Journal of Universal Computer Science, vol. 20, no. 12 (2014), 1649-1669
submitted: 31/5/14, accepted: 15/10/14, appeared: 1/11/14 © J.UCS

A Decentralized Infrastructure for Ubiquitous Learning Environments

Jorge Luis Victória Barbosa
(University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil
jbarbosa@unisinos.br)

Débora Nice Ferrari Barbosa
(Feevale University, Novo Hamburgo, Brazil
deboranice@feevale.br)

Jezer Machado de Oliveira
(University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil
jezeroliveira@gmail.com)

Solon Andrade Rabello Junior
(University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil
solonrabello@unisinos.br)

Abstract: This article proposes a decentralized infrastructure for ubiquitous learning environments called Global. The two main contributions of Global are its decentralized strategy and its extensible architecture based on software agents. Global can be specialized to create ubiquitous learning environments through the extension of its agents or through the addition of new agents. The article presents the infrastructure architecture, describing the agents and auxiliary components. Moreover, the article approaches the implementation and evaluation of Global through two applications dedicated to education. The evaluation showed that the infrastructure could be used for the development of decentralized learning systems, supporting specifically the characteristics considered as strategic for ubiquitous learning.

Keywords: Decentralization, Ubiquitous Computing, Ubiquitous Learning
Categories: L.3.0, L.3.6, L.7.0

1 Introduction

Approximately 20 years ago, Mark Weiser introduced the concept of Ubiquitous Computing [Weiser, 91], predicting a world where computing devices would be present in objects, environments and human beings themselves. These devices would interact naturally with the users without being noticed. Ten years later, Satyanarayanan reinforced the concept through a paper that would become a classic [Satyanarayanan, 01].

The evolution of technology has enabled ubiquitous computing [Caceres and Friday, 12], as mobile devices become smaller and more powerful. Moreover, the advent of wireless communication technologies has enabled devices to access services anywhere and anytime. Additionally, the increasing adoption of Location Systems

[Hightower and Borriello, 01] has been stimulating Location-based Services [Dey et al., 10] and Context-aware Systems [Knappmeyer et al., 13].

This technological reality is creating new opportunities for various areas, such as Accessibility [Tavares et al., 12] and Health [Vianna and Barbosa, 14]. Education, as well as other areas of knowledge, makes use of these new technologies to improve their practices and approaches. The application of these technologies in improving education strategies gave rise to a research front called Ubiquitous Learning [Wagner et al., 14].

Immersive technologies such as virtual worlds and augmented reality have been considered promising approaches to support and enhance learning [Serio et al., 13]. In recent years, various initiatives related to Immersive Education have been carried out [IED, 14]. Additionally, ubiquitous and immersive technologies have been integrated to support Immersive Ubiquitous Learning Environments [Liu et al., 09]. The full achievement of this approach is a challenge, because this kind of environment would enable to learn in everyday situations [Weiser, 91] through an immersive [Serio et al., 13] and context-aware way [Knappmeyer et al., 13]. In this sense, infrastructures for learning environments that fully support the features of ubiquitous computing [Satyanarayanan, 01] assume a prominent role.

There are several proposals for ubiquitous learning environments, such as LOCAL [Barbosa et al., 11], JAPELAS [Yin et al., 10] and GlobalEdu [Barbosa et al., 12], as well as the works proposed by [Chen and Li, 10], [Yang, 06] and [Zhang et al., 05]. Most proposals for ubiquitous learning environments use centralized structures [Yang, 06]. However, this approach is not adequate for the ubiquitous computing definition presented by [Satyanarayanan, 01], whose characteristics include the requirement for mobile networks, high availability, localized scalability and mobile access to information. In contrast, decentralized models support such characteristics more easily.

In traditional centralized models, the communication takes place in a pre-configured infrastructure through an intermediary (server) that is responsible for delivering the message, which hinders the creation of mobile networks. In turn, in decentralized models, communication between users takes place directly. By integrating this feature to the use of decentralized routing algorithms (*Ad Hoc* networks), it is possible to create mobile networks. Moreover, if a server goes offline or becomes inaccessible, so do the elements it manages or, at least, they lose some of their functionalities. When an element in a decentralized model is offline or inaccessible, only the functionalities that depend on it are compromised, thus ensuring greater availability. Localized scalability is the ability to prioritize interaction with closer users [Satyanarayanan, 01]. In decentralized environments based on *Ad Hoc* networks, the network topology itself favours localized scalability, because it can be controlled by the number of hops between nodes. In centralized environments, this management requires the participation of the server, thus reducing scalability.

Moreover, three other factors specifically stimulate the researches related to an infrastructure for ubiquitous learning environments: (1) even with the advancement in mobile device technologies, tools and APIs for development of ubiquitous systems are still quite limited compared to the development of desktop applications; (2) the development of ubiquitous applications is a costly process due to the variety of technologies needed for their implementation, such as wireless communication and

location systems, and (3) ubiquitous learning systems have common functionalities, especially profile management, communication between users and content management.

Considering this scenario, this article proposes a decentralized infrastructure for the development of ubiquitous learning environments, called Global. Global offers software agents to perform tasks that are common to the ubiquitous learning process. Customization to create environments is obtained by extension of these agents or by adding new ones. The text is organized into six sections. The second describes four ubiquitous learning environments that are related to Global, using mainly the parameter of decentralization. In this section, a comparison between related works creates the base to discuss the Global contributions. Section three presents the Global, mainly describing its architecture. The fourth section discusses aspects of prototype implementation. The fifth uses the prototype to evaluate the proposal by creating two learning applications. Finally, the last section presents the conclusions of the study, focusing on the Global contributions and future works.

2 Related works

This section describes and compares four ubiquitous learning environments, which are decentralized or present some decentralized characteristic.

2.1 Japelas

Japelas [Yin et al., 05] is a system that enables users to learn treatment expressions in the Japanese language. The students, carrying mobile devices, are assisted in the task of identifying the appropriate expressions for each context. Japelas considers users´ profiles and their location through RFID technologies. The environment is decentralized and the interaction among mobile devices takes place directly, using infrared communication (IR). Japelas only allows the contact between two users. A new version called Japelas2 [Yin et al., 10] is oriented towards supporting the interaction between many people, with a collaborative/social approach. Both systems use location technology and user profiles, but they focus on a specific application, namely, the correct use of Japanese polite expressions. Moreover, they do not include a structure that facilitates the addition of new languages.

2.2 Framework for social interaction with ubiquitous learning support

Zhang et al. proposed a framework for ubiquitous learning and social interaction between learners [Zhang et al., 05]. Moreover, they defined a model for building social interaction in ubiquitous learning environments, which is based on the *encounter function* proposed by [Jin, 03]. Jin´s approach is divided into three parts: *encounter*, *communication* and *collaboration*. The framework follows the same structure including three functional components with the same names. Based on these components the learners can collaborate through a *learning network*. The framework is indicated as a platform to support ubiquitous learning. However, it does not fit the definition of ubiquity established by [Satyanarayanan, 01], because it does not support location and context awareness.

2.3 P2P collaboration-oriented ubiquitous learning environment

Yang proposed a ubiquitous learning environment that supports point-to-point (P2P) collaboration [Yang, 06]. The model offers learning support through the creation of virtual communities where instructors and students discover, access and share resources. Moreover, the communities´ members can communicate between them and engage in group discussions. The environment uses the context awareness as an interactive model between learners and services. Accordingly, two ontologies are proposed, one to describe learners and one to describe services. Based on these ontologies, an interactivity model conducts a pattern matching to discover context-oriented services. Context detection is divided into two parts: client and server. At the client, the information is collected and, at the server, it is processed to define the learner context. The environment is not fully decentralized, and requires the intervention of a centralized server to determine the learner context.

2.4 GlobalEdu

GlobalEdu [Barbosa et al., 12] is an environment that supports large scale ubiquitous learning, mainly considering learners´ profiles and contexts. The GlobalEdu´s architecture is organized in three layers. The *Application Layer* is a Pedagogical Agent (PA) that assists learner interaction in the environment. The *System Layer* has a set of modules to assist the learner's educational process in the ubiquitous environment. These modules consider the learners´ contexts and profiles, mainly focusing on learning collaboration [Barbosa et al., 11] and content management [Barbosa et al., 13]. The *Execution Layer* should be a generic ubiquitous middleware to support the basic operations of the environment. Currently, GlobalEdu is running on a middleware called ISAM. GlobalEdu adopts a client-server model. The *Application Layer* provides peer-to-peer communication between PAs, allowing a decentralized approach. On the other hand, the *System* and *Execution* layers run in servers, characterizing a centralized approach.

2.5 Evaluation of related works

A related works comparison is showed on Tab. 1. Japelas [Yin et al., 05, Yin et al., 10] has a specific purpose, namely supporting the learning of treatment expressions in the Japanese language, and does not present structures that facilitate the addition of new features. Japelas is a decentralized environment, but its communication system is limited to the use of infrared. Furthermore, the system has a message routing structure and serves primarily to identify which formalism should be applied in the communication between users.

The framework for social interaction [Zhang et al., 05] enables the customization of its tutor, which can be tailored to a specific area, but lacks structures that facilitate the addition of new features. It is a decentralized environment, but fails to deliver two essential characteristics of ubiquitous systems, namely location sensitivity and context awareness. The P2P collaboration-oriented environment [Yang, 06] does not offer structures that facilitate the addition of new features. Moreover, the environment is not fully decentralized, and requires the intervention of a centralized server to determine the learner context, thus affecting the application scalability. Additionally, it maintains some of the information on the server, thus limiting mobile access.

GlobalEdu [Barbosa et al., 12] adopts a client-server model in which communication between system modules takes place through servers. The pedagogical agents communicate with the servers and also with one another.

The first column of Tab. 1 shows the main characteristics of ubiquitous computing as discussed by [Satyanarayanan, 01]. None of the related works supports all of them. Moreover, none is fully extensible. The following sections discuss how Global supports these missing aspects in ubiquitous learning environments. Specially, section 6 revisits the characteristics listed in the table and discusses the ways in which Global guarantees them.

	Japelas	**Zhang et al. Framework**	**Yang environment**	**GlobalEdu**
Objective	It supports the learning of treatment expressions in the Japanese language	It supports students learning, focusing on their social skills in a ubiquitous environment	It creates virtual communities where instructors and students can discover, access and share resources	It allows learners to build their knowledge anywhere and anytime, in a manner that is adapted to their profiles and contexts
Location sensitivity	RFID Tags	No tracking system	Combined technologies	It relies on the middleware used in the *Execution layer*
Communica-tion	Infrared	P2P	P2P and client-server	It relies on the middleware used in the *Execution layer*
Mobile networks	Partial, no routing algorithm	Yes	Yes	It relies on the middleware used in the *Execution layer*
Distributed security	Not Specified	Not Specified	Not Specified	It relies on the middleware used in the *Execution layer*
Context awareness	Yes	No	Yes	Yes
Localized scalability	Yes	Yes	Yes	It relies on the middleware used in the *Execution layer*
Mobile access to information	Yes	Yes	Partly, because part of the information is stored in the server	No, because all information is stored on the server
Extensible	No, it is focused only on the treatment expressions of the Japanese language	Although it is a framework, it only enables customization for its tutor	No	Partly, because it enables customization for the Pedagogical Agent (PA) in the *Application Layer*

Table 1: Related works comparison

3 GLOBAL Infrastructure

The Global architecture is showed on Fig. 1. There are six software agents which offer a set of features for learning support. They can be seen in the middle of the figure. The agents are supported by four components which can be accessed through

APIs. Each instance of Global runs on a device, and the absence of an agent compromises only the features that rely on it. The following sections describe the agents and components.

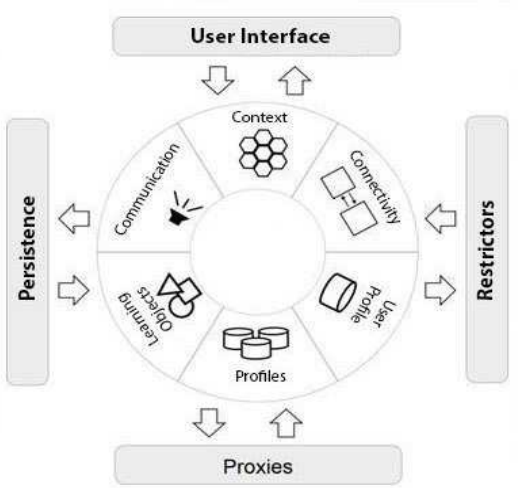


Figure 1: Global architecture

3.1 Connectivity Agent

Communication between agents is articulated by the Connectivity agent. The messages use the FIPA-ACL specification. In the exchange of messages between agents of the same instance, namely, when agents operate on the same device, the Connectivity agent performs a direct delivery to the recipient. In the exchange of messages between agents of different instances, the agent is in charge of finding the best way to deliver the message. This requires the agent to choose one of the available proxies that have access to recipient agent device and send the information through this proxy to the Connectivity agent of the recipient´s instance.

Each domain should provide a *webservice* with an authentication service, in case a device wants to prove the authenticity of another. This authentication is based on public and private keys, in which the private key is recorded on the device and the service in the domain provides the public key. This authentication is an optional process because it is possible the device does not have permanent access to the web.

3.2 Context agent

Contexts are pieces of information used to characterize the situation of a participant in an interaction [Dey, 01]. Global defines two sets of context characteristics: *Environment* and *Relationship*. The environment characteristics are collected by monitoring the environment, such as the position in space and time, for example, a context can be defined as a region linked with a specific time interval. The relationship characteristics, in turn, can be used to group users, such as by *status* (for example, teachers or students) or by areas of knowledge and interests.

Contexts may be derived from others, enabling a context to contain *subcontexts*. Furthermore, a context can be *centralized*, when only a user controls the context management, or *decentralized*, when any context user can manage the context.

In Global, the contexts can use these features to configure a wide variety of situations. For example, a class could represent a context centralized on the teacher, which would last six months. The teacher could create a context derived from the class, depending on physical location and time, which would represent a specific class, also defining the presence in the classroom through the time the student was in the context. During class, the teacher could apply a group assignment. Therefore, students could create decentralized contexts, forming groups whose duration would be the remaining time in the class. The decentralized contexts would allow for the exchange of information and materials within the group.

The description of contexts is based on the following metadata: (1) *Id*: unique identifier that represents the context; (2) *Description*: text-based description of the context; (3) *Ontology*: this metadata indicates the structure that represents the information in the context; (4) *Owner*: identifier of the context creator; (5) *Centered*: it identifies whether the context is centered on its creator; (6) *Restrictions*: it lists the *restrictions* of the context, as described in section 3.7; (7) *Information*: it forms an *information tree* on the context, as described in the next paragraph.

Each context has an area for the information storage, which is shared by the context users. If the context is not centralized on its creator, any participant can add or remove information. The synchronization of this information becomes complex due to the decentralized nature of Global. Moreover, the users can enter and leave the contexts anytime. Synchronization is based on a tree structure called *Global Information Tree.* This structure is similar to DOM (*Document Object Model)*, in which each tree contains part of the context information.

3.3 User Profile agent

The User Profile agent manages user information making it available to other Global agents. Global does not require the use of a specific pattern profile; instead, it uses the structure of the *Global Information Tree* to store profile information, leaving the possibility for each specialization infrastructure to use the profile model deemed most appropriate. Two types of nodes are used to build the *Profile Information Tree*: *public nodes* and *private nodes*. Only agents in the same instance of the User Profile agent can access private nodes. On the other hand, any agent can access public nodes.

Despite the fact that Global does not require a specific profile pattern, it implements the PAPI model. The choice was based on two characteristics of this standard: (1) *Flexibility*: PAPI can be extended and all its components are optional; (2) *Modularity*: the profile fields can be treated separately.

The user profile is divided into seven categories: (1) *Personal Information*: it stores basic user information; (2) *Security*: it stores user credentials, such as password, public key and private; (3) Information on Relations (*Relation)*: it defines a list of relationships between users; (4) Information Preferences (*Preferences)*: it describes users' preferences, in order to facilitate their interaction with the system; (5) Performance Information (*Performance)*: it stores the users' histories, containing their evaluations based on established metrics and who certified the assessment; (6) Information Portfolio (*Portfolio)*: it stores a set of references to the work done by

the users; (7) Information Interests (*Interests)*: it stores the users' interests through the LIP standard, since this kind of information is not part of PAPI.

3.4 Profiles Agent

Global is decentralized, so there is no centralized storage of users' information. This generates the need to manage the profiles of other users on the user's own device. This is the role of the Profiles agent, which manages access to the repository of other user profiles. The Profiles agent has four settings to update the profiles, which can be used depending on the needs of the application or the capacity of the device:

- *Request by demand*: when an agent requests a profile information, the Profiles agent requests it from the User Profile agent in the device owner's profile;
- *Request by demand with cache*: when a piece of information about a user is requested, the Profile agent contacts the User Profile agent in the device owner's profile, asking whether there is an update for the information; if there is no update, it uses the local information that is available;
- *Information storage*: Profiles agent keeps information on other users. Every time a user is accessible, it updates outdated information;
- *Information storage with real-time update*: through the subscription structure, the Profiles agent requests to the User Profile agent to report owner changes in the profile, thereby keeping data up-to-date in real time.

3.5 Learning Objects agent

The Learning Objects agent acquires, stores and distributes learning objects. At Global, the metadata specification of these objects follows the LOM standard. Global uses the following categories: (1) *General*: it brings together general information describing the object; (2) *Lifecycle*: it gathers information that describes the characteristics related to the history and current state of the object; (3) *Technical*: it gathers the object's requirements and specifications; (4) *Educational*: it gathers the educational and pedagogic characteristics; (5) *Rights*: it contains the intellectual property rights and terms of use for the object; (6) *MetaMetadata*: it gathers information about the metadata instance that describes the object; (7) *Relation*: it gathers the definitions about a learning object's relationship with other learning object; (8) *Annotation*: it provides comments on the educational use of the learning object and information about when and by whom the comments were created; (9) *Classification*: it describes the learning object in relation to a specific classification system.

Learning objects can be made available in two ways: (1) by the very user that adds a learning object, for personal use or for sharing with other users; (2) by requesting of other users, because the agent allows users to request the sharing of learning objects.

With regard to the decentralization of Global, we adopted a distributed model for shared objects, based on P2P networks, in which a *bitTorrent* network is created between users who have or want a particular learning object. Fig. 2 shows the operation of the learning objects acquisition process. An agent makes a request to acquire the object (step 1). The Learning Objects agent initiates the acquisition of the object through the Connectivity agent (Step 2). The Connectivity agent creates a P2P

decentralized *bitTorrent* network to download the object (step 3). The *bitTorrent* network returns the information to the Learning Objects agent (step 4). Finally, the information is persisted on the device through the Persistence component (step 5).

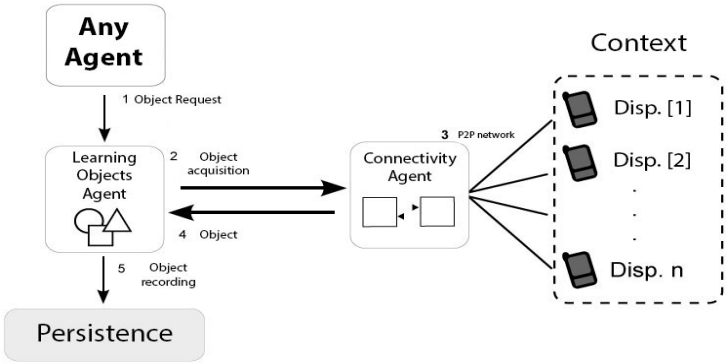


Figure 2: Sharing learning objects

3.6 Communication agent

The Communication agent manages the receiving and sending of messages. The agent can be used either by the user through the interface, as by other Global agents. Messages are text-based notifications that are stored in a *mailbox* for later reference. The Communication agent supports both the sending of messages to a specific user and the sending of messages to a specific context, in which all the users who are in the context will receive the notification. If the message recipient is not available, the agent stores the message until the recipient becomes available again.

3.7 Components

Global has four components that complement the agents' functionalities. The Persistence component is in charge of maintaining the persistency of information. It provides an API for recording and retrieving information, abstracting the I/O operations performed by the application. The API is similar to an object-oriented database. The component also provides a structure for exporting and importing data, allowing backup or migration, in case the user wants to switch devices, so as to maintain the settings and information from the previous device.

The User Interface component is in charge of providing the means for the user to access system functionalities. This component is modular and configurable, thus facilitating the customization of the system to a range of devices, both in the display size range and in the availability of interaction, including buttons, keyboard, touchscreen and multitouch.

The Proxies component represents a possible means of communication between devices. For example, a proxy can be implemented to exchange messages using a bluetooth connection and another can be implemented to exchange messages through a web server, accessed by a Wi-Fi or 3G network. Each proxy is in charge of identifying new connections, lost connections, keeping the list of available users and managing both the sending and the receiving of messages.

Global uses *restrictions* and *restrictors* to manage the contexts. The Restrictors component monitors the environment and identifies the status changes in restrictions, notifying the listeners. The restrictors are associated to the initialization of the Context agent, and can be adapted dynamically, depending on the device's resources. For example, *GlobalPosition* type restrictions may be associated with a location restrictor using GPS or by antenna triangulation.

Fig. 3 illustrates the restrictions, where a context has a *time-based* restriction for the days of the week, a *logical operation* restriction "or" used to create a restriction formed by other two *time-based* restrictions for the times of operation, one *physical position* restriction that represents the area of operations for the context and a *user list* restriction that represents the users who can participate of context.

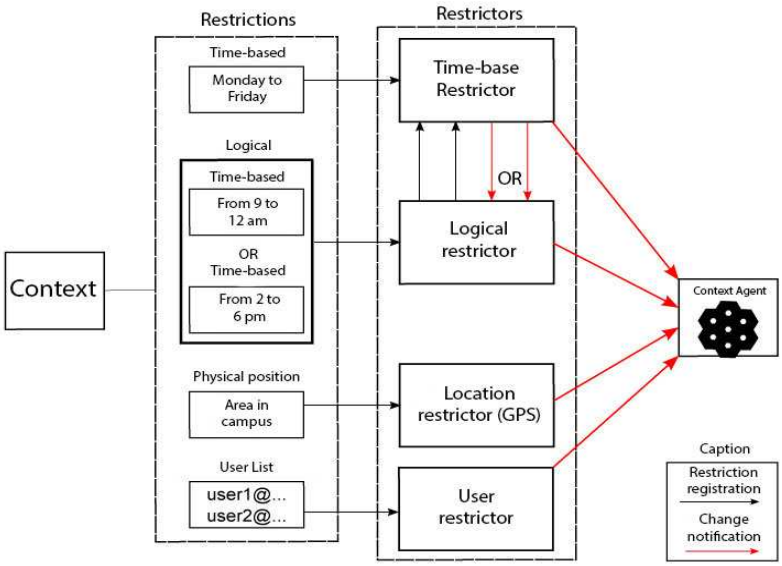


Figure 3: Example of restrictions and restrictors

4 Implementation Aspects

The prototype was developed in J2ME (*Java 2 - Micro Edition)*, MIDP 2.0 (*The Mobile Information Device Profile)* and CLDC 1.1 (*Connected Limited Device Configuration)*, and can be used in any smartphone that supports this setting. The Connectivity and Context agents were implemented following the model specifications and the other agents have implemented subsets of their specifications. Third-party libraries were used to simplify the development of the support APIs. The Persistence component was developed with the aid of the Floggy library, which is a persistence framework for mobile devices. The User Interface was implemented using LWUIT (*Lightweight UI Toolkit)*, a framework that provides visual components for J2ME applications. Furthermore, it provides an L10n standard API, which facilitates support for multiple languages.

Additionally, two *proxies* were implemented, one using bluetooth technology and another based on webservices for Wi-Fi/3G. In addition, four Restrictors were developed: (1) *Time-based*: it maps time intervals as hours, days and weeks; (2) *GPS*: it allows the creation of restrictions by physical position, using the GPS API for J2ME; (3) *Logical*: it allows the use of logic-based operations, such as "and", "or" and denial, between restrictions; (4) *QR Code*: it is used for indoor physical location, where GPS has limited accuracy or for devices without GPS.

5 Evaluation Aspects

According Edwards et al., the evaluation of an infrastructure is problematic since it is not visible to the end user [Edwards et al., 03]. The authors also emphasize that it is only possible to assess the functionalities of an infrastructure by building applications that use them and then evaluating these, thus obtaining an indirect assessment of the infrastructure. Considering this evaluation strategy, two applications were created from the Global prototype. The following items describe the applications and the qualitative aspects assessed:

- *Ubiquitous learning environment* (section 5.1): this application was implemented with the same functionalities of an existing centralized environment called LocalEdu [Hahn and Barbosa, 08]. The implementation enabled to assess the feasibility of Global for developing ubiquitous learning environments, even when they need centralized functionalities;
- *Collaborative learning environment* (section 5.2): Global was integrated with a multi-agent collaborative system, called CoolEdu [Rabello et al., 12]. Thus, the capacity of Global for integrating new functionalities by adding new agents was assessed. In addition, its capacity for developing a fully decentralized learning environment was also tested.

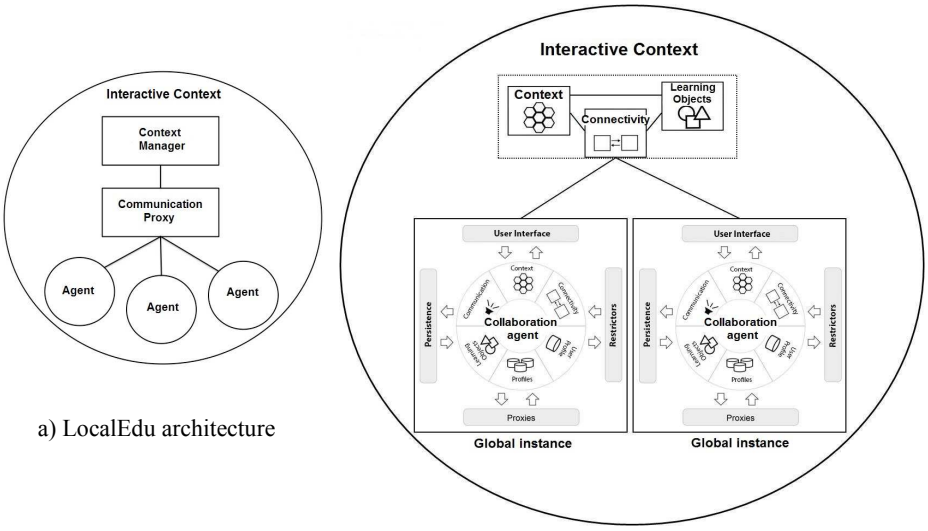
5.1 Ubiquitous Learning Environment

Hahn and Barbosa proposed a ubiquitous learning environment called LocalEdu [Hahn and Barbosa, 08]. The environment supports the organization of ubiquitous learning spaces in the form of interactive contexts, which are abstract representations of physical contexts. These contexts are defined by the presence of three components (Fig. 4a): (1) *Assisting agents*, which accompany and represent learners within the system; (2) *Context manager*, which manages the interactive context; (3) *Communication proxy*, which enables communication between the *agents*, as well as between the *agents* and the *Context manager*.

The *Assisting agents* move through the interactive contexts, aided by a tracking system. During this movement, learners interact and receive recommendations for customized learning objects. The learners' profiles are stored in the agents and follow the PAPI standard. Communication between agents is based on the FIPA-ACL specification. The interaction between agents and context takes place in two manners: (1) when a similarity between learners' profiles is detected, agents are used to foster collaboration among learners [Barbosa et al., 11]; (2) when learning objects relevant to learners are detected, learners are informed about them (Content Management [Barbosa et al., 13]).

As shown in Fig. 4a, two functionalities of LocalEdu are performed in a centralized manner. First, it has a *Communication proxy* to exchange information between the learners' devices. Second, the context and content management are centralized through an associated object repository for each *Context manager*. On the other hand, Global is completely decentralized. Specially, it does not have a repository for learning objects. The content management within a context takes place in a distributed manner among the participants of the context. Therefore, a special instance of Global was created, formed by specializations of the Context, Learning Objects and Connectivity agents. This new instance behaves as a centralized repository for objects. The implementation of this centralized functionality shows the flexibility provided by the Global. Regarding communication between devices, it was kept the decentralized communication via Proxies and Connectivity agents.

Moreover, it was necessary to create an agent to indicate similarities among learners' profiles and for referral of learning objects. This agent was named Collaboration. Fig. 4b shows the ubiquitous learning environment created using Global. The LocalEdu agents are formed by an instance of Global with the addition of a Collaboration agent, which is responsible for stimulating collaboration.



a) LocalEdu architecture

b) LocalEdu implemented with Global

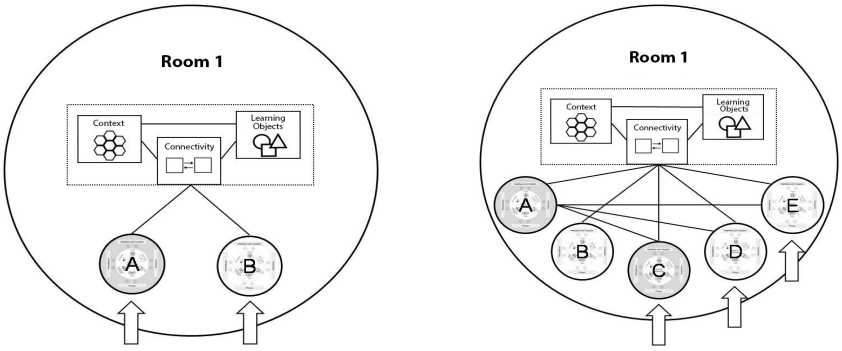
Figure 4: LocalEdu and its implementation using Global

The same scenario applied in the LocalEdu evaluation [Hahn and Barbosa, 08] was used to evaluate the implementation using Global. The scenario involves a simulated library where five learners search for material to solve an assignment related to the *Data Visualization* discipline. The scenario consists of two rooms, mapped for two contexts. The students carry mobile devices that are mapped to *Assisting agents*. The books available in each library context are mapped as learning objects. The Global simulation was implemented using the *Sun Java Wireless Toolkit* emulator suite. Seven instances of the emulator were initialized running Global, five

of which represented the students using *Assisting agents* (students A, B, C, D and E) and two represented the interactive contexts of a library (*Room 1* and *Room 2*). The movement of students was simulated through an XML script, representing the displacements. The position coordinates were obtained by mapping a real area, having been used in the simulation proposed by [Hahn and Barbosa, 08]. The simulation involved four moments.

On **Moment 1,** two students (A and B) arrive at *Room 1* in the library (Fig. 5a). Student A, in addition to the *Data Visualization* assignment, has an assignment that requires searching for course material in the *Logic* discipline. Based on the current interactive context, learning objects related to the discipline of *Data Visualization* are provided. On **Moment 2** students C, D and E, from the same class as A and B, arrive at *Room 1* (Fig. 5b). They enter the room and begin to search for material on *Data Visualization*. All learning objects offered by the interactive context with some relevance are recommended to the students. Additionally, the three newcomers receive indications informing that student A is interested in material on the *Logic* discipline. Student C is interested on the same material. On **Moment 3,** students C, D and E discuss the recommended materials. During the discussion, they come to an agreement about what materials will be used for the assignment. On **Moment 4,** student C looks for student A, and they discuss the *Logic* assignment. They decide to go to another room (*Room 2*), containing material related to the discipline (Fig. 6). While the new interactive context recommends learning objects, students A and C analyze the material and discuss the possibility of performing the task using the recommended material.

The Global proved to be suitable for the implementation of the LocalEdu's functionalities [Hahn and Barbosa, 08]. Tab. 2 shows the additional implementations needed. It became necessary to implement only specific characteristics of LocalEdu, such as the content management through contextualized learning objects and the algorithm to detect similarity between learners' profiles to foster collaboration.



a) Moment 1 b) Moment 2

Figure 5: Simulated scenario

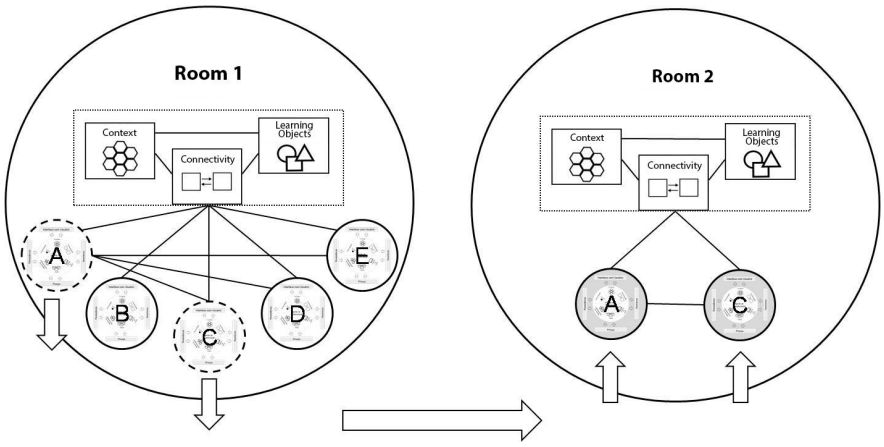


Figure 6: Moment 4

Resource	Required implementation	Provided by Global
Profile	Learner interests	PAPI Model
Context Management	-	Context Management
Context content	Information such as list of keywords and list of available learning objects in the context	*Global Information Tree* to treat content in the contexts
Communication between agents	-	Connectivity agent and proxies
Determining profile similarity	Algorithm to profiles comparison	Profile information through the Profiles agent
Management of Learning Objects	Adequacy of learning objects information	Management of learning objects
Referral of Learning Objects	Algorithm for referral of learning objects	Profile information and learning objects

Table 2: Specializations to implement a ubiquitous learning environment

5.2 Collaborative learning environment

CoolEdu [Rabello et al., 12] is a model to foster collaboration in decentralized ubiquitous learning environments. CoolEdu focuses exclusively on collaboration support, requiring that the environment to which it is integrated provides support for generic ubiquitous learning services. Like Global, CoolEdu consists of a multi-agent system, and its architecture is organized into three agents: (1) Social agent, responsible for stimulating collaboration; (2) Collector agent, responsible for managing the user profile; (3) Interface agent, responsible for the communication with the learner.

CoolEdu runs in decentralized ubiquitous learning environments, and it is up to

the environment to provide support for ubiquitous learning services. These services consist of features that are not supported by CoolEdu, but which are necessary for its operation. The services are organized into four groups (Fig. 7).

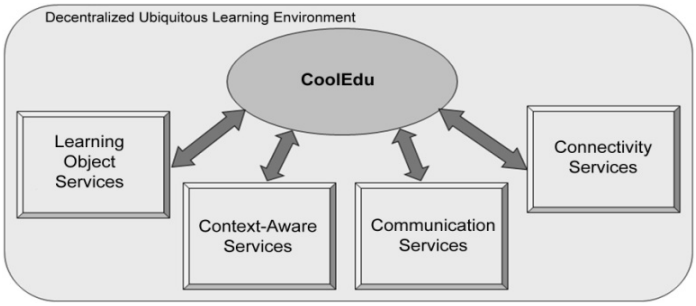


Figure 7: CoolEdu environmental requirements

Global implements the services requested by CoolEdu. Thus, the CoolEdu/Global integration created a collaborative and decentralized ubiquitous learning environment. Moreover, this integration allowed for the evaluation of the Global infrastructure's capacity to support the integration of new functionalities through the integration of new agents. The proposed integration involved two adjustments.

The **first adjustment** considered that CoolEdu defines the user as having two types of relationship with a context, namely, the user is in a context or the user is part of a context. When a user discovers a new context, the Social agent analyzes whether the context is interesting to the user and, if so, asks whether the user wants to be part of the new context. The adaptation of this behavior in Global involved the creation of a new Boolean restriction called *ActiveContext.* When a new context is discovered, the Social agent is notified and the users are asked whether they would like to join the context. In case they do, the *ActiveContext* restriction is marked as true, otherwise, it is marked as false. Thus, if *ActiveContext* is marked as true, the user is part of a context.

For the **second adjustment,** it was considered that both CoolEdu and Global support profile management. Global achieves this through the User Profile and Profiles agents, while CoolEdu achieves this through the Collector agent. Both use the PAPI model as a profile standard, but CoolEdu adds specific fields to analyze the collaboration. In order to avoid duplicate profile management, the Collector agent took on the role of both Global agents. Upon the initialization of Global, during the agent registration process, the Collector agent was registered three times, namely, as the Collector itself, as the User Profile agent and as the Profiles agent. Tab. 3 summarizes the adjustments needed to integrate Global with CoolEdu.

The integration was evaluated using a simulation based on the emulator of the *Sun Java Wireless Toolkit* suite. Six instances of the emulator were initialized running Global integrated with CoolEdu, each representing a learner (instances A, B, C, D, E and F). Profile information and learning objects being used by the learners were preloaded upon the startup of each instance. The movement of learners was simulated through an XML script, representing the displacements.

Resource	Required implementation	Provided by Global
Context Management	Treatment of the two types of relationship with a context (users are in contexts or they are part of contexts)	New Boolean restriction called *ActiveContext*
Profile Management	Adjustment of agents roles to avoid the duplication of profile management	Flexibility in the register of new agents to reconfigure the environment, in this case, *User Profile* and *Profiles* agents were replaced by the CoolEdu's *Collector* agent

Table 3: Adjustments to integrate Global with CoolEdu

The simulated scenario was motivated by the train that connects two cities localized in the south of Brazil, called Porto Alegre and Novo Hamburgo. This train passes near two universities and two faculties and transports daily thousands of students. The scenario involved the moments during the movement of students between their homes and classrooms. During this period, the train passes through four stations ("Mercado", "Rodoviaria", "Sao Pedro" and "Unisinos" stations). The Tab. 4 shows the nine steps of the simulation.

On **step 1** the students moved from their homes to the station. On **step 2**, the students enter in the train context through the station called *Mercado*. The Context-aware service (Global) of user A notifies the Social agent (CoolEdu), which requests from the Collector agent (CoolEdu) the profiles of users present in the context. The Collector agent returns the profiles of users B, C, D, E and F. The Social agent compares the profiles, finding a common interest among users A, C and D (in this case, an appreciation for the game of chess). The Social agent of learner A calculates the *interaction indexes* [Solon et al., 12] for users C and D, reaching the values 0.76 and 0.84, respectively. These indexes indicate the potential of interaction between the learners through a discrete value from 0 to 1. The calculation is based on: (a) the learners´ previous relationship; (b) the similar personality aspects shared by learners; (c) the knowledge domain involved in the interaction; (d) the number of previous contexts shared by the users.

Using the highest index as the criterion, the Social agent notifies the Interface agent (CoolEdu), which tries to send a suggestion of contact between users A and D. However, the Interface agent of user D denies the request to send the message because this user has configured the context *train* as unavailable to receive collaboration suggestions. The Social agent is notified of this denial by the Interface agent, and requests contact with the second user in the queue, namely user C.

On **step 3**, the Interface agent of user A sends a message to the Interface agent of user C, suggesting that learners A and C contact one another. The Social agent of user C analyzes the user's learner profile and accepts the suggestion. On **step 4** after exchanging messages with the student A, the learner C decides to create a context restricted by a public interest (chess) and by area (area near the train). Thus, the context supports a way to occupy the free time during the train journey, exploring activities related to chess. On **step 5**, all learners are notified of the creation of the

new context through their Context-aware services (Global). On **step 6**, the user D's Context service, which had initially denied the interaction, notifies the Social agent about the new context. Noticing that the context created is related to user D's interests, the Social agent requests the Interface agent to notify user D. The user decides to join the context created by user C. Additionally, user D has, among his learning objects, a tutorial on advanced chess techniques. Through the Context-aware services, the user makes the tutorial available in the context. On **step 7**, the Collector agents of users A and C notify them about the new learning object in the context. On **step 8**, the user C chooses not to acquire the new object. User A, in turn, accepts the file download corresponding to the new object and in the **step 9** initiates acquisition through the Learning Object service (Global).

The simulation involved the basic functionalities of the Global/CoolEdu integration. The evaluation showed that Global is able to interact with other multi-agent systems and even replace their agents with other implementations. Adaptations were required in contexts and profiles, but the rest of the integration did not involve significant changes. Moreover, the implementation showed the capacity of Global for developing fully decentralized learning environments.

#	Location	Actors	Description
1	Learners' homes	Learners A, B, C, D, E, and F	Learners go to the *Mercado* train station.
2	*Mercado* Station	CoolEdu A, C, and D	Learner A's CoolEdu locates learners C and D by their interest in chess. Users A and C are connected. Learner D is left out, because the privacy settings prevent the contact.
3	*Mercado* Station	Learners A and C	Learners A and C receive a suggestion to talk about chess and start a chat conversation.
4	*Rodoviária* Station	Learner C	Learner C creates a context with educational (chess) and environment (train) information.
5	*Rodoviária* Station	CoolEdu A, B, C, D, E, and F	All CoolEdus are notified of the creation of a new context.
6	*São Pedro* Station	Learner D	Learner D receives notification and accepts to participate in the context, making a Learning Object available to all members of the context.
7	*São Pedro* Station	CoolEdu A, C, and D	Learners A and C are notified of the availability of a new LO by their CoolEdus.
8	*Unisinos* Station	Learners A and C	Learner A requests the download of the new object, while learner C rejects it.
9	*Unisinos* Station	CoolEdu C and D	CoolEdu C begins the download of the object from learner's D device.

Table 4: Steps of simulation to evaluate the integration CoolEdu/Global

6 Conclusion

This paper introduced Global, an infrastructure for ubiquitous learning environments. Seeking a better match to the concepts of ubiquitous computing established by [Satyanarayanan, 01], Global proposes a decentralized architecture based on *Ad Hoc networks.*

When compared with the related works described in section 3, Global supports all the characteristics of ubiquitous computing listed in Tab. 1. Moreover, Global is a fully extensible infrastructure. These features implicate that Global can be used to develop flexible and decentralized ubiquitous learning environments as demonstrated in the section 5. The following items summarize the ways in which Global ensures these characteristics:

- *Location sensitivity:* it is ensured by the *GlobalPosition* restriction, which represents the user's physical position;
- *Communication between devices:* it is ensured by the Proxies component, which is a generic architecture for communication between devices. Global supports Bluetooth and Wi-Fi by default;
- *Mobile networks:* through a routing algorithm present in the Bluetooth proxy, messages can be routed between devices, thus creating mobile networks;
- *Distributed security:* it is ensured by the exchange of encrypted messages, and by the structure of public and private keys supported by the Connectivity agent;
- *Context awareness:* the Context agent, combined with the restrictors, is able to map characteristics of the environment, thus creating contexts. By monitoring of the changes to these characteristics, Global is able to change their behavior, thus ensuring context awareness;
- *Localized scalability:* Bluetooth technology operates in a localized manner. The limitation in the number of hops in the *Ad Hoc* network ensures greater localized scalability;
- *Mobile access to information:* all information is stored on the device itself and is accessible at any time, thus ensuring mobile access to information;
- *Extensible:* Global uses a multi-agent architecture. An agent does not need knowledge about the structure of another to interact, because the knowledge of the communication protocol is sufficient. Thus, in multi-agent systems, agents can be added, removed or replaced without affecting the system as a whole. This feature supports modularity, thus facilitating the addition, removal and modification of system functionalities.

During the development of Global, the following proposals for future works were identified: (1) despite the fact that communication between agents uses FIPA-ACL, the content does not follow a pattern, and the use of a content standard for communication between agents, in the case FIPA RDF, is intended; (2) adding new restrictions and restrictors, such as RFID readers; (3) adding new communication *proxies;* (4) extending the prototype implementation, focusing on agents that have not been fully implemented; (5) creating a user interaction agent using advanced techniques, such as emotive agents. Additionally, the work could be enriched with an evaluation involving users. In this sense, a possible strategy would be to execute the scenario described in the section 5.2, replacing the simulated learners by users. After the execution, the subjects would answer an assessment questionnaire based on the

Technology Acceptance Model (TAM) proposed by [Davis, 89] and applied and expanded by [Yoon and Kim, 07] in a study on the acceptance of wireless networks. In the TAM model, user satisfaction is measured through perceived usefulness and perceived ease of use. The TAM has been considered a standard for the evaluation of the acceptance of new technologies [Marangunic and Granic, 14].

Acknowledgements

This work was financed by Capes/Brazil (http://www.capes.gov.br) and CNPq/Brazil (http://www.cnpq.br). We would like to acknowledge the support of both institutions. Finally, we would like to thank Unisinos (http://www.unisinos.br) and Feevale (http://www.feevale.br) for embracing this research.

References

[Barbosa et al., 11] Barbosa, J. L. V., Hahn, R., Barbosa, D. N. F., Saccol, A.: A Ubiquitous Learning Model Focused on Learner Integration; International Journal of Learning Technology, 6, 1, 62-83, 2011. Doi=10.1504/IJLT.2011.040150.

[Barbosa et al., 12] Barbosa, J. L. V., Barbosa, D. N. F., Wagner, A.: Learning in Ubiquitous Computing Environments; International Journal of Information and Communication Technology Education, 8, 3, 1-14, 2012. Doi=10.4018/jicte.2012070108.

[Barbosa et al., 13] Barbosa, D. N. F., Barbosa, J. L. V., Bassani, P. B. S., Rosa, J., Martins, M., Nino, C.: Content management in a ubiquitous learning environment; International Journal of Computer Applications in Technology (IJCAT), 46, 1, 24-35, 2013. Doi=10.1504/IJCAT.2013.051385.

[Caceres and Friday, 12] Caceres, R, Friday, A.: Ubicomp systems at 20: Progress, opportunities, and challenges; IEEE Pervasive Computing, 11, 1, 14-21, 2012. Doi=10.1109/MPRV.2011.85.

[Chen and Li, 10] Chen, C. M., Li, Y. L.: Personalized Context-Aware Ubiquitous Learning System for Supporting Effectively English Vocabulary Learning; Interactive Learning Environments, 18, 4, 341-364, 2010. Doi=10.1080/10494820802602329.

[Davis, 89] Davis, F.: Perceived Usefulness, Perceived Ease Of Use, and User Acceptance Of Information Technology; MIS Quarterly, 13, 3, 319-340, 1989. Doi=10.2307/249008.

[Dey, 01] Dey, A. K.: Understanding and Using Context; Personal and Ubiquitous Computing, 5, 1, 4-7, 2001. Doi=10.1007/s007790170019.

[Dey et al., 10] Dey, A., Hightower, J., Lara, E., Davies, N.: Location-based services; IEEE Pervasive Computing, 9, 1, 11-12, 2010. Doi=10.1109/MPRV.2010.10.

[Edwards et al., 03] Edwards, W. K., Bellotti, V., Dey, A. K., Newman, M. W.: The challenges of user-centered design and evaluation for infrastructure; In Proc. Conference on Human Factors in Computing Systems, 2003, 297-304.

[Hahn and Barbosa, 08] Hahn, R., Barbosa, J. L. V.: An Architecture of Personnel Assistant to Ubiquitous Learning Environments; In Proc. Brazilian Symposium on Educational Informatics, 2008, 736-745.

[Hightower and Borriello, 01] Hightower, J., Borriello, G.: Location Systems for Ubiquitous Computing; IEEE Journal, 34, 8, 57-66, 2001. Doi=10.1109/2.940014.

[IED, 14] Immersive Education Initiative, 2014, http://immersiveeducation.org.

[Jin, 03] Jin, Q.: Learning Community Meets Peer-to-Peer Networking: Towards an Innovative Way of Person-to-Person E-Learning Support; In Proc. Int. Conf. on Knowledge Economy and Development of Science and Technology, 2003, 1-8.

[Knappmeyer et al., 13] Knappmeyer, M., Kiani, S.L., Reetz, E.S., Baker, N., Tonjes, R.: Survey of Context Provisioning Middleware; IEEE Communications Surveys & Tutorials, 15, 3, 1492-1519, 2013. Doi=10.1109/SURV.2013.010413.00207.

[Li et al, 05] Li, L., Zheng, Y., Ogata, H., Yano, Y.: Ubiquitous Computing in Learning: Toward a Conceptual Framework of Ubiquitous Learning Environment; International Journal of Pervasive Computing and Communication, 1, 3, 207-215, 2005.

[Liu et al, 09] Liu, T. Y., Tan, T. H., Chu, Y. L.: Outdoor Natural Science Learning with an RFID-Supported Immersive Ubiquitous Learning Environment; Educational Technology & Society, 12, 4, 161-175, 2009.

[Marangunic and Granic, 14] Marangunić, N., Granić, A.: Technology acceptance model: a literature review from 1986 to 2013; Universal Access in the Information Society, Published online, 1-15, 2014. Doi=10.1007/s10209-014-0348-1.

[Rabello et al., 12] Rabello, S. A., Oliveira, Jezer M., Wagner, A., Barbosa, J. L. V., Barbosa, D. N. F.: CoolEdu: a Collaborative Multiagent Model for Decentralized Ubiquitous Education Environments; IEEE Latin America Transactions, 10, 6, 2273-2279, 2012. Doi= 10.1109/TLA.2012.6418132.

[Satyanarayanan, 01] Satyanarayanan, M.: Pervasive computing: vision and challenges; IEEE Personal Communications, 8, 4, 10-17, 2001. Doi=10.1109/98.943998.

[Serio et al., 13] Serio, Á. D., Ibáñez, M. B., Kloos, C. D.: Impact of an augmented reality on students' motivation for a visual art course; Computers & Education, 68, 586-596, 2013. Doi= 10.1016/j.compedu.2012.03.002.

[Tavares et al., 12] Tavares, J. E. R., Barbosa, J. L. V., Costa, C. A., Yamin, A. C., Real, R. A.: Hefestos: a Model for Ubiquitous Accessibility Support; In Proc. Int. Conf. on Pervasive Technologies Related to Assistive Environments, 2012, 1-8. Doi=10.1145/2413097.2413131.

[Vianna and Barbosa, 14] Vianna, H., Barbosa, J. L. V.: A Model for Ubiquitous Care of Non-communicable Diseases; IEEE Journal of Biomedical and Health Informatics, 18, 5, 1597-1606, 2014. Doi=10.1109/JBHI.2013.2292860.

[Wagner et al., 14] Wagner, A., Barbosa, J.L.V., Barbosa, D.N.F.: A model for profile management applied to ubiquitous learning environments; Expert Systems with Applications, 41, 4, 2023-2034, 2014. Doi=10.1016/j.eswa.2013.08.098.

[Weiser, 91] Weiser, M.: The computer for the 21st century; Scientific American, 265, 3, 94-104, 1991.

[Yin et al., 05] Yin, C., Ogata, H., Yano, Y, Oishi, Y.: JAPELAS: Supporting Japanese Polite Expressions Learning Using PDA Towards Ubiquitous Learning; The Journal of Information and Systems in Education, 3, 1, 33-39, 2005.

[Yin et al., 10] Yin, C., Ogata, H., Tabata, Y., Yano, Y.: Supporting the acquisition of Japanese polite expressions in context-aware ubiquitous learning; International Journal of Mobile Learning and Organization, 4, 2, 214-234, 2010. Doi=10.1504/IJMLO.2010.032637.

[Yang, 06] Yang, S. J.: Context Aware Ubiquitous Learning Environments for Peer-to-Peer Collaborative Learning; Educational Technology & Society, 9, 1, 188-201, 2006.

[Yoon and Kim, 07] Yoon, C., Kim, S.: Convenience and TAM in a ubiquitous computing environment: The case of wireless LAN; Electronic Commerce: Research and Applications, 6, 1, 102-112, 2007. Doi=10.1016/j.elerap.2006.06.009.

[Zhang et al., 05] Zhang, G., Jin, Q., Lin, M.: A Framework of Social Interaction Support for Ubiquitous Learning; In Proc. Int. Conf. on Advanced Information Networking and Applications, 2005, 639-643.gv