each federated AgCAT system is completely independent from the other federated systems, allowing for different institutions to be included easily in the federation.

This work presents the functional structure and organization of the AgCAT system, showing the system's architecture, aspects of its prototype, and main results obtained until now. The next two sections present a literature review concerning the main topics related in the present work focusing on the metadata standards supported by AgCAT and the multi-agent technology that supports the system. The following section describes the multi-agent architecture of the system, the organization of its agents, particular details about the formation of the directory federation, and the metadata harvesting process. The last section presents the prototype of the system, and its first results.

## Metadata

For the Learning Technology Standard Committee (LTSC) at the Institute of Electrical and Electronics Engineers (IEEE, 2002), a learning object is any entity, digital or not, that can be used, reused or referenced during a learning process. A learning object is digital or non-digital (mock-up, image, film, etc.) content that can be used for an educational purpose, including, internally or through association, suggestions of contexts in which it should be used. Such a view is also adopted in the present study, despite being restricted to the case of digital entities.

The main property of LOs are their re-usability. Such a characteristic can be achieved through modularity, interoperability, and recovery. Modularity describes the degree of separation and subsequent recombination of LO components. Interoperability is the ability to operate in heterogeneous platforms. Recovery is related to the ability to be found due to its description of properties and functionalities. These characteristics guide the efforts of several research groups and entities

aiming to propose standardizations to enable development and use of LO worldwide. Within this context, the following initiatives stand out: IEEE LTSC (IEEE, 2002), IMS Global Learning Consortium (IMS, 2009), and Advanced Distributed Learning Initiative (ADL, 2001).

The purpose of adopting open standards is to obtain platform independence in relation to exhibition, or execution of objects, enabling the use of different operational systems and hardware platforms to make object content available. A metadata LO standard can be seen as a specification of a heading that provides information about the object. The data elements that comprise this heading are the metadata regarding the LO. Therefore, such a standard does not interfere with learning object content or rules as it only groups metadata. For this reason, these standards have been widely used in distance learning resources such as CAREO (Australian), the ABED and RIVED (Brazilian) repositories, and in several standardization initiatives, for example, the ADL/SCORM certification. In this respect, the Brazilian government, through the Secretary of Distance Learning of the Ministry of Education and Culture (MEC), brought forth an initiative to generate a diversity of interactive multimedia educational resources in the form of Learning Objects, which already resulted in the development and publishing of hundreds of didactic resources to computer and general use. These resources were developed by several teams of teachers and students of higher education institutions, available in the BIOE of MEC (BIOE, 2010) and RIVED (RIVED, 2009) repositories.

Among the metadata standardization initiatives, the IEEE-LOM (*Learning Object Metadata*) is considered as an open and internationally recognized standard, which facilitates the search, evaluation, construction, and use of LO. The IEEE-LOM is specified by norm IEEE Std 1484.2.1–2002 and provides a data model for the metadata, normally codified in XML. Such standard aims at specifying the syntax and semantics of information (metadata) concerning LOs. These specifications enable cataloging educational material (through their metadata), considering the diversity of cultural and linguistic contexts of LO creation and reuse. Thus, the objective is to ensure efficient ways of identification, (re)use, management, interoperability, sharing, integration, and recovery of these objects.

In practice, the IEEE-LOM standard defines a library of metadata that can be freely combined to create the information heading of LOs. According to this standard, if the heading of some object, which is formed only by elements defined in the IEEE-LOM standard, then the object is considered “strictly complying with the standard.” Otherwise, if, in addition to data elements defined in the IEEE-LOM standard, the heading has other type of elements (extensions of IEEE-LOM), then the object is considered only as “complying with the standard.”

In addition to re-usability, another important aspect that can be used to evaluate and compare metadata standards for learning objects (or metadata standards for other types of objects and content) is the degree of coverage of the information stored in metadata in relation to the applications intended for the objects. Within this context, there is a low degree of coverage offered by the Dublin Core Metadata Initiative (DCMI) unqualified metadata standard for the Web contents of educational and pedagogical applications, once compared with the IEEE-LOM standard.

On the other hand, within the context of multimedia content conversion and adaptation into different platforms of digital content availability, such as Web, mobile devices, digital TV, and game consoles, there is a relative lack of coverage offered by the IEEE-LOM standard compared to the MPEG-7 standard or other similar standards of metadata for multimedia content. Similarly, there was also a lack of coverage in the current metadata standards for Learning Objects, including IEEE-LOM, IMS-LOM, ADL, and DCMI relative to accessibility requirements for people with special needs and also relative to specific educational issues within the Brazilian context.

The project OBAA (Learning Objects supported by agents) was developed by the UFRGS University in partnership with the UNISINOS University in response to a request by MEC and the

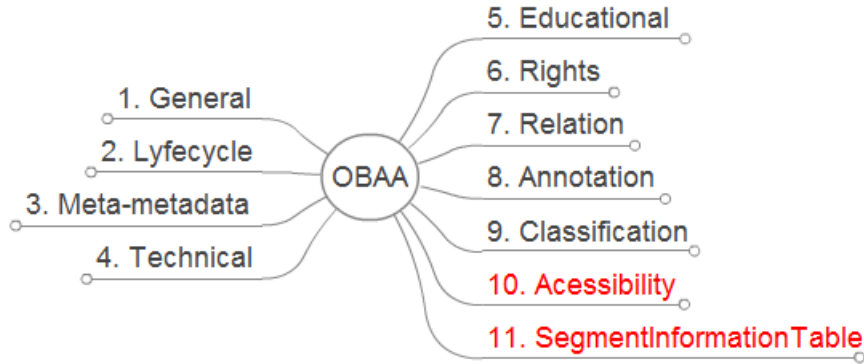
Brazilian Ministries of Communication and Science and Technology for research projects capable of dealing with multi-platform interoperability issues of digital content in the educational context. The main goal of the research was to establish a standardized specification of the technical and functional requirements of a platform for production, editing, management, and distribution of interactive digital content, chiefly LOs, to be used in education applications. The specification should allow the interoperability of the contents among Web and Digital TV environments. The objective of the OBAA project was to answer this need, also considering mobile devices and accessibility requirements. It reached these goals relying extensively on the convergence among the multi-agent systems, Learning Objects and ubiquitous computing technologies, allowing the authoring, storage, and recovery of LOs in varied contexts and through different digital platforms.

The OBAA metadata proposal (Viccari et al., 2010) is one of the main results of the OBAA project and it defines an extension of the IEEE-LOM standard. This proposal provides several new metadata, which allow objects' interoperability among multiple digital platforms beyond the Web platform, supporting new platforms like Digital TV and mobile devices. It also provides specific metadata for accessibility and pedagogical issues.

The proposed metadata intends to ensure freedom to the developer of pedagogical content so that the professional encounters no technological restrictions. The proposed set of metadata establishes a wide structure for cataloging, enabling different forms of application according to the needs of each LO designer. The metadata proposal followed the interoperability and functionality requirements presented below:

- *Adaptability*: enables the same description of an object to be used in an inter-operable manner, adapting to the characteristics of each platform, that is, the system presents a different interface according to the device. Initially supported by the Web, Digital TV (DTV), and mobile platforms.
- *Compatibility*: the metadata standard should maintain compatibility with the current panorama of international standards since it is important to interact with services previously developed through international standards.
- *Accessibility*: considering the right of universal access to knowledge, it is necessary to enable access to LOs by all members of the society, including citizens with special needs.
- *Technological Independence*: the metadata standard should be flexible in order to support technological innovation, allowing extensions without losing compatibility with the previously developed content.

The OBAA metadata proposal is an extension of the IEEE-LOM 1484.12.1 metadata standard, adding new pedagogical requirements in addition to technological foundations to enable use of LO in DTV and mobile devices. New metadata elements were added to technical and educational metadata categories of IEEE-LOM, and two new categories of metadata were created: accessibility and multimedia segmentation metadata (see Figure 1).



**Figure 1: Metadata Groups of the OBAA Proposal.**

## Agents and Multiagent Systems

Artificial agents are computational entities that have autonomous behavior, belong to an environment, and can communicate with other agents in the same environment, whether artificial or human. According to the definition by Wooldridge (2002), an agent is a computer system situated in a particular environment that is capable of autonomous execution in order to attain its objectives. Agents are characterized, among other things, by autonomy, proactive behavior, and the ability to communicate and work together forming multi-agent systems. Autonomy implies that agents can carry out complex tasks independently. By being proactive, agents can take the initiative to accomplish a given task even without an explicit stimulus from a user. As they are communicative, they can interact with other entities to get help for each task and target.

The BDI (Beliefs – Desires – Intentions) cognitive model for agents assumes that the intentions of agents are derived from beliefs and desires, and that the behavior of the agent is clearly implied by its intentions. The BDI model is one of the cognition models of the Mental State approach for agent modeling. In this model, the set of beliefs represent provisional knowledge of the agent, which can change with the passing of the time. Beliefs define what the agent knows about the environment, what it knows about other agents, and what it knows about itself. Beliefs are specified by logical properties concerning other agents, the environment, and about the agent itself. Agents should update their beliefs to reflect changes detected (perceived) in other agents, the environment, and themselves. They must maintain the consistency of the beliefs after this update.

Desires specify the state of affairs the agent eventually wants to bring about. One particular state of affairs is specified by a logical property to be held in this future state and by a list of attributes that define the admissibility criteria of the desire. The admissibility criteria attributes specifies the agent’s beliefs regarding desire. They define, at least, the priority of the desire, the ability of the agent to achieve the desire, and the estimated possibility of the desire to become true. We believe that the purpose of the agent, in the cognitive model of agents that we are using, is explicitly stated as the set of highest-priority desires of the agent.

The fact that an agent has a desire does not mean it will act to satisfy it. Acts are governed by intentions that are characterized by a choice of a state of affairs to achieve and a commitment to make this choice (here we follow the definition of Cohen & Levesque, 1990). Intentions are related to desires by admissibility criteria attributes. The agent will choose those desires that are possible, according to these attributes and to the agent’s current base of beliefs. It is important to note that intentions are also beliefs of the agent. One particular intention is a compromise in that the agent has to reach a specific possible future, that is, the agent believes that the state of affairs it wants to achieve does not hold now and that the agent must work to reach that state. It means

that before an agent decides what to do, it will be engaged in a reasoning process, confronting its desires with its possibilities, defining its intentions, and then planning its actions in respect to this intention.

In other words, an intention poses a decision problem (or a planning problem) for the agent. The agent should solve this problem and decide the course of actions, or plan of actions, to be followed in order to achieve the intention. A plan of actions is composed by a set of actions structured by sequence, iteration, and test/choice order relations (operators). These plans do not need to be fully specified from the beginning; they can be partial and the agent can start to follow the plan and reassess or complete it during execution.

The interaction of the agent with its environment is done by actions and perceptions. An action is an alteration in the external environment caused directly by the agent. From an intentional point of view, it also represents a way to attain an end (intention). Therefore, internally, the agent should know (believe) the basic effects produced by possible actions and what the relations of these actions to their intentions are. Agents detect properties in the environment, or more commonly, changes in these properties through perceptions. Independent of the agent, these changes may occur or they can be caused by actions executed by the agent or by other agents, but the only way the agent has to detect them is through its perceptions. Perceptions produce updating in the base of beliefs of the agent, yet the exact update produced by a particular perception depends on the current state of beliefs of the agent.

Agents form the basic element of computation in multi-agent systems that can be simply defined as systems formed by several agents working together. The fundamental characteristic of the agents in a multi-agent system is the ability that the agents have to communicate. It is through communication with other agents that a particular agent seeks to achieve its goals. Communication here is understood as occurring at the knowledge level where it is assumed that agents exchange knowledge (or more specifically beliefs) with each other. The traditional theory of agent communication follows the epistemological and linguistic basis provided by the Speech Act Theory of Searle, which defines an intentional semantics of communication centered on the perspective of speaker agents. Beliefs are communicated between agents by the use of communicative acts (or illocutionary acts), which are actions, from the point of view of the speaker agent, and perceptions, from the point of view of the hearer agent, destined only for communication purposes. There are several distinct kinds (forces) of communicative acts, depending on the intended purpose of the acts. The most common acts are assertive acts intended to make the other agent believe in a particular assertion, directive acts intended to make the other agent execute an action, commissive acts intended to make the other agent carry out a commitment, and other similar acts.

To be of value for communication purposes, these acts must be represented in a particular language and this language must be the same for all agents in a multi-agent system. This language is called the Agent Communication Language (ACL) of the system. The main form of communication among agents is through an Agent Communication Language (ACL) (Chaib-draa & Dignum, 2002; Labrou, Finin, & Pen, 1999). Most ACL used today is the FIPA-ACL language, specified by FIPA (*Foundation for Intelligent Physical Agents*) (FIPA, 2002). FIPA is an international non-profit organization aimed at developing software standards for agent-based systems. More recently, FIPA has become an IEEE working group, keeping the same goal of development of communication standards for agents. The set of all specifications of FIPA is divided into five distinct categories: (a) Agent Communication, (b) Agent Message Transport, (c) Agent Management, (d) Abstract Architecture, and (e) Applications. The FIPA-ACL language specifications (FIPA, 2002) and the remaining specifications defined in the Agent Communication category are the main specifications of the FIPA model for multi-agent systems. Specifications defined in the Agent Management category are also important for this model and used in the present work be-

cause they establish a reference model for the creation, registration, location, communication, migration, and extinguishing agents.

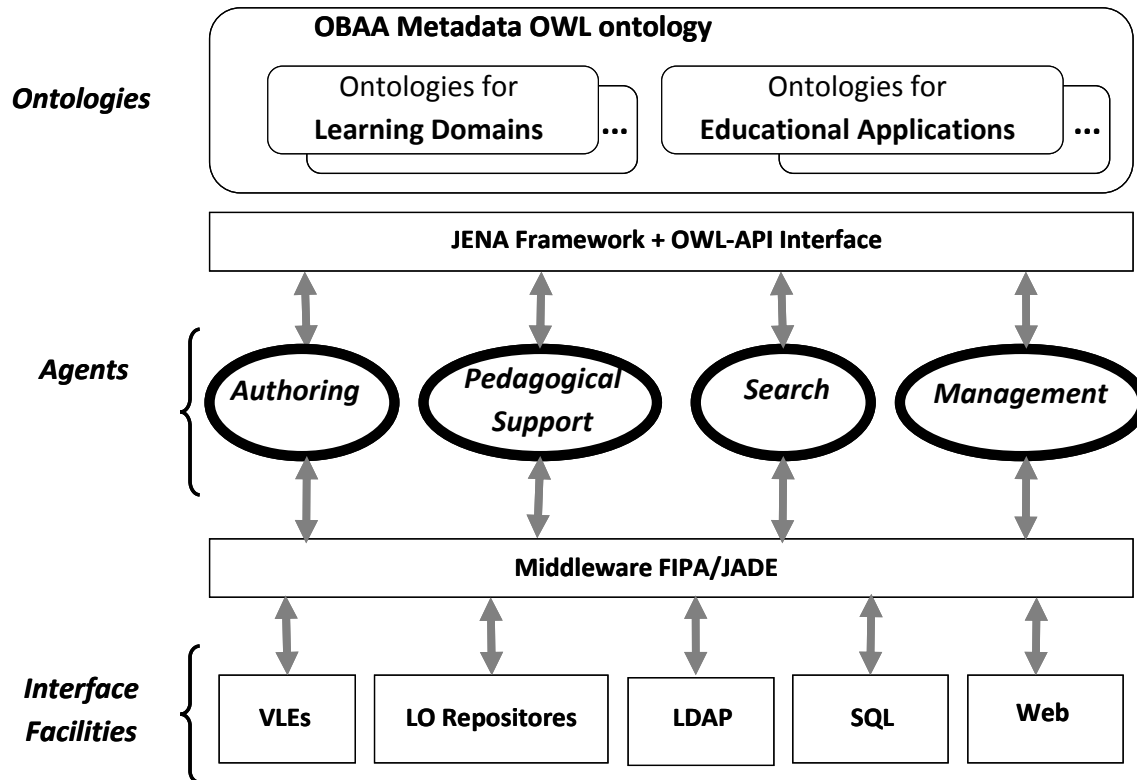
The JADE agent platform (Bellifemine, Caire, & Greenwood, 2007) is a software development environment based on FIPA standards that simplifies the development of these systems, handling several issues related to the communication and the life cycle of agents, as well as helping in debugging activities like the execution monitoring of agents. JADE contributed to the dissemination of the use of FIPA specifications due to the fact that it provides a set of software abstractions and software tools that grant developers the ability to apply FIPA specifications without a deep technical knowledge of these specifications. The JADE platform offers the following set of graphical tools to aid programmers in debugging and monitoring agents: RMA (*Remote Management Agent*), *Dummy agent*, *Sniffer agent*, *Introspector agent*, the DF (*Directory Facility*) agent, and *LogManager agent*.

The FIPA multi-agent systems management model is implemented by two JADE services: the AMS (*Agent Management System*), which supervises the general operation of the platform, and the DF, responsible for the directory service (yellow pages) of the platform. Both services are implemented as agents of the JADE platform. The DF agent registers most any agent caring to offer its services in the platform. It also allows agents to search for other agents and their services in the system. Additionally, the DF accepts requests from agents who want to be notified whenever a record of service or modification is made. Multiple DF agents can be launched simultaneously in order to distribute the yellow pages service over various domains. If necessary, DF agents located in different JADE platforms can be integrated into a federation (can be federated) of DF agents, allowing the spread of some agent requests to the whole federation of DF agents.

## AgCAT Architecture

The AgCAT catalog service is part of a general infrastructure for agents known as MILOS (*Multi-agent Infrastructure for Learning Object Support*) (Viccari et al., 2010) that is being designed by our research group in order to provide various types of services to support the life-cycle of OBAA compatible LO, including localization, authoring, use, management, content adaptation, and conversion for different devices. The main goal of MILOS is to support all requirements and functionalities specified in the OBAA metadata proposal. The architecture of MILOS is divided into three main layers of abstraction (see Figure 2):

- *Ontology Layer*: this layer is responsible for the specification of knowledge that will be shared among all agents involved in infrastructure.
- *Agent Layer*: this layer implements, through a set of multi-agent systems, the several operations related to LO life-cycle.
- *Interface Facility Layer*: this layer implements the communication services necessary for MILOS agents to inter-operate with web servers, virtual learning environments, LO repositories, databases, directory services and other types of educational legacy applications.



**Figure 2: MILOS Infrastructure**

The Ontology Layer, besides the OBAA metadata ontology, contains the ontology for learning domains and educational applications supported by MILOS.

The Agent Layer implements are supported for activities of authoring, adaptation, management, publishing, localization, and use of LO compatible with OBAA. These activities were distributed in four large multi-agent systems:

- *Search System*: supports searching LO.
- *Pedagogical Support System*: supports the pedagogical use of LO in educational contexts.
- *Authoring System*: supports LO authoring activities, including aid for platform adaptation.
- *Management System*: supports storing, managing, publishing, and distribution of LO in distinct platforms.

The Interface Facilities Layer provides the facilities interface that will allow non-MILOS learning environments and educational applications to gain access to MILOS agents and also permit MILOS agents to have access to LO repositories, directory services, databases, and Web servers.

The AgCAT system contributes to the MILOS infrastructure in the context of the Search System, being a prototype of this system. AgCAT agents will be responsible for obtaining, cataloging, and searching LO metadata for MILOS agents and users. The architecture of the AgCAT system is formed by three types of software agents (see Figure 3):

- *Finder agent*: this agent provides the search service to AgCAT users and other MILOS agents.



- *Librarian* agent: this agent obtains metadata from LO repositories and stores these metadata in the local catalog database.
- *InterLibrarian* agent: this agent is responsible for establishing the federation of LO catalogs.

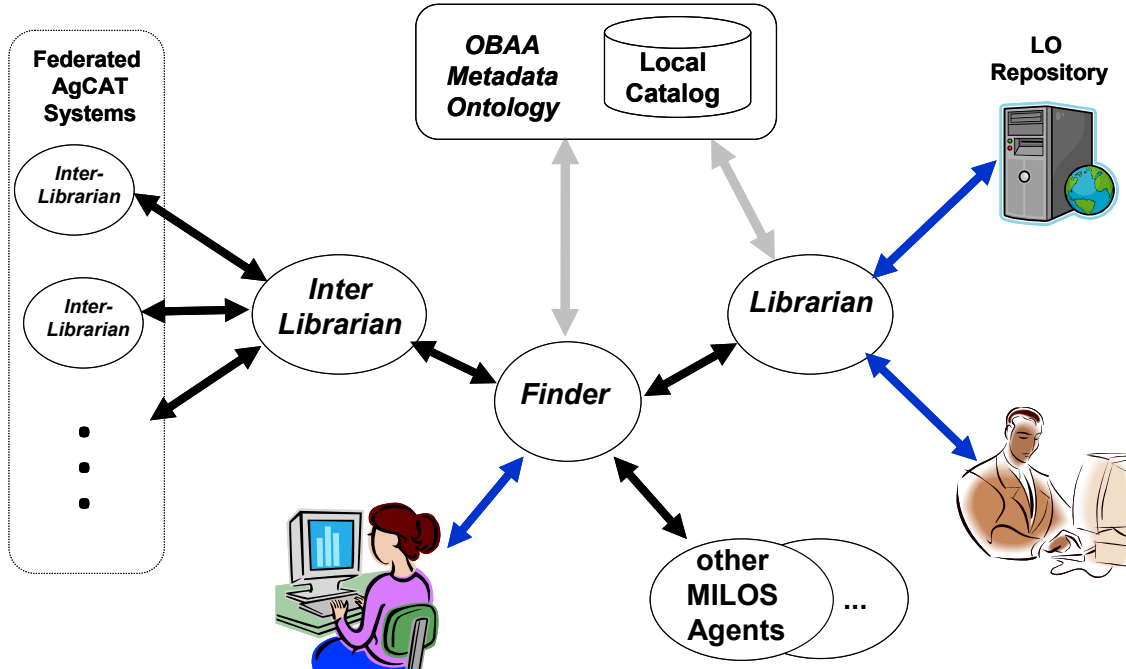


Figure 3: AgCAT Architecture

### Search of Learning Objects

The *Finder* agent provides search service to AgCAT users, allowing these users to keep an eye out for educational content in distinct repositories. By means of a Web interface, the *Finder* agent agrees to the retrieval of objects registered by *Librarian* agent or by other catalogs in the federation. Besides its Web interface, the *Finder* agent also provides a FIPA-ACL (FIPA, 2002) interface, answering the FIPA query messages from other agents (Figure 4).

The search mechanism implemented by the *Finder* agent permits objects to be found by logical expressions which basic predicates compare operators with the values of metadata elements. These expressions assent the conjunction (AND), disjunction (OR) and negation (NOT) of the basic predicates. The mechanism translates these expressions in equivalent SPARQL queries, which are used to retrieve the appropriate metadata from the local database or for other catalogs federated to the current AgCAT system. A user-friendly version of these logical expressions is provided by the Web interface of the agent. This interface also provides a basic search page where a user can seek out an object by specifying some of its properties. Other agents can also search for Learning Objects by using FIPA-ACL query messages. The language contents of these queries can be direct SPARQL, or the user-friendly version adopted in the Web interface.

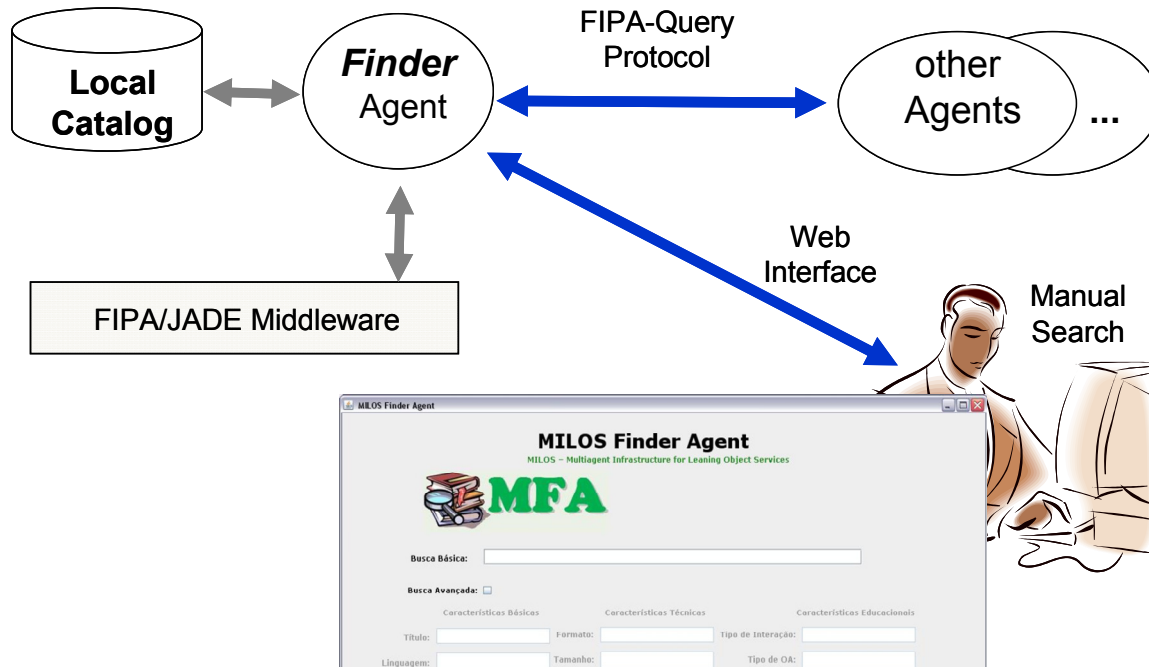


Figure 4: *Finder Agent Interface.*

## Federated Search

The third agent used in the AgCAT system is the *InterLibrarian* agent, which implements the federation of LO catalogs. This agent interacts with other *InterLibrarian* agents situated in remote FIPA platforms. The agent is configured with a list of federated *InterLibrarian* agents. A federated *InterLibrarian* agent is a recognized agent that does not belong to the current FIPA platform. The *InterLibrarian* agent in the local platform will accept a propagated query from federated agents checking with the local Finder agent if there are objects that satisfy the query. Independent of the result of the local query, the *InterLibrarian* will also propagate the query to the other federated *InterLibrarian* agents. The local *InterLibrarian* controls the destination and origin of queries, redirecting the response to the appropriate querying agent. The local Finder agent can also use the service of the *InterLibrarian*, asking it to propagate a query to the other federated agents.

The *InterLibrarian* agents from various FIPA platforms can be integrated into a federation, making it possible to propagate a search in an AgCAT system for the other systems throughout the federation. The federation provides a single distributed yellow pages book for Learning Objects. Figure 5 shows an example of a federation of directories covering some LO repositories.

Queries passed on to other agents in the federation are enclosed in the FIPA *propagate* messages, along with auxiliary parameters. The *x-search-id* parameter is mandatory and will uniquely identify the query in the entire search process. Other parameters permit some fine-tuning of the search process. This includes, among other things, the maximum number of responses returned by the search (*x-max-results*) and the maximum depth of the tree of federated agents being formed in the search process (*x-max-depth*). Whenever an agent receives a request to perform a search, it allows the search to be propagated to the other agents of the federation only if the *x-max-depth* is greater than 1. Furthermore, it is necessary for the agent to not receive a query with the *x-search-id* from a previous request. When the search is propagated to other agents, the value of the *x-search-id* parameter is not to be changed and the value of the *x-max-depth* is decreased by one unit.

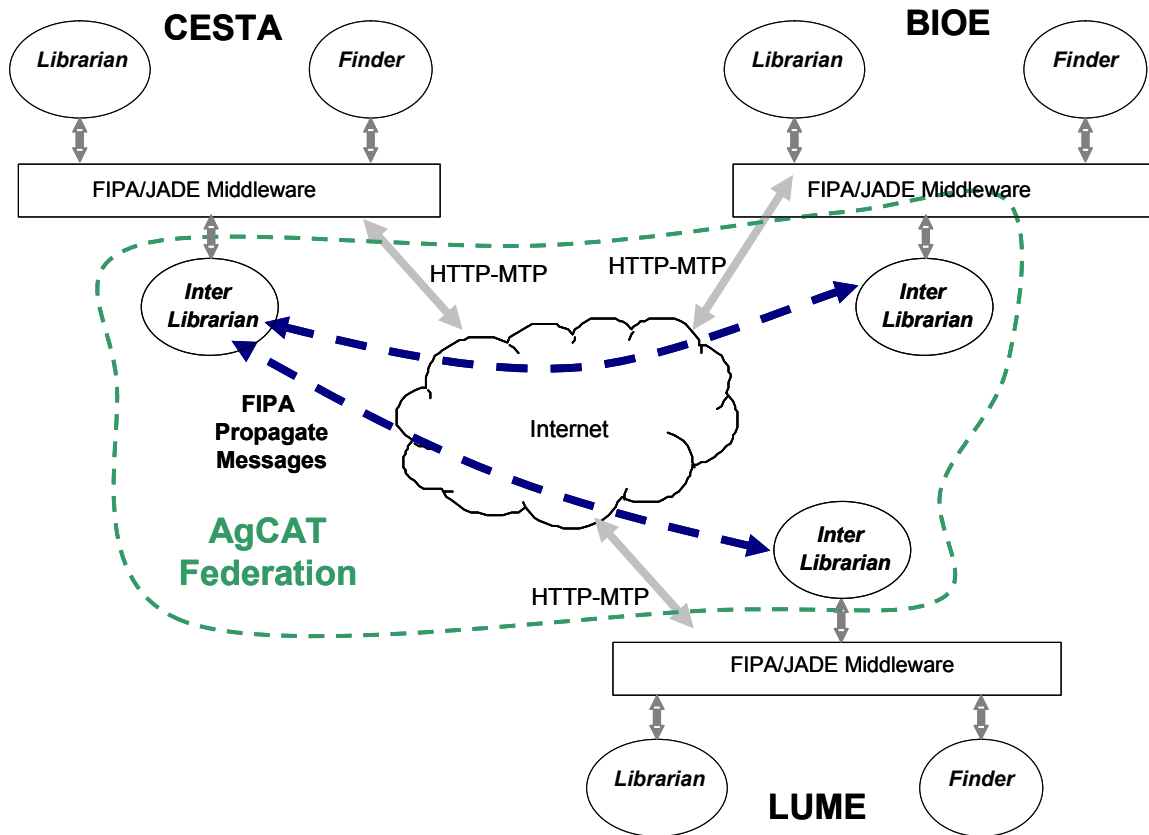


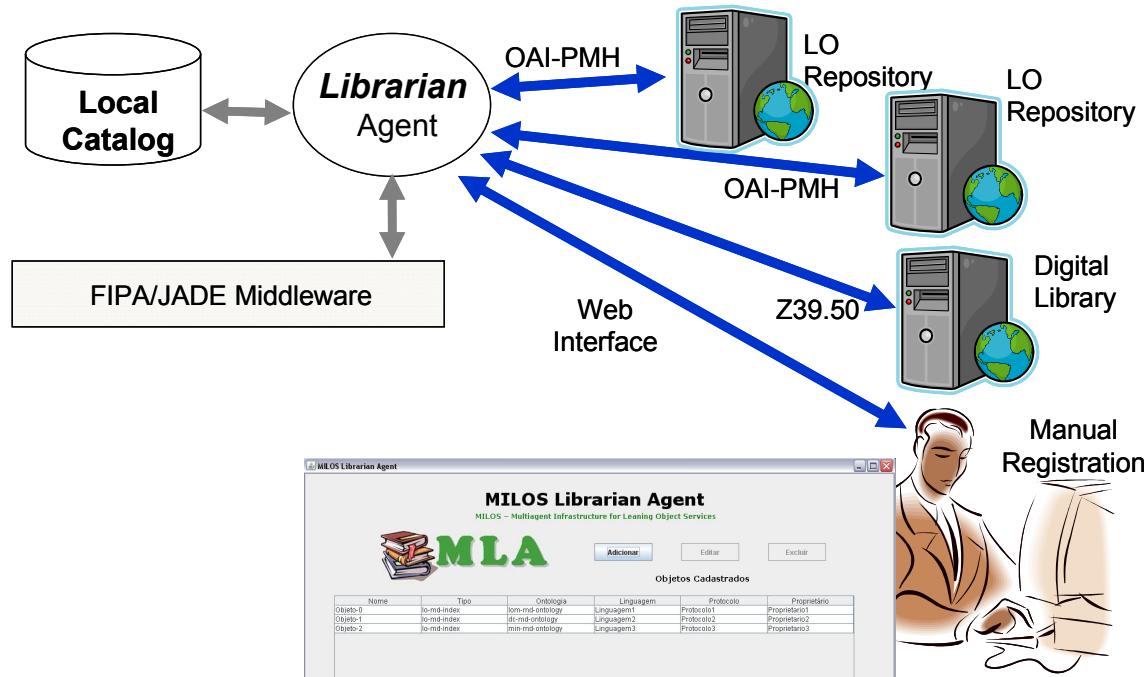
Figure 5: AgCAT Federation Example.

To transport the FIPA-ACL messages across distinct FIPA platforms, as in the case of a federation, it is necessary to use a MTP (*Message Transfer Protocol*). The current version of the JADE platform provides support for two MTP: HTTP (*Hypertext Transfer Protocol*) and IIOP (*Internet Inter-ORB Protocol*). HTTP has become the standard MTP for cross-platform communication, replacing the IIOP (Grimshaw, 2009). By default, the HTTP-MTP is activated on the main container in any other containers MTP is activated. Thus, it creates a server socket in the main container, which waits for connections via HTTP. The communication of agents in the same platform makes use of RMI (*Remote Method Invocation*).

### Metadata Storing and Harvesting

The main function of the *Librarian* agent is to keep a local database of LO metadata, forming the local catalog of metadata. The *Librarian* agent is able to attain the metadata from LO repositories, digital libraries and other metadata catalog services. To do so, this agent maintains a list of the repositories under its responsibility, as well as the configuration information about what kind of protocol must be used to extract metadata information from these repositories. It also provides a Web interface to its users, granting the manual cataloging of LO metadata. This interface lets the *Librarian* agent users register, edit, or delete any LO metadata stored in the local catalog database. The *Librarian* agent will periodically check if there is new or updated metadata information in the repositories. The new or updated metadata information is passed on to the local catalog database. To be fully conforming to the OBAA ontology, the local database will be implemented through a database of RDF triples, compliant to the RDF representation of OWL objects. SPARQL will be the preferential query interface of this database, permitting to formulate arbi-

rary logical queries on the metadata stored in the database. Figure 6 shows the main interfaces of this agent, including the user interface.



**Figure 6: Librarian Agent Interfaces**

To acquire metadata from LO repositories, it is necessary to use a certain kind of metadata harvesting protocol. There are several options to achieve metadata harvesting activities, starting at generic protocols based on directory services like the LDAP (Lightweight Directory Access Protocol), or database technologies similar to SQL queries over ODBC connections, passing through older protocols like ANSI / NISO Z39.50, suitable enough to access the information of digital libraries and reaching to standard harvesting protocols like OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) (OAI, 2009) specifically designed for retrieval of metadata. Initially the Librarian agent will support OAI-PMH and LDAP protocols. In the future, we also intend to add functionalities to the Librarian agent letting it access other types of catalogs of digital content, such as the catalogs of libraries that can be accessed through the protocol ANSI/NISO Z39.50.

The OAI-PMH is a protocol designed to provide interoperability between digital repositories, defining how it should be possible to acquire metadata from these repositories. Its main function is to facilitate the sharing of existing metadata in repositories that support the protocol. The OAI standard (OAI, 2009) defines two basic entities: the data provider and service provider. The data provider function is to search for metadata in databases and make it available to service providers through the OAI-PMH protocol. On the other hand, service providers can *harvest* the metadata for its users (OAI, 2009). In this case, the term harvesting refers to the search of metadata objects in repositories.

According to these definitions, the *Librarian* agent is a service provider using OAI-PMH to harvest metadata from LO repositories that support this protocol. OAI-PMH requests are sent by service providers to data providers, in order to harvest for metadata under the responsibility of a data provider. The response to an HTTP request that has been sent by the harvester consists of an

XML document containing the metadata available in searchable repository. The OAI-PMH defines six verbs to be used in specifying the conditions of a query and is represented as parameters in an HTTP request. The request message is formed by the repository URL combined with the request verb.

Below is shown an example of a query to the BIOE repository pertaining to the Brazilian Ministry of Education using the OAI-PMH:

`http://objetoseducacionais2.mec.gov.br/oai/request?verb=ListRecords&metadataPrefix=oai_dc`

The URL is annotated with three brackets below it:
 

- A bracket under `http://objetoseducacionais2.mec.gov.br/oai/request?` is labeled "Repository URL".
- A bracket under `verb=ListRecords&` is labeled "Verb".
- A bracket under `metadataPrefix=oai_dc` is labeled "Argument".

In this example, the verb *ListRecords* requests that all records available in that repository be retrieved. The *metadataPrefix* argument specifies that only Dublin Core metadata must be retrieved.

## The AgCAT Prototype

The AgCAT prototype was developed in JAVA with the help of the JADE Framework. Only the *Librarian* and the *Finder* agents were developed for the first version of the prototype. The user interface of these agents was restricted to a local (non-web) graphical interface. This interface makes it possible to configure the agents, perform the manual registration of LO in the local database, or search for LO according to the value of metadata. The functionality of the *InterLibrarian* agent was mapped in the Directory Facility (DF) service of FIPA. The local database of metadata was stored in the DF database, allowing not only access to metadata in the local DF, but also metadata from remote DF through the federation of collective DF. The standard DF service of FIPA already supports a type of distributed yellow pages, formed by the federation of DF services from several FIPA platforms. The search for a particular agent (or a particular object) is distributed on all DF services in the federation. The federation of available DF services in FIPA platforms is similar to the federation of *InterLibrarian* agents in the AgCAT systems granting a fast proof-of-the-concept implementation of the federation.

### Development of the Prototype

The development of the prototype started with the creation of a simple software agent able to register its services in a FIPA platform and its DF service. The development of the *Librarian* agent was incremental based on the addition of functionality to the initial agent. In the first place, its user interface was added, which allowed the registration of LO metadata directly to the DF service. Through this interface, it was also possible to delete previously registered objects and to edit these objects. Finally, functionality was implemented to harvest metadata information from LO repositories using the OAI-PMH protocol. The *Finder* agent was implemented soon after the *Librarian* agent became operational. This agent basically performs a search in the DF service for LO entries that satisfy the specified items in the search interface.

This type of solution offered some technical advantages, mostly because it makes use of a standardized service that already supports the concept of a federation of directories. But for this solution to work it is necessary to map the concept of LO in something that can be registered in a FIPA DF. The FIPA DF service only registers information about agents and their associated services. According to the IEEE-LOM definition, LOs are not necessarily agents or a service, but a digital (or non-digital) entity that can be used for educational purposes. Thus it was necessary to find out how to map LO metadata elements in modules that can be registered in the DF service. Interpreting an LO as an agent, even though possible, is technically a complex solution to this problem because it implies the definition and implementation of unnecessary functionality of

agents in LOs. Thus, for cataloging purposes, it was assumed that an LO could be interpreted as a service offered by a given agent if the following guidelines are followed: 1) the type of service indicates that this is a LO entry and not a common agent service, 2) the name of the service provides a unique identifier for LOs, and 3) other information about the service (the properties of the service) detail metadata information concerning LOs. A given LO will be represented as a “service” by an agent. The type of service is fixed and identified by the label *lo-md-entry*, indicating that what is being offered by the service is actually an LO entry composed of the metadata records represented as the *properties* of the service. The service name corresponds to a unique identifier for LO within the agent associated with the service.

Using this technique, all metadata elements from OBAA were mapped in agent services. In order to address the various standards for metadata, the compatibility metadata profiles defined in OBAA have been put to use, which allows mapping all IEEE-LOM and Dublin Core metadata in OBAA metadata.

## First Results

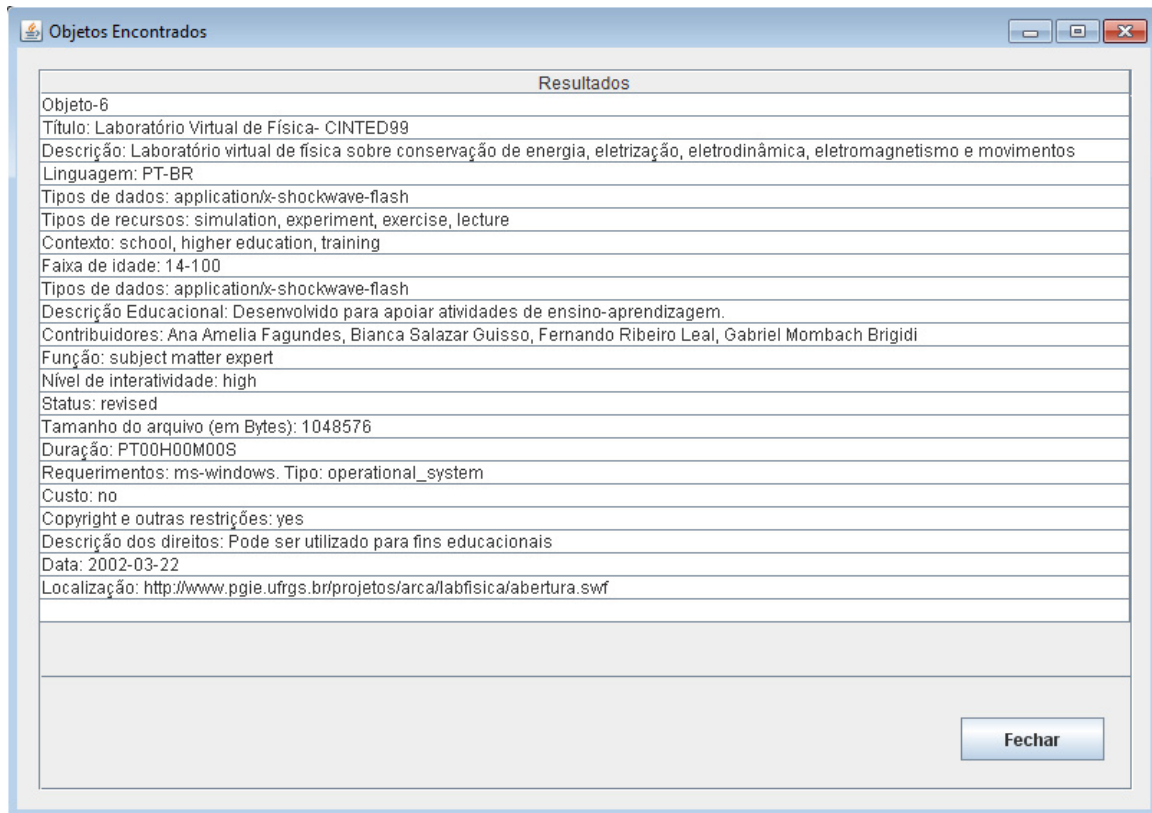
The initial tests of the AgCAT system were focused on the validation of the system’s search features and on the verification of the robustness of the FIPA DF service when used as a catalog of LO. Concerning the search tests, the *Librarian* agent imported metadata information from the CESTA repository at UFRGS, which contained 400 Learning Objects by the time the tests were executed. Figure 7 illustrates the main user interface of the *Finder* agent.



Figure 7: User Interface of the *Finder* Agent

An example of the search for a specific subject can be visualized as shown below. In this case the basic search features of the *Finder* agent are put to use in the search for a particular word held in the title, description, or keywords (subject) metadata regarding the object. The search item speci-

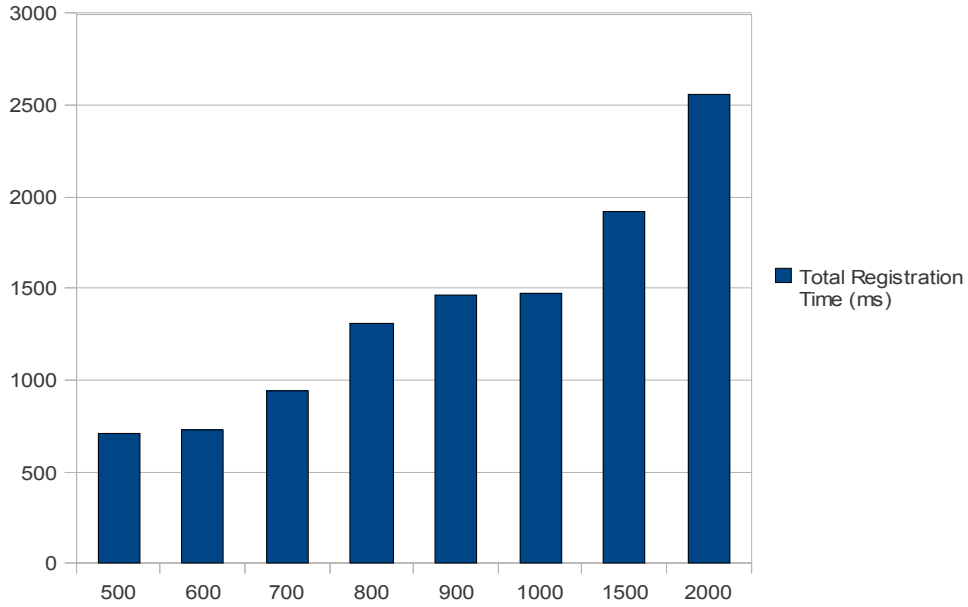
fied in the example (see Figure 7) is the word “eletrodinâmica” (electrodynamics). Figure 8 shows the result for this particular query.



**Figure 8: Search Results Obtained for the Example Query**

In order to verify the robustness of the FIPA DF service, a series of tests were performed in which a fixed number of LOs were registered in the service. A required time for each registration was recorded for every test and compared with other tests. The JADE implementation of the FIPA DF service was tested with the number of 500, 600, 700, 800, 900, 1000, 1500, and 2000 LOs. These objects were stored in files that could be read by the *Librarian* agent. In each one of these tests, the *Librarian* agent imported the metadata from one of these files and registered it in the DF service recording the necessary time to make the registration. Figure 9 shows the total time necessary to register the metadata.

These tests proved, at least for registration purposes, that the JADE implementation of the FIPA DF service is able to store the metadata information of a reasonable number of Learning Objects. It also maintains the linearity for the necessary time for the registration process. However, more tests are indispensable in order to check if this linearity is maintained in both search and update processes.



**Figure 9: DF Service Registration Time Tests**

## Conclusion

The AgCAT system forms the core of the MILOS multi-agent system responsible for searching repositories of learning objects. Consequently, it provides basic and advanced search facilities to other agents of MILOS as to its users just as well.

The initial prototype of AgCAT demonstrated that it is possible to create a metadata searching service based entirely on agent technology. This was an important result, because it demonstrated the feasibility of the strategy adopted to create an agent-based federated catalog of learning objects.

However, several important features of the AgCAT system were not supported by the prototype. As a consequence, the prototype is continually being developed to support all requirements of this system.

The most important features currently being developed are the full support of federated catalogs based on the *InterLibrarian* agent, the parsing of SPARQL queries by the *Finder* agent, and the incorporation, in the *Librarian* agent, of the ability to harvest metadata through customizable LDAP requests or SQL queries. In particular, the need for an *InterLibrarian* agent in the AgCAT system was a direct consequence of the inability of the FIPA DF service to catalog other things rather than agents and services. The *InterLibrarian* agent will, nevertheless, overcome this inability of FIPA platforms.



## References

- ADL. (2001). *Sharable Content Object Reference Model (SCORM) Version 1.2: The SCORM Overview*. Alexandria: ADLnet. Available at <http://www.adlnet.org>
- Bellifemine, F., Caire, G., & Greenwood, D. (2007). *Developing multi-agent systems with JADE*. John Wiley & Sons.
- BIOE. (2010). *BIOE – Banco Internacional de Objetos Educacionais*. Retrieved August, 2010, from <http://objetoseducacionais2.mec.gov.br/>
- Chaib-draa, B., & Dignum, F. (2002). Trends in agent communication language. *Computational Intelligence*, 2(5),1-14.
- Cohen, P., & Levesque, H. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42, 213-261.
- CORDRA Management Group. (2009). *An introduction to CORDRA - Content Object Repository Discovery and Registration/Resolution Architecture*. Retrieved August, 2009, from <http://cordra.net/introduction/>
- FIPA. (2002). *FIPA Communicative Act Library Specification, Std. SC00037J*. Retrieved July, 2009, from <http://www.fipa.org/specs/fipa00037/>
- Grimshaw, D. (2009). *Tutorial 4: Using the HTTP MTP for inter-platform communication*. Retrieved June, 2009, from <http://jade.tilab.com/doc/tutorials/JADEAdmin/HttpMtpTutorial.html>
- IEEE. (2002). *Standard for learning object metadata - IEEE 1484.12.1-2002*. Learning Technology Standards Committee of the IEEE.
- IMS. (2009). *Instructional management systems - Global learning consortium*. Retrieved July, 2009, from <http://www.imsglobal.org/>
- Labrou, Y., Finin, T., & Pen, Y. (1999). Agent communication languages: The current landscape. *IEEE Intelligent Systems*, March-April, 45-52.
- OAI. (2009). *Open Archives Initiative - Standards for web content interoperability*. Retrieved August, 2009, from <http://www.openarchives.org>.
- RIVED (2009). *RIVED - Rede Interativa Virtual de Educação*. Retrieved August, 2009, from [http://rived.mec.gov.br/site\\_objeto\\_lis.php](http://rived.mec.gov.br/site_objeto_lis.php)
- Taylor, C. (2003). *An introduction to metadata*. Queensland University, Australia. Retrieved July, 2009, from <http://www.library.uq.edu.au/iad/ctmeta4.html>
- Vicari, R., Gluz, J. C., Passerino, L. M., Santos, E., Primo, T., Rossi, L., . . . Roesler, V. (2010) The OBAA proposal for learning objects supported by agents. *Proceedings of MASEIE Workshop – AAMAS 2010*. Toronto, Canada.
- Weiss, B. (1999). *Multiagent systems: A modern approach to distributed modern approach to artificial intelligence*. The MIT Press.
- Wooldridge, M. (2002). *An introduction to multiagent systems*. John Wiley & Sons.

## Biographies

### **Carla Fillman Barcelos**

Computer Engineering at Vale do Rio dos Sinos University (UNISINOS), Brazil



### **João Carlos Gluz**

PhD in Computer Science at Federal University of Rio Grande Sul, Brazil. Associate Professor at Vale do Rio dos Sinos University (UNISINOS), Brazil.



### **Rosa Maria Vicari**

PhD in Computer Science at University of Coimbra. Associate Professor at Federal University of Rio Grande do Sul (UFRGS), Brazil.