Architectures for developing multiuser, immersive learning scenarios

DOI:
10.1177/1046878112443323

Document status and date:
Published: 01/12/2012

Document Version:
Peer reviewed version

Cover sheet

Final version sent September 27, 2011
8830 words

Nadolski et al.
Exploring architectures for swift and easy development of multi-user immersive learning scenarios

Rob J. Nadolski
Centre for Learning Sciences and Technologies (CELSTEC), Open University of the Netherlands, Valkenburgerweg 177, 6419 AT, Heerlen, The Netherlands.
Telephone: +31 45 5762389
e-mail: rob.nadolski@ou.nl

Hans G.K. Hummel
Centre for Learning Sciences and Technologies (CELSTEC), Open University of the Netherlands, Valkenburgerweg 177, 6419 AT, Heerlen, The Netherlands.
Telephone: +31 45 5762651
e-mail: hans.hummel@ou.nl

Aad Slootmaker
Centre for Learning Sciences and Technologies (CELSTEC), Open University of the Netherlands, Valkenburgerweg 177, 6419 AT, Heerlen, The Netherlands.
Telephone: +31 45 5762667
e-mail: aad.slootmaker@ou.nl

Wim van der Vegt
Centre for Learning Sciences and Technologies (CELSTEC), Open University of the Netherlands, Valkenburgerweg 177, 6419 AT, Heerlen, The Netherlands.
Telephone: +31 45 5762288
e-mail: wim.vandervegt@ou.nl

Exploring architectures for swift and easy development of multi-user immersive learning scenarios

Abstract

Multi-user immersive learning scenarios hold strong potential for life long learning as they can support the acquisition of higher order skills in an effective, efficient and attractive way. Existing virtual worlds, game development platforms, and game engines only partly cater for the proliferation of such learning scenarios as they are often inadequately tuned for learning. First, this study aims to identify architectures that more effectively support the development of multi-user immersive learning scenarios. Second, this study takes up the challenge to define and assemble more flexible architectures that cater for fast and easy development, which will become important in the current period of economic breakdown. Third, this study describes how such architectures should enable research into guidelines for multi-user immersive learning scenario design and development. This study outlines a method for defining and setting up such architectures by using experts and existing literature.

KEYWORDS: architectures; immersive learning scenarios; serious games.

Playing video games has become a very popular form of entertainment. Besides fun, games may also offer players a chance to take on new roles and experience worlds in which they can (collectively) learn to solve problems. Having recognized the power of video games as an instructional medium, designers are now increasingly creating games for educational purposes. Very different from the brightly packaged drill-and-practice software of the past, these games offer something new to learners: entire worlds in which they can actively participate, where what they know is directly related to what they are able to do and who they become (Gee, 2003; Gee & Levine, 2009). Their deployment enables social interaction and role play, personalised and experiential learning, and learner empowerment through increased interactivity (Scopes, 2009). This game genre holds potential for life long learning as it *can* support the acquisition of higher order skills in a more effective, efficient and attractive way when compared to common alternatives like excursions or classroom teamwork. Such alternatives usually offer less freedom of time, place and pace, and demand a high tutor load during exploitation with large student numbers. The development of contemporary serious games is more costly then is the case for these alternatives, but adequate instrumentation for game development could largely resolve this inefficiency. When we talk about multi-user immersive learning scenarios (ILS), we refer to serious games that are centred around authentic tasks, that provide artificial and/or real environments that challenge and make learners curious, that have appropriate and unambiguous outcome goals, and provide learners with clear, constructive and encouraging feedback. Authentic tasks are immersive in nature enable the acquisition of higher order skills in Higher Education(HE), often involving collaboration between learners.

Many developers choose general purpose software to downsize the costs of multi-user ILS-development (Kapp & O’Driscoll, 2010). However, existing virtual worlds, game development platforms, and game engines appear to only partly cater for the proliferation of multi-user ILS. As software engineers tend to overemphasize technical utilities that can only be used by software experts, the resulting game frameworks are often inadequately tuned towards learning. In addition, they often have steep learning curves for both learners and teachers (De Freitas, Rebolledo-Mendez, Liarokapis, Magoulas, & Poulovassilis, 2010). Most attempts by software engineers to develop instrumentation for fast and easy development of multi-user ILS are still in the requirements analysis stage. Hu’s (2010) recent work seems more dedicated towards multi-user ILS, using UML to illustrate the structure and intrinsic relations of the various subsystems. Nevertheless, his rather abstract, high level perspective towards suitability for all game-genres and its systems engineering approach might prevent an easy uptake by educational experts. Furthermore, as existing game frameworks often lack logging facilities, they impede research on the actual effects of multi-user ILS. Aforementioned shortcomings hinder the uptake of multi-user ILS. An easy to use and adapt, as well as extendible assembly of software

components will be needed to develop affordable multi-user ILS (Law & Kickmeier-Rust, 2008; Michael & Chen, 2006; Moreno-Ger, Burgos, Martínez-Ortiz, Sierra, & Fernández-Manjón, 2008).
It is a challenge to define and assemble a flexible architecture to cater for the fast and easy development of multi-user ILS. Such development entails pedagogical and game design, content authoring, and multi-user ILS-deployment. Developing multi-user ILS often requires interdisciplinary teamwork to balance pedagogical and gaming principles (Killi, 2005). The architecture requires authoring and deployment of mechanisms for personalisation and contextualization. Furthermore, it should enable evidence based research into guidelines for designing and developing effective and efficient multi-user ILS.

This study describes software-architectures that fulfil three aims: (1) to develop multi-user ILS, (2) to swiftly and easily develop multi-user ILS, (3) and to enable research into multi-user ILS.
The article is structured as follows. First, considerations and characteristics of multi-user ILS are clarified (section 1). Next, the method for this study is described (section 2): requirements, selection-criteria, blue print, and a mapping for multi-user ILS-architectures. These architectures also enable research into multi-user ILS. Finally, the concluding section (section 3) resumes the method and its limitations, sketches the current state of the art in multi-user ILS-development, and includes some suggestions for multi-user ILS-research.
It should be stressed that the method described in this study can be applied for another set of ILS or different genre of serious games. Of course, this would lead to a different set of game genre characteristics, to a different blue print and other outcomes for adequate software architectures. Indeed, before actually starting to develop and deploy specific games, one should critically examine what kind of games one wants to use and in which specific contexts. Although a lot of taxonomies for game genres exist, these are heavily disputed; categories tend to blur as new genres include characteristics from several others (Aldrich, 2005). It is recommended to articulate own characteristics and link them to existing research. The next section describes this process for *our* definition of the ILS-genre in which authentic tasks play a key role.

1 Multi-user ILS – characteristics and considerations

Multi-user ILS enable participants to learn by experiencing, exploring, practicing and reflecting when actively engaged in collaborative and playful learning. In this, they can have huge but manageable learner control. Educational content should be at the heart of serious game-play, so that learners can use knowledge and apply skills while playing in a carefully designed learning environment. This concurs with Driscoll's (2005) view on effective instruction in virtual worlds. According to this view learning should take place through exploration and reflection, be embedded in an environment where learners interact through role play, be proactive and collaborative with learners testing and comparing multiple perspectives, and simulates real life in authentic contexts (through real world activities, behaviours, actions and events). By including collaboration in the learning scenario, multi-user ILS will transform learners and learning in three ways: (1) transforming a participant from a passive recipient to an empowered actor, (2) transforming content from information that learners have to remember to a tool that learners can apply to reach certain targets, and (3) transforming context from an assurance that "this knowledge will be relevant in the future" to an actual reality where learner's actions have immediate consequences. Important learning theories that justify multi-user ILS are *experiential learning* and *inquiry-based learning* (Kolb, 1984; Dewey, 1997). Both emphasize the need for (simulations of) realistic experiences within inspiring learning environments while executing authentic tasks. Exploring and reflecting on phenomena, testing hypotheses and constructing objects, learning by doing and reflecting are basic characteristics of these learning experiences (Dewey, 1997; Kolb,1984; Kebritchi & Hirumi, 2008).
A central concept in *our* definition of multi-user ILS is *authenticity*. This concept is represented in authentic tasks and their characteristics: dealing with real world problems, necessity to collaborate and reflect, offering competing solutions and diversity of outcomes, involving different perspectives, being complex and ill-defined (Herrington, Oliver, & Reeves; 2003). Authenticity in ILS is foremost determined by the fidelity of the tasks themselves, and not by the fidelity of the task environment.

Authenticity largely depends on the 'suspension of disbelief' involved. The poet Coleridge suggested that if a writer could bring "human interest and a semblance of truth into a fantastic tale, the reader would suspend judgment concerning the implausibility of the narrative". In other words, authenticity is largely determined by its recipients and their opinion should be that they are indeed dealing with authenticity. Learners solve authentic problems in an environment that might need high fidelity to invoke adequate and meaningful learning experiences. *Fidelity* refers to how something relates to its real counterpart: how close does the experience resemble real life? Stone (2008) proposes a fidelity model with three areas: environment (3D), task, and user interactivity. How lifelike an environment appears does not seem that important, except for simulations of phenomena. Many multi-user ILS can do without such simulations. ILS-tasks need high fidelity: they are authentic. Finally, the lifelikeness of user interactivity is important if acquiring motor skills is essential, which is rarely the case within multi-user ILS. Although the lifelike of the environment is often unimportant, 2D/3D can easily simulate the Virtual World(VW) in which the user is a key-player. Positioning learners in VW-alike settings with the possibility to change camera position and orientation will enhance their presence and increase vividness of the experience. *Vividness* describes the impact of the environment on our senses (Steuer, 1992). Incorporating more senses intensifies the experience and may influence retention of learning (Wood, 2004). There is little evidence of learning benefits that can be attributed specifically to 3D (McFarlane, Sparrowhawk, & Heald, 2002; McLellan, 2004). Dalgarno and Lee (2010) intended to isolate the distinguishing characteristics of 3D in Virtual Learning Environments(VLEs) and their potential learning benefits. They argue that representation-fidelity (3D) along with interactivity will lead to more immersion, and consequently a strong sense of (joint) presence. Nowadays 3D becomes more affordable and better matches the users expectations who now experience 3D leisure games. De Freitas et al. (2010, p 78) point out that: "There is evidence that regular gamers find the graphics of virtual worlds too low level, and can experience negative transfer as a result", and as a consequence 2D/3D seems indispensible for multi-user ILS. VWs provide a rich space for constructivist learning, facilitating more learner engagement than most online courses (Cheal, 2009).

As mentioned before, authentic tasks in multi-user ILS often need to contain elements of collaboration through communication affordances. Richer and more effective collaborative learning can be facilitated by text, voice, emoticons, gestures (Dalgarno & Lee, 2010). Different levels of ILS-communication are: (a) within ILS (influencing ILS-flow and learning flow within ILS), (b) alongside ILS (influencing learning flow within ILS), and (c) about ILS (influencing 'off topic' learning flow). Central to the concept of *immersiveness* is the level of involvement and commitment: active and interactive *participation* in learning through affordances for increased *interaction*. ILS are preferably deployed in a VW-alike manner since this offers rich opportunities for interaction and community sense, which both enhance learning engagement and reduce negative feelings to drop out (Wallace & Maryott, 2009). Immersion is the state where you cease to be aware of your physical self. It is often accompanied by intense focus, distorted sense of time, belief in the (fantasy) context, and effortless action. Participation means acting with objects or – more often - with a group of learners that communicate, and negotiate together as they would do in the workplace (Hummel et al., in press). Such virtual collaboration often results in artefacts (e.g., business plan, product design, film). Interaction refers for example to communication, control of environment and objects. Interaction is conditional for engagement that in turn is conditional for immersion. The optimal learning state is that of being in *flow*: a mental state of immersion and clarity (Csikszentmihalyi, 1990). Please note that immersiveness and flow are not sufficient for characterizing multi-user ILS. Just to mention a few examples, really immersive – but not ILS because it does not deal with authentic tasks - might be using a first-person shooter for the automation of multiplication tables or playing Tetris® to practice mental rotation. It is of utmost importance to orchestrate adaptation of separate ILS-elements so that learning goals can be reached by continuous flow (i.e., the optimal learning state).

Adaptation can foster transfer through immersed higher order learning. Learner preferences, learning style, presentation diversity as well as more contextualization are supposed to lead to deeper and more meaningful learning. Improved transfer of knowledge and skills to real situations can be achieved through contextualization (Clark & Mayer, 2008). Motivation is the fuel to maintain self-directed learning. Several studies have shown that three factors contribute to motivation: autonomy (self directed), mastery (the urge to get better), and purpose (e.g. Ryan & Deci, 2000).

Finally, in line with the previous paragraph, ILS often nurture users' *identity*: how does the user present and see himself through embodied actions and social actions within the learning environment? (Gunter, Kenny, & Vick, 2008). Social presence is thought to make learning more of a human experience. The use of avatars for representation can increase a learner's sense of social presence and allows them to engage with immersive experiences with a greater sense of control (Nowak & Biocca, 2004; De Freitas et al., 2010). In such a context, learners learn how to investigate and pose solutions, and learn what it means to *be* a scientist, historian, etcetera when resolving meaningful dilemmas. Rather than working on problems in which they must imagine the implications of their decisions, learners actually experience the consequences of their own actions themselves (Barab, Gresalfi, & Arici, 2009).
The opportunity to create learning environments being safe, ethical and not possible in real life hints at the enormous potential of multi-user ILS. Studying cell behaviour, walking inside a volcano, abstractions in micro worlds can be part of multi-user ILS. Cheal (2009, p1) contests: "Students in a virtual world can actively create their own projects as three-dimensional environments without the restrictions of gravity, scale, economics, identity, or distance" .

Aforementioned multi-user ILS characteristics and affordances (i.e., multi-user ILS-qualities that allow a user to do specific actions) can be summarized as:

- Experience, exploration, reflection and playful gaming
- Authentic tasks to be carried out in inspiring learning environments (2D/3D)
- Collaboration and communication with others through role play
- Immersion
- Active and interactive participation
- Personalisation and adaptation
- Self-directed learning
- User identity (2D/3D)

2 Method identifying suitable multi-user ILS-architectures

The method and resources of this study are light grey-shaded in Figure 1 (Phase 1). Main outcome is the identification of software-architectures that fulfil three aims: (1) development of multi-user ILS, (2) swift and easy authoring of multi-user ILS, (3) enable research into multi-user ILS.
This study consists of six steps and results in several candidate architectures for multi-user ILS. First, the characteristics of multi-user ILS determined the requirements for architectures for multi-user ILS. Second, these requirements were condensed into a blueprint of a multi-user ILS-architecture. Third, ten selection criteria for multi-user ILS-architectures were identified. Fourth, a first sifting of candidates removed technically non-viable candidates. For this sifting, documentation on websites, manuals, and studies about the candidates were used. Fifth, the four most important selection criteria, aforementioned documentation, and the blueprint were used to evaluate remaining candidates (about 90). The goal was a second sifting of candidates that would be subjected to a more detailed inspection by having them installed and tested. Preferred candidates could cover a range of the blueprint. Sixth, the four most promising combinations for multi-user ILS-architectures were chosen. Steps 7 and 8 can further constrain these to one architecture to be used within research, but these steps were not yet conducted.

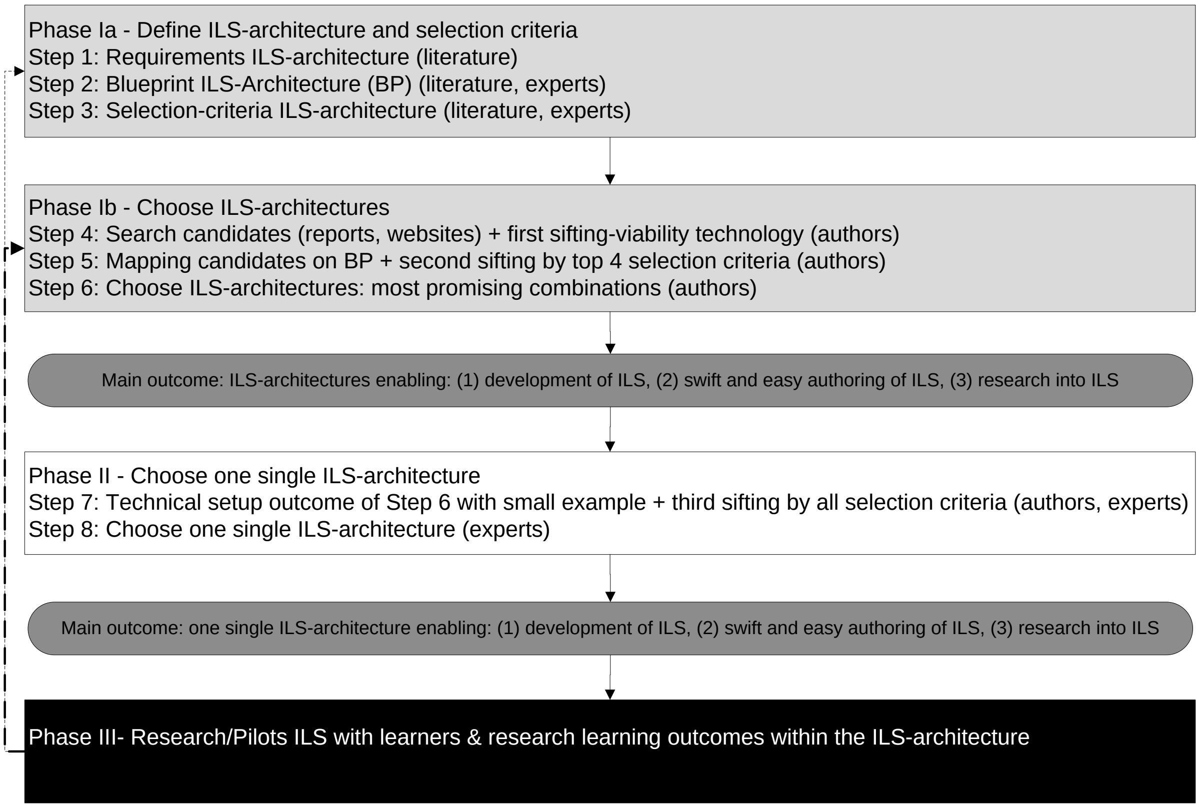

Figure 1. Method to choose one single architecture for swift and easy development and research into multi-user ILS.

While mapping candidates onto the components of the blueprint, we gained more insight about the extent to which they could cover the whole architecture. Not surprisingly, during mapping several gaps were identified. Either because no candidate did meet the criteria, or – even worse – because there were no easy to integrate candidates for some components at all. Interchangeable alternatives could also be identified.

2.1 Step 1: Requirements multi-user ILS-architecture

Requirements can be functional or non-functional (e.g., performance, quality standards). Three generic requirements were identified. A first generic requirement is to focus on a technically reliable, easily extendible *components-based* architecture with adequate performance. Using services for integrating candidates seemed a better option than starting from scratch. We accepted that candidates might not already meet all requirements, but that it would be feasible to adapt them accordingly. A second generic requirement is to look for a single architecture that integrates various technical approaches, and preferably is also usable for mobile platforms. Software-developers often prefer to do the technical setup via dedicated installers: which components are needed, and how should these work together? Finally, a third generic requirement was that the architecture should support data robustness. Existing content should be safe guarded from future technology changes. Preferably all that needs to be created after such changes is a new interface, or at worst an export and translation of the XML-definition of the ILS into a new format.
Hereafter, the implications of the three aims of this study towards more specific requirements for the ILS-architecture are given.

2.1.1 Aim 1: Development of multi-user ILS

Evidently, the resulting architecture should enable to produce and procure ILS. These requirements are nurtured by looking at the earlier described characteristics for multi-user ILS (see section 1, last paragraph). Here, the requirements are restricted to ILS-deployment as ILS-authoring requirements will be treated later (see Aim 2).

Most salient requirements for an architecture for multi-user ILS-architecture are that it:

- should support various means of communication and collaboration;
- should enable a variety of adaptation (personalization and contextualization) means; and
- should include VW-alike functionality (2D) and might sometimes need 3D.

2.1.2 Aim 2: Swift and easy development of multi-user ILS

The architecture in itself is to a large extent pedagogic-agnostic. It should be flexible and enable swift and easy usage by pedagogical experts. Development of multi-user ILS encompasses different stakeholders that use several editors for authoring. Subject matter experts might model their domain and identify domain assets (resources). Instructional designers might use a story-editor and level-editor to specify learning flow for various ILS-roles (playing or non-playing characters). Furthermore, they might specify pedagogical rules for several kinds of (embedded) learner support or emergent learner support (AI-rules). Other adaptation-rules can incorporate storyline assets (in-world artefacts), probably produced by graphical designers. Subject matter experts and instructional designers often work closely together as multi-user ILS are interspersed with pedagogy and domain assets. It should be possible to integrate different types of authoring output. Authors should be able to deploy, test and modify a multi-user ILS-scenario with various ILS-roles. Software-developers are preferably not needed during authoring. However, they should be able to derive and built scenario templates from existing ILS to further streamline the authoring. This will be based upon logging-data. In addition, authors should be able to easily repurpose, maintain, and reuse existing ILS. In sum, various high level editors are needed in which multi-user ILS content can be produced, enriched, tested, and examined for improvements.

2.1.3 Aim 3: Enable research into multi-user ILS

Although some research shows multi-user ILS capabilities to improve learning outcomes, it is often unclear *why* (and how) multi-user ILS do achieve this. A comprehensive and systematic overview of mechanisms that drive multi-user ILS-effectiveness is missing (Ritterfeld, Cody, & Vorderer, 2009; Wilson et al., 2009). Because of this, there is too much uncertainty regarding transfer effects: does better within-multi-user ILS performance also lead to equivalent real world behaviour (outside ILS)? Users actions (overt or covert) should be traced and subjected to analysis of logging data. This can inform design and provide feedback to optimize the flow and learning within multi-user ILS. The architecture needs various means to gather such personal user-data without interfering or even disturbing the user experience. In sum, the architecture should enable gathering user-data for research into multi-user ILS.

<u>2.2 Step 2: Blueprint for multi-user ILS-architecture</u>

The requirements are reflected in the architectural blueprint for multi-user ILS. This is **not** a static picture but evolves over time if new insights and functionalities become available. The blueprint is used for searching candidates and mapping. Different stakeholders will use the ILS-architecture: end-users (learners – teachers), authors, administrators, and software-developers. The ILS-architecture entails three main parts: (1) ILS-deployment, (2) authoring, and (3) authorization and matchmaking (see Figures 2 and 3). ILS-deployment is divided into (1a) representation, (1b) communication (interaction between end-users), and (1c) ILS-engines (calculating adaptation). The depicted sizes of all aforementioned parts are only meant to present a basic graphical structure.

The three main parts of the ILS-architecture and their users are described below.

ILS-deployment aims at end-users actually involved in learning and will support activities in one or more ILS-roles.

The part *authoring* constitutes of components which can be used by various authors, like instructional designers, subject matter experts or graphical designers. End-users during deployment can often also do some 'authoring'. Theoretically, ILS-deployment could even encapsulate all authoring. However, this seems unlikely to happen soon, which is reflected by the dotted arrow in Figure 2 from 'ILS-deployment' to 'authoring', indicating a rather weak relation.

The part *authorization & matchmaking* is used by administrators who define which end-user can access which ILS-instance, in which ILS-role(s), in which time frame, etcetera. If learners themselves can choose or change ILS-roles, a connection from 'ILS-deployment' to 'authorization &

matchmaking' is needed. Administrators can also have tasks like system maintenance by creating backups, monitor system stability, and examining error log files.
Finally, software-developers should have a negligible role during ILS-development, but are crucial in developing, configuring, maintaining and improving the ILS-architecture. Several components support them in doing this. For example, an ILS might need different engines for doing computations (see Figure 3). An (interpreting) programming language can enhance various engines, and even combine or sequence them using data on user's progress.

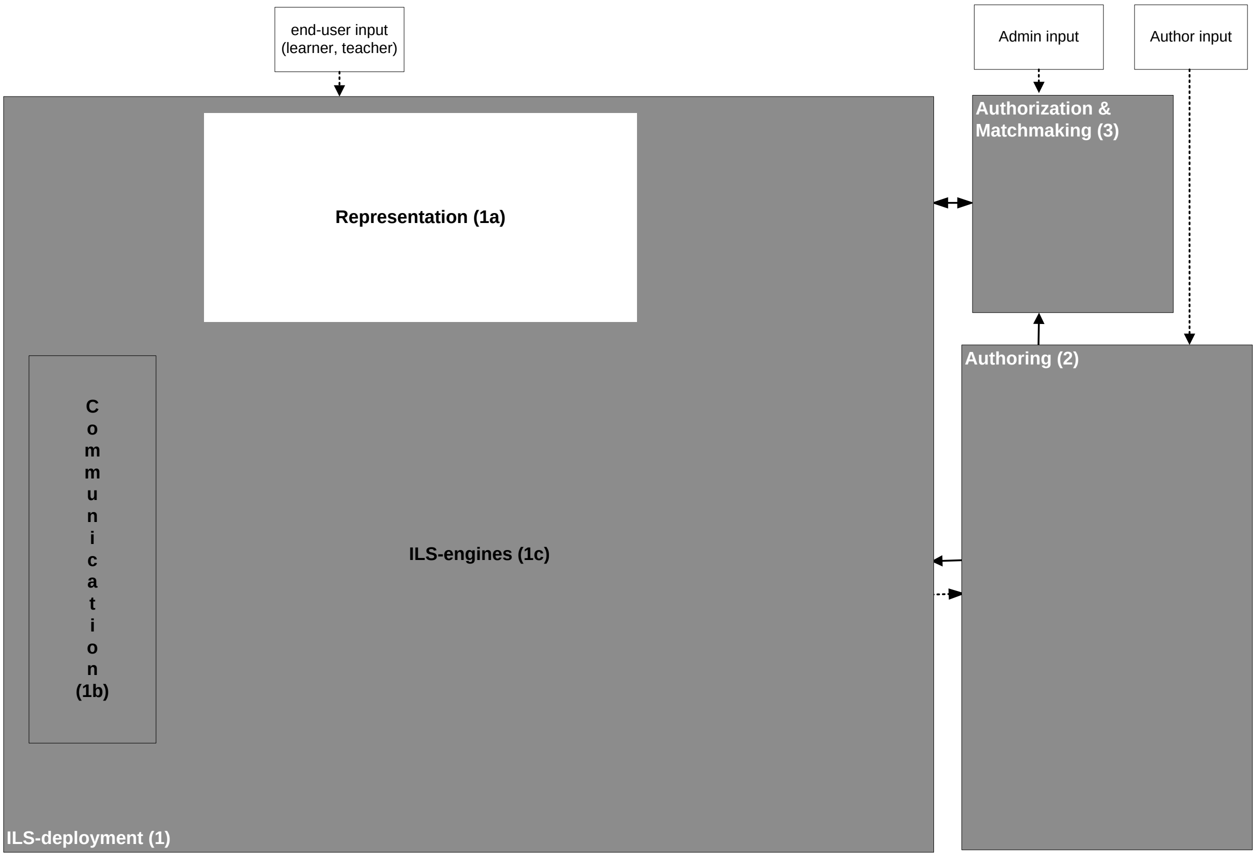

Figure 2. Main parts of the multi-user ILS-architecture.

Technical setup – requirements

In the multi-user technical setup, users can connect by means of a hosting server or location service for matchmaking (i.e., who plays with whom). Users can also interact by communication means like chat. The (mobile) device for ILS-deployment is a simple player whereas major components controlling the ILS are server-located. This eliminates portability issues as one can control the server-type independent of connected client-types.

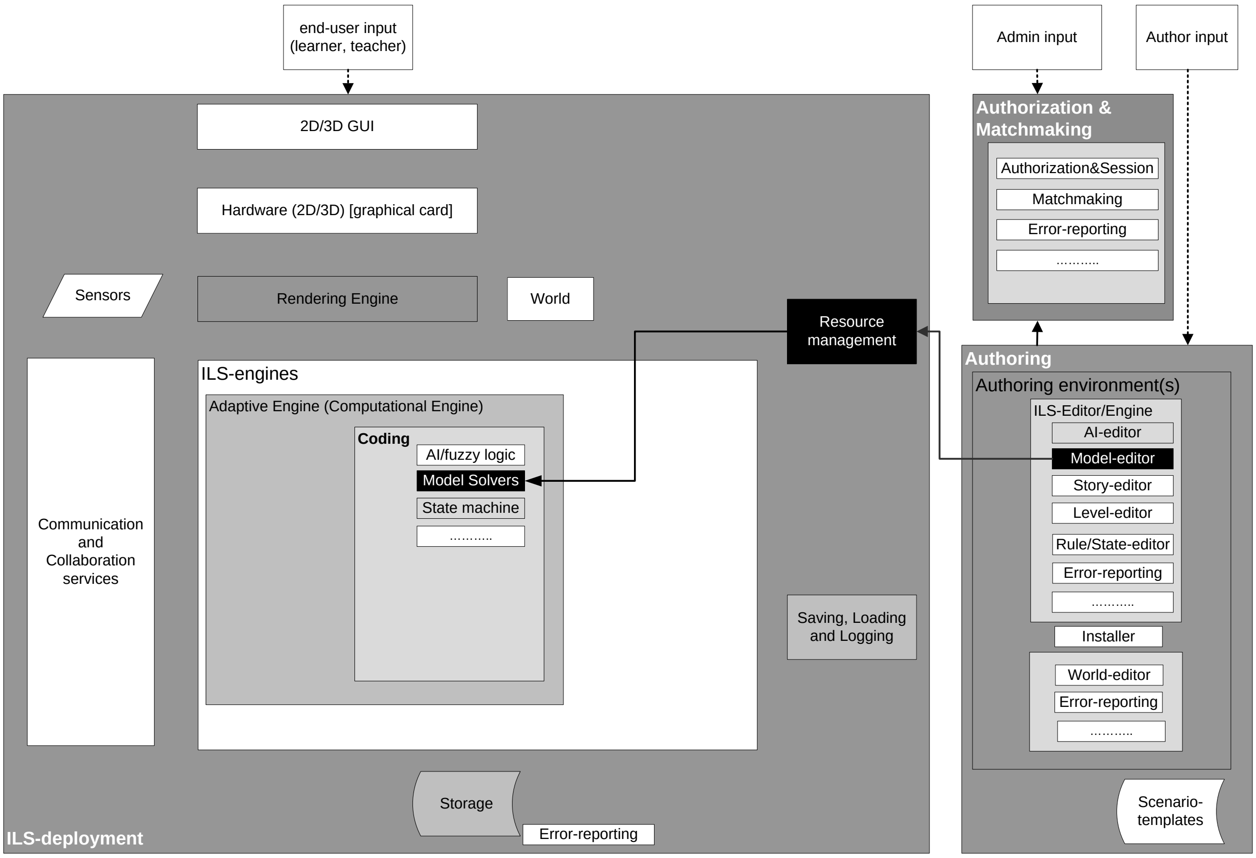

Figure 3. Example displaying salient relationships between the components Model-editor and Model-solvers in the multi-user ILS-architecture.

Figure 3 shows a more detailed ILS-architecture. For reasons of brevity and simplicity, only one authoring-component is illustrated and described in more detail: the ‘model-editor’. This example also clarifies the approach and possible issues at specific component-requirements. With the model-editor an author specifies the simulation-model(s) that learners can use. The model-editor results (i.e., resources) are transferred to the component ‘Resource management’, from which it will be consumed and interpreted by the component ‘model-solver’ in the ILS-engines-part. This can induce triggers at other ILS-engines (e.g., State-machine). Model solvers usually contain the core calculation model that could range from a complex set of (partial) differential equations down to simple calculations. Ideally such a component could solve an external set of (textual) formulas like in MathCAD®. Using a component to solve externally defined sets of equations prevents the need to fully program the model and makes model changes much easier. Fruitful approaches could be using Mathematica, .Net/Link of J/Link bridges between respectively .Net and Java, or to use Matlab Builder NE that compiles Matlab programs into .NET components. Alternatively, using discrete simulations will allow learners to participate more easily and to understand the underlying models through visualization. Unfortunately most of them (e.g., SimQuest, Powersim, and NetLogo) are stand-alone applications and are therefore not suited to be embedded within programming languages such as C# or Java enabling dynamic models.

2.3 Step 3: Selection-criteria multi-user ILS-architecture

As a third step in the method, literature, (generic) requirements, and about a dozen experts were consulted to derive a first set of selection criteria for a blueprint ILS-architecture using existing candidates. These criteria were subsequently presented to another small group of experts (n=6, equal number of software developers and instructional designers) at the Open University of the Netherlands (OUNL). They were asked to individually rank the criteria provided, and to mention some additional criteria themselves. Criteria are shortly described before presenting their ranking results.

a. supports multi-user ILS
ILS were typified by keywords mentioned in the last paragraph of Section 1. ILS for each domain (topic) should be feasible. This criterion equals the first aim of this study: development of multi-user ILS.
b. supports (personalized & contextualized) learner support
Learner support can be embedded, provided by peers or self (e.g., via monitoring, awareness). Strictly speaking, this criterion is covered by criterion (a), but as adequate learner support is seen as the educational glue of ILS it seemed worthwhile to mention it apart.
c. technical openness
Technical openness is dealt with by software-developers and indicates that the architecture is easy to integrate, and to adapt or extend by appropriate testing and debugging facilities, control-variables configuration, or user-management. Example: easy integration with existing VLE's/Learning Management Systems (LMSs). This criterion encapsulates the third aim of this study: enable research into ILS.
d. easy authoring
Authoring is primarily done by pedagogy experts (e.g. instructional designers and subject matter experts). They should be able to specify pedagogical adaptation rules, author simulations (either model, or world), have easy authoring-preview for multi-user scenario development and testing. This criterion equals the second aim of this study: swift and easy ILS-development.
e. user experience
The criterion of user experience was considered to involve (i) usability (easy to use by end-users), (ii) performance (acceptable), and (iii) scalability (till 50 end-users simultaneously) [(i), (ii) and (iii) are dependent]. This criterion confirms with the first generic requirement.
f. standards compliant
SOAP, REST, Web-based. Simple Object Access Protocol (SOAP) provides access to remote objects. SOAP provides the envelope for sending Web Services messages over the Internet. Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. This criterion confirms with the first and third generic requirement.
g. high sustainability
The architecture-technology should have a broad range of implementations and offer good support for software-developers and authors, for example through large and active communities (developers-authors), good documentation, workshops etcetera. There should be made use of proven technology (stability). Mentions further developments involving large and active communities/partners. This criterion is common sense.
h. development platform - technical framework
Preferably C#, not C, C++ or other less common programming languages. The target platform at least PC, preferably also PDA/Smartphone. This criterion confirms with the second generic requirement and shows a preference for somewhat easier and more common programming languages .
i. sources available (not necessarily open source)
Adequate resources are needed to assist software-developers understand the configuration, maintenance, and improvement. This criterion confirms with several generic requirements.
j. costs
Total cost of ownership: licenses, support, professionalization - recurring or not.

Respondents did not add other selection criteria. Table 1 presents group ranks of aforementioned criteria. This outcome might be representative for institutions already inclined to apply and research ILS in their educational offerings. However, other institutions in HE might use different weights or even different criteria because their ILS-policy and staffing is quite different from the OUNL.

Table 1. *Criteria for multi-user ILS-architecture and their ranks*
(1 = highest, 10 = lowest, draw: same rank numbers) (n=6).

Criterion	**Rank**
a. supports ILS	1 (+)
b. supports (personalized & contextualized) learner support	5
c. technical openness	2 (+)

d. easy authoring	2 (+)
e. user experience	5
f. standards compliant	10
g. high sustainability	4 (+)
h. development platform- technical framework	7
i. sources available (not necessarily open source)	9
j. costs	8

Notes: Top four ranked criteria are indicated by (+). These are used for a second sift of candidates (step 5).

2.4 Step 4: Search candidates and first sifting

A first sifting of candidates enabled us to remove the technically non-viable candidates. Documentation on websites, manuals, and studies were used to search and sift candidates for this fourth step (e.g. De Freitas, 2008 and Appendix A). Installing and testing all candidates (over 500) was clearly unfeasible. It was decided that candidates were technically non-viable if there (i) was not yet a 1.0-version (ii) was not a recent question-answering discussion, (iii) was an outstanding unanswered question for a long time, (iv) were no recent bug fixes, or (v) was made use of an atypical – seldom used – programming language. Furthermore, it seemed that several candidates had been acquired by other companies. Such candidates were also judged as technically non-viable, although they would better be qualified as 'not existing anymore'. Finally, if candidates could only cover a very small part of the blueprint, it seemed too time-consuming and costly to make them technically viable.

2.5 Step 5: Mapping candidates on blueprint and second sifting

Remaining candidates from step 4 (about 90) were evaluated on (a subset of) the four highest ranked criteria (step 3), again using aforementioned documentation to hold them against the blueprint (step 2).

Table 2. *Candidates that cover several parts of ILS-architecture after second sift.*

Name of Alternative	Crit. 1	Crit. 2	Crit. 3	Crit. 4	Fit on BP
Unity (http://unity3d.com)	+	-	+	++	o
SecondLife-OpenSim www.secondlife.com http://opensimulator.org/wiki/Main_Page	+	-	+	++	o
Caspian Learning www.caspianlearning.co.uk	o	+	--	o	-
Blender (www.blender.org)	+	-	+	+	-
3D-Gamestudio www.3dgamestudio.com	+	-	+	+	-
DX-studio www.dxstudio.com	+	-	+	+	-
EMERGO www.emergo.cc	+	+	+	o	o

Notes: Criterion 1 = supports ILS, Criterion 2 = easy authoring, Criterion 3 = technical openness, Criterion 4 = high sustainability.
Possible scores for each criterion :
-- (very low); - (low); o (neutral), + (good); ++(very good)
Possible scores for goodness-of-fit on Blueprint ILS-architecture (Fit on BP):
-- (<20%); - (20-40%), 0 (40-60%), + (60-80%); ++ (>80%)

Table 2 includes the seven remaining candidates after this second sift. These were chosen for a closer – at least technical – inspection (see Appendix for whole list).
Although no candidate had a satisfying score on goodness-of-fit on the blueprint, some combinations resulted in acceptable scores. Most candidates could not ideally cover the part 'authorization & matchmaking', but this can easily be covered by existing LMS. Figure 4 shows a mapping from EMERGO on the blueprint. Rastered rectangles indicate that a candidate not ideally covers components' functionality. Filled rectangles indicate that the candidate ideally covers – some parts – of components' functionality. The picture should not be seen in a strict sense in that filled rectangles

exactly cover the underlying sub components, but as a rough indication for the size of the part for which the underlying component is ideally covered. Figure 5 adds all other chosen candidates showing their strengths and weaknesses, and – maybe even more important – identifies the gaps which justify new software development for certain components. Unity and the SecondLife/OpenSim combination have similar functionality. The platforms 3D-Gamestudio and DX-studio have similar functionality and address a subset of functionality from the aforementioned duo. Taking the equal criteria scores from this quartet into account, 3D-Gamestudio and DX-studio are crossed out. Caspian Learning is crossed out as this cannot address multi-user ILS and hardly offers technical openness. Finally, Blender might offer additional functionality as compared to Unity and SecondLife/OpenSim.

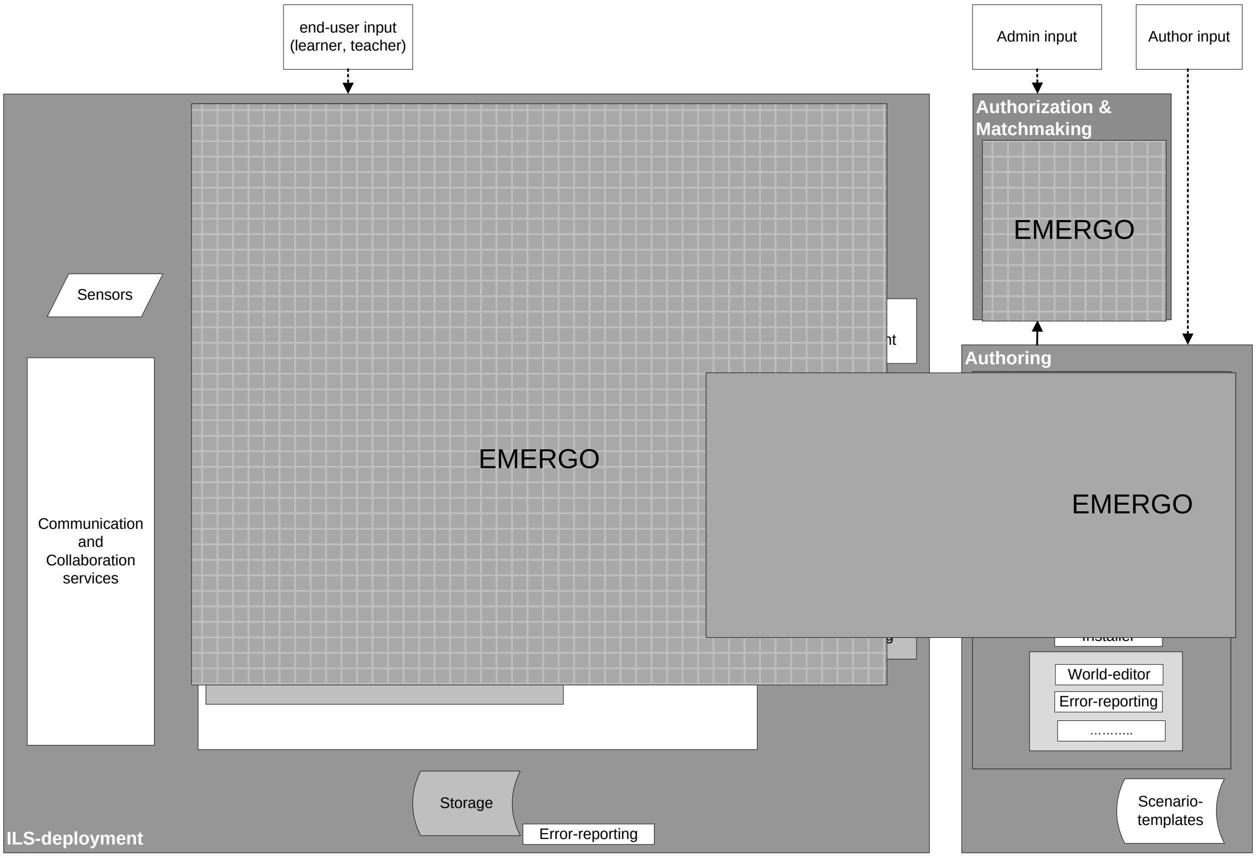

Figure 4. Mapping EMERGO onto the blueprint.

2.6 Step 6: Choose multi-user ILS-architectures

Figure 5 shows four promising combinations for ILS-architectures:

a. Unity, EMERGO
b. Unity, EMERGO, Blender
c. SecondLife/OpenSim, EMERGO
d. SecondLife/OpenSim, EMERGO, Blender

It also shows that some parts are not completely covered, especially within Authoring. Only EMERGO partly covers authoring, others are rather weak. This might not be surprising as EMERGO has been explicitly developed with easier authoring in mind (Nadolski et al., 2008). Furthermore, ILS-deployment-communication and ILS-engines can improve faster ILS-development.

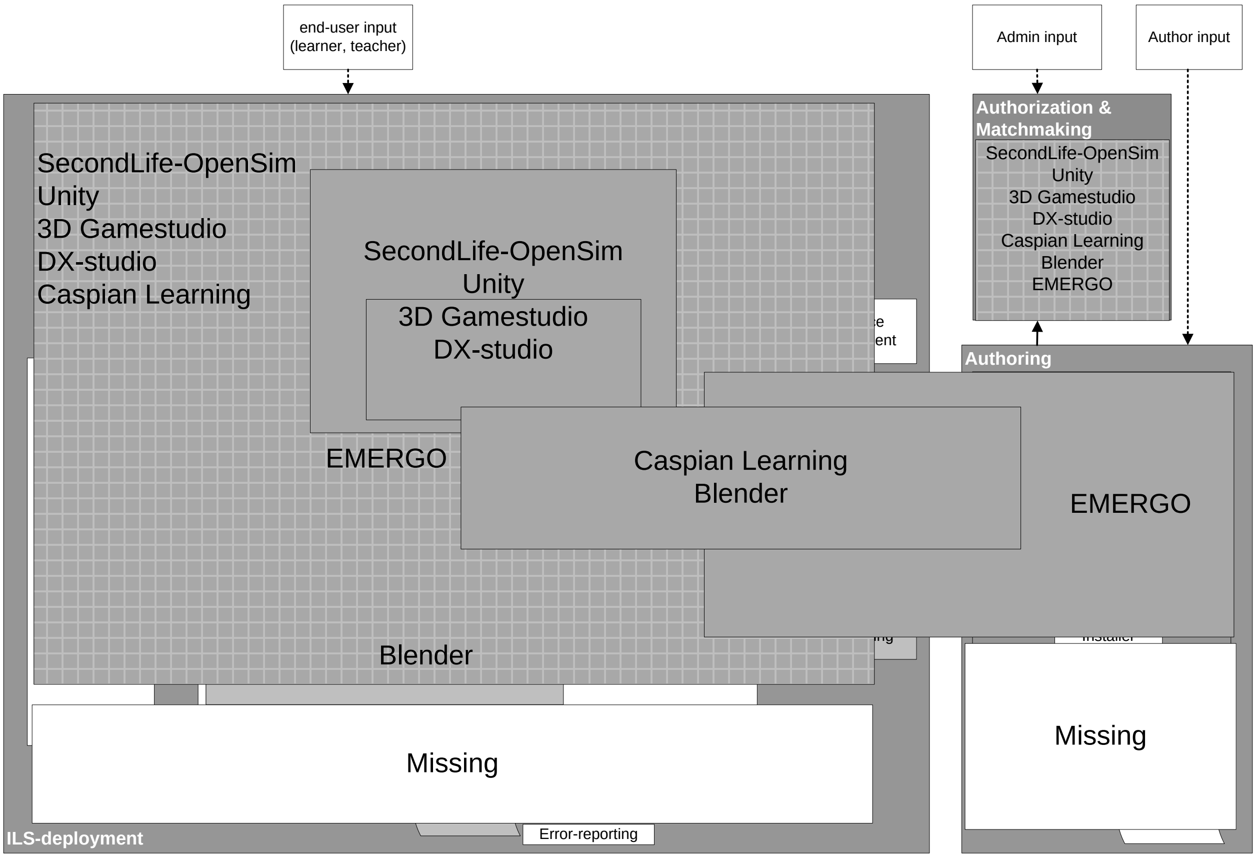

Figure 5. Mapping candidates onto the blueprint, expanding Figure 4 with all other candidates from second sifting and identifying some gaps (i.e., 'Missing').

3 Conclusion and discussion

This study followed the first six steps of the method described, showing the potential of components-based architectures for both swift development of multi-user immersive learning scenarios and enabling ILS-research. After applying the characteristics of multi-user ILS to determine requirements for multi-user ILS-architectures (step 1), such requirements were used to identify a blueprint for a multi-user ILS-architecture (step 2). Ten selection criteria for multi-user ILS-architectures were identified (step 3). We used these criteria for a first sifting of candidates by using documentation on websites, manuals, and studies about the candidates, removing technically non-viable candidates (step 4). The four most important selection criteria, aforementioned documentation, and the blueprint were used to evaluate the remaining candidates for a second sifting using a more detailed inspection by actually installating and testing them (step 5). Preferred candidates were required to cover all the components of the blueprint. While mapping candidates onto blueprint components, interchangeable alternatives but also several gaps were identified. Either because no candidate did meet the criteria, or – even worse – because there were no easy to integrate candidates for some components at all. In the final step, four most promising combinations for multi-user ILS-architectures were identified: (a) Unity, EMERGO, (b) Unity, EMERGO, Blender, (c) SecondLife/OpenSim, EMERGO, (d) SecondLife/OpenSim, EMERGO, Blender (step 6).

Although current solutions are not yet ideal, they provide a proof of concept to guide further articulation of a flexible multi-user ILS-architecture. A follow up to the current study is needed to gradually develop ILS-examples (see Figure 1). Such examples can be used for extending and improving the multi-user ILS-architecture and could ultimately lead to a single multi-user ILS-architecture. Alternatively, one could first aim at a single multi-user ILS-architecture by performing steps 7 and 8 (Phase II). Furthermore, research pilots focussing on multi-user ILS learning outcomes where learners will use such ILS-examples are needed (Phase III). The outcomes of such design-based research will influence further development of the multi-user ILS-architecture. In the future new and easy to embed technological developments (like smart objects, emotion detection, speech-to-speech translation) will be incorporated in these architectures.

A first limitation of the current study is that it will not lead to longstanding and generic solutions. On the one hand development of software for multi-user ILS-architectures will continue whereas on the other hand future societal-technical developments are somewhat unpredictable. In our view, a solution should have a minimal lifespan of 3-years to justify investment in development, licences and training costs. It is recommended to use various means for scouting ILS- software development (see Appendix) and consult forecasts by main trend watchers (e.g. Gartner, The New Media Consortium).

A second limitation of the study is that its method and main findings might be partly biased by its respondents (all from OUNL), their context and view on multi-user ILS. However, the method can be repeated and applied by others taking their context and view on ILS into account. In fact, the main findings from step 6 are regarded less important than the way in which they were derived. When applying the method from this study it is recommended to involve different stakeholders (e.g., game and software developers, game and instructional designers, researchers, teachers and trainers, learners and players) as this offers an excellent way of exchanging their expertise and sharing costs in their division of work. A drawback might be that it takes considerable time to gain mutual understanding.

A third limitation of the study is that its method for sifting candidates might lightly favour commercial software over open source alternatives. Commercial software sometimes pays more attention to their promotion through documentation and helpdesks than open source software. However, the availability of documentation and helpdesks remains important when searching for easier and faster ILS-development, so as such their availability stays important when sifting alternatives. A recommendation for sifting is that one should not exclude open source alternatives too early because of sparser documentation than their commercial counterparts.

A final limitation for the findings in this study might be that its respondents were more or less familiar with EMERGO (its development started in 2006). Although it cannot be ruled out that this might have influenced their responses, the findings demonstrate that EMERGO could only partly meet respondents' own criteria and could only partly cover the blueprint. As mentioned earlier, it should be acknowledged that EMERGO explicitly aims at easy authoring. Other criteria for its development were quite implicit as it should serve many different Dutch educational institutes not specifically interested in or active in serious games development, but working in a broader HE area where acquiring higher order cognitive skills using IT solutions is key.

3.1 Maturity in ILS-development

ILS-design and development often require a multi-disciplinary approach. Especially game design and instructional design need to move ahead from some misconceptions in both disciplines (Shelton & Wiley, 2007; Gunter et al., 2008, Kebritchi & Hirumi, 2008). Creating ILS requires considerable balanced thought and planning at every stage of design and production to match media with appropriate content, integrate and intertwine content closely with game play, and support learning through carefully crafted feedback (Gunter et al, 2008; Law & Kickmeier-Rust, 2009; Harteveld, Guimarães, Mayer, & Bidarra, 2010). Developers of educational games contend that immersiveness might ask for more frequent adaptation than on a task by task level. ILS have a very strong focus on narrative, and authentic learning tasks are intricately embedded in the game's narrative. Reordering of

tasks would therefore result in a, most probably, implausible rearrangement of narrative plot elements. This needs further research as was conducted in the 80Days-project (www.eightydays.eu). Combining ILS-engines for adaptive and interactive digital storytelling seems necessary. This might also improve reusability of ILS-resources, an additional boost for minimizing ILS-development costs.
More and better researched frameworks and more mature easy to use tools will become available to support teachers in applying ILS as part of their daily pedagogical repertoire (Aldrich, 2010). Indeed, this might have a real impact on the current paradigm of organising formal learning, and finally take advantage of the available educational technology.This might lead to more effective and more efficient learning, lower costs and higher involvement of all stakeholders.

3.2 Suggestions for research of ILS-applications

The kind and amount of learner support in ILS needs optimization to warrant continuous flow. Few is known about how to reach a balance between structured activities and more open-ended explorations. In addition, the social power of ILS needs further exploration (see e.g., De Freitas et al., 2010). Research should focus on the rate and efficiency to which within ILS learned skills are transferred to practical situations. Although there is some face validity that doing something 'virtually' will bare relation to this in reality, there appears to be ample room for researching transfer-issues when applying knowledge and skills in different, not previously encountered settings (e.g., De Freitas, 2009).
ILS seem to offer a potentially powerful alternative for the acquisition of higher order skills in an effective, efficient and attractive way which justifies investment for improving ILS-architectures. One might have higher initial set up costs to enable the execution of activities virtually, but in the long term repetitive exploitation costs can be minimal (Aldrich, 2010). Indeed, the motive for using ILS could even be stronger when the initial set up costs could be dramatically reduced. Please note that ILS-deployment can involve some running costs but can also be a free 24/7 resource for learning.
A study by Bardon and Josserand (2009) shows that the pedagogical benefits are a prerequisite rather than the core motivation for adoption or implementation of games. This exemplifies that we should argue for multi-user ILS from a combination of various motives (both economic, pedagogical, and for evidence based research). We feel this would improve the chances for a major breakthrough of ILS. We are looking forward to provide our contribution in this direction, for instance as partner of the recently started Games and Learning Alliance (www.galanoe.eu).

4 References

Aldrich, C. (2005). *Learning by doing: the essential guide to simulations, computer games, and pedagogy e-learning and other educational experiences*. San Francisco, CA: John Wiley & Sons.

Aldrich, C. (2009). *The complete guide to serious games and simulations*. Somerset, NJ: Wiley.

Barab, S.A., Gresalfi, M., & Arici, A. (2009). *Why educators should care about games*. Educational Leadership, September 2009, 76-80.

Bardon, T., & Josserand E. (2009). Why do we play the games? Exploring institutional and political motivations. *Education + Training*, *51*(5/6), 460-474.

Cheal, C. (2009). Student perceptions of a course tought in Second Life (2009). *InnovateOnline*, *5(5).* Accessible on http://www.innovateonline.info/pdf/vol5_issue5/Student_Perceptions_of_a_Course_Taught_in_Second_Life.pdf (last accessed, May 17, 2010)

Clark, R.C., & Mayer, R.E. (2008). *E-learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multi-media Learning* (2nd edition). San Francisco, CA: John Wiley & Sons

Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. New York: Harper Perennial.

Dalgarno, B., & Lee, M.J.W. (2010). What are learning affordances of 3-D virtual environments. *British Journal of Educational Technology, 41*(1), 10-32.

De Freitas, S. (2008). *Serious Virtual Worlds: A scoping study*. Prepared for the JISC e-Learning Programme by SGI. Available at:

http://www.jisc.ac.uk/media/documents/publications/seriousvirtualworldsv1.pdf (last accessed, September 28, 2009)
De Freitas, S. (2009). *Research insights from the serious games institute: new models for applied research?* Available at: http://repository.alt.ac.uk/453/1/Webinar_serious_games_20090616_Sara_de_Freitas.ppt (last accessed, June 1, 2010)
De Freitas, S., Rebolledo-Mendez, G., Liarokapis, F., Magoulas, G., & Poulovassilis, A. (2010). Learning as immersive experiences: Using the four-dimensional framework for designing and evaluating immersive learning experiences in a virtual world. *British Journal of Educational Technology*, *41*(1), 69-85.
Dewey, J. (1997). *How We Think*, New York: Dover Publications.
Driscoll, M.P. (2005). *Psychology of learning for instruction*. USA: Pearson Education, Inc.
Gartner Inc., http://www.gartner.com/technology/home.jsp
Gee, J.P. (2003). *What video games have to teach us about learning*. New York: Palgrave.
Gee, J.P., & Levine, M.H. (2009). Welcome to our virtual worlds. *Educational Leadership, 66*(6), 49-52.
Gunter, G.A., Kenny, R.F., & Vick, E.H. (2008). Taking educational games seriously: using the RETAIN model to design endogenous fantasy into standalone educational games. *Educational Technology Research & Development, 56,* 511-537.
Harteveld, C., Guimarães, R., Mayer, I.S., & Bidarra, R. (2010). Balancing play, meaning and reality: The design philosophy of LEVEE PATROLLER. *Simulation & Gaming, 41*(3), 316-340.
Herrington, J., Oliver, R., & Reeves, T.C. (2003). Patterns of engagement in authentic on line learning environments. *Australian Journal of Educational Technology, 19*(1), 59-71.
Hu, W. (2010). A common software architecture for educational games. In Zhang et al. (Eds.): *Edutainment 2010* (pp. 405-416), LNCS 6249.
Hummel, H.G.K., van Houcke, J., Nadolski, R.J., van der Hiele, T., Kurvers, H., & Löhr, A. (in press). Scripted collaboration in serious gaming for complex learning: effects of multiple perspectives when acquiring water management skills. *British Journal of Educational Technology*.
Johnson, L., Levine, A., Smith, R., & Stone, S. (2010). *The 2010 Horizon Report.* Austin, Texas: The New Media Consortium.
Kapp, K.M, & O'Driscoll (2010). *Learning in 3D: adding a new dimension to enterprise learning and collaboration.* San Francisco: Pfeiffer.
Kebritchi, M., & Hirumi, A. (2008). Examining the pedagogical foundations of modern educational games. *Computers and Education, 51*, 1729-1743.
Killi, K. (2005). Digital game-based learning: Towards an experimental gaming model. *Internet and Higher Education 8(3),* 13-24.
Kolb, D.A. (1984). *Experiential learning: Experience as a source of learning and development*. Englewood Cliffs, NJ: Prentice-Hall.
Law, E. & Kickmeier-Rust, M.. *80 Days: Immersive Digital Educational Games with Adaptive Storytelling.* Available at: http://www.eightydays.eu/80days-goes-public/scientific-publications.html (last accessed 21 September 09)
McFarlane, A., Sparrowhawk, A., & Heald, Y. (2002). *Report on the educational use of games.* Cambridge: Teachers Evaluating Educational Media.
McLellan, H. (2004). Virtual realities. In D.H. Jonassen (Ed.). *Handbook of research for educational communications and technology* (pp. 461-497). Mahwah, NJ: Lawrence Erlbaum.
Michael, D. & Chen, S. (2006). *Serious games: Games that educate, train and inform*. Boston, MA: Thomson.
Moreno-Ger, P., Burgos, D., Martínez-Ortiz, I., Sierra, J.L., & Fernández-Manjón, B. (2008). Educational game design for online Education. *Computers in Human Behavior, 24*(6), 2530-2540.
Nadolski, R. J., Hummel, H. G. K., Van den Brink, H. J, Hoefakker, R., Slootmaker, A., Kurvers, H., & Storm, J. (2008). EMERGO: methodology and toolkit for efficient development of serious games in higher education. *Simulations & Gaming, 39*(3), 338-352.
Nelson, B.C. et al. (2007), Robust Design Strategies for Scaling Educational Innovations: The River City Case Study. In B.E. Shelton and D.A. Wiley (Eds.), *The Design and Use of Simulation Computer Games in Education*. Rotterdam, The Netherlands: Sense Publishers.

Nowak, K., & Biocca, F. (2004). The effect of agency and anthropomorphism on users' sense of telepresence, copresence, and social presence in virtual environments. *Presence: Teleoperators & Virtual Environments*, *12*(5), 481-494.
Ritterfeld, U., Cody, M., & Vorderer, P. (2009). *Serious games: mechanism and effects*. New York, NY: Routledge.
Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, *55*(1), 68-78.
Salen, K., & Zimmerman, E. (2004). *Rules of play*. Cambridge, MA: MIT Press.
Scopes, L.J.M. (2009). *Learning Archetypes as tools of Cybergogy for a 3D Educational Landscape: A structure for eTeaching in Second Life*. University of Southampton. (accessible from http://eprints.soton.ac.uk)
Shelton, B.E., & Wiley, D.A. (2007). *The design and use of computer games in education*. Rotterdam, The Netherlands, Sense Publishers.
Steuer, J. (1992). Defining virtual reality: Dimensions determining telepresence. *Journal of Communication, 42*(4), 73-93.
Stone, R.J. (2008). *Human Factors Guidelines for Interactive 3D and Games-Based Training Systems Design*. Available at: http://www.hfidtc.org/pdf/HFIDTC-2-8.7.2-1-HF-Guidelines-for-SG.pdf (last accessed, June 1, 2010).
Wallace, P., & Marriot, J. (2009). The impact of avatar self-representation on collaboration in virtual worlds. *Innovate: Journal of Online Education*, *5*(5). Available at: http://www.innovateonline.info/index.php?view=article&id=689 (last accessed May 29, 2009).
Wilson, K.A., Bedwell, W.L., Lazzara, E.H., Salas, E., Shawn Burke, C., Estock, J.L., Orvis, K.L, & Conkey, C. (2009). Relationships between game attributes and learning outcomes: Review and research proposals. *Simulation & Gaming*, *40*(2), 217-266.
Wood, E.J. (2004). Problem-based learning: Exploiting the knowledge of how people learn to promote effective learning. *Bioscience Education, 14*(3). Accessible on http://www.bioscience.heacademy.ac.uk/journal/vol3/beej-3-5.aspx (last accessed, June 1, 2010)

Biographical notes

All authors work at the Centre for Learning Sciences and Technologies (CELSTEC) of the Open University of the Netherlands, and are currently involved in projects developing immersive learning scenarios.

Rob J. Nadolski is assistant professor technology enhanced learning with a focus on effective competence-based education and training using innovative educational technologies such as serious games, more specifically addressing improved learner support via technology means (like sensors).

Hans G. K. Hummel is associate professor technology enhanced learning with a research focus on effective workplace-based education using innovative educational technologies such as serious games and communities of professional practice

Aad Slootmaker is senior software developer and his main activities concern the technical design and development of interactive multimedial computer programs and serious games as well as consultancy for VLE's/PLE's.

Wim van der Vegt is senior software developer and his main activities concern the technical design and development of interactive multimedial computer programs and serious games as well as consultancy for VLE's/PLE's.

Appendix – Examined alternatives for second sifting in Step 5

About 500 alternatives were examined on their suitability for inclusion in the blueprint ILS-architecture by applying the four criteria ranked highest (Step 3). Those alternatives were selected from searches in databases of scientific literature and by searching the Web. We used several combinations of search terms, like “architecture”, “games”, “serious games”, “video games”, “virtual worlds”, “game engine”, “game platform”, “game framework”, “educational games”, “instructional games”, “learning games”, ”social games”, “systems”, ”edutainment”, “simulations”, “simulation engine”, “modelling”. Furthermore, visits at various (inter)national conferences addressing serious games, informal talks, attending demo’s, and personal experience with some alternatives could sometimes complement the information from the searches. Additionally, several popular books and magazines about (serious) games development were skimmed for alternatives. Finally, (inter)national mailing lists and newsgroups on serious games were consulted for finding alternatives. Documentation on websites, manuals, and studies were used to examine candidates. Documentation that included rubrics as (a) listed functions and features, (b) demonstrators, and (c) user experiences (from various stakeholders) enabled a faster decision than documentation that missed one or more of such rubrics. In a couple of cases, documentation was very sparse or even completely missing. If there was no documentation, the alternative was excluded for further examination as these alternatives were regarded highly unlikely to pass a second sifting round. Please note that the downsides of this approach are mentioned in the discussion section.

Only 90 of all alternatives are included in this appendix as the others had at least one ‘not applicable score’ on one the four criteria. It was not always possible nor needed to score all four criteria. Sometimes, information was missing to apply a criterion, or the application of one criterion already indicated that other - already evaluated and similar - candidates were clearly superior to the one under evaluation (indicated by *). Each candidate was examined by minimally two reviewers (one technical, one pedagogical).

Notes:
Criterion 1 = supports ILS; Criterion 2 = easy authoring; Criterion 3 = technical openness
Criterion 4 = high sustainability
Possible scores for each criterion :
not applicable (n.a.) ; -- (very low); - (low); o (neutral), + (good); ++(very good); ? (unknown)

Possible scores for goodness of fit on the BluePrint ILS-architecture (Fit on BP):
not applicable (n.a.) (not functional or not in use anymore); -- (<20%); - (20-40%), 0 (40-60%), + (60-80%); ++ (>80%)

Name of Alternative	Crit. 1	Crit. 2	Crit. 3	Crit. 4	Fit on BP
Active Worlds www.activeworlds.com	+	--	+	+	o
Active Worlds Education UniV. www.activeworlds.com/edu/awedu.asp	+	--	+	+	o
Adventure Game studio http://www.adventuregamestudio.co.uk/	o	o	+	+	-
AgentAttack http://www.agent-attack-it.de/					n.a.
Ataris Virtual World www.mcvuk.com/interviews/9/Ataris-Online-Revolution	o	?	--	+	--
Areae-Metaplaces www.areae.net	o	o	?	--	o

Axiom 3d http://axiomengine.sourceforge.net/	o	--	+	o	-
Baja Engine http://www.bajaengine.com/	o	--	o	-	--
Blender http://www.blender.org	+	-	+	+	-
Blitz3D, BlitzMax http://www.blitzbasic.com/	+	-	o	+	-
Box2D http://www.box2d.org/	o	--	+	-	--
3DCakeWalk http://www.3dcakewalk.com/	o	-	-	-	-
C4 Engine http://www.terathon.com/c4engine/index.php	o	o	o	+	-
Caspian Learning http://www.caspianlearning.co.uk/	o	+	--	o	-
Citypixel.com www.citypixel.com	o	o	?	o	-
Club Penguin www.clubpenguin.com					n.a.
Croquet project www.opencroquet.org	+	--	o	+	o
Cybertown www.cybertown.com	+	?	o	+	o
Cyworld www.cyworld.com					n.a.
DarkbasicPro http://wiki.gamedev.net/index.php/DarkbasicPro	+	-	+	+	o
Delta 3D http://www.delta3d.org/)	+	-	o	o	o
DreamSpark https://www.dreamspark.com/default.aspx	o	-	o	+	o
DXFXNA http://dxframework.org/wiki/index.php/DXFXNA	o	-	+	-	-
DXFramework http://dxframework.org/	o	-	+	-	-
DX Studio http://www.dxstudio.com	+	-	+	+	-
E76 Sim Engine http://www.capricorn76.com/	+	-	+	-	-
e-Adventure http://e-adventure.e-ucm.es/lang.php?lang=en	+	o	+	o	o
Elemental Engine http://www.phatyaffle.com/index.php?option= com_content&task=view&id=91&Itemid=110	+	?	?	-	o
EMERGO www.emergo.cc	+	+	+	o	o
Entropia Universe www.entropiauniverse.com	+	?	--	o	o
Extreme Optimization Mathematics Library	+	--	+	-	--

http://www.extremeoptimization.com/ Mathematics/Default.aspx					
Forterra Systems OLIVE www.forterrainc.com	+	o	-	+	o
Gaia Online www.gaiaonline.com	o	-	-	+	o
Game maker http://www.game-maker.nl/	+	o	+	+	o
3DGameStudio http://www.3dgamestudio.com/)	+	-	+	+	-
Great Northern World http://mdm.gnwc.ca/	+	?	+	o	o
Habbo www.habbo.com	o	-	-	+	o
Hipihi www.hipihi.com/index_en.html	+	o	-	-	o
Horde3D http://www.horde3d.org/)	+	o	?	o	o
Irrlicht http://irrlicht.sourceforge.net/	+	o	+	++	-
IMVU www.imvu.com	+	-	-	?	o
INNOV8 http://www-01.ibm.com/software/solutions/soa/innov8.html	+	o	+	+	o
JmonkeyEngine http://jmonkeyengine.com/	+	-	o	++	-
Jogre http://jogre.sourceforge.net/	-	-	+	+	-
Kaneva www.kaneva.com	+	-	+	o	o
Lego Universe http://universe.lego.com					n.a.
Media Machines www.vivaty.com	+	-	o	o	o
Media Grid: Immersive Education http://immersiveeducation.org					*
MetaVerse http://metaverse.sourcefourge.net	+	-	+	+	o
Microsoft Virtual Earth www.microsoft.com/virtualearth					n.a.
moove online www.moove.com	o	-	o	+	-
MultiVerse www.multiverse.net	+	-	+	+	o
Musiclounge www.themusiclounge.com					n.a.
NASA World Wind http://worldwind.arc.nasa.gov					n.a.
Nicktropolis					n.a.

www.nick.com/nicktropolis					
Ogoglio www.ogoglio.com	+	o	+	--	o
OGRE http://www.ogre3d.org/	o	-	+	++	-
OpenSim http://opensimulator.org/wiki/Main_Page	+	-	+	++	o
PhyreEngine http://www.phyreengine.com/	+	-	-	+	-
Physics2D http://physics2d.googlepages.com/home	o	--	+	-	--
Play station home http://en.wikipedia.org/wiki/Play station_Home					n.a.
Project DarkStar http://www.projectdarkstar.com/					n.a.
Protosphere www.protonmedia.com	+	o	-	+	o
Pulse!! http://elianealhadeff.blogspot.com/2008/06/pulse-update-serious-games-improving.html)	+	o	o	o	o
Quake Engines http://www.idsoftware.com/	o	-	+	-	-
Quake - IdTech2/3	o	-	+	-	-
qwaq forums www.quack.com					n.a.
RAGE http://en.wikipedia.org/wiki/Rockstar_Advanced_Game_Engine					*
RealmForge crafter http://en.wikipedia.org/wiki/RealmForge					n.a.
SAGE engine http://en.wikipedia.org/wiki/SAGE_engine	o	-	-	--	-
Sauerbraten http://sauerbraten.org/	+	-	+	o	-
Second Life www.secondlife.com	+	-	+	++	o
SimQuest http://www.simquest.nl/	+	o	-	+	-
SmartFoxServer http://www.smartfoxserver.com/overview.php	o	-	+	+	--
Spring http://spring.clan-sy.com/	+	-	+	+	-
Teen Second Life http://teen.secondlife.com					n.a.
there.com www.there.com	+	o	-	+	o
Torque http://www.garagegames.com/products/torque-	+	-	+	o	o

3d					
Truevision3D http://www.truevision3d.com/	+	-	+	+	o
Unigine http://www.unigine.com/	+	-	o	+	o
Unity http://unity3d.com/)	+	-	+	++	o
UnReal-3 http://www.unrealtechnology.com/technology.php	+	-	-	o	o
Uni Verse www.uni-verse.org					n.a.
Unype www.unype.com					n.a.
Virtual Heroes ALT Platform http://www.virtualheroes.com/index.asp	+	o	o	+	o
Visual3D.net http://www.visual3d.net/)	+	o	-	+	o
Whyville www.whyville.net					n.a.
WILL interactive http://www.willinteractive.com/	+	?	--	+	+
(Open) Wonderland https://lg3d-wonderland.dev.java.net now: www.openwonderland.org	+	-	+	o	o