FABULA PLATFORM FOR ACTIVE E-LEARNING IN MOBILE NETWORKS

B A Khan
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
BASIT(at)IDI.NTNU.NO

Mihhail Matskin
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
Misha(at)IDI.NTNU.NO

ABSTRACT

Connectedness has become a common term of today’s world. The ever-increasing availability of network access has opened new possibilities for individuals to collaborate and share information with one another. With the communication infrastructure in place software support can be used to describe, publish and discover the learning resources. An individual carrying a mobile device has the opportunity to get access to a dynamic and collaborative environment full of similar individuals. In such an environment there is a huge potential to learn form others through sharing of experiences and conducting shared tasks. Learning methodologies can take significant advantage of the capabilities of information technologies and can take the learning experience a step further. However there is still a great need of e-learning systems, such a system can make use of communication infrastructure. This paper is written within the scope of project FABULA (FABULA.Con, 2007). By the virtue of this paper we present the work done to construct a service based e-learning architecture for FABULA system. This architecture uses Web-services and software agents to support efficient collaboration and cooperation for learning activities.

1. INTRODUCTION

The availability of communication infrastructure is not enough for e-learning, there is a need for well designed learning systems. Such a system will provide the supporting services and mechanisms to conduct learning and collaborative activities. Since more and more information become available *we urgently need techniques to help us make sense of all this, to find what we need to know and filter out the rest; to extract and summarize what is important* (Davies John, 2007). Thus such a system needs to take an active part during the learning and collaborative activities while dealing with the challenges such as open environments, heterogeneity and dynamism. A service based approach towards the system design can result in an interoperability (both syntactic and semantic), open, extensible and cooperative system (Jon Mason, 2004) (Wilson, 2005); where each service is intended to fulfill a specific learning goal. Further more in-order to reduce the complexity of the system, categorization and layering of system components (i.e. services) assists in paying due importance to particular aspects of system functionality. We adopt the same approach in project FABULA, where the technical goal is to construct a web-services based e-Learning system for mobile networks. Such a system would operate in the context of a city. The main aim is to support the inhabitants of the city (i.e. pupils) in fulfilling a specific learning need within a specific situation through the use of information and communication technology. By virtue of such a system it would be possible to transform the communication infrastructure of the city into an active medium for learning.

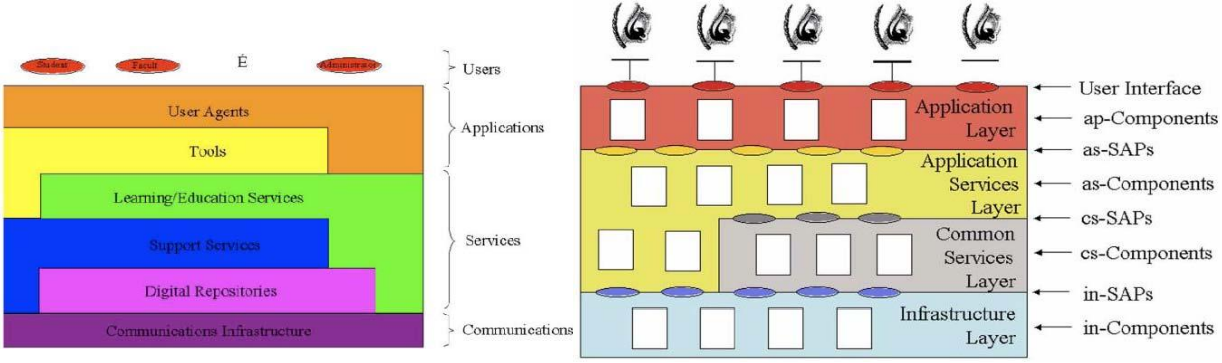
2. EXISTING LEARNING PLATFORMS

There are several e-learning Frameworks / Platforms / Architectures, which provide guidelines for designing the architecture of an e-learning system. Design of an e-learning system can benefit by using the existing frameworks, these frameworks can help in identifying the components and services in an e-learning system. Scott Wilson (Wilson K. B., 2004) suggested that Framework(s) can be use to identify the common services of a system.

One such framework is **IMS Abstract Framework (IAF)**. The core aspects of IAF (Con, 2008) (Smythe, 2003) are Interoperability, Service-orientedness, Component-based design, Layering and Binding neutrality (XML, WSDL, Java, etc). IAF acts as a device which enables IMS to describe the context within which it develops interoperability specifications of its e-learning technology. IMS-IAF provides different views / architectures / models of e-learning system from different perspectives. The idea is to identify different components of an e-learning system by examining it from different angels (Smythe, 2003).

IAF-Logical perspective on Architecture is a collection of six layers; each layer uses the functionalities provided by the layer underneath it. This logical ordering identifies and provides information about all the main entities/components that an e-learning system is likely to have. The main components of an e-learning system as depicted in figure 1 below.

Figure 1. Logical architecture of an e-learning system & layers of an e-learning system (Smythe, 2003)



IAF-Physical perspective on Architecture separates the various components related to the actual deployment of the system. This view deals with the communication infrastructure and service categories. Service delivery engines, delivery devices and federated digital repositories are other main components in this perspective.

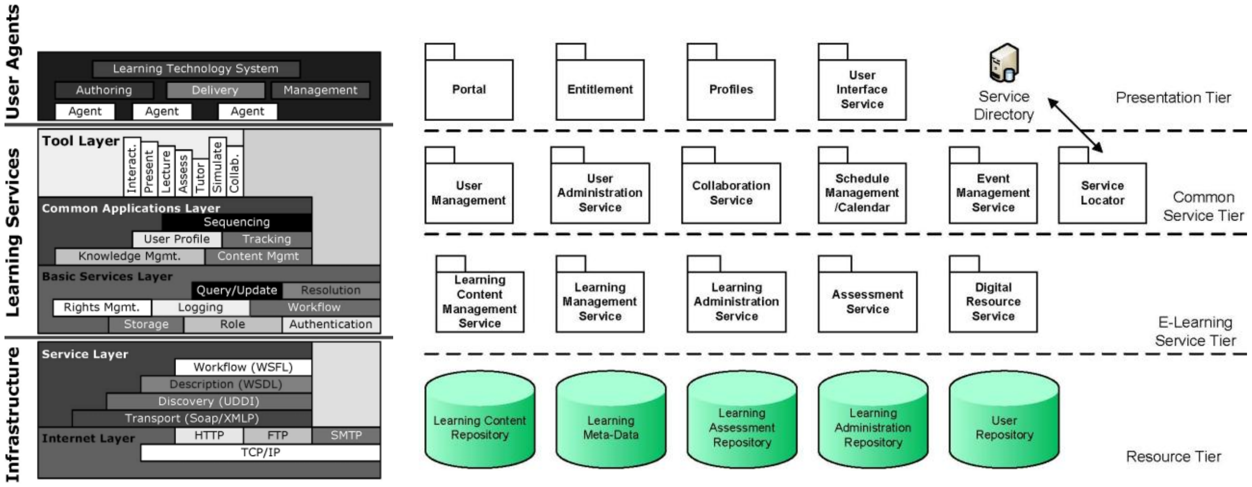
IAF-Functional perspective on an e-learning system is divided into two different perspectives, content perspective and the individual perspective. Content perspective deals with the issues related to the content that is available for learning; these include the repositories, learner's profile management and catalogs. Personal perspective of functional model deals with the issue related to the management of information about the learner.

E-learning framework and E-Framework are the initiatives by U.K's Joint Information Systems Committee (JISC) to produce an evolving and sustainable, open standards based, service oriented technical framework to support the education and research communities (Easterby). *The main set of guideline principals of E-framework are service oriented approach to system and process integration, development promotion and adoption of open standards (such as IMS, OASIS, W3C, IEEE), community involvement in development of the e-framework and open collaborative development* (Members, 2006).

The Learning Systems Architecture Lab (LSAL) conducts research focused on the design and creation of Internet-based technologies for education and training. LSAL came up with a stack E-learning system's layer called *E-learning Services Architecture* (LSAL, 2001). The whole system is basically divided into three main layers. Namely *User Agent layer*, *Learning Services layer and Infrastructure layer* as depicted in Figure 2.

Another such framework is proposed by SUN Microsystems which is **Sun's E-learning Framework;** which is broken down into four layers each performing a critical function with-in an E-learning system (SUN, 2003). The major advantages of Sun's framework are the component based development, faster development, reduced development cost, and security. The framework divides the system into four tiers, namely; *Presentation Tier:* consists of navigation and presentation logic, *Common Service Tier:* constitutes of the services which are common among the learning application, *E-learning Service Tier:* Consists of components that support the non-learning, delivery-related administration functions, and *Resource Tier:* The assessment system that measures student performance against specific learning goals.

Figure 2. E-learning system stack (LSAL, 2001) & SUN's E-learing Framework (SUN, 2003)



A common pattern that we have identified after looking at these frameworks is the fact that.

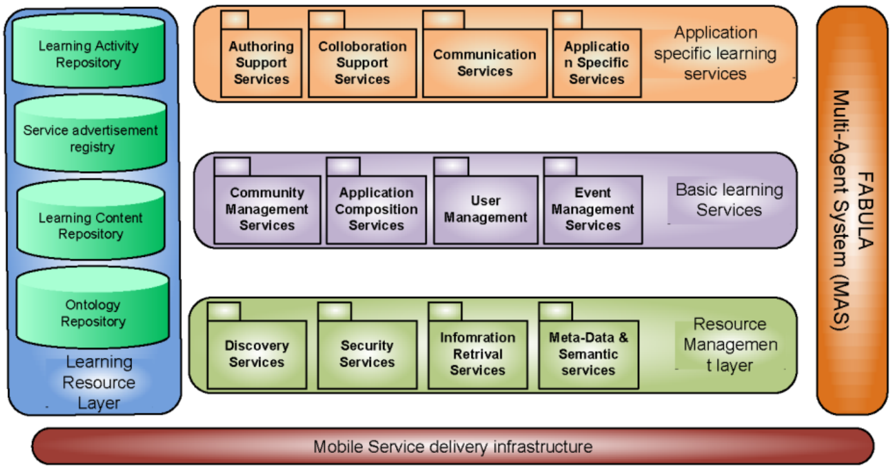
- *Services are always layered over each other; the services in the lower layer provide support to the services above them.*
- *Digital repository is used to manage (store) the learning contents within the system.*
- *Software agents are active components of the system that perform several different functions on behalf of the user*

Building on the input from a framework as a reference model, assists in finding those services which should be combined to realize a particular functionality. FABULA system architecture also follows the same guidelines; however we pay greater emphasis on the more active entities (i.e. software agents) of the system. Among others our main goals while designing FABULA are to create a service based learning system, support dynamic deployment of services, support location and context aware service delivery to the users, discovery and selection mechanism for learning services. In FABULA we have also adopted a layered model, but we believe that having services alone in the system is not enough to assist the user during the learning activities, more elaborated architecture of FABULA is presented in the next section.

3. FABULA SYSTEM ARCHITECTURE

FABULA is built on service based approach for providing the promised functionality. As shown in **Figure** 3 in order to reduce complexity, the system is divided into layers of services. Each layer consists of different categories of services. Such a layering helps to separate the services in the system based on their role during the system execution. The overall framework is divided into six main layers as depicted in **Figure** 3.

Figure 3. FABULA service architectures


Learning Activity Repository
Service advertisement registry
Learning Content Repository
Ontology Repository
Learning Resource Layer
Authoring Support Services
Colloboration Support Services
Communication Services
Applicatio n Specific Services
Application specific learning services
Community Management Services
Application Composition Services
User Management
Event Management Services
Basic learning Services
Discovery Services
Security Services
Infomration Retrival Services
Meta-Data & Semantic services
Resource Managemen t layer
FABULA Multi-Agent System (MAS)
Mobile Service delivery infrastructure

In the architecture of FABULA, as shown in **Figure** 3, every component of the system is generalized to be a service, yet based on roles of services in the system we can make some differentiation among the services of the system. Therefore we look at our system from two different perspectives, *active service perspective* and *passive service perspective*. Passive services are services with some associated description, described in some service description language (SDL), and published in a directory repository. These services can be searched and invoked when needed. On the other hand passive services become active services when they are hosted by a software agent. Software agent or multi-agent system performs the core system functionalities and therefore can be considered as active components of the system. In simpler terms a passive service is just a service, *which is described, published, searched when needed and invoked on demand if available*. On the other hand, a software agent is considered to be the host of passive services *(i.e. when hosted by agent, they become active services)* in the system. In practical terms software agents can host many passive services of FABULA system. Thus each agent has role(s) which are fulfilled by passive service(s) it hosts. Referring to **Figure** 3 the FABULA Multi Agent system (F-MAS) operates at all the layers of the system. Other noticeable items in the system architecture are *Mobile Service Delivery Infrastructure* and the *Learning Resource Layer*. Learning Resource Layer is intended to hold the persistent data in the system. Resource layer and service delivery infrastructure are not discussed in this article.

4. PASSIVE SYSTEM PERSPECTIVE

These services can be dynamically deployed, removed and discovered within the system. Based on the nature of functionality passive services are organized at three different levels in the system, at these levels services are categorized into twelve main categories. An important thing to mention is the fact that all service categories in the *Basic Learning Services* and *Resource Management services* are considered to be life line services in our system. This is to say that we believe that services in these categories are extremely important for system's execution.

Before discussing the services of FABULA system, we describe some ontological terms which are used at many place later in this paper.

- **Learning application:** can be defined as a *work flow or a process* consisting of many activities (services invocations) to achieve a *Learning Goal*.
- **Learning Activity:** is the abstract representation of sequences of steps that shall be done to achieve Learning Goal. This is to say that a Learning goal can be translated to a *Learning Activity*.

4.1 Application specific learning services layer

Include the services which will assist FABULA users in editing documents and other learning artifacts. Services in this layer could be very use-case specific, for example services required for learning in class room environment can be very different from the service required for learning in a museum.

Authoring support services: Include the services which will assist FABULA users in editing documents (Create, update, delete) and other learning artifacts. The main utility of such services will be during the time when several users will edit a learning artifact. This category of services may include.

Document Editing Services (Create, update, delete)	Concurrent Artifacts Editing management services	Calendar Management Services

Collaboration support services: Consist of complementary services; these services may never play a direct role in the Learning Application. However these services may be invoked by the user in an ad-hoc manner for collaboration. These services may include (Canova Calori and Divitini, 2008).

Social network representation and management	Place representation and management

Communication services: These services also consist of complementary services and may never play a direct role in the Learning Application. But they can be invoked by the user in an ad-hoc manner for communication. Category of communication services may include the following services

Text messaging	Picture messaging	Voice communication

***Application specific services*:** These are the services which may be dynamically added to the system (i.e. advertised in the *Semantic* Service Advertisement Repository). Since we look at FABULA as a system that could be used in many different learning contexts then for each context different services will be required. Thus these services will be completely specific to the context of application, and hard to predict. With the addition of such services system may start to support more learning use-cases or may start to perform better for other Learning Applications.

4.2 Basic learning services layer

Services at this layer are the intermediate level services, they do not directly interact with user - neither they directly deal with the learning content. These services use the functionality of the services in the lower layer - furthermore they support the services in the upper layer at the run time. Accessing or providing functionality to/from all the services in other layers is done through clearly defined *Service Access Points (SAP)* (i.e. Service Interface). Thus, maintaining *autonomy* and *heterogeneity* in the system. These services perform the functionalities which are common to many learning processes.

Community management services: this category of services consists of services related to managing user community, since FABULA is all about collaboration among its users. These services are vital; the sole purpose of these services is to manage different structures of user communities. Services such as recommendations about the social connections and contacts are belongs to this category.

Application composition services: services in this category are the responsible for constructing the Learning Applications on demand of user***. User management services:*** manage the storage, retrieval and search of user profile. ***Event management services:*** we consider FABULA to be an event driven system. Services in this category consist of services which manages different events of FABULA. The main idea to have such an event based system is due to the fact that FABULA is a distributed services based system, thus there is no possibility of integrated programming style (function calls and exception handling). Events will provide necessary abstraction to deal with various kinds of flow primitives as needed by Learning Applications or dependencies among service invocations.

4.3 Resource Management services layer

This layer provides the most basic system level services for FABULA. Theses services handle the data that is stored in and accessed from the repositories. Services in this layer are very much static in the system.

Discovery services: Provide information about the other services (basically services in the upper layer) which are available in the FABULA system. These services directly operate on the service advertisement repository. ***Security services:*** Category contains all the services related to the security of the system. These services take care of the issues related to the identity and permission management of the user or its agent.

Information retrieval services: This category of services includes the services associated to the content management and retrieval from the content repository. This category include the following service

<table>
<tr><td colspan="3">Service for storing, retrieval, organization and searching the content in the content repository</td><td colspan="3">Services for content reputation and recommendation (social content filtering and collaborative filtering)</td></tr>
<tr><td colspan="2">Cataloguing Service</td><td colspan="2">Content Archival Service</td><td colspan="2">Digital right management Service</td></tr>
</table>

Metadata and semantic annotation service*:* This category of services work over the ontology repository. The services in this category are required to do semantic reasoning with the concepts of FABULA ontology. The learning content saved in the content repositories can be associated with semantic metadata and ontology can be queried through these services. Services in this category include

Semantic reasoning service	Metadata creator service
Terminology service	Ontology query service

5. ACTIVE SYSTEM PERSPECTIVE

As discussed earlier active service perspective of system considers the system to be a collection of software agents, hosting different services, communicating and collaborating with one another. Inorder to manage the complexity of F-MAS we integrate the concept of AGORA (Matskin, 2000), (Matskin M. , 1999) (Matskin M. a., 1998). AGORA can be considered as a conceptual node or a meeting point for different agent for cooperative work. For each cooperative work (act of communication, coordination or negotiation) point there can exist an AGORA node. The AGORA node provides some *default agents*; these agents provide the services needed to support the coordination, negotiation and shared activity execution within the context of that AGORA. Each AGORA node can also contain registered agents which use the functionality provided by the AGORA node. Registered agents represent participants of cooperative activity and default agent corresponds to the management services which are created by default when a new AGORA is generated.

Presence of default agents in AGORA based MAS insures that the architecture of registered agents can be kept simple. Since the default agents takes care of complex details of enforcing coordination decisions, security checks, negotiations, message interoperability and translation for agent communication in the MAS. Having such facilities allows the design of registered agents to be more simple and oriented towards the useful processing.

Using the AGORA approach to MAS design requires to first understand the problem to be solved at hand and then to identify the participants in the cooperative activity. The next step is to identify the cooperation point among the participants, once the cooperation points are identified; the next step is to map the participants to agents and finally create agent for each participant and AGORA for each cooperation point. For FABULA depending on the scenario (learning in museum or hunting for treasure in the city) the structure of the F-MAS may differ, this is due to the fact that different use-case scenarios may have different cooperative points (i.e. AGORA nodes) for agent work.

In FABULA system we consider that all the users in the system are represented by FIPA (FIPA, 2000) compliant software agents which are running on the local device of the user. The user agent may perform different kind of offline processing for its user. We believe that all the learning activities in FABULA will take place as collaboration between different individuals organized in groups. We strongly believe that a common set of AGORAs can be identified in almost all scenarios. Such as

- **FABULA system AGORA :** coordinates the activities of FABULA system, includes the agents for providing the services in *Resource Management Layer*
- **Group AGORA :** coordinate the activities of individual belonging to different groups
- **Inter Group AGORA :** coordinate the activities when participants from one group want to communicate with the participant of the other group

6. EXAMPLE OF AGORA BASED TREASURE HUNT GAME

In this section we present a simple use-case scenario; and use it to elaborate the FABULA architecture presented in the previous sections. Our scenario assumes that the following are the participants of the learning activity

- *A teacher*
- *20 Students*

Our example scenario revolves around treasures, these treasures are hidden at different places in some surrounding, each treasure can have a number of clues associated to it, and these clues contain the information about the location of treasure in the surrounding. These treasures are arranged in sequential order, only after finding the first treasure, students can search for the next treasure. In order to get the information hidden in the clue students have to use their common/general knowledge in doing so students have to collaborate; for collaborating while being mobile, requires some services to be available to the students.

Suppose that the teacher logs into the FABULA and system presents the teacher with a screen where she can create the new treasure hunt game (i.e. learning application). Students are divided into 4 groups, consisting 5 students in each group. After creating the game she informs students by broadcasting a message and informing them about their groups and they are invited to play the game. During their hunt for the treasure, each group has several services available to them for collaboration and coordination of their activities. The system continuously infiltrates the learning activity, by providing the participants of the game with useful advices and thus helps them during their learning activity.

6.1 Designing the F-MAS for our example

By simple inspection we can identify that there are 21 participants (a teacher & 20 students) in this cooperative activity, as described earlier all participants in a cooperative learning activity are represented in the system by their software agent. For the purpose of this paper we consider a limited number of cooperative points. All the cooperative points are mapped to AGORA nodes. By inspection we find out that following AGORA nodes can be identified

Inter-group AGORA: Since students are organized into groups, each student group (i.e. where cooperation will take place) can be mapped to an AGORA node. There can exist 0 to n instances of such AGORA in the system, depending on the number of students and group size; in our case there will be 4 such AGORA nodes. Each participant in the group will be presented by its appropriate software agent. Such an AGORA node will provide the services which are needed for the group communication and coordination. Majority of services (if not all) in this AGORA node will come from the *Application Specific Learning Service Layer.* Services to manage the representation of social network and the management of activities within this social network, including communication and collaboration; such as text messaging, voice messaging, chatting, document editing etc… are provided by the default agents in this AGORA node.

Teacher AGORA: This AGORA node will allow teachers to communicate and coordinate their activities. This node is identical to *inter-group AGORA*. Since we do not consider more then one teacher in this scenario we will not discuss the functionality of this node any further.

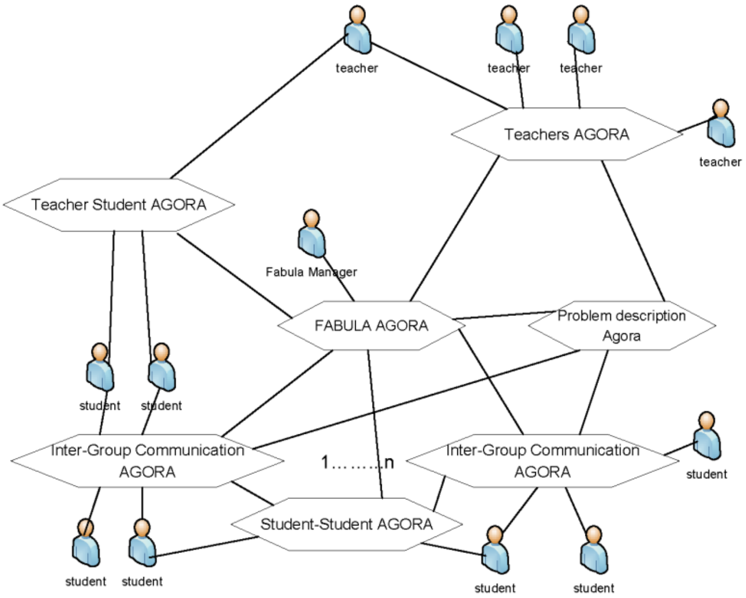
Problem description AGORA: Such a cooperation point exists to take care of the activities of the teacher while describing the treasure hunt problem. This AGORA provides the teacher with all the services and functionality required by the teacher to describe the treasure hunt game. The main set of services provided by this agora will come from *Basic Learning Service Layer* and *Resource Management Services Layer.*

Teacher Student AGORA: Such a collaboration node (i.e. AGORA) is required when the students wants to communicate with their teacher. The main set of services in this AGORA will come from *Application Specific Learning Service Layer and Basic Learning Service Layer.* Service such as community management service will be required when two communities need to communicate with one another.
Student-Student AGORA: Such a collaboration node is required when students from one group of students want to communicate with another group of students.
Default AGORA: This AGORA is the starting point for FABULA system. It controls all the system level activities of the system. It contains most of the services in *Resource Management Service Layer and Basic Learning service layer.*

6.2 Detailed processing of AGORA based system

Assume the teacher enters the FABULA mobile learning system. As soon as she logs (registers) into the system the Default AGORA creates a new agent for her in FABULA system and she gets access to a newly created personal agent. We assume that the users of FABULA might already have a user agent running on their mobile devices. Thus in such a case the newly created personal agent can act as a proxy of the actual user agent in the FABULA system. FABULA system presents the teacher with a screen to create a new treasure hunt game and thereby creates a new AGORA ("Problem builder AGORA"). The newly created *Problem description AGORA* is intended to provide the teacher with the necessary services which are required by the teacher to create a learning application. Once a learning application is created the teacher asks the system to invite the students to start the treasure hunt game.

Figure 4: AGROA based Treasure hunt



As a result of this activity, a new AGORA is created for each group of students. This *Inter-group communication AGORA* contains all the required services that the teacher allowed the students to use (when teacher creates the game she can decide upon these issues) during the run time of complete game. This AGORA provides a context within which certain services are available to the students of the system. In case students from a group want to contact the teacher and seek help during the learning activity. In such case there will be created a new *Student-Teacher AGORA* that will facilitate the communications between teacher and students. *Student-Student AGORA* will be created to facilitate the coordination and communication among the student groups. Figure 4 depicts the organization of AGORAs for treasure hunt game.

REFERENCES

Book

Davies John, R. S. 2007. *Semantic Web Technologies Trends and research in ontology based system*. WILEY.

Conference paper or contributed volume

Con, I. 2008. IMS Global Learning Consortium, Inc. *(IMS/GLC). Retrieved 04 02, 2008, from IMS: http://www.imsglobal.org/aboutims.html*

Canova Calori, I. and Divitini, M., 2008. *Reflections on the role of technology in city-wide collaborative learning. Proceedings of mLearn2008 Conference*. Ironbridge Gorge, UK, pp. 74-81

Easterby, S. (n.d.). 2008 The e-Framework and the ELF. *Retrieved 10 02, 2008, from http://www.elframework.org/news/e-framework*

FABULA.Con. 2007. FABULA - *the city as an arena for learning*.

FIPA. 2000. Foundation for Intelligent Physical Agents. *Specification [FIPA]*.

Jon Mason, K. B. 2004. Trends and issues in e-learning infrastructure development. *A White Paper for alt-i-lab 2004. Prepared on behalf of. DEST (Australia) and JISC-CETIS (UK)*

LSAL. 2001. E-learning system stack. *Carnegie Mellon University*.

Matskin, M. . A 1998. An Architecture for Multi-Agent Support in a Distributed Information Technology Application. *International Workshop on Intelligent Agents in Information and Process Management , 47-58*.

Matskin, M. A. 2000. Agora: An Infrastructure for Cooperative Work Support in Multi-Agent Systems.

Matskin, M. 1999. Multi-Agent Support for Modelling Co-operative Work. *Proceedings of the IFIP TC6 WG6. 7 Fifth International Conference on Intelligence* in Networks , 418-432.

Members, J. C. 2006. Briefing paper on the e-Framework and Service-Oriented Architecture. *JICS. JISC*.

Smythe, C. 2003. IMS Abstract Framework: White Paper, version 1.0, edt. *IMS Global Learning Consortium* .

SUN. (2003). E-Learning Framework, Technical white paper . *SunMicrosystem, INC*.

Wilson, K. B. (2004). Service-oriented frameworks : Modelling the infrastructure for the next generation of e-learning systems. JISC-CETIS (UK).

Wilson, S. (2005). Can web service technology really help enable coherent diversity in e-learning*? JISC.*