Educational Technology & Society 5(4) 2002
ISSN 1436-4522

A Framework for Intelligent Virtual Training Environment: The Steps from Specification to Design

Christos Bouras [1,2], Vassilis Triantafillou [1,3], and Thrasyvoulos Tsiatsos [1,2]
[1]Computer Technology Institute
[2]Computer Engineering and Informatics Dept., Univ. of Patras, Greece
[3]Environment and Natural Resources Management Dept., Univ. of Ioannina, Greece
bouras@cti.gr
triantaf@cti.gr
tsiatsos@cti.gr

ABSTRACT

The growing need for communication, visualization and organization technologies in the field of e-learning environments has led to the application of virtual reality and the use of collaborative virtual environments. This paper presents a system architecture to support such environments, defined by user needs and using state of the art technologies.

Keywords
e-learning, 3D virtual environments, avatars, intelligent agents, translation

1. Introduction

A lot of research has been developed in the last few years concerning e-learning systems leading to the set-up of many research projects the last few years (Singhal, Zyda, 1999). Although research projects try to fit in technological advances, and to produce up to date technological e-learning systems, commercial products already available, focus more on the already available technological solutions to provide efficient e-learning solutions. The main concern is to ensure that end users are satisfied when using e-learning systems in order to learn together. On the other hand a important question to be answered is: How we can tie the up-to-date technologies and standards to learning theories in order to produce efficient and marketable systems for e-learning? In other words, we have to achieve the right balance among user requirements, learning methods and available technologies (Bouras et. al., 2001a) with a view to implement a system that is user centered, usable and effective. In order to deal with this problem we need to define and follow a well defined system development process. Few tools can be found that support collaborative e-learning by applying a wide range of up-to-date technologies therefore there is a strong need to formulate a development process to facilitate the analysis and design phases of the development of such a system. This framework describes in detail the methods, tools and results of each step of the development process from user requirements towards the realization of the system architecture taking under consideration the application of mature standards and technologies. This process has been adopted in the development of INVITE (Intelligent Distributed Virtual Training Environment), a project funded by the Information Society Technology (IST) Program of the European Commission.

2. The INVITE approach

Most of the influential organisational theories referring to the characterisation of the new emerging “knowledge based learning organisation”, emphasise that while training characterised by hierarchical structure of the declining organisational modes, collaborative learning is the dominant mode of organisational learning necessitated by the new more horizontal much more flexible and dynamic structure of these new organisations. This mode is to large extent being enhanced by the immanent nature of ICT basic to the new emerging organisational structures. Therefore an innovative Internet based learning environment for distance organisational learning, should focus on open, collaborative processes and not on training. Training, although diminishing, still has very height importance in most organisations. Therefore the optimal solution for such a platform would be to allow both kinds of learning and the many variations in between but to encourage those modes considered to be essential for the leaning organisation in the global economy.

“Collaborative learning” is any kind of group learning in which there are at least some meaningful learning interactions between the learners (“horizontal interactions”).

“*Collaborative e-learning*” is any kind of group learning that takes place mainly in a virtual environment. Web based collaborative learning has significant advantages not only in comparison with face-to-face classical approach, but also an individual web-based learning approach. These advantages include among others a dramtic increase in student participation rate, enhanced student satisfaction from the educational process combined with higher motivation, better combination of self-reflection and interaction among students (Lehtinen, 1999). Collaborative e-learning plays also a role in developing a sense of belonging to a community individual learners-one that is missing in individual on-line learning methods (Clark, 2000). Collaborative e-learning improves interaction among students, increasing individual participation and involvement, as well as being instrumental in creating group spirit (Nachmias et. al., 2000) achieving a “significant difference” in learning efficiency (Gilbert & Han, 1999).

Current synchronous e-learning tools are mostly based on video-conferences, and seem to have been designed to virtually represent the concept of frontal learning. A general problem of these tools is the reduced social presence of the participants that are represented in windows by means of live pictures which mostly are not fluid, occur with time delays, have a low resolution and are quite small. Thus, participants are rather given a feeling of distance than a feeling of nearness and group awareness. Current e-learning scenarios are mostly resource based, communication is seen in many cases as an add-on feature and often supported only asynchronously. Available synchronous tools focus on tutor-centered scenarios, such as presentations and lectures. These limitations have caused a severe gap in the realisation of e-learning scenarios: many companies and educational institutions view e-learning rather critically and currently see only limited deployment potential in their environment. In addition to the support of synchronous learning sessions, tools for preparing sessions and reflecting them must be integrated. Furthermore, the integration of tools for user and content manipulation as well as interfaces for integrating standardised learning content will make the INVITE a completed system to support e-learning services. Besides audio, written chat or using a system for application sharing, they can additionally communicate by gestures, mimics and their positioning within this virtual space. Learners act intuitively and do not spend any effort in thinking how to operate their learning environment. For supporting various learning scenarios, tools for monitoring learner actions and recording sessions, tools for structuring learners into groups and control their communication channels, display learning material as well as tools for evaluating and reflecting the learning progress in the group will be developed and implemented. Automatic translation will enhance the quality of international co-operation. Low social presence of participants during computer-aided communication is a grave weak point in current learning platforms. Missing conversational stimuli and few possibilities to realistically transmit the activities of the communication partners lead to passive behaviour of participants, the so-called lurking phenomenon. Possibilities for interaction and communication are too restricted, and presence and activities of participants cannot be experienced sufficiently. The use of virtual 3D rooms and avatars for moving around in these rooms offers a new approach. Three aspects are briefly described in the following illustrating how interaction can be performed and experienced in a more realistic way in order to provide the ground for effective synchronous learning settings (Schwan et.al, 1996).

3. System Development Process

For the development of a system for e-education such, as INVITE, it is essential to engage persons with different expertise such as end-users, pedagogues and software developers in different fields (Virtual Reality-VR, databases, computer networks, intelligent agents, on-line text translation, etc.). In order to minimize the development cost and manpower while engaging actively end users into the development process we introduced a user centered design philosophy, which led us to use an iterative and incremental development process along with rapid prototyping techniques. According to (Goransson 2001), an iterative and incremental development process specifies continuous iterations of design solutions together with users. An iteration includes:

- a proper analysis of the user requirements and the context of use,
- a prototype design phase and,
- a documented evaluation of the usability of the prototype.

The evaluation will result in suggestions for modifications in the following prototype design. INVITE aims to improve and upgrade the current educational and training methods in such a way that they will be friendlier to the users, more imminent and more interesting and adaptable to their needs. It combines some of the most recent technological developments in the area of network based educational and training platforms. Both these arguments where needed to be backed up by end users.

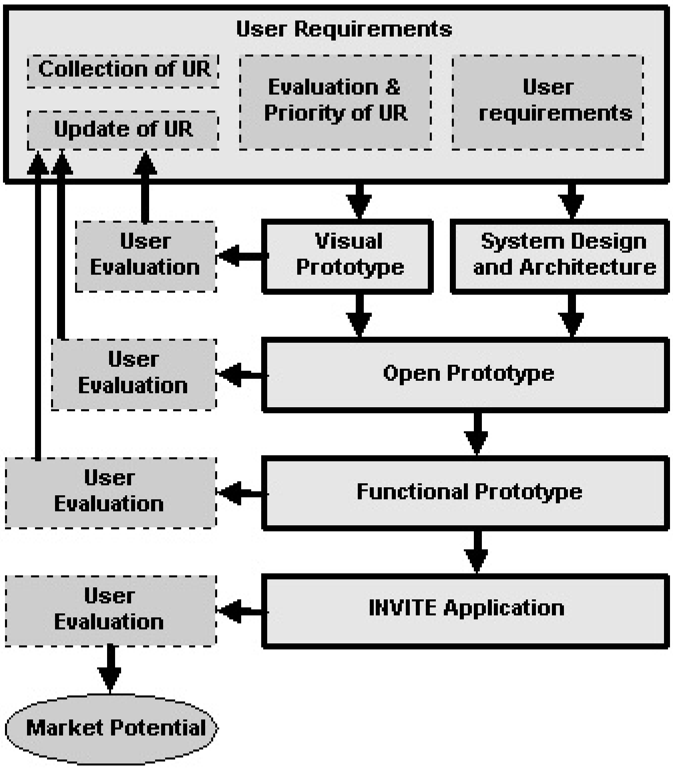

Figure 1. System Development Process

In the INVITE case, the system will be developed within three iterations of the software development cycle (Figure 1):

1. **Visual Prototype:** The first iteration is the prototype design, (rapid prototyping) which results in the "Visual Prototype", in order to ensure early user feedback about the application. The main reason for choosing rapid prototyping techniques is that our system aims to exploit new technologies (such as distributed virtual environments and intelligent agents) and that the end users have to be persuaded about the significance and the added value of the usage of advanced technologies. The visual prototype gives the end-users the opportunity to understand this and to approve or to discard the use of new technologies. In the cycle, usability and end-user tests are planned in order to gain concrete feedback of the end-users concerning the concept of the system, and acceptance scores. The output of the first evaluation cycle e.g. is a detailed and specific list of recommendations for the interface development and a ranked list of requirements for functionalities.

2. The output of the second iteration will be developed within two phases. These phases will have as results two intermediate versions of the system:
   - **Open Prototype:** This version of the system is the first intermediate system, which integrates basic functionality that can be used in order to support collaborative e-learning services. This functionality could be text chat, application sharing, users' representation by photo-realistic avatars and simple user-system interaction through intelligent agents that provide help for the system usage. The creation of the open prototype is necessary, because it will act as the basis for the end-user evaluation, which is focused on "how the users can learn using a 3D virtual environment". The results of this evaluation will be used in order to produce the second intermediate system (Functional Prototype), in which specific e-learning scenarios will be realized.
   - **Functional Prototype**: This version of the system is the second intermediate system. As described above, this system will provide full INVITE functionality in order to realize collaborative and tutor-centered e-learning scenarios. This prototype will be the basis for end-user evaluation, which is focused on the real usage of the system, according to specific e-learning scenarios.
3. **INVITE application:** This will be the final version of the system. Actually, it integrates the necessary changes according to end-user evaluation. This version of the system will be evaluated by end-users and experts in order to prove the ma rket potentiality of the INVITE system.

Usability engineering starts with the assessment of user-requirements and from that the deviation of system specification. This step guides the technical development and is not at all technically driven. The usability evaluation follows a formative approach by assessing problems and end-user-requirements with the existing system. However, the results of the evaluation indicate directions for further improvements of the design and the technical development. In each phase new functionality will be added to the system according to the users' demand. Each phase of the previously described system development process is very critical for the successful production of a user-centered system. Due to the limitation of space, in this paper we describe in more detail both the steps and the methodology, which we have followed in order to define the functional specifications and the system architecture.

3.1. Steps for Defining the System Specifications

The most significant part in the development cycle of the system is the definition of the functional and technical specifications. In order to achieve a user centered system that will be effective and realizable we need to take into account the users needs as well as to integrate possible technologies in a uniform way. For this reason we follow the steps described below.

Step 1: Description of User Scenarios. Based on the user processes defined and the requirements of the end users for setting up a learning environment we have defined two kinds of e-learning scenarios: (a) collaborative learning scenarios based on collaborative learning techniques (e.g think-pair share and jigsaw), and (b) tutor-centered scenarios. Along with the usage scenarios use-cases have been created in order to have a clear view of the user and system interaction and the deployment of learning scenarios in the INVITE platform.

Step 2: Functional Specification: In order to translate the user requirements into functional specifications an intermediate step was needed. This step is the mapping of user requirements to system features. The system features are far enough from the technical solution but concrete enough to get the right functionality. This step is very useful for: (a) the early development of the visual prototype, (b) to identify as soon as possible the basic system components and to prevent development risks, and (c) to guarantee the implementation of the user requirements. The feature list is influenced by the user scenarios (a list of features, necessary to perform the tasks described in the scenarios) and discussions with potential users. The feature list describes the features of the system in a general way, and from the task-oriented viewpoint of the user. In this step we map the features of the system into functions. The functions are also described from the user perspective - but not task oriented. The functions are more individual functions and might be used for several tasks, together with other functions, in several contexts. As soon as there is a function mentioned in the function list, the function needs to be specified in detail with its technical background interface.

Step 3: Technology and Standardization Monitoring. Based on the functionality and the training scenarios we present the technologies and standards that can be used as a reference point for the implementation of the INVITE system components. These results will be the basic input for the definition of the system architecture and will help in defining the final technical solution.

Step 4: System Architecture. The interaction of all functions leads to a description of a system architecture. For the description of the system architecture we applied the UML (Unified Modeling Language) (Booch et. al. 1999) notation.

Step 5: Implementation. This step concerns he development process of each prototype system. In this step results of the preliminary technology evaluation are taken into account.

Step 6: Evaluation. The developed application will be evaluated subsequently during the project. The feedback from the evaluation will lead to new requirements and necessary changes of the application. These changes are reported back in order to implement possible add-on the architecture and the relative development process.

4. Definition and Presentation of User requirements

Gathering user-requirements for a platform, built on innovative presentation means, cannot be based on mere user questioning. Especially for 3D technology, only few of the potential users of INVITE are aware of this technology, even less had contact with it and then in a different context, mostly in games or chats. On the other hand, the potential of the 3D technology has been identified, but it has also been noted that this potential is not implemented in current platforms for communication and learning. For this reason, INVITE has taken the approach to base the user requirements on a broader fundament of research, including recommendations stemming from an analysis of current learning theories as well as the investigation of current learning models in companies and analyzing deficits of current platforms. 3D is not an a priori requested technology, however the worst disadvantage seen by users is the lack of "personal contact". In other words the end-users require more awareness of other people in the e-learning virtual environment (Bouras, Tsiatsos 2002). As has been shown, it is 3D technology, which has high potential to make progress in the area of virtual presence. The use of 3D technology in INVITE has always to be put into this light: increase group awareness and the feeling that learning in INVITE is a "personal contact". The general recommendations for the system, which stem from pedagogical considerations, can be summarized as follows: (a) Enhancing Collaborative e-learning as an educational tool; (b) Empowering users as employees; (c) Empowering users as individuals.

These general recommendations are complemented with rather explicit recommendations for the realization of the system resulted from the user requirements analysis. These explicit recommendations are divided in technological (concerning the user interface and the system), structural (concerning the organization of users and e-learning courses) and procedural (concerning the interaction among the users, as well as the user-system interaction).

4.1. Technological recommendations

INVITE will concentrate on synchronous learning, but will offer a tool, which can be interfaced and combined with available learning platforms. The approach is to build a platform for synchronous e-learning, which can be interfaced with standardized content management and/or instructional management systems. As resulted from the user requirements analysis an original user-friendly user interface should be developed, which provides a high degree of virtual presence. Furthermore, the system should provide a web-based environment, which supports the known Internet communication and collaboration tools (emails, chats, forums, interactive blackboard, application sharing, etc.) either in the platform itself or by interfaces with standard products. A wide use of multimedia channels that will enable users to convey their thoughts and even physical gestures to their peers as vividly as possible and in real time should be used. Real-time audio transmission is a pre-requisite for the application. The platform should be as simple as possible access-wise, in order to allow students to take part in the discussion as quickly and in real-time as possible.

4.2. Structural Recommendations

INVITE should support different basic scenarios of organizational training used today in organizations of which the extremes are: (a) the small learning group made of 3-5 researcher or engineers, and (b) the "intense program" organized in large groups (40 or more). The platform should allow for the basic collaborative learning processes, enable several groupings (and sub-groupings) of students over a certain topic with changing participation. The environment should support an unlimited number of users/students. However, pedagogical research recommends groups that would not be larger than 15 members, preferably divided to sub-groups of 2-5 members. The platform should provide basic collaboration facilities, such as the ability to divide tasks between several group members on-line. The platform should allow tutors and users in general to prepare/configure the environment for learning sessions if they need to, choosing the features and values to be operated for a specific session, including integrating documents before a synchronous learning meeting starts. The platform should allow students and tutors to define their respective roles as tutor, facilitator, evaluator etc. in relation to a specific scenario and be able to change these roles during the learning process.

The user study showed that most workers as well as organizations prefer to arrange group learning most of the time. However, in order to support also those learners, which prefer to learn alone, INVITE, should support, within the same 3D environment (either synchronous or asynchronous) individual work within a group context (for instance, to post a presentation in the 3D room, to be later addressed by others).

4.3. Procedural Recommendations

INVITE platform should encourage discussion and / or group work, which in most cases is based on case studies, examples or written material. These material need to be searchable, collectable, it must be possible to discuss, integrate and elaborate them with other learners. INVITE should support also more close ended, high structured and high level of tutor control on the learning process and the communication channels. In most of the organizations the learning process is controlled to a large extent. Standard moderation and group learning techniques, such as brainstorming, card-moderation or meta-plan should be supported. It is suggested that the reflective nature of the INVITE platform would be able to supply some kind of non-judgmental evaluation of the learners' cognitive and social performance, as well as the group process during the collaborative learning assignments. Quantitative, as well as qualitative evaluation tools should be provided. Quantitative tools may include different kinds of traditional tests and submitted papers. Qualitative tools may include, for example, peer-evaluation general assessments. The platform should also contain tools, which allow monitoring and reflection of the learning process, as an individual as well as a group. INVITE should provide an easy way to move from one level of self-reflection to another – i.e., from self-reflection to group reflection, and vice versa. Or – allow a simultaneous reflection on both levels.

5. Definition and Presentation Functional Specifications

The system features where derived based on the user requirements, the user scenarios and the experience of the researchers. These system features give a first idea to the designers and developers of the system about the future functionality of the system. The system features have been grouped in the following categories: (a) User Interface Features; (b) User Communication and Collaboration features; (c) User representation features; (d) e-learning community features; (d) Content features; (e) Evaluation features; (f) Translation features. This feature has been used in order to built the Visual prototype. In order to go beyond, and to implement the Open Prototype, a detailed list of functions should be defined. These functions are described in the following paragraphs.

5.1. INVITE Community Model

To get a complete list of system functions, we have built a model of places, members and agents. This allows us to ensure for each place, each member, and each agent service the completeness of functionality. Furthermore the roles that can be assigned in each user of the system are defined and the rights on both the places and the rest of users.

5.1.1. Places

The real world metaphor leads us to structure the Digital Virtual Environment (DVE) into Places. This does not necessarily express a spatial visual (geometric) representation. The place as a metaphor rather describes an entity, where typical features are found, and accessed by the user. Using places, special themes in the e-Learning context are easily established and clearly structured. The hierarchy that is presented below is used not only for navigation reasons for the user but also for the users' manipulation. Users can own places, meet at places, use different (place specific) communication/collaboration tools, and add properties to places. Depending on the configuration of "rights and roles", places can either be "public" (unlimited access) or "private" (limited access) or "group places" (group access).

A Place is a 3D or 2D room, which can be private (only for personal use), public (accessible for all), accessible only by a group of users etc. The places are the main entities of the environment and the most of the members and agents actions are taking place there. The places offer specific functionality to the users according to their role in the system. Also places can be logical, which means that are groups of other places. These logical places are used in order to give the owner of an INVITE platform the chance to host many course offerings to a high number of different user groups. Examples are: Provider Organisations (PO) and User Organisations (UO). The hierarchy of places can be described as follows:

Level 1: **Entry Point**: This Entry Point is the first place of the community. There the users can registered to the system and to become members or they can enter in the system after a quick authentication procedure (if they are already members). The main functionality is login or register. Unregistered users can get only a small text about the offers.

Level 2: **Personal Desk**: Personal Desk is the central point of the community for each Member. From this place the users can access all the Courses in which they are members or to enrol in other Courses. Here the users can access their own belongings that consists out of messages, documents, calendar information and profile information. The users can get an actual information from course providers and see who else is online.

Level 3: It includes both the Provider Organizations and the User Organizations. **Provider Organizations** : A Provider Organisation (PO) is characterized by the ability to offer courses. It has a person that fills the role of maintaining the course structure called Course Manager and a number of persons that fill the role of Tutors. The decision to let users participate in courses is given to the role of the course manager. It is possible, that tutors are also course managers. The Platform Owner can enable members to become course manager of one organisation. The course manager can enable members to become tutors. **User Organizations:** In order to group members to their organization, the User Organisation (UO) is created. A UO is characterized by the ability to provide a person to fill the role of the User Manager. The user manager confirms member request to be part of the organisation. The user manager has the ability to enable a member to become a confirmed member. Those members become confirmed members instantly. The platform owner can enable a member to become a user manager for one organisation. Only confirmed members are able to register for a course.

Level 4: **Course Category:** Course Categories are groups of Courses or Course Sub-Categories with similar topics.

Level 5: It includes both the Courses and the Course Sub-Category. **Course Sub-Category:** Course Sub-Category is a place with the same structure as the Course Categories, but they have as an upper level another Course Category or Course Sub-Category. This structure gives the possibility to extent the tree of the Course Categories, Course Sub-Categories, and Courses. **Courses**: Courses are the most significant component of the community's hierarchy. They are containing the 3D learning spaces and additional functionality such as text chat, audio chat, etc.

5.1.2. Member

Member is a user registered to the system through the entry point. Each member is assigned to a set of 'attributes' that defines his specific 'role' in the learning environment (member, learner, tutor etc.). Depending on that role the member can own places, and properties, and has access rights to communication / collaboration tools. Each member has his/her Personal Desk. According to his/her profile, the member can see only a part of the INVITE courses, course categories and Provider organizations. This solution (distinguishing between provider and user organization) is more complex than necessary for the situation of user organizations and provider organizations. Focusing on a marketable INVITE platform, the possibility to host the INVITE-platform and to give several independent organizations the possibility to use the platform is essential and was mentioned in several presentations to prospects.

5.1.3. Roles and Rights

Any learning group as a real life group needs different roles to achieve their learning and working goal. Each role is assigned to individual tasks, and people are empowered to fulfil a task in their role. To have a role means to have special rights and duties. The roles of a user in the INVITE community are the following: Member, Learner, Junior Tutor, Tutor, User Manager, Course Manager, and Platform Owner.

Member: A member needs to provide e-mail where he can be contacted. A member can use the entrance chat to get help from a help agent, can enrol for a public course, receives the newsletter he can receive public messages, read the public calendar. To change to any different role, a user must first be a confirmed member and then the user can be promoted to another role. A member will find the functionality to apply for becoming a confirmed member in one of the member organisations in his personal profile section. A member can choose an avatar from a wide range of predefined avatars, upload a self-made avatar or use.

Learner (confirmed member): Learners have read and write access to a course, can find courses categories where s/he is already enrolled using the course finder on his personal desk. Learner is a confirmed member, whose identity is confirmed by a member manager of a user organisation. A learner can access the training session room (course), get the documents prepared for this session. A learner can also upload to the personal desk and distribute the uploaded document to the associated sessions. Some functions in a session are restricted for learners, like "make the microphone open to the group", "allow private chat" etc.

Junior Tutor: Junior Tutor (short name: jrTutor) is a learner who can have tutor rights temporarily (during the course). S/he is, therefore, temporary co-owner of a course (where s/he is enrolled) during the lesson. A jrTutor is a learner with a certain experience level, who can support a group in the learning process. S/he can do the same as a tutor while a tutor is in the same course. A jrTutor tutor helps learners and members to find their way in the community and can support groups of members. A jrTutor has the right to moderate the chat in a session room to help the tutor, or changes permanently to that place, if the courses owner allows it.

Tutor: Tutor is the owner of the course and sub-categories of courses. A tutor can organise and structure the session, drop learners from a session, mute them, give the floor to others etc. (switch on and of modules of a session room). A member or jrTutor becomes a permanent tutor, until the tutor cancels the status. Tutors might be grouped according to different learning subjects using the "interests tree" if an organisation has many course categories for the subject and many tutors for each course category. A tutor can create new courses in the same sub-category for follow up sessions or parallel courses with other members.

User-Manager for an Organization: The user manager (short name: UO manager) promotes a member to confirmed member (learner).

Course-Manager for an Organization: The course manager (short name: course manager) promotes a member to a tutor. The courses manager can maintain the courses structure of a provider organisation. Owners of this role can open new session rooms, employ Tutors, Experts, assign members and learners to courses, as well as add general information about courses.

Platform owner: Platform owner (short name: SysAdmin) has all rights and can modify everything, gives the role of user manager and course manager to a member. SysAdmin has no access to the content of the user files or to the private part of user profiles.

5.1.4. Agents

To support the user best in the INVITE community, a number of services are necessary. Some of the services are dedicated to specific places or members, and some are not. We call those services 'agents'. From a technical point of view agents are artificial, pre-programmed DVE processes who react to specific events in the learning environment.

Some Agents have a visual representation and appear as an avatar. This can help the user to understand that a certain service is available in a course room, and to use most of the communication functionality even for communicating with the agent. In other scenarios agents' services are performed in the background (e.g. search agents, calendar reminder), and not visualised in real-life metaphors. An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors (Fabri, 1999).

5.2. Definition and Presentation of System Functions

In this section the functionality offered by the system is presented in a detailed way. The functions are ordered in personal desk place, course place and agent functions. The prototypes of INVITE will subsequently integrate this functionality. Figure 2 is an abstract (due to limitation of space) class diagram of the INVITE system funations. These functions are described in detail in the following paragraphs.

5.2.1. Personal Desk -Related Functions

Personal Desk is the central point of the community for each *Member*. From this place the users can access all the *Courses* in which they are members or to enrol in other *Courses*.

The following functionality is available in the Personal Desk Place:

Course and Member Finder: The users can search for courses, register for a Course, search for members, search for documents.

My Courses: MyCourses is the area of the Personal Desk where each user can see the specific courses, course categories, course sub-categories, and provider organizations in which is a learner, tutor, and or course manager. Actually MyCourses area is a tree of the courses in order to facilitate the user to find the desirable course that he wants to participate or the course category/sub-category that he wants to administer. If a user is course manager of a course category, a relative icon is appeared next to the name of the course category in the MyCourses list.If a user is tutor of a course, a relative icon is appeared next to the name of the course in the MyCourses list.

Messages: The users can communicate asynchronously using Instant messaging and Message board functionality:

Instant Messaging: Members can send messages to other members online without having to be in the same place. In general the following instant message actions are supported: Send an instant message to other users; Send short non-text notifications to other users; Check for unread instant messages; Retrieve a message; Delete a message; Display a list of all members who are currently online.

Message Board: INVITE provides two different message board types: (a) The group message board in the learning room belongs to the Course Place owner (normally to the tutor). (b) The World message board for all members belongs to the Provider Organization (PO). The Message Board Functionality can be divided in the following functions: Sort messages; List the messages in an hierarchical list; Create Create/reply to a message; Read a message; Delete a message; Configure a message board.

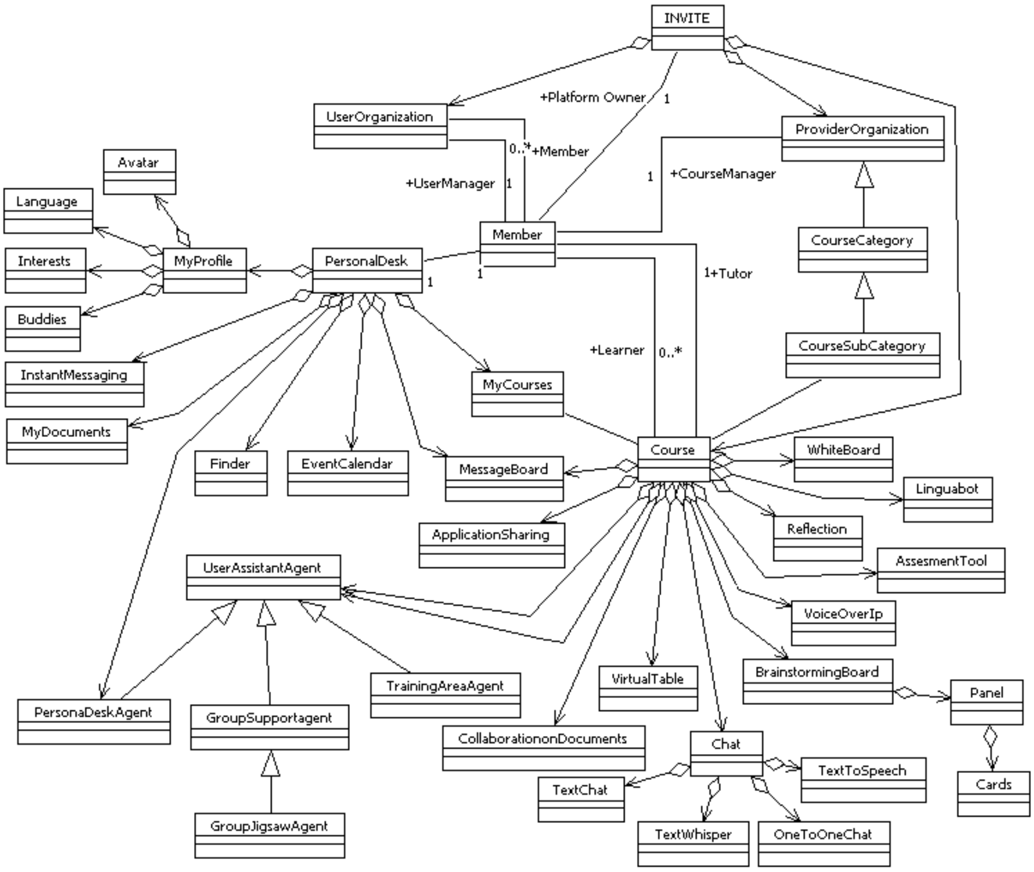

Figure 2. INVITE functions

Member Profile: Each member in his/her personal desk can edit/view his/her preferences in the INVITE community. The following functions are provided: View/Update member' s general information; View /Update member' s interest; View/Select read/write Language; View /Update Filters for instant messaging; View /update buddy list; View/Upload/Change my Avatar.

Event Calendar: INVITE offers four different calendar types to the member on the personal desk: (a) The private calendar belongs to the learner. It is the interface for the user to edit entries for private use. Those dates can be published to the public calendar depending on the rights. (b)The events scheduled for the user: This calendar type merges both private calendar entries, and calendar entries of all courses the member belongs to. This calendar is generated every time it is viewed. (c) The group calendar, which is containing the events regarding the UO the member, belongs to, and is administrated by the member managers. The member manger can decide for each entry, if it should also be visible in the public calendar. Default is local. (c) World calendar, which is containing all the events set to public state by the member managers in the UO calendars, and so the world calendar is administrated by all the member managers collaboratively.

The above calendars offer the following functionality: Create a new event; Display an event; Update/edit an event; Delete an event; Configure a calendar; Adding a reminder; Edit/Delete a reminder.

My Documents: My Documents is the area where the user can upload, edit, copy, search, delete learning content as well as to see related info about learning content.

Personal Desk Agent (is described later in this paper).

5.2.2. Course Place Functions

The course place provides the following functionality:

Chat: The chat functionality includes text -chat, text to speech, and text whispering.

VoiceOver IP: Using push-to-talk or automatically non-silence detection the users are able to communicate to the specified partner (one-to-one chat) or to all members.

Application Sharing: A user is able to share the window of his local computer a document/application with the rest of participants in the same course.

Collaboration on Documents/applications: The members are able to collaborate on a shared document/application giving the other participants in the same room the control of the shared application.

Whiteboard/Paintboard: The user can initiate the use of whiteboard clicking on the respective "whiteboard" button in the 3D room. Then the Whiteboard opens in a pop-up window not only in the user desktop but also in the other participants.

Virtual Table: INVITE course 3D places contain (according to the learning scenario) a virtual table, which is the main tool for the uploading and the representation of the learning material. According to the learning scenario the virtual table can have different functionality. The functionality of the virtual table is the following: (a) Drag and Drop learning content into the table from the My documents; (b) Presentation Board -Present learning content (pictures, slides, video, 3D objects) on the table; (c) Brainstorming Board.

Content open for the participants from the table in a popup window: (a) Open a document (not VMRL compliant file); (b) View SCORM content; (c) Show reflection data; (d) Prepared Questions.

Expel disruptive students from the course: The tutor is to click on the name of the disruptive student (in order to select him), and then he can see his/hers profile (by right click on the name of the student). After that a window opens which contain the button "Exclude".

Linguabot: Linguabot is represented by an avatar and can interact with the users through the text chat communication channel. The user has a private chat communication with the linguabot. The user writes text into the chat area. S/He has also defined the language that prefers to read and write (in his profile). The linguabot after receiving the textual chat input from each member, analyses each member profile where the necessary translation parameters are defined. Then it sends translated text to all chat members, or direct to the specified member (by one-to-one chat).

Avatar gestures: The users' photorealistic avatars support animations in order to offer the users the capability to express mimics (agree, didagree, etc.), feelings (happy, sad, etc.)and actions (hi, bye, appluse etc.).

Activity Logging: In order to support the users' and the groups of users' self reflection and self-awareness, we provide the users the capability to see their log of activities. The parameteres that are logged are usage of chat, instant messaging, audio channel, avatar gestures, upload of documents etc. There are two type for activity logging: (a) Member Activity logging, and (b) Group Activity logging.

5.2.3. Agents functions

The following agent types have been identified for the INVITE environment:

PersonalDeskAgent: The following System functions have been identified during the analysis phase: WelcomeUser, SuggestUser, ProvideHelpOnUIItem, EscortUser, GoToWebResource.

TrainingAreaAgent: TrainingAreaAgent will provide help to the user when invoked but also will act proactively on certain situations. Will be based on a dynamic structure in order to augment proactive functionalities in the following versions of INVITE.

GroupSupportAgent: This agent type will be used for the user support in using the Collaborative Group Learning techniques offered by INVITE.

6. Technology and Standards

The technological approach used is the integration of DVEs (Distributed Virtual Environments) with intelligent agents in order to achieve a more friendly and efficient way for training. The DVEs accomplish the need of interaction among the users with a friendlier user interface as well as the presentation of the knowledge. The intelligent agents help the users to act within the system and to explore complex and unstructured amo unt of multimedia data.

Components/functionality	Technologies	Supported Standards, Protocols, Specifications
Integration of Avatars	Avatarme' s AvatarBooth (for capturing photo-realistic avatars), Avatar Server, CGI for conversion of ame formatted avatars to VRML formatted avatars	H-anim
Voice over IP	Microsoft DirectPlay application programming interface (API). Direct X 8a	UDP
Streaming Video	Both RealNetwork' s solution and Microsoft's Solution can be used	MPEG, RealMedia
Document Repository	XML, FairCom Database, SCORM run time environment Blaxxun Platform 6 API	SCORM, OpenOffice.org-XML
DVE functionality - 3D User Interfaces	ISB, ISA, Java, VRML, 3DS Max, Blaxxun Platform 6 API	VRML 97, VRML-EAI, VRML-MPEG4, X3D
2D user interfaces	XML, HTML, CGI, Blaxxun Platform 6 API, Javascript	HTML 4.0
Intelligent Agents	Java, EAI, JDBC, CGI	AIML, XML
Translation system	Linguatec' s Linguabot, Personal Translator 2001, e-translation server, Java, Java-Remote Message Invocation (RMI)	OLIF
Application Sharing	NetMeeting Based solution. NetMeeting 3 API.	T.120

Table 1. INVITE supported standards and used technologies

In order to make the best choice of the technologies, we define some qualitative criteria for the whole system: (a) **Scalability**: The ability to support a maximum number of simultaneous users which could vary according to the specific settings of each virtual world. (b) **Consistency**: Consistency is realized by distributing and synchronizing user input as well as user-independent behavior in order to achieve the impression of a single shared world; (c) **Various content support**: Several forms of data should be supported and they should be embedded in the DVEs. (d) **Customization:** The total design and outlook of the virtual environments should be customizable according to the needs of their specific themes. Also the users should be able to exploit various subsets of functionality according to the theoretical learning models or scenarios. (e) **Interoperability**: One of the main criteria in the choice of the technology is interoperability. Interoperability of a system can be achieved if the system uses technologies that follow widely accepted standards.

The basis for INVITE will be existing software products, where available. This is effective, especially for a marketable product. The work of making compatible and integrating existing technologies and standards we think that it is useful and effective. INVITE uses where it is possible existing technologies in order to be compatible with other systems integrating, updating and/or modifying them in such a way in order to accomplish the project objectives and the users needs. Furthermore INVITE does not use existing platforms as they are. The starting point for the implementation is the user requirements and in the next steps the evaluation results and the updated user requirements. The technological solutions are chosen in order to offer the desired functionality and satisfying the criteria of interoperability and system independence as much as possible. For this reason the development work is based on creating new software components based on specifications or on API's. The range of technologies available for developing e-learning oriented virtual environments is more varied than ever. The main components of the system their functionality and the useful technology and standards for its implementation are depicted in Table 1.

7. System Architecture

In order to serve a collaborative e-learning system, the INVITE architecture is defined by some basic principles (Bouras et. al. 2001b). The system should be based on a variety of communication protocols, scalable, platform independent and based on open standards. INVITE system is based on several components that provide all the needed functionality. These components are the following: The client, the INVITE community Server (built on blaxxun's Virtual Worlds Platform), the Agent Server, the T.120 server, the Voice Server, the Avatar Server, the Language server, the Document repository, the Application Programming Interfaces and the Streaming Server. On the server side the Virtual Worlds Platform acts as an integration platform through an extended API. Figure 3 presents the general INVITE architecture.

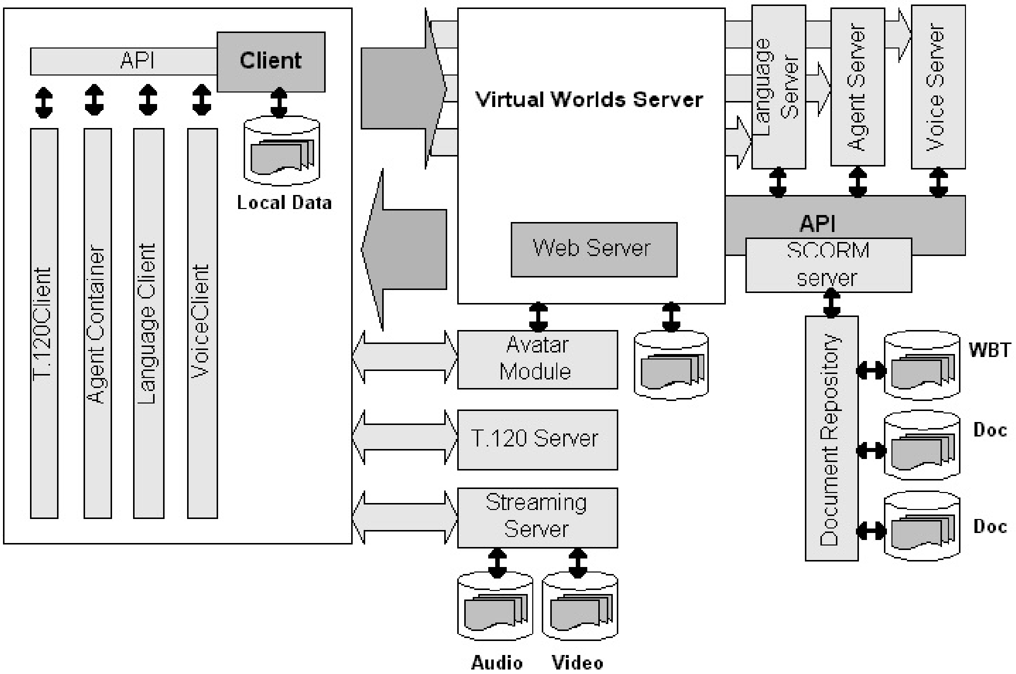

Figure 3. INVITE system architecture

In the following paragraphs the components of the system are described in detail.

Client: Actually the client of the INVITE system is a web browser (Internet Explorer 5.5 SP2 is a tested and recommended solution). In order to provide the user all the desired functionality it is necessary to use specific small external applications (plug-ins and ActiveX controls). These external applications are the BlaxxunContact, and the T.120Client. BlaxxunContact is a plug-in and is installed locally on the users' local PC. blaxxunContact is used to view and interact with the virtual world. It is used in order to send the necessary shared events for the client to the server in order to view the motion of avatars, the gestures, the actions on the shared objects. Also blaxxunContact integrates the voice client and the agent container through a client API. Therefore is used to offer the interaction with the agents and the text chat functionality. Though the voice client the Voice Over IP functionality is supported. This client is used to record the audio stream from the user and to forward it to the VoiceServer. Furthermore it is able to receive a stream from the forward server and playback. T.120Client is used to provide application and data sharing, and collaboration on documents.

Virtual Worlds Platform Module: The Virtual Worlds provides server and client technology for synchronous and asynchronous communication. There are text chat, message boards, user registration, user profiling and tracking, encryption, instant messaging, list of friends, question and answer modules, 2D and 3D user home space, public meeting places, 2D and 3D shared objects. Objects can be interactive web content, like 3D-VRML or HTML and other data formats specified in MPEG4. In INVITE many of these services have been adopted in the INVITE needs as well as new have been implemented. Examples are new users' structure, new templates for creation of INVITE training categories, etc.

3D Worlds Module: The client software is able to visualise dynamic 3D data in high quality and performance. The 3D Worlds module is part of the blaxxun' s Virtual Worlds Platform and provides functions for representing relations in 3D space, like participant groups or any data that is useful to be viewed in 3D. The functionality is adaptable using the client API.

Translation Module: The translation module can translate online chat and stored documents. It can translate on demand, just in time, or as a background service for later use. Translation and look-up is possible for several languages and also for specific use of terminology. For this module INVITE will use Linguatec' s solution based on the Linguatec Personal Translator technology. Linguatec has combined their machine translation engines with a scalable server architecture and many different clients to allow for a processing and offloading of the task of language translation from the HTTP server or the application server to one or many different dedicated translation servers.

Avatar Module: For the purpose of immersive creation of learners, tutors and other roles participating in the learning environment as well as 3D visualisation of participants as avatars there is a photo-realistic representation available on the Internet, hosted by the AvatarMe server. The AvatarMe Server provides user-specific access to personal avatars, which can be accessed from outside the system. The avatar server is used in order to store avatars either on the AvatarMe's format (.ame) or in VRML format (.wrl). Also it is used in order to convert avatars that have created form AvatarBooths (in ame format) to VRML compatible avatars. A third functionality that avatar server provides is the creation of avatars from images with the use of the avatar editor generator. The H-Anim standard is a starting point for the Avatar functionality.

Streaming Video Module: The 3D Worlds module offers the possibility for streaming media content. Depending on other system requirements, the streaming servers from Real networks and Microsoft are supported, and content can be viewed on a screen in the 3D environment or in 2D with the client plug-ins from the server vendor. The technology for streaming content will change with the implementation of MPEG-4. The 3D client supports also the MPEG-4 standard.

Agent Module: The agent module consists of the agent container on the client side and the agent server. The technology used to implement this module is blaxxun server agents and AIML. The agent server controls the agents in the virtual community. It encapsulates all functionalities implemented by the INVITE user interface agents. It includes all the components needed for the user agent interaction, inter-agent coordination and proactive agent behavior. The Agent Controller and the AIML Parser are modules of the agent server.

Agent Controller: All INVITE agents are programmed using special agent scripts that implement agent reactions on certain events raised by the environment. The Agent Controller component coordinates the flow of events from the environment to the respective agents and also disseminates the events generated by the agents to API. In addition, the Agent Controller component coordinates the parsing of the input chat stream performed in the AIML Parser component. The Agent Controller component is used in order to provide an additional layer that can implement complex multi agent behavior based on the ‘atomic’ agent reaction scripts.

AIML Parser: This component is used to parse the user text chat stream, and provide the appropriate text response. The response is based on the AIML structure of the agent glossary and information patterns. In this way the INVITE agents can understand a large number of user text patterns and provide feedback in the proper context.

Content Management Module - Document Repository: The content management module stores, archives and retrieves documents. Meta-Data support the retrieval of content. Existing content from CBT and WBT can be integrated, if the content was created according to standards like XML DTDs describing meta data for learning content. The highest priority for implementing a standard interface will have the SCORM specification (SCORM server module). The SCORM server is an API that it is used for uploading and retrieving SCORM compliant learning content. Also the SCORM server performs a conformance test on the SCORM content when it is uploaded to the server in order to proceed with the uploading of the file or not. If the users' file is SCORM compliant, the SCORM server unpacks it (the zip file) and localizes the Manifest file. The Manifest, an XML document with special structure, contains, among others, metadata describing the package as a whole and metadata describing the instructional resources contained into the package. An XML parser, implemented as part of the SCORM server, reads the manifest file in order to find the location of the metadata. Hereafter, the metadata is collected and stored in the database and the original file is sent to the Web Server.

T.120 server: The T.120 server is responsible for providing multipoint T.120 conferences where the users can connect (using the T.120Client) in order to use application and data sharing as well as to collaborate on documents. In INVITE the WhitePine's MeetingPoint (Meeting Point Conference Server-MPCS) T.120 server is used. This server gives the administrators the capability to create T.120 conferences and to control the members' access on these conferences. These T.120 conferences can be related in every course place in a way that every course place to have its own T.120 conference. This has as result the participants in a course, which are entered in the course place to participate in the same T.120 conference.

Voice Server: The voice server is responsible for providing multipoint voice over IP services. The voice server will support parallel audio sessions in order to provide Voice over IP functionality in different groups of users, in parallel. The voice server receives the streams of the voice clients, mixes them and replicate the audio stream back to the voice clients.

INVITE voice server implementation is based on the Microsoft Direct X8 Game Voice Server. The connection between the chat user is done by the blaxxun contact plug-in. The management, who talks to whom, is controlled by the chat scene mechanism.

8. Conclusion- Future Work

INVITE is a system, which aims to support group-learning processes and offers the advantage of creating nearness and social presence, thereby making participants aware of the communication and interaction processes with others. Besides audio, written chat or using a shared application system they can additionally communicate by gestures, mimics and their positioning within this virtual space. The high innovation potential within the project also brings high risks, since the project explores scenarios and technologies currently not used in the mainstream of learning applications. The unique innovative feature – and risk – in the area of learning scenarios is the focus on collaborative learning: Though research unambiguously demonstrates the advantage of collaborative learning, the current practice in companies is in many case still tutor-driven, closed models. However it has also be shown that this might be because of organizational issues, which can be changed, deploying an environment like *INVITE*. The second unique innovative feature – and risk – lies in the deployment of 3D technology. The use of a 3D-environment is not a required feature by the users a priori, i.e. user do not perceive the use of this technology as an essential feature of the platform. However, they express other requirements, which seem to be best realized with 3D technology: multiple communication channels, and the realization of an environment where interaction and representation is as close as possible to real learning situations. These two basic innovative aspects, collaborative learning and 3D environments, result in a number of further features of the system, which are currently not available in any other products on the market:

- Facilitation of group building: Support of learning in big groups, splitting to small teams, rejoin, change roles during the learning session, etc.
- Reflective tools, including session recordings, analyzing group behavior, but also standard reflective tools as assessments and tests.

Generation of photo-realistic avatars based on photos of the members to get just-in-time self-similar avatars of the participants.

Interaction with all kinds of electronic material within a 3D environment.

Enhanced perception available to users the users due to the use of multi-user 3D environments, gestures, navigation, and multi-modal interaction (person-to-person, person-to-group of persons, person-to-learning content).

Role-dependent virtual screens (e.g. tutor gets detailed reflection data on his virtual screen, learner gets his personal reflection data).

Linguabot: translation service illustrated by a translation avatar; interaction by inviting to a private chat.

User support and monitoring within the environment through intelligent agents.

In this paper we have presented a methodology we adopted in the INVITE project in order to define an architecture of a system, which targets the offering of collaborative e-learning services using collaborative virtual environments. The design of this architecture is based on the requirements of the target groups, who are professionals of certain organizations. To achieve this we have applied a specific system development process and a methodology in order to derive the functional specifications and formulate the system architecture. This methodology is based on a step-wise approach, which aims to translate user requirements into technical specifications in order to satisfy the relative user requirements and to provide a user-centered system. Some results of the application of this methodology are described in this paper. Our next steps are to proceed with the evaluation of the open prototype.

References

AvatarMe, http://www.avatarme.com

Blaxxun Interactive, http://www.blaxxun.de.

Booch, G., Jacobson, I., & Rumbaugh, J. (1999). *The Unified Modeling Language User Guide*, Reading, MA: Addison Wesley.

Bouras, C., Triantafillou, V., & Tsiatsos, T. (2001a). Aspects of Collaborative Learning Environments using Distributed Virtual Environments. *Paper presented at the ED-MEDIA 2001 conference*, June 25-30, 2001, Tampere, Finland.

Bouras C., Triantafillou, V., & Tsiatsos, T. (2001b). Architectures for supporting e-learning through CollaborativeVirtual Enviroments: The case of INVITE. *Paper presented at the IEEE International Conference on Advanced Learning Technologies*, August 6-8, 2001, Madison, Wisconsin, USA.

Bouras, C., & Tsiatsos, T. (2002). Extending the Limits of CVEs to Support Collaborative e-Learning Scenarios. *Paper presented at the IEEE International Conference on Advanced Learning Technologies*, September 9-12, 2002, Kazan, Russia.

Clark, J. (2000). “Collaboration tools in Online Environments”, http://www.aln.org/alnweb/magazine/Vol4_issue1/Clark.htm.

Fabri, M., Gerhard, M., Hobbs, J. D., & Moore, J. D. (1999). Agents for Networked Virtual Learning Environments. *Paper presented at the Neties’ 99 Conference,* March 18-19, 1999, Krems, Austria.

Gilbert, J. E., & Han, C. Y. (1999). Adapting Instruction in Search of a ‘Significant Difference’. *Journal of Network and Computer Applications*, 22, 149-160.

Goransson, B. (2001). Usability Design: A Framewotk for Designing Usable Interactive Systems in Practice. *IT Licentiate theses 2001-06*, ISSN 1404-3203, Department of Human Computer Interaction, Information Technology, Uppsala University, Uppsala, Sweden.

INVITE (Intelligent Distributed Virtual Training Environment), http://invite.fh-joanneum.at.

Lehtinen, E, Hakkarainen, K., Lipponen, L., Rahikainen, M., & Muukkonen, H. (1999). Computer Supported Collaborative Learning:A Review,
http://www.kas.utu.fi/papers/clnet/clnetreport.html .

Linguatec, http://www.linguatec.de.

Nachmias, R., Mioduser, D., Oren, A., & Ram, J. (2000). Web supported Collaboration in Higher education Courses. *Educational Technology & Society*, 3 (3), 94-104.

Schwan, S., & Hesse, F. (1996). Communicating and Learning in Virtual Seminars: The Uses of Spatial Metaphors in Interface Design. *Journal of Universal Computer Science*, 2 (6), 503-513.

Singhal, S., & Zyda, M. (1999). *Networked Virtual Environments: Design and Implementation*, New York: ACM Press.