Integrating a Web3D Interface into an E-learning Platform

Francesco Di Cerbo
Free University of Bozen
Gabriella Dodero
Free University of Bozen
Laura Papaleo
Department of Computer Science, University of Genova
ICT Department, Provincia di Genova

Abstract

In the paper, we present an application of Web3D technologies, X3D with X3DOM and HTML5, to enhance our e-learning platform DIEL. DIEL exploits the social translucence concept in order to create a virtual and dynamic learning space where avatars of community members can interact while implementing any social constructivist learning paradigm. While supporting collaborative learning communities, the addition of high-performance 3D graphics into a DIEL web page, without requiring any plug-in at client side, allows a complete and meaningful interaction with the e-learning services, that benefits the users’ experience in the exploitation of proximity relationships with respect to other users and resources.


Keywords: e-learning applications, Web3D technologies, X3D, X3DOM, Web applications

1 Introduction

As a consequence of advances in computer hardware and Internet connection speed, web content has grown richer and more interactive for users. One such area is Web3D which has seen the emergence of ISO standard technologies such as VRML (Virtual Reality Modeling Language) and its XML-based (eXtensible Markup Language) successor X3D (eXtensible 3D). X3D, in particular, presents opportunities for further integration with existing web technologies for extraction and display of meaningful data from 3D models. It combines both 3D geometry and runtime behavioral descriptions into a single file, encoded in a particular format such as XML. X3D provides means of associating behaviors and dynamic scripts with 3D objects, so that users can interact with those objects [Hamza-Lup et al. 2006].

In the context of global communication, these technologies are becoming more stimulating to create the possibility of collaborative spaces for e-learning and simulation. Recently, with the popularity of e-learning increasing among academic and training institutions, the use of Web3D content presents an opportunity to enhance learning as to create 3D virtual environments where students and teachers can share knowledge. For this reason, web-based knowledge transfer is becoming a field of research which deserves the attention of the research community, regardless of their domain of expertise [Hamza-Lup and Sopin 2008].

In this scenario, we foresee a research trend in studying and designing e-learning collaborative systems where interactions among users can be done in a 3D environment, moving towards 3D virtual worlds with avatars, objects, scenarios where the student can actively learn in a game-like immersive virtual reality.
In this paper we present our interactive e-learning platform focusing the attention on the design, development and implementation of an innovative 3D client interface which uses the X3D standard in combination with the promising future web technology WebGL [Kronos Group 2010] and X3DOM [X3DOM Project 2010]. Our e-learning platform, called DIEL, has been developed as an extension to a popular, free-open source software, Moodle [Moodle 2010]. It supports the creation of a virtual learning space with the aim of providing new ways of user interaction and data representation in a web-based real-time environment. In a virtual learning space, teachers and students cooperate in the creation of shared knowledge, building meanings and concepts where each individual has his/her own relevance within the entire learning process. Users with the role of teachers coordinate interactions, steering the learning community towards a specific educational goal.

It has been demonstrated by a number of authors ([Sancho et al. 2009], [De Lucia et al. 2009a]) that by introducing a much more enjoyable, immersive and interactive interface, students are attracted by the novelty, and this could provide the conditions for transforming the sometimes passive actors (i.e., students and teachers) into involved, very active actors [Wilson 1993]. However, defining actual benefits from the use of learning environments is still a difficult matter, since applying proficiently virtual spaces in learning and teaching activities requires an alchemy of several factors, the most important of them is the responsibility, up to the teacher, to populate the environment with suitable materials and an ad-hoc designed strategy for learning activities, in order to meet the intended learning objective. For this reason, the virtual environment should be designed in order to assist teachers in the challenge, providing not only a set of ad-hoc services, but also using metaphors easy to adapt to the context. Our work tries to follow this principle, supporting the creation of a customized virtual learning space, that could be used to provide a virtual representation of a learning path (see [Di Cerbo et al. 2009]).

The 3D interface is at an initial state of development, but the results are stable enough to start an interesting evaluation phase of the usability of the interaction mechanisms, as well as of the response from teachers and students in using a new paradigm for web-based e-learning communication.

The rest of this work is organized as follows. In Section 2 we briefly introduce the distance learning concepts and the web-based e-learning field where our application can be placed. Section 3 is dedicated to the description of the necessary technologies for the 3D interface, while Section 4 describes the interactive e-learning tool based on Moodle which integrates our interface. In Section 5 the game-like 3D interface is presented with all the necessary implementation details. Finally, in Section 6 some concluding remarks are drawn.

2 Web-based E-learning Applications and similar works

E-Learning could be defined as an effort to support learning processes, exploiting ICT services. Computer Supported Collaborative Learning (CSCL) aims at studying technologies that enable an active cooperation in a virtual community of learners.

Web-based e-learning applications, as the one we present in this work, are part of a wider world of distance learning applications as shown in the following diagram (Figure 2).

The schema has been defined in [Anohima 2005] with the intent to describe and classify the mass of words and concepts around e-learning emerging technologies.

Recently, a great attention has been put on studying how to use 3D virtual environments, or metaverses, in education [Monahan et al. 2008], [Zhang et al. 2010], [De Lucia et al. 2009a].

A number of studies has been focusing on understanding the effectiveness and the benefits brought by the use of metaverses: for instance, [Abbastista et al. 2009], [De Lucia et al. 2009b] approach the problem in different ways, conducting empirical studies that compare virtual with real-life learning experiences. Their results show a positive tendency of the sample considered towards the use of Second Life in learning experiences, that seems to be proficiently usable, instead of direct face-to-face communications.

In other experiences, for instance in [Sancho et al. 2009], despite the good reactions to a similar metaverse used by the users, it is underlined that such technologies have a limited impact with respect to the learning strategies adopted in a learning experience. Indeed, one of the most interesting research directions is how to exploit the 3D metapheres in supporting meaningful approaches for learning communities.

The approach described in the paper shares some features with the previously mentioned works: among them, the use of the same Learning Management System (LMS), Moodle, to exploit its services in terms of learning activities. This characteristic, however, is also a source of distinction: DIEL is seamlessly integrated into Moodle, without requiring any additional software layer as a graphical interface. Moreover, in the design of the tool, we referred to earlier experiences, like for instance [Pfister et al. 1998], where the virtual learning space was modeled and functional to ease the fruition of contents; the benefits of such an approach are the provisioning of a representational and mnemonic method on mapping semantic onto spatial relations, as well as the facilitation for group and community interactions.

The choice of a web browser as the only required mean to access the structure is strategic, and it aims at supporting different visualizations on top of the same metaverse structure. Our analysis of the recent technological advances is that web browsers are becoming a mean for the development of applications more and more similar to native ones. While it was possible to obtain similar results with cross-platform technologies like Adobe Flash, for instance, HTML 5, WebGL and AJAX represent a valuable alternative, not based on
any commercial plugin or proprietary standard. Moreover, through the mentioned standards also mobile devices could be used to access the tool, and with their continuous raising of performance levels, the quality of the experience of mobile users is getting closer to those using a desktop or laptop.

## 3 Authoring and Visualizing Web3D Content

Today several technologies exist to create 3D content for the web. Most of them are plug-in based systems, which means they depend on an additional piece of software running inside a web browser. This adds more complexity for the end user to use such 3D content. The user has not only to deal with plug-in installation but also with potential security issues and incompatibility.

In this section we present a brief summary of the most famous technological frameworks used for creating 3D content on the web, focusing our attention on X3D, WebGL and X3DOM which are the basic technologies on top of which we built our 3D interface.

One of the first technologies able to support animations on the web has been Java, thanks to the use of applets which give the possibility to run small programs inside a web page. For instance, with Java, it is possible to use projects such as Java3D [Java3D 2010], suitable for creating and representing 3D content on the web. Today applets are quite uncommon, sometimes replaced by more powerful standards. Flash [Adobe System Inc. 2010], for example, is a multimedia platform designed to create rich, interactive animated web content and is commonly used on the web. It is based on a proprietary technology owned and distributed by Adobe System and runs inside the web browser using the a specific plug-in. Flash content gets distributed using a binary SWF file format. In order to create such SWF files a commercial authoring software is needed. Until version 10 of Flash there was no real 3D support, but with the release of version 10 basic manipulation of 3D object was added. Currently there are several applications that are using these new features of Flash providing support for creating 3D content and representing it on the web. Unfortunately this is a commercial solution and not an open standard.

On the other side, O3D [Google Inc 2010] is a Google attempt to establish an open standard for 3D graphics on the web. The architecture is based on two layers: a lower-level layer and a higher-level layer. The lower-level is implemented as a browser plug-in which communicates with the graphical subsystem like OpenGL or Direct3D, of the underlying operating system. The higher-level is a JavaScript API providing access to the underlying graphical subsystem. O3D uses the concept of scene graph for representing 3D content. At present O3D supports all major web browsers such as Google Chrome, Safari, Firefox and Internet Explorer.

### 3.1 X3D

X3D [Web3D Consortium 2010] is an open ISO standard developed by the Web3D Consortium with the aim of creating the next-generation of specifications for representing three-dimensional graphics. It is the successor of VRML, including a large number of new and extended features. A major difference between X3D and VRML is the use of XML as syntax. X3D benefits in many ways of using XML since it is easy to read for both humans and computer systems, it is well supported, it is license free, the data is well structured and it is web compatible. XML also allows a direct extension of the X3D language, providing in this way a real mechanism for extensibility. Applications that can parse and render X3D scenes are called X3D browsers and are often implemented as a plug-in to be integrated into web browsers. These X3D browsers expose a so called Scene Access Interface (SAI) for manipulating the content of a scene. X3D also provides bindings for programming languages like ECMAscript and Java. The X3D specification defines not only support for multiple file encodings but also multiple programming language bindings whereas the DOM binding is only a recommendation. Internally X3D uses a scene graph to model objects in a three dimensional space. In X3D there exists also the notation of so called profiles. A profile defines a set of functionalities and components of X3D. Five profiles exist whereas each profile includes the previous one: Core, Interchange, CAD Interactive, Immersive, Full. Profiles give the advantage that, by using them, an X3D browser application has not to implement the full specification, achieving intermediate support by implementing only a subset of profiles. In X3D it is not only possible to model basic 3D objects it is also capable of modeling very complex real-time 3D scenes with user interaction, dynamic scene graph manipulations, animations and event based programming.

### 3.2 WebGL: Web Graphics Library

WebGL [Kronos Group 2010] is a new cross-platform, royalty-free web standard for a low-level 3D graphics API. It is currently developed by the Khronos Group, a consortium founded by industry leading companies like Intel, AMD, Apple, Google, Mozilla, IBM and many more with the focus on creating open standard API specifications for a wide range of platforms and devices. WebGL was created with the aim of bringing plug-in free 3D graphics into modern web browsers. Major browser manufactures like Apple (Safari), Opera (Opera), Google (Chrome) and Mozilla (Firefox) already implemented the first draft of WebGL into the development version of their browsers. From a more technical point of view, WebGL is a DOM API, exposed through the HTML5 canvas element. It is fully integrated with the web page DOM interface and can be used by DOM compatible languages, like JavaScript. It is based on the OpenGL ES 2.0 specification [OpenGL 2010], uses the OpenGL shading language and, for our purposes, it is generally similar to the OpenGL API.

Recently, the Virtual and Augmented Reality Department of the Fraunhofer Institute proposed X3DOM [Behr et al. 2009] as an open-source framework developed with the aim to bring the X3D standard to HTML5 on top of WebGL. X3DOM has been presented as a framework for integrating and manipulating (X3D) scenes as HTML5-DOM elements, which are rendered via WebGL thus not requiring plug-ins or additional software for displaying the X3D content. The system allows the declarative definition of 3D-scene description and runtime behavior. The first stable version of the framework has been released very recently, on March 2010. Examples and API can be found in [X3DOM Project 2010].

## 4 The DIEL e-learning platform

DIEL [Di Cerbo et al. 2008a] is an e-learning platform developed for providing new ways of user interactivity and data representation in a web-based real-time environment. The project uses a client server architecture, where the client side is based on a standard HTML5 browser [W3C HTML5 2010] and the server side is implemented as a standalone PHP application, embedded into the open source course management system Moodle. More precisely, DIEL is a project that aims at creating a dynamic and interaction-friendly graphical interface (a virtual learning space) for community portals. The actual implementation focuses on the e-learning system
Currently, a DIEL user lives chats, and assignments. inside a room. Examples are forums, wiki, textual or audio-video resources using (4) to be either students or teachers, interacting in the environment and through the use of the social translucence concept [Erickson et al. 2002]. The social translucence principle primarily states that if all members of a web community could be aware of what every other member is doing, the whole community would modify its dynamics accordingly, as for real life interactions. The paradigmatic example is very simple but effective: if one person is going to open a door, knowing that on the other side another person is going to do the same, she would open it carefully. Probably, if she had not this information, the door would be opened in a different way, risking to hurt the person who is on the other side. The social translucence principle applied to this example would mean to design a door with a window, in order to allow to see on the other side. DIEL is applying the social translucence principle, letting the learning community be aware of every activity held by every member; if a student accesses a resource, her avatar will be displayed close to it. This allows members to group, to discuss or to study together, in a structured way or not.

In the DIEL virtual environment, individuals are free to move and interact, find contents and insert opinions, without a fixed interaction stereotype. The organization of a learning activity inside DIEL becomes much like the exploration of a virtual set of rooms, each of them dedicated to a specific learning activity. Moodle resources appear as objects inside the room, and moving from room to room is achieved by traversing doors. When planning a learning path for the class, the teacher naturally maps materials and activities into a corresponding path which traverses interconnected rooms. Number, purpose and content of each room are defined by the teacher, while the learners are free to explore them. The software can be applied in pure e-learning environments, or as support tool in blended teaching [Di Cerbo et al. 2008b].

As described so far, in DIEL, courses are represents as rooms which are connected together with a predefined topology. The main components are: (1) rooms, which describe the virtual learning space, (2) doors, which connect two different rooms, (3) users, which can be either students or teachers, interacting in the environment and using (4) resources, identifying any source that a user can inspect inside a room. Examples are forums, wiki, textual or audio-video chats, and assignments.

Currently, a DIEL user lives in the environment as a 2D avatar (see Fig. 4) that is able to move around in the learning space (a set of virtual rooms) and that has the ability to interact in real-time with both other online users and available resources. The following representation is generated using an AJAX-based 2D interface which communicates as often as needed with the DIEL server. The server is responsible for ensuring real-time coherence among all users, as well as monitoring the activities in the learning space.

An experimental extension of DIEL towards 3D interfacing has been developed around one year ago using FFilmation, a 3D graphics engine based on the Adobe Flash technology (see Fig. 5). This prototype was developed in order to compare the adequateness for educational purposes of 2D and 3D representations of the same applications and services. The prototype was tested with 30 high school students, with the aim to verify the usability and user satisfaction of 2D and 3D DIEL interfaces in e-learning. These students have filled in a questionnaire; those with greater expertise in videogames have criticized as too elementary the graphical interface features (this consideration goes well beyond the intended design of 3D graphical interface in DIEL, which was never conceived to be a competitor of videogames!). However they all believe that, by substituting current classroom activities with others based on the 2D/3D version of DIEL, probably the same learning objective may be achieved with more personal satisfaction; more interestingly, they all gave a concrete benefit to interactions between peers with respect to a traditional e-learning environment, like the Moodle original one. Other students, maybe because of a more passive attitude, reported an occasional lack of orientation, especially with the 3D view. We took into account all these problems and requests, and this early experience represented the elicitation of user requirements for the design of the final 3D interface using an advanced and flexible technical infrastructure.

**5 Web3D Interface for DIEL**

In this section we present our new interactive 3D interface for the DIEL e-learning platform presented above. We decided to use X3D, WebGL and X3DOM for HTML5 in the implementation, since we believe that together they are the most promising technologies near-
changed the with introduction of WebGL and the X3DOM project.

In order to be able to represent

In the past, the only drawback X3D had,

any on-line games nowadays allow to export characters and other

With a new X3D client, a user will be able to upload his avatar, as

With WebGL this is going to be removed since it will be built in the

availability on mobile devices;

easy to extend;

should be well documented to speed

possible software with free/open source license is preferred;

as for teachers, as for students.

one of the main limitations of DIEL has been,

the need of additional software to render 3D content on the web.

With WebGL this is going to be removed since it will be built in the

pre-rendering or pre-animation. Another powerful feature

results in a very modular modeling of 3D virtual scenes. X3D is

also a real-time 3D authoring language, which means it handles the

whole animation process of a three-dimensional object without the

need of pre-rendering or pre-animation. Another powerful feature

of X3D is the ability for X3D scenes to include references to other

external scenes, and to load them. This mechanism allows us to

develop a main scene composed by several smaller scenes. This

results in a very modular modeling of 3D virtual scenes. X3D is

useful also to overcome one of the main limitations of the old Flash 3D

interface, namely the impossibility to customize scenes and avatars.

With a new X3D client, a user will be able to upload his avatar, as

many on-line games nowadays allow to export characters and other

resources in the X3D format. Moreover, the use of X3D resources

adds flexibility to scene generation; in fact, we defined a rigorous

Object-Oriented architecture of elements that could be used by the

server to model a DIEL scene, but we left to an inversion of control

pattern the binding of objects to their X3D graphical representation.

In this way, different instances of the class Door could be associ-

ated to different X3D specifications, according to a dynamic context

managed at server side. In the past, the only drawback X3D had,

and that prevented us to adopt it from the beginning, was the lack of

native support in web browsers. In order to be able to represent

X3D content in the browser a third party plug-in was needed. This

changed the with introduction of WebGL and the X3DOM project

[X3DOM Project 2010].

The implementation of the 3D interface can be divided into two

parts. One part consisting of modeling 3D scenes and objects using

X3D. The other part of developing a foundation framework pro-

viding essential client services like server communication and user

interface interaction.

5.1 Foundation Framework

The foundation framework at client side was developed as an

AJAX based web application using the Closure JavaScript library

[Google 2010], in order to gain a modular and object-oriented de-

sign. The implementation is organized in the following namespaces

and classes:

- diel.client (AbstractClient, X3DClient)
- diel.net (AbstractResponseHandler, RequestType, Response-
Handler, ResponseType, XhrIoHandler)
- diel.ui (AbstractUIManager, X3dUIManager)

This foundation part of the 3D interface has a very simple task: it

is responsible for only handling basic operations, and everything

related to 3D is handled by the X3D standard.

From an architecture point of view we can divide the client into

three main layers: (1) Client, (2) Communication and (3) User in-

terface.

The Client layer defines the main application class, called

X3DClient. This class is responsible for instantiating parts of the

other layers and glue them together. The class implements an Ab-

stractClient interface to ensure loose coupling between the com-

ponents and to improve the flexibility of the system. It also en-
capsulates references to the user interface and to the communica-

tion manager of the client. The main responsibility of the client

layer is to provide a real-time communication between the DIEL

server and itself. In a standard client-server architecture, a direct

communication schema would be adopted in order to minimize net-

work and computational overheads; however, the notation of Sock-

ets does not exist in the DOM API reference of web browsers. In

order to achieve a nearly real-time communication, the client uses

a mechanism called polling, through the well-established AJAX

calls. Polling is a technique where the client sends requests to server

at constant or variable intervals without waiting for a response, in a

fully asynchronous manner. To this respect, emerging standards and

methodologies like Web Sockets [W3C 2010] and Comet [Cra-

ne and McCarthy 2008] could represent a significant improvement,

in terms of performance and resource allocation.

The Communication layer is responsible for handling the whole

communication process between the client and the DIEL server.

Communication is based on a protocol, consisting of a set of oper-

ations [Di Cerbo et al. 2008b] embedded in messages using JSON

(JavaScript Object Notation): an example is shown in Listing 1.

Figure 6 shows a simplified sequence diagram of the communica-

tion process, involving DIEL client, server and a database.

```json

define the 3D interface has a very simple task: it

The implementation of the 3D interface can be divided into two

parts. One part consisting of modeling 3D scenes and objects using

X3D. The other part of developing a foundation framework pro-

viding essential client services like server communication and user

interface interaction.

5.1 Foundation Framework

The foundation framework at client side was developed as an

AJAX based web application using the Closure JavaScript library

[Google 2010], in order to gain a modular and object-oriented de-

sign. The implementation is organized in the following namespaces

and classes:

- diel.client (AbstractClient, X3DClient)
- diel.net (AbstractResponseHandler, RequestType, Response-
Handler, ResponseType, XhrIoHandler)
- diel.ui (AbstractUIManager, X3dUIManager)

This foundation part of the 3D interface has a very simple task: it

is responsible for only handling basic operations, and everything

related to 3D is handled by the X3D standard.

From an architecture point of view we can divide the client into

three main layers: (1) Client, (2) Communication and (3) User in-

terface.

The Client layer defines the main application class, called

X3DClient. This class is responsible for instantiating parts of the

other layers and glue them together. The class implements an Ab-

stractClient interface to ensure loose coupling between the com-

ponents and to improve the flexibility of the system. It also en-
capsulates references to the user interface and to the communica-

tion manager of the client. The main responsibility of the client

layer is to provide a real-time communication between the DIEL

server and itself. In a standard client-server architecture, a direct

communication schema would be adopted in order to minimize net-

work and computational overheads; however, the notation of Sock-

ets does not exist in the DOM API reference of web browsers. In

order to achieve a nearly real-time communication, the client uses

a mechanism called polling, through the well-established AJAX

calls. Polling is a technique where the client sends requests to server

at constant or variable intervals without waiting for a response, in a

fully asynchronous manner. To this respect, emerging standards and

methodologies like Web Sockets [W3C 2010] and Comet [Cra-

ne and McCarthy 2008] could represent a significant improvement,

in terms of performance and resource allocation.

The Communication layer is responsible for handling the whole

communication process between the client and the DIEL server.

Communication is based on a protocol, consisting of a set of oper-

ations [Di Cerbo et al. 2008b] embedded in messages using JSON

(JavaScript Object Notation): an example is shown in Listing 1.

Figure 6 shows a simplified sequence diagram of the communica-

tion process, involving DIEL client, server and a database.

```json
{
    "type": "complete",
    "user": "admin",
    "additionalParameter": "",
    "data": {
        "0": {
            "parameter": "type", "value": "Room"},
            "parameter": "name", "value": "Foyer"},
            "parameter": "data", "value":
            "<group id='room'>shape >... <shape></group>" },
            "1": {
            "parameter": "type", "value": "Avatar"},
```
The main class of this layer is represented by the XhrIoHandler. The class is responsible for sending requests to the DIEL server using an XMLHttpRequest API call. XMLHttpRequest is a Javascript API available inside every modern web browser providing a mechanism for making HTTP/HTTPS request through JavaScript code. The XhrIoHandler provides the ability of handling multiple requests issued concurrently. It uses internally a XhrManager provided by the Closure JavaScript Library. This manager creates a predefined number of XHR objects and collects all requests in a priority queue. Every time a XHR object is available it takes the request with the highest priority and processes it. This allows us to issue multiple XHR requests simultaneously and to prioritize important ones. Moreover XhrIoHandler is also responsible for forwarding any response message received from the server to the ResponseHandler. The handler parses the response message and processes it or forwards it to other classes of the client.

The User Interface layer is based on an abstract principle called Inversion of Control [Fowler 2004]. From a technical point of view, Inversion of Control means a decoupling of the execution flow of certain operations from the effective implementation. The caller of an operation has no control of how and when he will get an answer. It is also often referenced as the Hollywood principle: don’t call us, we’ll call you. In our solution this means the user interface layer is not responsible for creating or manipulating three-dimensional objects since that is all handled by X3D. Javascript classes from the user interface eventually get called by X3D objects for handling certain operations or events. The main implementation of this layer lives in the X3dUIManager class. It provides methods for setting up an initial interface and for adding or removing predefined X3D elements, received from the server into the HTML 5 DOM tree. Moreover it implements event listeners which will be triggered by user interactions with X3D objects. X3dUIManager has relationships with other classes (represented in Figure 7), that are responsible for implementing the different aspects of the 3D interface.

Figure 6: A simplified sequence diagram of the communication process handled by the Communication layer.

Figure 7: Javascript classes defined for DIEL User Interface responsible for implementing the different aspects of the 3D interface and used by the X3dUIManager.

5.2 X3D scene modeling

X3D scenes can be modeled using the XML syntax. As any valid XML document also X3D uses the notation of nodes and attributes, whereas nodes can have a parent-child relationship. Every valid X3D scene consists of a X3D root node, with attributes for the version and profile used by the scene. The root node has two children: an optional Head node, containing meta information about the scene and a Scene node, containing all scene graph nodes, used for modeling a X3D environment.

Modeling complex X3D scenes can be a difficult and hard process. Fortunately for educational purposes too complex scenes are not needed. Moreover, we defined fixed values for common used elements in a three-dimensional environment such as: viewpoint position, light, scene background to further simplify the modeling process.

We said above that DIEL uses the notation of a room for representing a course. For simplicity, in a DIEL X3D scene we use a floor notation instead. Listings 2 and 3 below show the X3D scene and Javascript code, necessary for modeling such floor object. The listing is simplified with respect to the real one, as the scene is generated dynamically at client side, according to the directives specified by the DIEL server and conveyed with messages shown in Listing 1.

Listing 1: A part of a JSON message exchanged between DIEL 3D client and server.
Listing 2: Simplified XHTML and X3D source code for generating a X3D client view: informations regarding the texture to use, as well as all relevant parameters can be stored on a database and easily injected into the source code using Dependency Injection.

```html
<html>
<body>

Listing 3: Simplified Javascript source code for generating a X3D client view:

```javascript
/*
 * A Floor.
 */

```java

//namespace.Floor.prototype.toX3d = function() {
var node=document.createElement("transform");
var shape=document.createElement("shape");
if (this.texturePath != null) {
    var appearance=document.createElement("appearance");
    var texture=document.createElement("imagetexture");
texture.setAttribute("url", this.texturePath);
appearance.appendChild(texture);
shape.appendChild(appearance);
}
var box=document.createElement("box");
shape.appendChild(box);
node.appendChild(shape);
return node;
}

Figure 8: Rendered X3D scene representing a floor object integrated inside Moodle.

The scene is composed of a Shape node containing a Box, adjusted in size to resemble a floor. Moreover, the Shape node uses an Appearance element for setting the color of the Box. Figure 8 shows a rendered output of the scene inside DIEL using the X3D client.

In order to simulate the concept of users in a room, we created a second X3D scene consisting of an avatar, that can be controlled with a mouse, and a set of resources. By combining the two X3D scenes we are able to simulate, with our X3D client, a typical DIEL scenario where multiple users are together in one room. In Figure 9 the rendered scene is shown.

The DIEL 3D interface has been integrated into Moodle as a separate course format, and this allows the DIEL server to access to Moodle internal libraries and interfaces, for example access to the authentication system or to the database. Using the course format for integrating the X3D client into Moodle has also the benefit that it gets embedded directly into the user interface, keeping the structure and specific visualization objects of Moodle in their usual places. Figure 9 shows the actual integration into Moodle; users are presented with a very well-known structure. More important, the integration layer allows 3D and 2D visualization to expose resources and services directly managed by Moodle. Such resources are displayed as objects (if the course format is 3D) or icons (if the course format is 2D); internally, they are made accessible via URLs. Since the client is just a browser, the entire visualization and linking of resources is totally transparent to it. For instance, a teacher may upload didactic materials, or create wiki, forums, assignments, and so forth, as in any other Moodle course. A student could decide to use one of such resources, simply moving her avatar to the corresponding object or icon.

Because of our choice to use quite recent Web3D technologies, the 3D interface is still at an initial stage of development. As already mentioned, X3Dom is currently under development, and reached its first official release only on April 25th, 2010. The current implementation of our X3D client allows to create dynamic scenes, where avatars, doors and resources are displayed, according to the configuration and topology specified for each course. Users can move in the virtual environment with their mouse, and interact with Moodle through the objects, and with each other via chat (textual or audio-video).

Figure 9: The 3D interface integrated in Moodle, showing a learning room. Two avatars (students) interact with the environment. On the walls and on the floor different resources are located.

6 Concluding Remarks

We presented a Web3D interface for supporting interactions in social constructivistic learning communities of students and teachers, inside a web-based e-learning environment. Our interface has been developed using X3D and the recent frameworks, WebGL and X3DOM, among the most promising technologies to establish a standard for representing 3D content on the web. At the time of this writing, WebGL is already included in the alpha version of the major web browsers. Taking advantages of the extensible 3D standard in combination with future web technologies, we ended up with a very flexible and extensible client application.

The main objective of our work is to study the impact of virtual 3D environments on students’ learning processes. To show how we
plan to do such a study, we detailed the e-learning platform called DIEL, and how its 3D interface was developed. Our preliminary observations with DIEL users confirmed other experiences available in literature, that is, 3D e-learning environments increase the quality of learning in web-based scenarios, by supporting multiple user access and users interactions, and by allowing the users to receive feedback from their own actions as well as from others, thus maximizing the intake of information. With respect to other e-learning experiences, the originality in our approach lies in the possibility to use a browser as the one and only piece of software that is needed at client side. This enables a higher degree of portability of the e-learning environment and its potential for accessing it via mobile devices as well. The second unique feature of DIEL, as briefly mentioned before and described in more details in [Di Cerbo et al. 2008a], lies in its automatic data logging of activities performed.

One of our future directions for improvements will be in the interface functionalities, by allowing, for example, more sophisticated elements in the rooms and more fluid movements for the users. Users will be able to select their own avatars, thus encouraging their attitude to study by interacting: avatars are an important issue in users’ experience, in contextual text-based and audio-video chats.

To this purpose, we are developing a tool to allow users to easily customize the avatars to be displayed, and for teachers, to customize objects associated to resources. We are also considering how to embed videos directly in the 3D interface, in panels or surfaces, exploiting HTML5 and the X3Dom library: such a functionality would be important to implement user videochats. As the coverage of X3D standard by X3Dom grows, as well as the support for HTML5 “device” specification, it shall be straightforward to embed such new functionalities inside DIEL.

A further enrichment to DIEL would be to encode the teachers’ knowledge, associated to the creation of learning path, into an on-demand learning environment and its potential for accessing second life: The case of SecondDMI. Computers & Education 52, 1, 220233.


Hamza-Lup, F., Davis, L., and Zeidan, O. 2006. Web-based 3d planning tool for radiation therapy treatment. In Web3D ’06:


