Realistic Immersive Virtual Agent-Based Learning Environment (RIVALE)

Case Study for Technology Enhanced Learning of Requirements Elicitation Skills

Master's Thesis at

Graz University of Technology

submitted by

Pirmin Riedmann

Supervisor: Univ.-Doz. Dipl.-Ing. Dr.techn. Christian Guetl

Graz University of Technology

Co-Supervisor: Dr. John Venable

Curtin University, School of Information Systems

Institute for Information Systems and Computer Media (IICM)

Graz University of Technology

A-8010 Graz, Austria

Graz, January 2014

Realistische Immersive Virtuelle Agentenbasierte Lernumgebung

Fallbeispiel für Technologie gestütztes Lernen von Fähigkeiten in der Anforderungserhebung

Masterarbeit an der
Technischen Universität Graz
vorgelegt von

Pirmin Riedmann

Betreuer: Univ.-Doz. Dipl.-Ing. Dr.techn. Christian Gütl
Technische Universität Graz

Mitbetreuer: Dr. John Venable
Curtin University, School of Information Systems

Institut für Informationssysteme und Computer Medien (IICM)
Graz University of Technology
A-8010 Graz, Austria

Graz, Jänner 2014

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…… ……..

date (signature)

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am …… ……..

(Unterschrift)

Abstract

Computer supported learning has been an active field for decades but also has a long history of mistakes. There is a wide range of approaches all of which focus on problems in teaching knowledge, from as simple approach as providing a wiki to newer approaches like 3D virtual learning environments (3DVLEs), where further research is required. The latter approach (3DVLE) is very encouraging, making it possible to simulate situations which are hard or even impossible to realize in the real world because of time, cost or safety reasons. Further, such systems could be accessed from different locations at different times, depending on the setup.

This thesis presents research to develop and evaluate a 3D virtual learning environment, developed to offer collaborative learning and practicing sessions in groups on requirement elicitation techniques. Classes on requirement elicitation mainly offer a lecture on theoretical aspects and some abstracted exercises. A new system was needed to overcome problems of teaching some of these techniques, like interviewing or document reviewing. It was important to find a way to simulate, motivate and teach in a way where students can learn by themselves, with as little input from a supervisor as possible and in a realistic way.

To overcome the issues mentioned above, a 3D virtual learning environment was designed to simulate a realistic fictional case study, called *Carol's Corner Store,* in which students can move, explore, and take actions to complete their learning tasks. This research also aims to implement the realistic training world with game elements to further motivate the students and also elaborates on the requirement elicitation techniques used to accomplishing the learning activities. It also provides a deeper look into the design and implementation of the different modules, which had to be developed and extended for the selected 3DVW framework *Open Wonderland,* to meet the requirements. Open Wonderland was chosen after evaluating different 3D virtual environments which could be used for educational reasons. This thesis points out the reasons for this and presents the designed world. It has three main areas: (1) a tutorial area, where people learn how to interact and use the client program, (2) a representation of an office complex, where students obtain their assigned tasks, as well as read more about requirement elicitation and (3) Carol's Corner Store, where students use the learned techniques.

A big part of this research revolves around chatbots, which are used in the project as an interview simulation system. Different chatbots systems were evaluated and *Pandorabots* was determined as the best solution, providing free online bot hosting. *Pandorabots* hosts *Artificial Linguistic Internet Computer Entity (A.L.I.C.E.)* implementations, using *Artificial Intelligence Markup Language (AIML)* files and has freely available knowledge bases and free AIML edit possibilities.

A first evaluation was conducted with teachers and students who recently finished a system analysis class to learn from their experience. It was conducted with (1) preliminary questionnaires, (2) a session of about two hours with observation, (3) post-questionnaires, and finally with (4) an

interview with the participants. The valuation showed that most users liked the world's look-and-feel, had a lot of fun with the prototype, and got motivated from different aspects. The evaluation also brought up different problems with the Chabots and OpenWonderland, which future versions will have to solve and about which more research still has to be continued. However, all participants were sure that this kind of program will improve learning and teaching for students as well as for supervisors.

Kurzfassung

Das Forschungsgebiet des computerunterstützten Lernens ist schon seit Dekaden aktiv, aber genau so lang ist die Geschichte an Fehlern, die gemacht wurden. Es gibt eine große Bandbreite an möglichen Lernmethoden, von einfachen Ansätzen wie Wikis bis hin zu virtuellen 3D Welten. Speziell die virtuellen 3D Welten sind vielversprechend. Diese machen es möglich Situationen/Szenarien zu simulieren, welche in der realen Welt auf Grund von Zeit, Kosten und Sicherheitsgründen nicht durchzuführen wären. Weiterer Vorteil ist, dass solche Systeme zeitunabhängig sind und man von verschiedenen Standorten darauf zugreifen kann, abhängig vom Setup.

Diese Thesis beschreibt den Forschungsablauf und den Testvorgang von einer virtuellen 3D Lernumgebung. Entwickelt zur Bereitstellung von gemeinschaftlichem Lernen und Üben in Gruppen von Requirement Engineerings Techniken. Die herkömmliche Lernmethode ist eine Kombination aus reiner theoretischer Wissensvermittlung und abstrakten Übungen. Eine Aufgabe der Forschung war es bekannte Probleme dieser Lernmethode zu minimieren oder komplett zu entfernen. Wichtig war es Lösungen zu finden, Lernsituationen zu simulieren, Studenten zu motivieren und ebenso ihnen einen Weg zu zeigen, wie sie mit geringem Aufwand von Professoren lernen können und das so realistisch wie möglich.

Aus zuvor genanntem Grund wurde eine 3D Lernumgebung entworfen, die eine realistische Umgebung darstellen soll, in der sich die Benutzer frei bewegen können und an einer fiktiven Case Study teilnehmen, in diesem Falle bei „Carol's Corner Store". Diese Forschungsarbeit zielt darauf ab eine realistische Trainingsumgebung zu erstellen, die mit Spielelementen unterstützt wird. Diese Umgebung soll Motivationsaspekte für die Teilnehmer bieten und zur Verbesserung der verwendeten RE Techniken beitragen. Die Design- und Entwicklungsschritte der verschiedenen Module werden genau beschrieben, außerdem, welche neu entwickelt und erweitert wurden aus dem gewählten 3DVW Framework. Nach der Auswertung von verschiedensten, virtuellen 3D Umgebungen hat sich das kostenlose Open Wonderland für die Vermittlung von Lerninhalten als beste Lösung herausgestellt. Die Thesis weist die Gründe auf für diese Entscheidung und präsentiert die designte Welt. Es gibt drei Gebiete im Programm. (1) Eine Startwelt, in der die Benutzer eine Einführung in die Welt und den Client bekommen. (2) Eine Welt, in der Aufgaben gegeben und weitere Informationen über die fachlichen Inhalte erlesen werden können. Diese wird hier in einem Büro dargestellt. (3) Und Carol's Corner Store in der der Benutzer die gelernten Techniken im anwenden.

Ein wesentlicher Bestandteil der Forschungsarbeit sind Chatbots, die im Projekt als Interviewsimulationssystem verwendet werden. Hierbei überzeugte Pandorabots, welches gratis hosting von Onlinebots anbietet. Pandorabots verwendet die Artificial Linguistic Internet Computer Entity (A.L.I.C.E.) als System, welches Artificial Intelligence Markup Language (AIML) Files einsetzt. Als weiteren Service bietet die Seite gratis Bot-Wissensdatenbanken und gratis AIML-Editor-Möglichkeiten an.

Die ersten Tests wurden mit Professoren und Studenten durchgeführt, welche im Gebiet tätig sind oder kürzlich erst entsprechende Fächer abgeschlossen haben, um möglichst viel von ihrer Erfahrung zu lernen. Es wurde ein (1) vorbereitender Fragebogen eingesetzt, gefolgt von einer (2) zweistündigen Programmtestphase mit Benutzerbeobachtung, mit anschließendem (3) Fragebogen und zuletzt einem (4) Interview. Die meisten Benutzer mochten das Aussehen und die Handhabung, hatten viel Spaß mit dem Prototyp und wurden durch verschiedene Aspekte motiviert. Die Tests brachten aber auch verschiedenste Probleme mit den Chatbots und Open Wonderland zu Tage. Auf jeden Fall waren sich alle Teilnehmer einig, dass diese Art von Lernprogramm das Lernen verbessert, für Studenten aber auch aus Sicht der Lernbeauftragten.

Acknowledgement

I would like to thank first my supervisor Univ.-Doz. Dipl.-Ing. Dr.techn. Christian Gütl, without whom this work would not be possible. It was his engagement and his international connections that brought me to Australia and to this thesis. The project would not have success without the effort, time and motivation he put in along the way. Thank you very much for making this all possible.

Further I would like to express my deep gratitude to my co-supervisor Dr. John Venable, who is researching at the Curtin University in Australia. He came up with the idea for this project. I really want to thank him for all his support and his patience throughout the entire time and I know that I was sometimes a lot of work for him, so I am even more thankful for all the ongoing input and effort he put into this work. Every Master student can only dream to get such a supervisor.

I also would like to thank Dr. Vanessa Chang who was my second co-supervisor. I am grateful for all the conversations, the support and the input she gave me even while she had tons of other work to do. It was more than helpful!

Special thanks also to the whole department “School of Information Systems”, especially Julia Kivuyo who helped me a lot with all kind of problems I had besides the project and Dr. Thorsten Reiners for our chats and discussions on the topic. I thank everyone there, for a very helpful, welcome and great environment and the chance to be a part of an international research project.

Finally, I also want to thank very much my family who made all this possible and my girlfriend Janine who helped and supported me on each single step on this way.

Contents

1 Introduction

The world was always changing, but since the introducing of the World Wide Web, human life went through a, until then, unknown speed of change. The WWW made the world a smaller place and paved the way for vast amounts of different applications becoming an integral part of our life. Two of these new great opportunities are gaming and eLearning. One of the big challenges is to combine these two and use the advantages of both to improve learning and teaching methods for the new "internet generation", also called Generation Y. This Gen Y is surrounded by wired and wireless technologies and technology is part of their life (Chang & Gütl, 2010). Therefore the outcome of gaming and E-Learning put together are serious games, which aim to make not only learning as interactive and diversified as possible, but also to practice skills and knowledge. It is no surprise that over the years, education researcher have sought to understand how important games are for education and how to make education more effective with the help of games (Looi, 2005). Games and 3D virtual world simulations have become a huge industry (Looi, 2005), with about 25$ billion revenue in 2011 (ESA, 2013) and virtual 3D worlds are more and more dominating the gaming world. Games like World of Warcraft, Minecraft, Heroes of Newerth or Diablo 3 are dominating gaming charts (DFC Intelligence, 2013), but also more open 3D virtual worlds have success stories, such as Second Life which increased the registered number of users from 100,072 to 2,251,416 in only twelve months (Hebbel-Seeger et al., 2013). Therefore this project picks up such success factors and transform it in a new platform where teaching and learning is possible in a realistic way.

Picking up such success factors of gaming and E-Learning can be done with different approaches but only 3D virtual environment are able to overcome all the needs. As Schank (1997) stats: "*Failing in interesting ways should be a goal of training.*" Therefore, Open Wonderland an immersive and collaborative environment will be used to fulfil the high appetence for such an environment. OWL (Open Wonderland) is built with a module based system. It is comparably easy to expand the environment with new modules and supports already many interaction and collaboration functionalities such as in world chat, PDF-viewer, voice chat or whiteboards. The idea now is to offer a platform for higher education by simulating a realistic behavior and to develop a reactive system, putting it all together in a 3D virtual environment and adding some game elements for motivation and a better learning out come.

There are hundreds of topics which would need and fit fur such a platform, but in a first prototype version only system analysis, exactly requirement elicitation (RE) is operated, which heavily uses different interviewing techniques (Sabahat, Iqbal, & Azam, 2010). In RE the old approach was to give all students printed documents and to let them extract the requirements and there are only time and labour intensive teaching and practicing methods.

One of the main techniques of RE is interviewing. To simulate this in the best way, a natural language communication simulation was needed. An artificial intelligence agent, also called chatbot, is a method to achieve such characteristics, what is used in this research to simulate and practice interviewing a customer. Chatbots, such as A.L.I.C.E bots (Artificial Linguistic Internet Computer Entity)**,** are using natural language to interact with users (Shawar & Atwell, 2007). A.L.I.C.E is using Artificial Intelligence Markup Language (AIML) to generate answers based on a pattern and template matching approach and also provide a conditional branching and supervised learning to produce new responses (Kerly et al., 2006). This chatbots will then be presented to the players by non-player characters to round off a realistic simulation.

1.1 Goals and Objectives

The main goal of the project is to set up a *3D Multi-User Virtual Environment (MUVE)*, developed to have collaborative learning and practicing session in groups on requirement elicitation techniques. The environment needs to be extendable as easy as possible to support an ongoing developing process. The main focus of the curriculum in the project is requirement elicitation, where one main technique is interviewing. This is simulated via artificial intelligence agents, able to communicate in natural language. Those so called chatbots are virtualized in different variations, by using ontology's (implemented with AIML) as knowledge source. The first prototype will support a MUVE where up to three teams of students can learn and practice the basics of requirement elicitation, supported by a teacher which can assist the users in the game at any time. The virtual environment will also provide game elements and will give a feeling of moving in a real world, as much as possible.

Research tasks include:

- Evaluating of different MUVE with educational background and focus.
- Simulate artificial intelligence and natural language processing with chatbots, using ontologies as knowledge sources and integrate them to Open Wonderland.
- Setting up an Open Wonderland MUVE which allows interaction with each connected user as well as working in teams on different tasks, in a user friendly and easy understandable environment. What includes an introduction (tutorial) area.
- Give the project a serious gaming look and feel. It motivates students with game based elements.
- Sets up different modules supporting the different kinds of interview techniques as well as the different pedagogical approaches in teaching new knowledge e.g. visualisation or repeating of learned knowledge.
- Modelling and setting up user friendly worlds and areas.

1.2 Methodology

This section illustrates used methodologies of the written part of the work as well as the implementation part and gives an overview of the structure.

This project follows a *Design Science Research* (DSR) approach. Design Science Research is aimed to solving some type or kind of problems and can be seen as a "Technology Invention". It should produce knowledge of a new solution to solve problems. DSR should produce constructs, models, methods, instantiations, and better theories. It also should produce a kind of guideline and advice for practitioners to help choosing among different competing solution and with technologies as well as with implementing their choice. It also has to provide clear, precise and complete statements of knowledge that other researcher can test it and enhance it (see also Figure 1). (Venable, 2006)

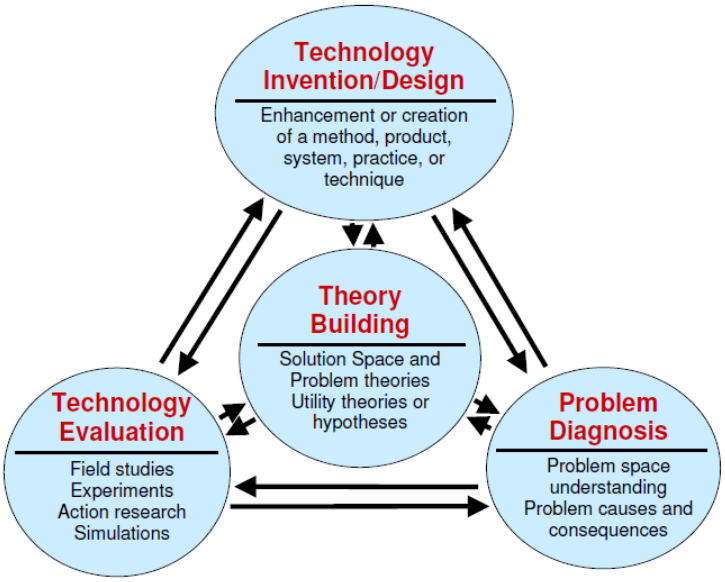

Figure 1: An Activity Framework for Design Science Research (Venable, 2006)

Venable (2006) stated: "*Theory should be a primary output and that theory and theorising need to play a central role in the advancement of Design Science Research*". He provides Muli-Method DSR Context Framework shown above, stating the theory building as core part. Thus, theorizing or theory building is a central activity. The theories may be modified many times during a single DSR project.

1.3 Structure

This thesis is structured in two main parts, the literature review and the implementation with the evaluation. It starts with the introduction and moves on to the main parts.

Chapter 2 will start with the application domain topic *Requirement Elicitation (RE),* defining terms like requirement, elicit and gather, followed by elicitation techniques, the process and will finish with related work and approaches used to teach RE. The chapter continues with virtual learning, were it starts with E-learning, focus into E-learning technologies and techniques. Furthermore, serious gaming will be introduced, followed by 3D virtual Worlds. In the end, bots, non-player-characters, chatbots and natural language processing is described.

Chapter 3 will introduce the requirements and design decisions. The chapter starts with the project vision and idea. Followed by a section identifying the requirements and lists different challenges the project has to face. The chapter continues on with the conceptual architecture naming the main stakeholders and representing an activity diagram how a work session for these stakeholders can and should look like. After this, the module, the capability system and existing modules are introduced. Based on this, the design decision on the modules to develop are made and explained. The chapter finishes with a look on the ALICE system and the conclusion drawn.

In Chapter 4 implementation details of the first prototype of RIVALE are outlined and discussed. The main focus is lies the development process of the modules used to implement the chatbot functions, the modules to provide story and task information and the work done on the world modules in which the users moves around with an avatar. The chapter finishes with an explanation on the chatbots system itself.

Chapter 5 starts with the user viewpoint, bringing up screenshots and details how a session looks. After that, the chapter continues with the evaluation of the first RIVALE prototype. The participants were teachers and students recently finished a system analysis class. The last part explains and discusses the evaluation outcome in detail.

Chapter 6 and 7 are the final chapters and conclude on the research results and lessons learned. The lesson learned is split in theory, development and evaluation. The results will also give an outlook into possible researches following this thesis.

2 Background and Related Work

The Internet and especially the World Wide Web is increasingly important for education and training. But it is still a big challenge to use the WWW and multimedia application in an appropriate way. Many of the multimedia training programs are not better then old, low-tech training programs, they simply look better (Schank, 1997) and are often a bit easier to access. WWW let us access a huge amount of data and knowledge which is often unstructured and inappropriate for teaching and learning in an advanced way. The problem is not that the data is not there, it is often just not presented in a way that it would support learning like it should. There are different ways to overcome these issues. One good approach is to provide information bound in a virtual world, allowing a high interaction from user side as well as from application side. The problem is that these systems are often a closed system with a fixed content. Newer researches try to provide these environments with dynamic and more flexible content from the World Wide Web and this is done with a new generation of information retrieval systems and selecting methods. Such systems are not only getting better in finding homepages, there are also improved ways to query ontology's and other data structures. This makes it very interesting to upgrade existing virtual environments with systems providing information and behaviour based on interaction coming from the Internet and other sources. One of these enhancements is to simulate humans in 3D virtual worlds also called NPC. NPC serve different purposes in games, from providing information, to helping with tasks and other things (Adolphs et al., 2011). Artificial intelligence can be based or supported by a type of information retrieval system, named before. Simulating a real time chat with a NPC in natural language, can give feelings to users as they would chat with a real person. Nevertheless, still a lot of research has to be done in this area and there are only a few research projects that have already worked on virtual characters with unrestricted natural language input and the integration of human-like dialogue capabilities in virtual worlds (Adolphs et al., 2011).

This chapter provides information about the state of art at the topics relevant and related to this work. The first section describes related work and background on the application domain requirement elicitation. The subsequent section will focus on virtual learning and the different types of virtual learning. It will also provide a deeper look into 3D virtual worlds and environments as well as Bots and there extension, chatbots. Chatbots are the most important part of this work and are also in focus in the implementation part. This section also covers serious gaming and the challenges developers are facing with it.

2.1 Application Domain

2.1.1 Requirement Engineering

There are many definitions for requirement. In general *"[...] each contract specialist, lawyer, engineer, systems engineer, manager, or anyone else involved in the transition of vision into product, has his or her own definition of a requirement"* (Harwell et al., 1993). Therefore the following list will state a view definitions related to requirement elicitation and system engineering.

The company TechMISLLC (n.d.) defines requirement as following: *"A requirement is a singular documented need of what a particular product or service should be or perform. It is a statement that identifies a necessary attribute, capability, characteristic, or quality of a system in order for it to have value and utility to a user. In the classical engineering approach, sets of requirements are used as inputs into the design stages of product development. Requirements are also an important input into the verification process, since tests should trace back to specific requirements. Requirements show what elements and functions are necessary for the particular project."*

The Author Chemuturi (2013) wrote: "The dictionary defines requirements as ''a need'', ''a thing needed'', ''a necessary condition'', ''a demand'', ''something essential to the existence or occurrence of something else'', and ''something that is needed or that must be done''. Simply stated, a requirement is a need of some person or process. A requirement is capable of being fulfilled."

The Author (Chemuturi, 2013) stats further: *"IEEE (Institute of Electrical and Electronics Engineers) standard 610 ''Glossary of Software Engineering Terminology'' provides three definitions:*

1. *A condition or capability needed by a user to solve a problem or achieve an objective,*
2. *A condition or capability that must be met/possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents.*
3. *A documented representation of a condition or a capability as in (1) or (2) above."*

2.1.2 Requirements Elicitation (RE)

Requirements Elicitation (RE) is seen as the process of collecting and discovering the requirements from a system, by communicating with customers, systems users and all other stakeholders. It is part of System Analysis. Another often used synonym for RE is *Requirement Gathering (RG)*. However, the Author (Chemuturi, 2013) stats that elicitation and gathering is not the same but is often used as combination to find all requirements.

The Author Chemuturi (2013) explains elicit and gather like this: "*The Dictionary meaning of the term ''elicit'' is to ''draw forth or bring out'' something that is latent or potential or ''call forth or draw out'' as information or response. This connotes a dialog in which information is drawn out from a party possessing the needed information*. *[…] The Dictionary assigns multiple meanings to the term*

''gather''. One of them is ''to bring together'' as in ''tried to gather a crowd''. Another meaning is to pick up or amass as if ''by harvesting/gathering ideas for the project''. Another one is ''to effect collection of'' as in ''gather contributions''. As you can see, the term ''gather'' connotes collecting things which are available but scattered over the place. "

Therefore elicitation is the first hand collecting from a person by using interviews. Gathering on the other side is an indirect collection of information from other sources then human beings such as documents, existing application, standards and guidelines. The following list is provided by (Chemuturi, 2013) showing techniques to elicit requirements:

- Personal Interviews
- Questionnaires
- Customer/market surveys
- Observation
- Brainstorming

However, in many sources RE contains also requirement gathering including techniques such as document analysis. But still the main techniques are with human interaction, because much of business or technical requirements are not documented anywhere, it is just in minds of stakeholders, in the feedback from end users what needs to be obtained and from a study of flowcharts and others that have to be made. A lot of projects fail because of wrong or pure requirement elicitation (Morgan, 2010).

Elicitation process

It is impossible to know all the details of the customers need up front. Therefore it is essential and it can be called keystone to extract the requirements for a successful system and project implementation. To reach this, it is necessary to be prepared for interviews and other techniques as best as possible. So the first step in the process is to gleaning a comprehensive and accurate understanding of the project's business needs. It is important for the analyst to have a strong understanding of the business needs, to lose not the scope during the elicitation process. It will also help to select the proper stakeholders and elicitation techniques (Morgan, 2010).

As next step, it is necessary to ensure that an adequate amount and mix of stakeholders are secured for the elicitation process duration. A good analyst must "*actively engage stakeholders in defining requirements"* (Morgan, 2010). Morgan refers to BABOK (Brennan, 2009) who lists the possible stakeholders as follow: "*a project's stakeholders may include customers/end users, suppliers, the project manager, quality analysis, regulators, project sponsors, operational support, domain subject matter experts, and implementation subject matter experts".* An analyst must choose the participants of the appropriate stakeholders based on the unique needs of the project. Then a technique has to choose, used on them.

After selecting a technique, the scope of the selected elicitation technique has to be clarified and all necessary supporting materials have to be gathered. Then all techniques have some differences in the preparation (Brennan, 2009). Next is a deeper look into interviews. As Babock (2009) said, he strongly believes in that the primary difference between an expert analyst and a novice lies in the ability to recognize situations and apply the proper tools, in the case of interviews: questions, that suit the situation. To be able to so, the analyst need to know already a set of questions that could fit in different situations. So it is also a part of preparation to bring a set of questions together, helping to elicit the requirements.

2.1.3 RE Teaching Methods

There are different traditional approaches to practice, evaluate and give feedback on elicitation skills. One way is to give an assignment based on some case study to ask the student to describe their way and what they would do to fulfill the task, in the particular situation. This can include e.g. to design questionnaires or interviews schedules. As Venable (2012a) stats *"[…] writing an interview schedule and actually carrying out an interview are very different things, which means that much potential learning about the interviewing task and needed skills does not occur."*

Another way is to hold mock interviews during a class sessions. In this mook students are interviewing a person (sometimes even other students) playing a role. Other students can observe the mock interviews and learn from what they see and what happens. They main problems in this approach is that not every student gets the actually conduct an interview. It is also often not very realistic because the interview might be somewhat fragmented/disjoint and the interviewer may have very little stake in what they find out from the interviewee. Another problem is that this approach does not confront the student with the issue that is may be difficult to arrange and schedule interviews (Venable, 2012a) and in the interviews the interviewee might almost always have an answer on an question but in really the questions might often not give an expected, useful answer.

Another approach is to have people who pretend to be people in the case study and they get then interviewed by a number of students outside of class. This addresses a view problems from the approaches named before. But the problem here is that people can become tired of that, that means when there are a lot of students who interview them, so that the workload and concomitant costs may be quite high.

2.1.4 Related Work

Teaching requirement elicitation is traditionally centered on theory and students rarely get involved in real projects to practice and use those skills. Today often contracts and projects are done over country boarders. This leads to even more problems by teaching RE regarding to cultural, language and time differences, inadequate communication, difficulties in knowledge management and trust

(Romero et al., 2008). Therefore training RE gets even harder in a classic theoretical approach. New ideas have to come up using new technologies to give a better learning and teaching environment. Such new approaches are manly based on E-learning techniques discussed in the next topic.

But there are not so many research projects where E-learning tools aiming on teaching and practicing RE and related topics such as System Analysis. Those examples are to the best of the author's knowledge. One would be the work from Romero et al. (2008) developed a virtual agent for teaching of RE in Global Software Development (GSD). Their aim was to propose a simulator environment which will be enable students and teachers to acquire a subset of skills necessary for RE in GSD by using an agent to learn interview techniques and computer mediated communications or an understanding of the cultures and customers of other countries. The students interact with NPCs and/or real humans to obtain the functional and non functional requirements.

Figure 2: Starting room of the fictional company MRE in the HyberCase project Kendall et al. (2007)

Another project is "HyperCase" done by Kendall et al. (2007) developed an educational tool to teach system analysis on an HTML site using Javascript components. HyperCase is an interactive system analysis design simulator focusing on situations and activities an analyst would encounter in the real world. The projects try to provide the student with a rich, realistic and entertaining experience. HyperCase depicts a fictional company, based on real-life problems from the author's consulting experiences. In the "game" the student is part of an analysis team where he has to fulfill a broad range of task and to be find out information such as "*How many people are employed by MRE*?" or *"Summarize what you learned from the interview with Taylor."* Figure 2 shows the starting screen of Hybercase.

Such projects open new ways for teachers and students in the future, but there is still a lot of research do to and there are still only little amount of good working programs available for the broad

range of educational institutes in the world, that have the pedagogical success what it should have. This leads to the next discussed topic “Technology Enhanced Learning”.

2.2 Technology Enhanced Learning

According to Schank (1997), the best way to learn is by doing, failing and practicing. For this we need to create a save environment where failing leads not to extra costs, learning is no safety problem and time is no issue at all. Virtual worlds serve this purpose. Virtual learning is learning and teaching environment where teacher and students are separated by time or space, or both. Teachers provide course material and content by computer bases applications like videoconferences, management applications or other (Pazhanisamy & Gopalakrishanan, 2011). Virtual learning can take place synchronously or asynchronously.

2.2.1 E-Learning

E-learning refers to all forms of learning where information and communication technology (ICT) or digital media is used as a presentation and distribution tool to enhance and/or support learning and teaching in tertiary education. This includes a broad range of different systems, from simply using emails to get access to course work, over using apps on the phone for sharing information, to an online course offered from the university entirely online (OECD, 2005). Tavangarian et al. (2004) suggest as a starting point the consideration of four issues for e-learning systems which must be supported individual:

- “*How can the teacher be supported in producing teaching material for standardized profiles?*
- *How should the material be presented to learner and which kind of interaction with the material will support learning?*
- *Which kind of feedback is useful and possible?*
- *How should teachers and co-learners be represented within the system?”*

E-Learning is not simply another technology innovation with a little impact on education. Teaching and learning possibilities has now almost infinite access to data and information via the WWW. E-Learning transforms the education of this century in a new dimension. It has the potential to fully integrate the benefits of freedom with connectivity. However, the challenge is enormous, because there are no simple rules or recipes to follow to implement an effective and pedagogical useful system (Garrison, 2004). There was the thinking, few years ago, that E-Learning will be the teaching way of this century. Now a day is mainly the believing in that the traditional way of teaching cannot be replaced. E-Learning is simply an addition to the learning process. With the combination of different teaching strategies, learning can be improved. Especially for people who prefer to learn and read with pc instead of traditional textbooks and books is E-Learning an advantage.

The advantages and disadvantages of eLearning and online education (depending on the used E-learning types and technologies, e.g. synchronous or asynchronous) are outline below.

Advantages

- Students can follow entire online courses at universities located overseas without being physically in the same country or even on the same continent, having no inconvenience and cost of living abroad (OECD, 2005).
- It is possible to attend courses from anywhere, such as home or the workplace. Students can, depending on the eLearning technology, work on their own, on a schedule that suits their individual needs (Athabasca, n.d), and that with a 24 hours / 7 days accessibility to the students and teachers (Zameer, 2010).
- The whole system requires less physical infrastructure and allows an ease time management as well as general improved cost effectiveness (Zameer, 2010).
- Can lead to more motivation for self study (Zameer, 2010).
- Help to improve computer and other skills which are used to access the courses and the material (Zameer, 2010).
- Improved interaction between students and teachers (Dalsgaard, 2006).
- Students can independently solve problems with provided tools and methods (Dalsgaard, 2006).

Disadvantages

- Hard to test if the work which is submitted is made by him or someone else (Zameer, 2010).
- Not every teacher has the knowledge to use the virtual education systems which leads to effectiveness leaks (Zameer, 2010).
- Cheating is easy and hard to detect (Zameer, 2010).
- Technology students have advantages over non technology students (Elearning-Companion, 2011).
- Employers as well as institutions of higher learning are still often less willing to accept degrees from only eLearning institutes because they are not as accepted as face to face environments (Elearning-Companion, 2011).
- Big leak of social interaction because every communication will be through electronic mediums, as well as speaking practices and discussions skills (Elearning-Companion, 2011).
- Not all courses and all teaching fields are available or possible online (Elearning-Companion, 2011).
- Sever and other technologies which are important, like Internet connection in general can fail and make the access of the online courses impossible (Elearning-Companion, 2011).
- Asynchronous communication can be a problem for fast exchange of question (Elearning-Companion, 2011).

Today universities are available who are only offering online education and completely abdicate campus like the Athabasca University in Canada with 32 000 students and increasing numbers (Athabasca, n.d). Today nearly 30% of college students are taking at least one online class (Friedman & Friedman, 2013).

But E-Learning still faces problems. The learning material must be customized (depending on the used E-learning types and technologies, e.g. synchronous or asynchronous) to get the best output results. Of course it is impossible to provide complete individual course material for everyone. One approach is to divide the learning community into various stages of advancement, what will reduce the number of versions needed (Tavangarian et al., 2004). Therefore the Authors Tavangarian et al. (2004) regard to the system offered and developed by Dreyfus (Dreyfus & Dreyfus, 1986) and Baumgartner (Baumgartner & Payr, 2001), specifying five different levels in which learner can be classified (see Figure 3).

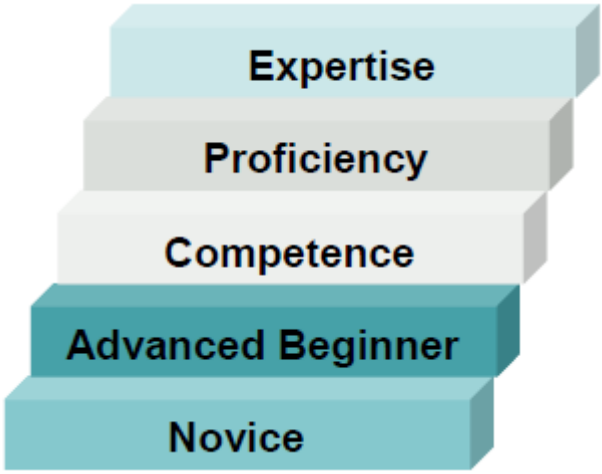

Figure 3: Learner Levels (Dreyfus & Dreyfus, 1986)

The levels differs the learners in different grades of intellectual and practical mastery of the subjects. This is a quit simplified and one dimensional modal and point of view and it is in general not enough to classify learners like this, more than one criterion is necessary to specify optimal (Tavangarian et al., 2004). This model can also be used to classify the users' technology knowledge need to interact in a proper way with the used E-learning technique.

2.2.1.1 Pedagogical Approaches and E-Learning Theory

There are different kinds of learning theories and pedagogical approaches in E-Learning may be considered by designing an E-Learning system. The E-Learning theory examines these approaches.

General is E-Learning theory the cognitive science principle of effective multimedia e-learning, this includes the following models (Moreno & Mayer, 1999):

- Social-constructivist
- Laurillard's Conversational Model
- Cognitive perspective
- Emotional perspective
- Behavioral perspective
- Contextual perspective
- Mode Neutral

There is no single best-practice e-learning standard. It is more a list of learning and teaching styles showing the ways how technology can be used and implemented. There are already a lot of researches in this direction, trying to find the best working solutions. An example is Moreno and Mayer focusing on representation of the content in E-Learning. (Moreno & Mayer, 1999) showed that in different studies where found that e.g. illustrations placed near text generated about 75% more useful solutions on problem-solving transfer question then other students did who had text and illustrations presented on separated pages. Or students were able to produce 50% more and creative solutions when the verbal and visual explanations where integrated than separated.

2.2.1.2 E-Learning Typs

E-Learning can be divided in a bigger variety of sub terms, all regarding to the different use of technology e.g. "hybrid learning", "Distributed learning" or "blended learning". E-Learning can assist or completely replace other teaching approaches, from none to full online distance learning approaches (Bates & Poole, 2003).

Synchronous vs Asynchronous

E-Learning can be synchronous or asynchronous. In fact, many people take only courses because of their asynchronous nature, what makes it possible to learn and attend a course when they want. It makes it easier to manage work, family and free time. Students tend to spend more time in thinking about their contribution as in synchronous way. On the other side is synchronous e-learning as a more social experience for student and teacher. It helps to avoid frustration and asking and answering in real time. However recent improvements in technology and increasing bandwidth capabilities have led to a growing offer of synchronous educational systems. Studies try to determine which approaches have the better results generally show no significant differences (Hrastinski, 2008).

Collaborative vs Competitive learning in E-Learning

Computer-Supported Collaborative Learning (CSCL) supports group work processes, such as students working together on learning tasks. The system can be implemented synchronously or asynchronously. CSCL uses blogs, wikis and document sharing/managing platforms. It also uses Web 2.0 technologies, sharing information and data between different people in a network, making the

learning process a social learning process. Study fields in CSCL are development psychology, social psychology, sociology, cognitive psychology and sociocultural perspectives (Hmelo-Silver, 2006). The other part is competitive learning. This approach should always follow a number of principles, such as having an only a symbolic or little value prize and a goal set clearly into the learning process and not into the winning. Advantages in competitions can be, higher motivation, self-esteem, recognition gain and that student put in more effort (Cantador & Conde, 2009).

2.2.1.3 E-Learning Technologies

E-Learning has used many technologies and most of them are combined in courses. Some of the following technologies are also using other presented technologies (like audio or text) and are combinations of them compound with new approaches. In this work the focus is on *Multi User Virtual Environments (MUVE).* Therefore the explained technologies are used in MUVEs.

The list will start with VLEs and MUVEs. The following technologies will be outlined if they are included or if there is a simular module with this concept available in the project Open Wonderland (OWL) (this does not mean that the function is used in the implementation of this project).

Virtual Learning Environment (VLE) & Multi User Virtual Environment (MUVE)

VLEs are web based e-learning education system. It provides a platform for student and teacher to simulate a classroom over distance, to share class content, homework, tests, assessments and external resources like homepages in real-time. It is traditional a synchronous way of learning and often provides a mix of E-learning technologies such as webcams, microphone and real-time chatting in a group setting. An extended way is a MUVE having the same functions as a VLE but is set up in a 3D world. There it is possible to show emotions via chat (smiles) or via the avatar (e.g. let avatar dance). Other technologies are used in MUVEs are whiteboards, text notes, screen sharing and so on. Research has shown that students, who decided to work with a MUVE, got the better marks as students who stayed with the classic methods (Hebbel-Seeger et al., 2013). New systems allow recording of the classes and to watch them later again, providing an asynchronous way of learning.

Audio

Radio was one of the first distance learning technologies used in educational classrooms. Newer technologies like the Internet allow streaming over bigger distance in often better quality. There are different ways to provide audio, like webcasts and podcasts which can be downloaded and saved and are not longer a synchronous way of teaching like radio e.g. is. There is also a difference in the direction (unidirectional or bidirectional) the audio communication works and is provided. There is in general no immediately interaction possibility (e.g. to influence the talk) if the communication is asynchronous. There are also systems which are synchronous but only provide audio unidirectional and so fore it is not possible to contact the teacher via audio, additional technologies must be used. However, there are systems providing synchronous and bidirectional audio transaction (e-teaching,

2011). VLE are normally provided such systems, also in OWL, which offers a voice chat to groups as well as to single other users. To use this is a headset or other voice input hardware required.

Video

Video teaching is also not a new concept. Older but still used approaches provide content on CD/DVDs or VHS tapes. Video teaching allows teacher to reach students which are visual learners. The Internet provides a better and faster update and there are many websites offering video sharing. YouTube and others are often used by teachers to broadcast there lecture. This is an asynchronous way to teach and allow no instant interaction with the audience. Other programs like Skype, Adobe Connect or webcam transmission programs, offering a synchronous and bidirectional way of lecturing, called videoconferences and also called teleteaching. It is a way where more teachers and students can communicate at the same time via audio and video. The communication is "simular" to face to face teaching, because communication is not restricted to audio, it also includes gesture and mimic (E-learningWiki, 2009; e-teaching, 2011). A kind of video teaching are interactive video games. There are video modules developed for OWL. To use video is a webcam or other video capturing hardware device required.

Text

There are different ways how text exchanges are used in E-Learning systems. Asynchronous ways are E-Mails, Mailing lists, Newsgroups, Forums and Blogging. Also services like Twitter can be used as a teaching technology. This systems allow it to always review the written information (as long as they are not deleted), synchronous systems on the other side, are not always have storing function included. Synchronous way is Chat (e-teaching, 2011). OWL includes a life chat, allows chatting with a group or a single other user.

Whiteboards

A whiteboard can be compared to a blackboard or flipchart. The users can work together in real time (synchronous) on sketches over a network. The whiteboard provides draw function as well as texting tools. Reason to use a whiteboard are the more people can work together at the same project, often whiteboard sessions are recorded, visualisation of teaching content, easy to use (E-LearningWiki, 2009). OWL provides such a module.

2.2.2 Serious Gaming

There are many definitions in the literature for *Serious Gaming* most of them agree that serious games are digital games used for purposes other than for entertainment, such as training, advertising, simulation and education (Susi et al., 2007), in the fields of healthcare, military, management, public policy, corporate games, education games etc.

Serious games open up new opportunities in a broad base of fields for complex skill learning regarding higher education (Westera, 2008). Games give a great way to improve learning outcomes. They are often cannot replace traditional teaching methods, because, for example a leek of social aspects, but they are a great tool to assist and support teaching in all educational levels. They e.g. support learning with increased visualization and challenged creativity aspect, help gathering new skills with computer and software applications, improve co-operation skills, problem solving strategies and critical thinking (Westera, 2008). Further potential benefits of games include improved self-monitoring, decision making, problem recognition and problem solving, better short and long term memory and general team skills. Other researchers report that gamers developing a more analogical thinking instead of a trail-and-error approach (Susi et al., 2007).

Serious games try to combine education with fun and game based aspects such as points for achievements or ranking list, this should have positive effects on the motivation and results in better learning outcomes. However this is not the main reason of taking games. In higher education, the main objective is to simulate complex problems and situations which are not easy to produce in reality, because of a safety, cost, time, etc (Susi et al., 2007). Learners require the experience of failure to learn better, but the risk in some situation is simple often too high. In real-life a wrong tool can cause death for the learner as well as for involved people. So it is understandable that for reducing the risk, a lot of learning situations are transferred to an abstracted model (Reiners et al., 2012). A digital game often simulate the closest possible and save simulation way to do so.

There is today a big industry of E-Learning, including Serious Gaming, which keeps growing. The global E-Learning market was at 32.1 billion Dollars in 2010. The Game-based learning market by his self reach a value of 1.2 billion Dollars in 2011. The growing rate is 15.4 % and at 2015 the revenue will have doubled to 2.5 billion Dollars (Ambient Insight, 2013). The e-Learning research company Ambient Insight published about this the Table 1, showing forecast for the enormous growing rate in the market.

Region	2011 Revenues in $US Millions	2016 Revenues in $US Millions	Five Year CAGR 2011-2016
North America	$286.73	514.83	12.4%
Latin America	$21.51	$77.22	29.1%
Western Europe	$83.15	$136.43	10.4%
Eastern Europe	$11.47	$36.04	25.7%
Asia	$813.18	$1,723.20	16.2%
The Middle East	$2.87	$6.18	16.6%
Africa	$10.04	$25.74	20.7%
Total	**$1,228.95**	**$2,519.64**	**15.4%**

Table 1: Growrate Game-based Market (Ambient Insight, 2013)

It is not surprising that also big players in the entertainment industry start to offer education programs. As example MinecraftEdu is an educational version of the very common most successful indi game, Minecraft, in the world. Minecraft offers very cheap licenses for schools. According to Joel Levin ("TheMinecraftTeachr"), how help starting the project, the company was very supportive of their effort to bring Minecraft to schools (TeacherGaming, 2013). Another example is the company Valve which 2012 lunched an education program called "Teach With Portals". There Teachers can get Portal 2 and a level editor for use in the classroom. They also offer a space to share and connect with each other to exchange ideas and sharing lessons (Valve, 2013).

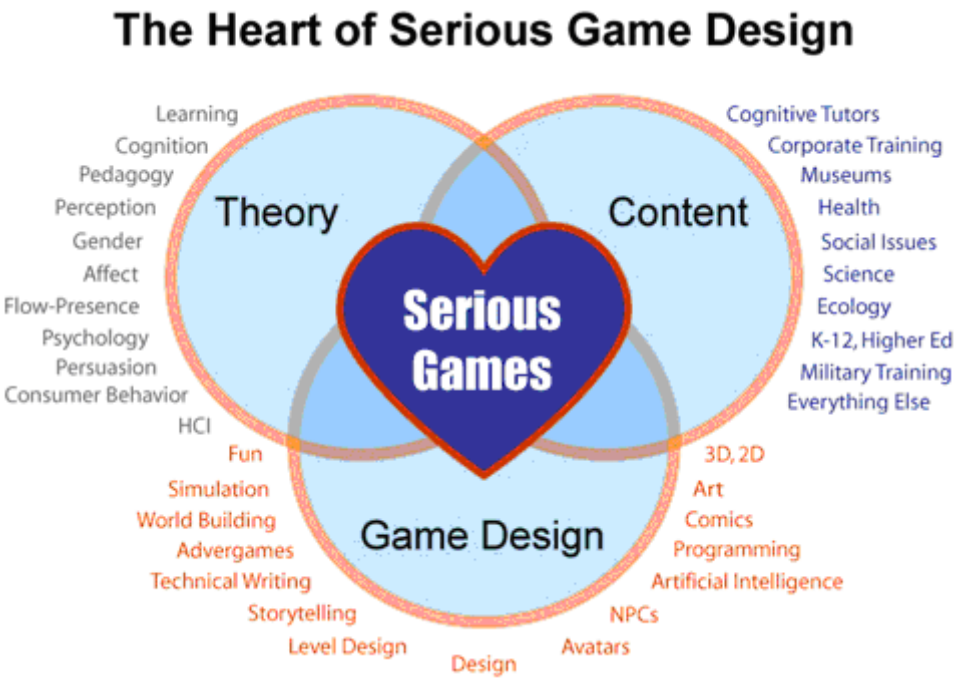

Figure 4: The Heart of Serious Game Design (Michigan State University, 2013)

Serious games developer facing new challenges. Their teams need to have the technical knowledge as well as the pedagogical and have to handle new combination of input streams. Today a lot of programs as well as university degrees are available just focused on eLearning and serious gaming, e.g. the Serious Gaming University (Serious Game University, 2013) or the Michigan State University (MSU), offering a fully online graduate certificate in serious games (Michigan State University, 2013). MSU provides also a chart about the most important facts a serious game designer has to take care about, shown in the Figure 4 called "The Heart of Serious Game Design". It consists out of three main parts, the theory, the content and the game design. Considering all parts will lead to a good serious game design.

2.2.3 3D virtual Worlds and Environments

Games and 3D virtual world simulations have become a huge industry (Looi, 2005), with about 25$ billion revenue in 2011 (ESA, 2013) and virtual 3D worlds (also MUVEs) are more and more dominating the gaming world. Games like World of Warcraft, Minecraft, Heroes of Newerth or Diablo 3 are dominating gaming charts (DFC Intelligence, 2013), but also more open 3D virtual worlds have success stories, such as Second Life which increased the registered number of users from 100,072 to 2,251,416 in only twelve months (Hebbel-Seeger et al., 2013). This *Multiuser User Virtual Environments (MUVE)* allow non-collocated users to interact in a virtual reality, represented by avatars (Gamage et al., 2009). MUVEs are an extended version of *Multi-User Dungeons (MUD)* (Dieterle & Clarke, 2008). MUDs are computer programs accepting a number of connections at the same time, presenting all in text. These MUVEs are manly used for entertaining and gaming approaches. However there are a lot of projects and research done with it in the educational domain, e.g. a “Case Study for Virtual Worlds and Cloud Computing” by Chang & Gütl (2010), “Evaluation of Collaborative Learning Settings in 3D Virtual Worlds” by Chang et al. (2009) or Second Life Education New Zealand (SLENZ) project (Gamage et al., 2009), to name only a few of the big amount of researches. These online virtual worlds provide new opportunities and challenges for technology-enhanced learning (Allison et al., 2010) and practicing, by combining educational, social and gaming aspects with each other. In the moment the interaction with this MUVEs are restricted to computer screens, mouse and keyboard input, but this will soon overcome this restrictions (Gütl, 2011).

The following section provides information about virtual world environments frameworks, which have been used in different educational projects and researches. This is only a tiny overview of the most important, giving a short history, features and technical information for each one.

2.2.3.1 Second Life

The 3D virtual World *Second Life* (SL) (illustrated in Figure 5), was released in June 2003 by the company Linden Lab, has celebrated his tens birthday recently. Linden Lab was founded in 1999 and was working from 2001 on a virtual World called “LindenWorld”. This was later renamed into Second Life and went into closed beta in November 2002. Since then the user number has rapidly grown and SL got one of the biggest MUVEs existing. For example the registered users raised, in 2006 from 100,072 to 2,251,416 in only twelve months (Hebbel-Seeger et al., 2013).This MUVE is build and extended by his residents how can buy land in order to form it as they want. It is possible by writing code to manipulate the environment, what is important for educational matters to modify the world as needed. It is also possible to make or purchase own clothes, participate in group activities, work, explore, play and interact socially in different ways (Messinger et al., 2009). There are different fields in which SL is used, such as advertisement, education, communication, entertainment and others. Basic accounts are free and it is simple to use, just downloading of the client is required. Just to buy land part (or even an island) a premium account is required, where he can modify then the landscape

and all other stuff. As everywhere, money rules the world, also in SL. The currency is called “Linden Dollar” (LindeLab, 2013). SL is not explicit made for education, even when there is already a bigger community around education in SL. SL is simple adopted in a way that it can be used as educational tool. LindenLab does not over special over to universities or other institutes for using it in education (SecondLifeWiki, 2013a). Therefore the following prices have to be considered as shown in the Table 2.

Figure 5: Screenshot Second Life (Schmitz, 2007)

Additional Land (over 512 sqm)	Parcel Size	Prims per Parcel (approximate)	Monthly Land Use fee
1/128 Region	512 sqm	117	US$5
1/64 Region	1,024 sqm	234	US$8
1/32 Region	2,048 sqm	468	US$15
1/16 Region	4,096 sqm	937	US$25
1/8 Region	8,192 sqm	1,875	US$40
1/4 Region	16,384 sqm	3,750	US$75
1/2 Region	32,768 sqm	7,500	US$125
Entire Region	65,536 sqm	15,000	US$195

Table 2: Second Life cost (LindeLab, 2013)

Technology and Requirements

SL has a client also called “viewer” and the server part. The servers are hosted by Linden Lab and using *Linden Script Language* (LSL). The server code is not open source and therefore there is not much information available, offered from LindenLab. There is a “Login Server”, handling verification of usernames and passwords, a “Spaceserver” handling routing of messages, a “Dataserver” handling connections to the central database as well as log, inventory and search database. There is also the “Simulator” which is the primary SL server process. Each simulator process simulates a region. There are about 31000 such regions and a bunch of other servers (SecondLifeWiki, 2013b). The client also called viewer is coded in C++ and will run on Windos, Mac OS X and Linux. It has some creation tools build-in. There is no plug-in system available what makes it more difficult to extend the world in advanced way. The viewer uses the Open GL Library.

Figure 6 shows the structure of SL. As soon a user is logged in over the viewer, he is connected to a simulator node (sim node). When he moves from one region to another, he also changes the server process he is using and connected with. The communication from viewer to sim node is over UDP. Such sim nodes have the possibilities for e.g. do HTTP calls or connect to mail servers as shown. There are other functions available right from the start way, such as voice chat and other. Implemented services as well as sim nodes are/can use the provided databases, running on MySQL.

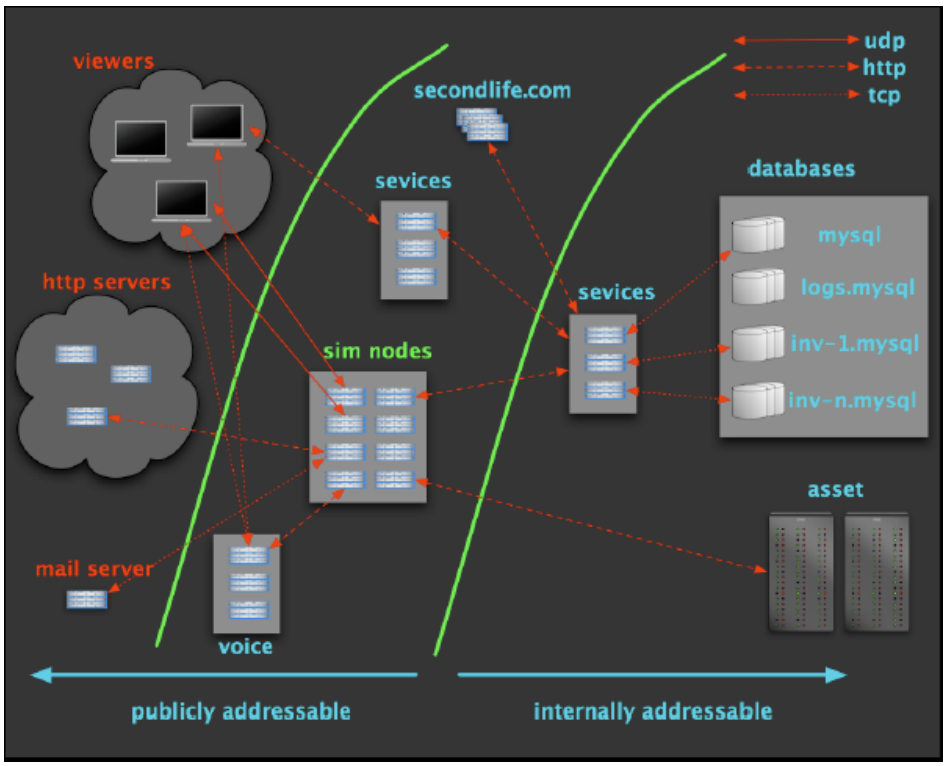

Figure 6: Second Life Architecture (Wilkes, 2008)

There are clients for Windows, Linux and Mac OS X available. The table 3 provides an overview on the Windows requirements.

Windows	Minimum Requirements
Internet Connection:	Cable or DSL
Operating System:	XP, Vista, or Windows 7
Computer Memory:	512 MB or more
Screen Resolution:	1024x768 pixels
Graphics Card:	NVIDIA GeForce 6600 or better OR ATI Radeon 8500, 9250 or better OR Intel 945 chipset

Table 3: Second Life requirements (LindeLab, 2013)

2.2.3.2 Open Wonderland

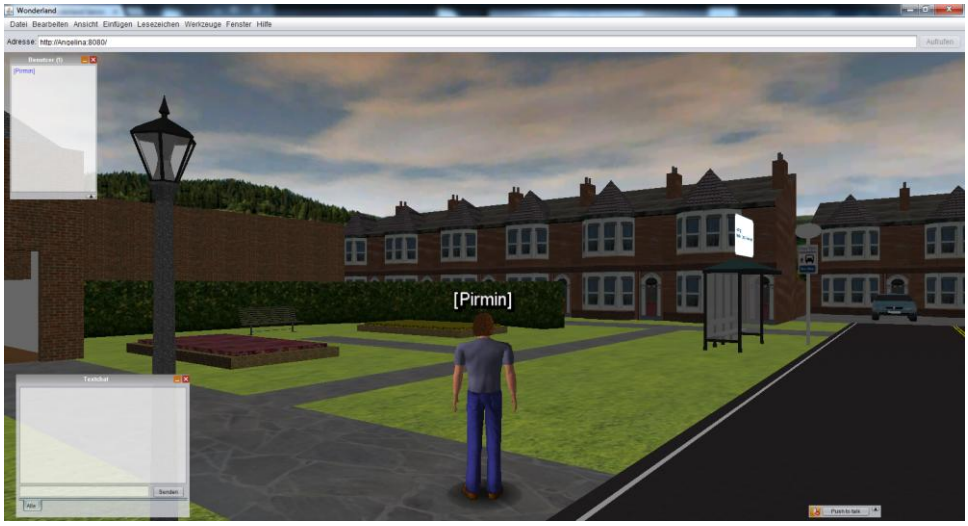

Figure 7: Open Wonderland

Open Wonderland (OWL) (illustrated in Figure 7), first called Project Wonderland has been funded by Sun Microsystems. When Oracle took Sun over they decided to close it down. After this, the project was renamed into Open Wonderland which is now a community supported open source project, written in Java. It is complete free of fees and published under GNU General Public License 2.0 licence. The Framework is made for creating collaborative 3D virtual worlds focused on education and business. It is completely extensible for developers and graphic artist to create new worlds and add new features to existing worlds, done with a plug-in system. This makes it possible to use Open Wonderland even without Java knowledge, since there are already a lot of plug-ins available. The community offer to share the modules different developer has made. OWL trys to provide an

environment that is robust enough in terms of security, scalability, reliability and functionality that organizations can rely on (OpenWonderland, 2012a).

Technology and Requirements

According to the CEO of OWL Yankelovich (Kaplan & Yankelovich, 2011), who is the raison why the project was not stoped completely and is now running community based, the design goals are: "*[...] enabling collaboration with a focus on synchronous interaction, providing an extensible toolkit based on open standards, and putting in place the infrastructure for federation to enable the 3D web.*" The Figure 8 shows various components and how they communicate.

Wonderland uses a client-server model to create collaborative virtual worlds (see Figure 8), written entirely in Java. The client provides a browser that turns these shard services in a 3D view of the world. OWL used different types of networking protocols optimized for different data types. This allows each protocol to be optimized for the type of data they handle. The following list provides an overview on the services (Kaplan & Yankelovich, 2011):

- "*Web services for authentication, downloading code, and world assets such as 3D models and textures;*
- *custom TCP-based protocols for communicating world data such as object properties and position;*
- *Session Initiation Protocol (SIP) and Real-Time Transport Protocol (RTP) for audio; and*
- *multimedia streaming protocols for video, application sharing, and screen sharing*."

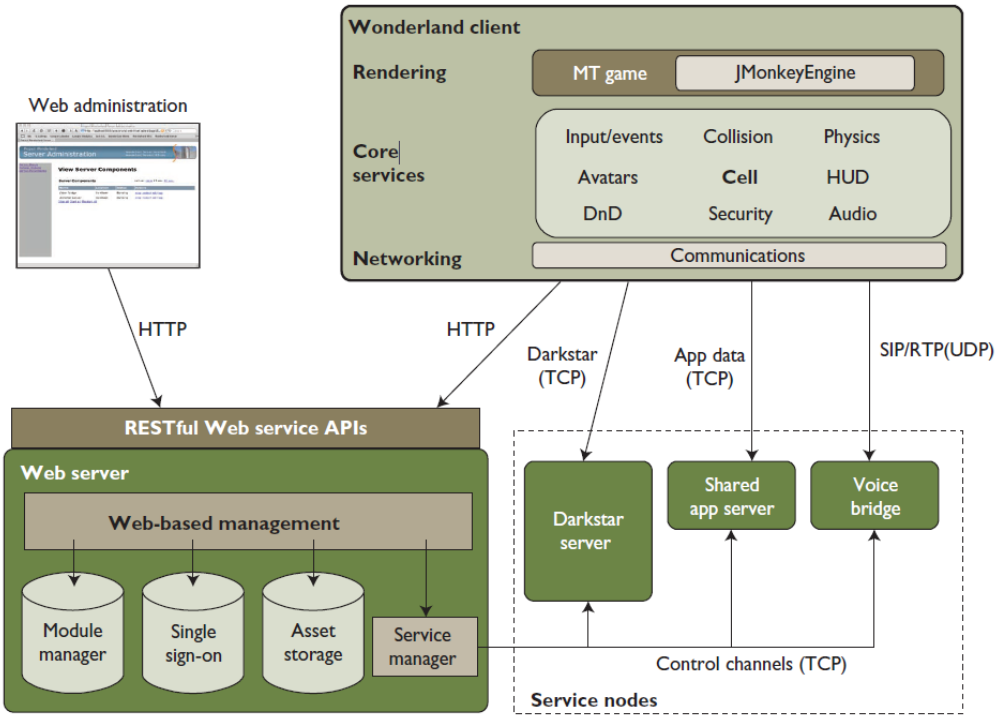

Figure 8: Open Wonderland network diagram (Kaplan & Yankelovich, 2011)

The Wonderland server is based on four cooperating services, all separate Java applications with own networking and storage mechanisms. This allows it to separate the services on different machines. Those four services are, the Web administration server, which is the main coordination point based on Glassfish Java EE Application server, then the Darkstar server, providing a platform specifically design for online games, the JVoiceBridge, providing server-side mixing of high-fidelity immersive audio and the Shared *Application Server (SAS)* allowing server-hosted application sharing.

The Wonderland client provides also some core services. It consists of two separate projects. JMonkeyEngine is a rendering framework for OpenGL-bases application in Java. This provides the basic scene graph, but it has some limitation, because it is only possible to work on a single thread at a time. MT Game is a subproject of Open Wonderland and solves the issues of JMonkeyEngine by adding multiprocessor capabilities. The client core service layer provides basic function such as loading and rendering 3D objects with correct positioning, moving objects and collision detection.

OpenWonderland has clients for Windows, Max OS X, Linux and Solaris. The Basic system requirements are in the Table 4.

	Requirements
Computer Processor	1,5 GHz
Computer Memory	1 GB
Graphic Card	Recommended Nvidia driver, 128 MB video memory, recommended 256 MB

Table 4: Open Wonderland requirements (OpenWonderland, 2012b)

2.2.3.3 OpenSimulator

OpenSimulator (illustrated in Figure 9) also called *OpenSim,* was founded in January 2007 by Darren Guard. At that time the Second Life client was released open source, which is used to connect on OpenSim servers, therefore the developers had only to focus on the server part. It is made to simulate a multi-user 3D environment. It is published under BSD License which allows commercial use of the Code, because the developer who changed code must not publish the changes (OpenSimulator, 2013).

Technology and Requirements

OpenSim is written in C#. The Server has three different instances defined as “modes”, the standalone mode where the client interact with one server, the grid mode including specific service abbreviated as UGAIM, where every letter stand for a service: User, Grid, Asset, Inventory and Messaging and as third mode, the hypergrid mode which is loosely connected set of simulators

without a global grid manager. In all modes supporting multi-user connections, where multiple users can simultaneously be presented within a single virtual space (Fishwick, 2009).

Figure 9: Open Simulator Screenshot, "Image of the moment" from OpenSim Website (OpenSimulator, 2013)

OpenSimulator requires .NET Framework 3.5 when running under Windows. It can run on Windows XP, Vista and Windows 7 as well as Windows server 2003 and 2008. .Net Framework 4.0 is not officially supported by OpenSimulator yet. There are different versions for 32 and 64 bit systems available. It also run on Linux and Mac OS X and requires then Mono 2.4.3 or later. There are few Mono version which end up in significant performance and scalability problems. The list can be found on their wiki (OpenSimulatorWiki, 2013a). The core requirements are shown in Table 5.

	Requirements
Computer Processor	Dual Core, 2.0 GHz
Computer Memory	2 GB for one Region

Table 5: OpenSimulator requirements (OpenSimulatorWiki, 2013b)

2.2.4 E-Assessment

E-Assessment is a field of research for Computational Science, Pedagogy and Psychology (Wesiak et al., 2013). E-assessment can be understood as the interaction between the student and computer during the assessment process. The test delivery and the feedback is done by computer (AL-Smadi & Gütl, 2008). It includes the use of information technology for any assessment-related activity. It can be used to assess cognitive and practical abilities.

Assessment has different strategies according to its purpose. The two main basic types are formative and summative assessment. Formative assessment is part of the learning process. It gives feedback to the teacher and the student to guide their efforts toward achieving the goals of learning process.

Summative assessment is performed in the end of a specific learning activity and used to judge the student progressing e.g. a final exam (AL-Smadi & Gütl, 2008).

Formative assessment in traditional way is a very high resource using method. It increases the staff workload a lot. This can be minimized with e-assessment approaches. Formative assessment is very important to get a current state of students. As said before, it gives feedback to both, student and teacher and helps to enhance their teaching and learning activities. Further system based evaluation help teacher to fair, reliable, efficient and effective in once (AL-Smadi & Gütl, 2008).

The Graz University of Technology designed an integrated framework for e-assessments that is based on different complex learning requirements such as collaborative learning, serious gaming and storytelling. The model called *Integrated Model For E-Assessment (IMA)* describes the components involved in the learning objectives, resources and assessments methods as well as inputs to learning experiences and interactions. This focus combination is expected to yield effective learning processes such as reflective and experiential learning as well as socio-cognitive learning. Figure 10 shows the IMA on an abstract level with its core-methodology (Wesiak et al., 2013). The core methodology consists of the following four main components:

1. The learning objectives, defined by the course instructor and didactical objectives such as gaining social competence or meta-cognitive skills.
2. *Complex learning Resources (CLR)* should be provided to support learner and to guarantee that he can achieve the learning objectives.
3. New forms of assessment should meet the demands arising from the CLR by considering educational objectives.
4. Evaluation should refer to the used assessment method. Results in this step can influent again the first three steps.

Beside this core methodology, several components influencing the learners experience have to be considered as shown in Figure 10 by big red arrows on the left and right side. "*These include educational aspects (e.g. different learning styles or social learning), psychological aspects (emotion or motivation), technical issues (e.g. adaptive learning or tool selection) and existing standards and specifications (e.g. best practices or ethical aspects).*" (Wesiak et al., 2013)

There are other researches, but still only a few, projects aiming more specific on immersive 3D environments. Especially formative evaluation supporting systems are hard to develop for 3D virtual Worlds. Therefore the most available solutions are made for one platform and lack the support of other 3D environments. However, this requires teachers to have high level of programming skills to implement assessment in teaching programs (Maderer et al. 2013).

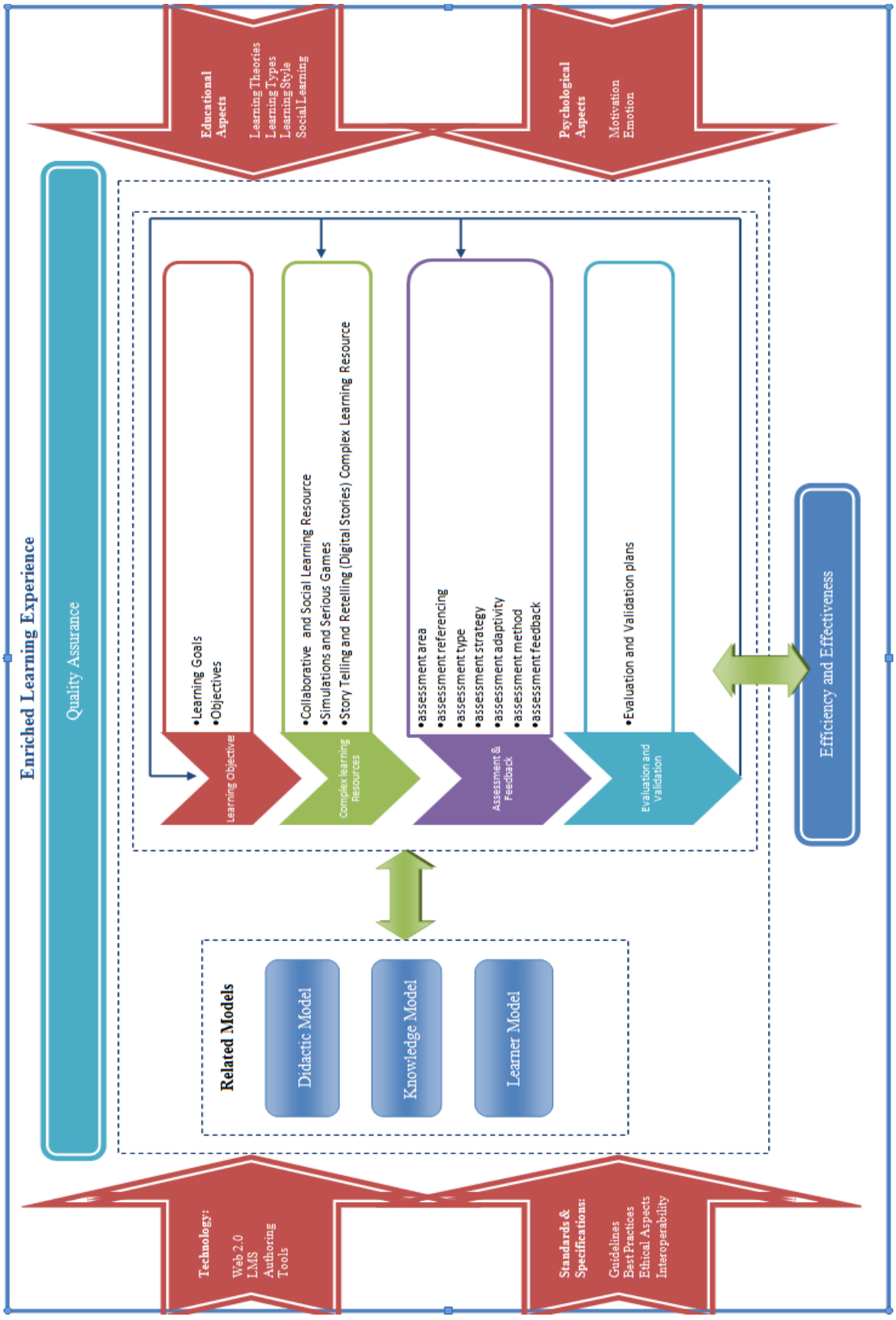

Figure 10: the abstract level of IMA with its core-methodology (Wesiak et al., 2013)

2.3 Bots

A Bot is a computer program that automatically solves tasks, without interaction of a human, running mostly over the Internet. The biggest uses of bots are the web crawlers, who automatically analyses web pages and content. They also can be abused to e.g. collect E-Mail addresses for advertisement reasons or to find security holes in software to attack systems (Wikipedia, 2013a). Another usage fields are in situation where a simulation of human behaviour is required, e.g. chatbots (AI4US Ltd., 2013). Those type of bots are often used in computer games to simulate a person in game, e.g. in a multiplayer roll play game to simulate a policeman. They are also called NPC (non-player character).

2.3.1 Non-Player Character

Non-Player Characters (NPC) are, like the name already expresses, characters in a game who are not controlled by a human but by a part of code, usually through artificial intelligence. The NPC is embodied, meaning that it is represented by an avatar in games or 3D environments. It can detect and react on different input mechanism from the user such as keyboard input, mouse positioning and mouse clicks. NPC interaction is part of gameplay of nearly every modern game and is a very important piece for every game story. They are often a kind of “support character” helping somehow (Wikipedia, 2013b). NPCs can respond verbally and nonverbally (Johnson et al., 2005). They are differing in the level of intelligence they got. Some will only be a visual extra in a game, others can have high developed artificial intelligence background where it is possible to communicate on various themes with them. A great challenge is to make NPCs believable. There are different approaches the most are hard to implement. One of these challenges is to make a chat with an NPC via a free input form, called chatbot (AI4US Ltd., 2013).

2.3.2 Intelligent Pedagogical Agent

Intelligent Pedagogical Agents are similar as NPCs used with a special purpose for education. They are autonomous software entities that support human learning by interacting with student and teachers and by collaborating with other similar agents (Devedzic, 2004). These agents provide scaffolding for student understanding game procedures, specific task as well as the game world by itself. It can present the story, have dialogues with the users or give task and information to the students. It can provide combinations of different stuff like narrative complexity and entertainment. These agents often using an artificial intelligence providing different reactions on students behavior and making the experience individual for every user.

Basically intelligent pedagogical agents providing the necessary infrastructure for knowledge and information exchange between server and client. It helps in locating, browsing, arranging, integrating and otherwise using educational material from different educational servers.

They can support individual or collaborative learning. The connected knowledge bases provide enough intelligence to provide a personalization of the learning tasks it supports. It can be seen as a learning tutor. Authors develop educational content on the server with important pedagogical issues, to ensure educational justification of learning, assessment and possible collaboration among the students (Devedzic, 2004).

2.3.3 Chatbots

Chatbots are computer programs aimed to simulate the conversation of a human being, using natural languages, can be visualized as a artificial person, animal or other creature (Shawar et al., 2002). This can be text based, spoken or a non-verbal conversation, based on an artificial intelligence generated response. The general chatbot workflow is shown in Figure 11. A user produces input (text or other), this is sent to the system, querying the database for matches and then there is returned a filter response to the user.

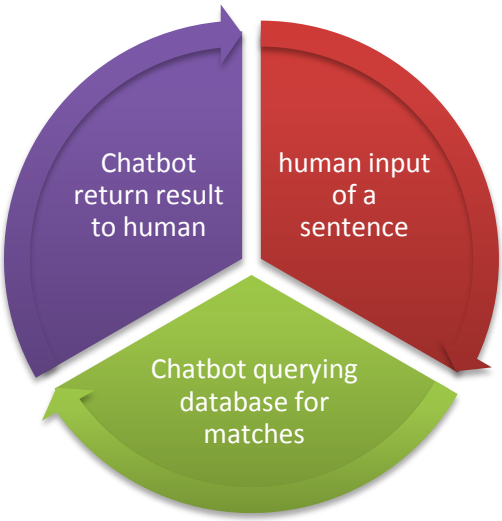

Figure 11: Chatbot basic workflow

This technology started in the 1960's trying to chat with users without that they recognize that they were chatting with a bot. Chatbots are not only developed to mimic human conversation, also entertain users, educate users, for information retrieval, business applications and e-commerce. The first chatbot was implemented by Joseph Weizenbaum called ELIZA in 1966, designed to emulate a psychotherapist. ELIZE was based on simple key word based matching responses. It proved to be incredibly effective in focusing attention and maintain conversation. At that time the bot was called Chatterbot, what is now an outdated term (AI4US Ltd., 2013). One of the most advanced and famous chatbot system now a days is ALICE, developed 1995 by Dr. Richard S. Wallace, using natural language stored in *Artificial Intelligent Mark up Language (AIML)* files that is a derivative of *Extensible Mark up Language (XML)* (Shawar & Atwell, 2007). Extended since then, ALICE won many honours and awards in various contests. The Jabberwacky chatbot is another example, aim to simulate natural human chat, focusing on entertaining in a humorous manner. It got no hard-coded

rules to follow and it learns from all his previous chats and produces an output by using a contextual pattern matching technique to select the response (Kerly et al., 2006).

There are much more chatbots available then the three named before and a lot of chatbot got improved a few times. The best one's are winning the few international competition, like the Leobner Prize started in 1991 from Dr. Hugh Loebner, the National Science Foundation (NSF, 2013), and the Sloan Foundation (Sloan, 2013). Offering a reward of 25.000 U.S. passing first an unrestricted Turing test and Dollar 100.000 U.S. Dollar for the program, being the most "human" program, that can pass a Turing Test using audio-visual input (Mauldin, 1994; Moor, 2000).

In 1950, Alan Turing asked the question "*Can machines think?"* and introduced the Turing test. Turing test is an "imitation game" in which "*an interrogator tries to determine on the basis of written interrogation of a man and a woman which is the man and which is the woman*" (Moor, 2000). The man in the test, imitate a woman and answer the question as he believes a woman would do. In the end a human has to tell if he was talking to another human or to a program (Moor, 2000). The Table 6 gives an overview of the most important and advanced chatbots. Almost all of them won the Leobner Prize.

Name	Year	Developer
ELIZA	1966	Joseph Weizenbaum
PARRY	1972	Kennenth Colby
RACTER	1983	William Chamberlain, Thomas Etter
A.L.I.C.E	1995	Richard Wallace
Cleverbot	1997	Rollo Carpenter
Jabberwacky	1997	Rollo Carpenter
Albert One	1998	Robby Garner
MegaHAL	1998	Jason Hutchens
Eugene Goostman	2001	Vladimir Veselov, Eugene Demechenko, Sergey Ulaen
Elizabeth	2002	Abu Shawar Bayan, Atwell Eric
UltraHAL	2007	Robert Medeksza
Elbot	2008	Fred Roberts
CHAT-L	2009	Bruce Wilcox

Table 6: Chatbot List (Shawar & Atwell, 2007; Loebner, 2013)

There are further a lot of homepages for searching specific chatbots or simple adding new one as well as categorizing them. One of those very interesting homepages is (AI4US Ltd., 2013) providing e.g. links to over 40 educational chatbots.

People want to use their language to communicate with other humans, also with computers. As Sharwar and Atwell (2007) quotes is the best way to facilitate *Human Computer Interaction (HCI)*, by allowing users "*to express their interest, wishes, or queries directly and naturally, by speaking, typing, and pointing*". This is a main reason why these systems were developed and still have a high demand. There are a lot of different term, have been used for chatbot. The homepage (AI4US Ltd., 2013) has a collection of over 160 with explanations for the most common, such as virtual human, conversational agent, virtual agent, virtual assistant or artificial conversational entity.

2.3.4 Natural Language Processing

Natural Language Processing (NLP) is theoretically motivated area of research and application that explores how computes can be used to understand, work with and manipulate natural language in text for or speech to get a useful output. NPL tries to get knowledge on human being in way that that information can be used for applications and programs, working with language, understanding the information and processing further useful tasks out of the data (Chowdhury, 2003). NLP try to achieving a human-like language processing, therefore is NLP considered as a discipline within *Artificial Intelligence (AI)*. However, the word "processing" should not be replaced with "understanding" because the goal of understanding could not be accomplished yet. In the early days the research field was called Natural Language Understanding (NLU), but this term has changed, because a full NLU would be possible to paraphrase an input text, translate the text in to other languages, would be able to browse the content means it would be possible to answer questions on the text and draw inferences from the text (Liddy, 2001). It is common to use statistical methods for different proposes, e.g. generating grammars and parsing or word sense disambiguation to parse content better. Nowadays a lot of sources available supporting this approach, such as WordNet or international consortia and research groups which offering free text and speech databases.

2.4 Summary

The first part of the chapter showed that it is hard to give a best practise approach for requirement elicitation. It is applied at by many different work fields and an important technique in the project management process. It is a technique to elicited requirements from customers and all stakeholders. It is important to be as best as possible prepared before it comes to one of the used methods, such as interviews, to get a maximum output from it. As example for an interviews preparation, it is a good idea to have a basic set of questions ready to do best.

E-Learning is a very strong growing market and offers a lot of opportunities to spread knowledge. Still quit new and exciting parts of it are virtual classrooms and serious gaming. Even when such new technologies cannot replace completely the classic way of teaching, they are still very supportive and helping to gather new knowledge as well as practice skills. They also have positive effects on motivation and help to learn content better. Serious gaming by itself tries to use more gaming aspects to even more stimulate the motivation from users. One big part of this approach is to make worlds as authentic as possible. This includes that this systems are offering computer programs that act like other humans, based on artificial intelligent (AI). The so called chatbots are one instant of this AI. This faces developer with new problems, because the most difficult part of creating of a bot is writing original, clever, sometimes humorous, interesting dialogue that will keep the client entertained and entranced (Wallace R. S., 2005). And it is even harder to write a chatbot which can be used as teaching tool.

MUVEs are one of the best ways to make a program as realistic as possible. It is possible to simulate classrooms as well as workflows. The entertaining gaming industry draws a great picture of the success of such MUVEs. Games like Minecraft, WOW or Guild Wars 2 are most played games. It is important for the project to get a good environment to be able adopt parts which make such games so successful. Therefore open VEs have been discussed and evaluated. Open Wonderland seems to be one of the best choices. Open Wonderland is written in Java, therefore it was important to get a Java based chatbot framework to simulate AI. Different solutions have been discussed and the online free hosting system Pandorabots looks like a perfect solution, where developer need only little effort to generate a good output.The next chapter will give the project conceptual model and design with deeper detail knowledge on used technologies.

3 Requirements and Design

This chapter discusses requirements and design decisions for the implemented project. There will be, in general, first the project vision, followed by the resulting requirements and then the design decision, to make the reason for the decisions clear. To underline this, all existing modules are described, as well as all new modules needed, for simulating a serious game styled world in a multi user virtual environment platform. The purpose of this chapter is to give a deeper look into the systems designs and show more details on the project requirements. It will start with the project motivation, followed by the requirements, then the design main implementation decisions with a general project structure, then more details about OpenWonderland and existing modules and finally more about the used chatbot system.

3.1 Project Vision

The main problem of RE is the labour and time intensive classic methods which are still the common way to teach RE showed in 2.1.3. The literature review showed that there are only a few projects targeting on an E-learning supported method and the most advanced is made in HTML. This brings up the main ideas and aim for this projects.

The main aim of the project is it to develop and evaluate selected portions of fictional case study that is performed within a 3D Virtual World environment. This will be used then as an exercise tool for students taking a system analysis and design class. The developed program will simulate the fictional case study in a way that students will practice using system analysis requirements elicitation techniques, such as interviewing, questionnaires, document review, form review, or observation, in order to find out about (an) organizational problem(s) and to determine the requirements for a new or modified information system that would help to solve the (fictional) organizational problem(s) (Venable, 2012a). The project initiator Venable further stated as following: "*The intention is to provide a more realistic experience than the traditional written (text and figures) case study commonly supplied with textbooks on systems analysis and design and to thereby support better learning as well as more realistic assessment of student skills in requirements elicitation.*" (Venable, 2012a)

To complete the experience and to get a better learning output, other techniques must be added like game based aspects for the motivation and highly collaborative interaction possibilities as well as tasks that the students also really have to use them.

The problem to be solved in relation to RE

Students taking an introductory *Systems Analysis and Design (SAD)* unit or course need to learn requirements elicitation skills, such as the following techniques:

- Interviewing
- Questionnaires
- Document review
- Form review
- Observation

This techniques have a different level of implementation difficulty as well as importance for RE in general. Therefore the main focus is on interviewing and document reviewing.

It would be a great benefit to practice all of the different requirement elicitation techniques, but this is not enough. In order to learn these skills well and to develop the ability to judge when and how to use each of them, as well as to choose between them, it is useful and actually important to be able to evaluate how well the students have developed these skills (summative assessment), and to give them a feedback to further their learning (formative assessment), means an first prototype must have such student evaluations functions.

To address all the issues discusses in 2.1.3 the aim of the project is to develop a virtual case study, in which a virtual 3D world is used to provide a realistic, virtual environment which a student or group of students can carry out a whole range of requirements elicitation activities. "*The longer term goal of the research is to develop a realistic, immersive experience for the students that is comprehensive in its ability to provide learning and experience for students in the whole suite of elicitation skills described above, and possibly others (e.g. participant observation using simulated existing systems).*" (Venable, 2012a)

The next topic will go now into the requirement analysis for the project. It will identify the requirements and list them.

3.2 Requirements Analysis

The above explained vision gives a great impression on what the research should focus. The requirement analysis needs to be considered not only requirements coming from the project idea and vision, but also pedagogical based requirements connected to the main requirements and requirements leading from social phenomenons. These are motivation for using a program e.g. games like World of Warcraft, Minecraft, Heroes of Newerth or Diablo 3 who are dominating gaming charts (DFC Intelligence, 2013) or the success story of Second life, which increased the registered number of users from 100,072 to 2,251,416 in only twelve months (Hebbel-Seeger et al., 2013). This will bring up the need for a 3D virtual world environment and it should be obvious that picking up the reasons for the motivation for many players on the planet to play the above named games is a big target and vision for every educational program developer. This is the reason for a deeper look into game theory and the resulting serious gaming. Possibilities like to login into the program from where ever and at any time the students want is a great way to give students the chance to practice at the

time that is best for them. This does not mean that the first target is to develop a system which can be accessed always, but it shows the possibilities for further versions and the future perspective. The technology which is improving to a higher level from day to day open new perspectives like before named, but also others like artificial intelligence. To combine this systems and use the advantages for educational approaches is a great ambition for researchers.

The requirements for the project can be divided in different parts regarding to the related literature and research area background. It further has to be considered two main stakeholders, the students and the instructors. Next will be explained the main groups of topics where the requirements will come from and introduces the requirements.

1. Requirement in the Application Domain Analysis Techniques

The project vision showed that a main goal of the project is to implement a platform where students can learn an practice requirement elicitation techniques in a way close to reality without a lot of cost and time effort, simulating for every student the same basic environment without having a lack of motivation of an interviewee or other circumstances which can harm the learning results as stated in the project vision. Therefore it is important to simulate different requirement techniques as realistic as possible to make-up a new way for learning and practicing requirements elicitation in a non harming and low on cost and organizational effort based teaching system. There are a lot of different requirement elicitation approaches, as discussed in 2.1.1. It is not possible to picture every technique in the first version of this research, thought that there are also many various ways of using the different techniques. Thus there must be set a core area of techniques want to be practices. The discussion showed that one of the most important techniques is the personal interview with the different stakeholders. Therefore the main focus has to be set at this technique. Also important is the document reviewing process. In a first version of RIVALE must be a basic document review interaction chance to give the student a second technique to get in touch with. As discussed the first step to a successful requirement elicitation process is always to be as prepared as possible for every used technique, for that reason students need a place in game where they can get more information and help to prepare their self's for the interview and the document review.

For students the main aspect regarding requirement elicitation will be the efficiency and quality of the interviews and provided documents for review as well as the sources where they can gather information and tips for handling the tasks. For instructors the main focus will be that students can learn and deepening their requirement elicitation skills without restrictions to cost, time, motivation and other problems appear from other named practice projections. The following simplified functional requirements can be identified:

1.1. Interview simulation
    1.1.1. The program should provide interview possibilities
    1.1.2. The interaction should be in natural language
    1.1.3. the interview should be hold in a way that students can extract information

1.1.4. Instructors should be able to extend, insert and edit system answers

1.2. Document review

1.2.1. The system should provide PDFs, DOCs, pictures or other representation possibilities to present documents

1.2.2. The documents should contain content where student can extract information

1.2.3. Instructors should be able to extend, insert and edit the documents

1.3. The system should be less time, cost and labour expensive then classic RE teaching methods

1.4. The system should provide extra information on RE techniques provided with documents, slideshows and movies

1.5. The instructor should be able to give the students information on RE techniques and other help with text chats and voice chats

1.6. Students should be able to learn RE collaboratively

2. Required Artificial Intelligent Pedagogical Agent

There are different ways to simulate an interview in a computer program. There is definitely the need for an intelligent pedagogical agent who is providing the necessary infrastructure for knowledge and information exchange. The most realistic would be an artificial intelligent pedagogical agent using voice input and output to communicate and being represented in world with an avatar. The current voice recognition systems are not this far developed, at least not the open available systems, as that this would work well enough. The second best simulation technique is done with an artificial intelligent pedagogical agent represented by an avatar using a text input chat. As discussed in the section 2.3 there are several such bots, called chatbots available today. They are implemented in different coding languages with different level of technical progress. The literature review 2.3.3 gives an overview showed that there is a quite long history of developing chatbots using natural language in a free from as input form. To make it as realistic as possible the chatbot has to be represented by an avatar so that every interacting person has the feeling of talking to a real person.

For students the most important aspects will be how easy the interaction is with the chatbot and that it is easy to get the requested knowledge. It might also be useful to have a log on the chat to be able to look up, already asked question, answers without spending a lot of time on writing down notes by hand or other methods. For instructors is especially an assessment method important to support as well as judge students for the used questions and techniques. It is also essential to have log of the chats to be able to upgrade the chatbot knowledge base. The following simplified functional requirements can be identified:

2.1. Interview realism

2.1.1. It should be a good real world representation

2.1.2. It should be hold with a fictive person, represented by an avatar, to give a feeling of interviewing a real person

2.1.3. The answers should fit to the question asked
2.1.4. There should only be a minimum of wrong answers to questions

2.2. Availability and support
2.2.1. The interview knowledge base should be always available and editable
2.2.2. It should be easy to edit the knowledge base even with poor coding knowledge
2.2.3. Updates should be available for all students at the same time
2.2.4. The underlying hosting system should be cheap

2.3. Interaction
2.3.1. It should be easy to start a interview
2.3.2. It should be possible to review interviews for students but also for instructors

3. Required Assessment Functionality

It is important for students as well as for instructors to have effective assessment functions, such function can save a lot of time, effort and teaching labour in systems. It is essential for a successful knowledge acquisition. E-Assessment has been discussed in the section 2.2.4 and showed the two basic types summative and formative assessment. Summative assessments should be done by evaluating the chat logs as well as the out coming documentation on the fulfilled tasks. An automatic system for recognizing the completion of a task and evaluating it would be required. One way formative assessment can be done would be to see the chats students have direct. Another way is to be able to follow students wherever they go and see what they do and how they do it. System functions where students can contact the teacher direct are also useful solutions. The following simplified functional requirements can be identified:

3.1. Summative assessment support
3.1.1. It should be possible to evaluate chatlogs from students
3.1.2. The system should have task which have a outcome which can be check in the end
3.1.3. An automatic system to recognize students process with task would be good

3.2. Formative assessment support
3.2.1. Observe students while interviewing should be possible for instructors
3.2.2. It should be possible to see and observe the way and actions students take also beside interviewing

4. Required Serious Gaming Approaches

The literature review showed the success of games using a virtual world more than once. Number like the 25$ billion revenue in 2011 (ESA, 2013) and the still exponential growing market is great indication in what way future learning products should go. Therefore the RIVALE prototype should have game based approaches to see if that can improve the educational targets. Gaming approaches

e.g. can be a high score board of the students having the best results in the programs. The following simplified functional requirements can be identified:

4.1. Gameification
  4.1.1. Task with point should be provided
  4.1.2. Leader boards should be available
  4.1.3. A story should be told
4.2. Design
  4.2.1. The visualization should be increased
  4.2.2. It should contain challenging creativity aspects
  4.2.3. It should be a good real world representation

5. Required 3D Virtual Environment

The requirements before have identified already different functionalities a multi user virtual environment needs or must have an extendable system to develop this functions. The literature review 2.2.3 has reviewed different, especially good for educational use, virtual 3D environments with their basic features, core system architecture and system requirements. A core requirement is that the simulated world is easy to use and it takes a minimal time to learn the core functionalities needed to interact with the different parts of the system. This will not only raise the student's motivation to use the system, it also will lead to a better outcome at the knowledge acquisition in a shorter time. The following simplified functional requirements can be identified:

5.1. World usage
  5.1.1. Instructors should be able to insert, edit and delete world content (not only documents)
  5.1.2. Students should be able to insert, edit and delete world content
  5.1.3. The client should be easy to handle
  5.1.4. The system should be self explaining

6. Non-Functional Requirements

Summarizing the requirements for the environment and the other significant parts, the main non functional requirements for the platform are as following:

6.1. Multi User

It must support, up to an average class size, user interaction at the same time. This would save instruction labour and makes it possible to practice and learn in bigger groups.

6.2. Collaboration

Collaborative function to fulfill task in a team and to be possible to communicate with each other in text based way as also by voice. To work on tasks functions like collaborative text editors, notepads and others must be provided. This raises the

students understanding, their motivation and spares instructors attention when problems occur, because the students can help each other.

6.3. Interaction

Students as well as instructors should be able to interact with elements in the world in various ways for a more realistic experience and to raise student's motivation.

6.4. Visualization

The virtual should support visualization function in an way that the developed world is believable and enjoyable. Especially the visualization but also the simulation will have a great impact on the outcome and his realism.

6.5. Usability

The RIVALE prototype must be a user-friendly environment. It should be easy to understand and to interact with to motivate not only students but also instructors to work with the system. This will lead to better outcome thru more efficiency, less used time for learning how the system work and reduced failure.

6.6. Extensible

It should be easy to extend the program for developers as well as for normal instructors. This is important to open ways for futures work on other prototypes.

6.7. Performance

Performance problems should be minimal. This can be done when the platform has a curtain level of scalability. Therefore the virtual environment should offer different server spaces, what are easy to handle for instructors without the need of an expert.

3.3 Conceptual Architecture

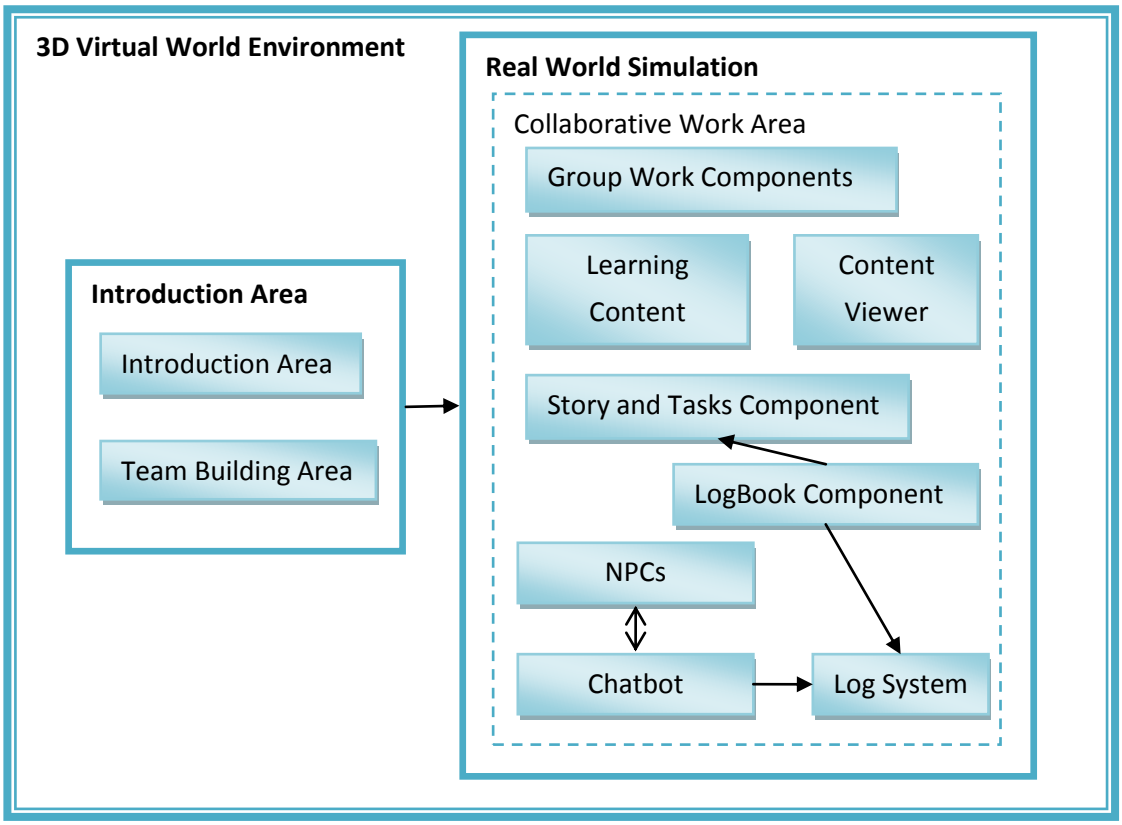

Figure 12: Conceptual Architecture

Figure 12 shows the conceptual architecture. The first RIVALE prototype should manly provide an artificial intelligence agent used as a chatbot, an introduction area for students who never used the program before, some story and task to get some game aspects as well as a type of guideline and this all should be possible in a collaborative way in a 3D virtual world. The students should be able to work on one task together, walk around in a group or alone, elicit and gather information and share them in the group. All this should be possible in the virtual world using just the provided tools, such as the group work components, and no other ways of communication and exchange. Therefore a module simulating an interview has to be developed which has a knowledge base and a connected log system to save the chats. Also a tool to tell the story and to give the students a task overview has to be developed in combination with presenting the chats done by the student (here “LogBook Component”).

The following two activity diagrams give the basic activities the above named modules have to support from point of view of the two main stakeholder, student and supervisor.

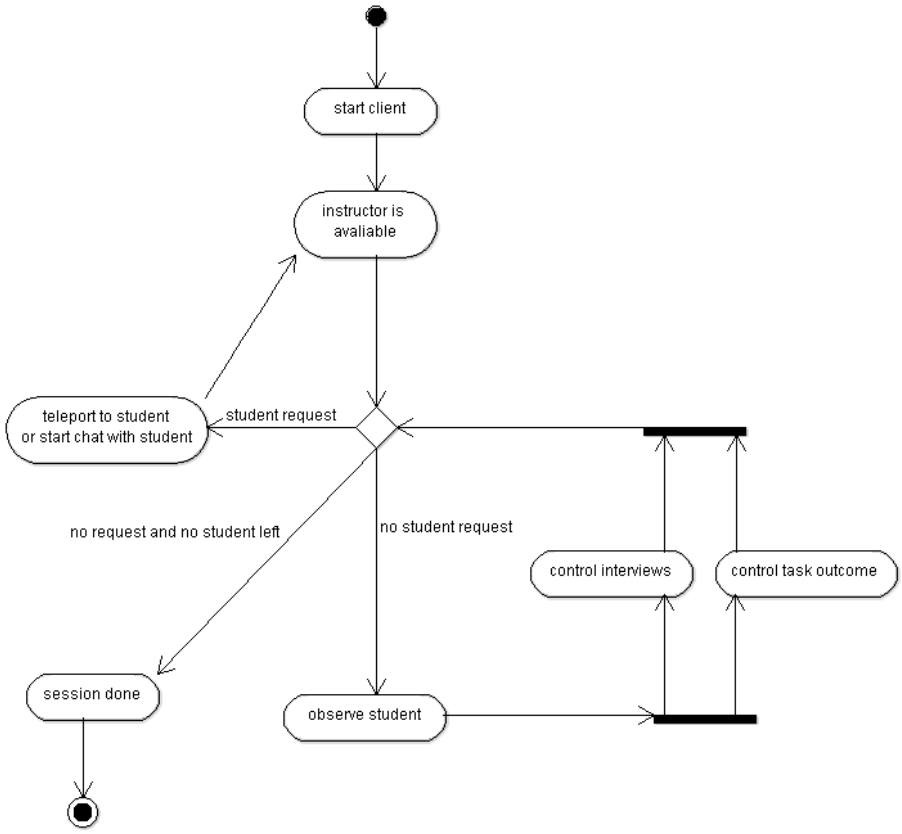

Figure 13: Shows how RIAVLE should work for supervisors

Figure 13 shows how the work process for a supervisor should look like. A supervisor enters the world like every student and will be represented by an avatar like everyone else. He can inform the students that he is free for new requests for help from students. Students can then contact the supervisor ingame via the collaborative chat system. If there is no request, the supervisor can observe student behavior, how they are doing in the interviews and the findings of the single student and the groups by using the log system and task component.

Figure 14 shows how a possible work session for a student looks like. The student enters the world by a client program. Some students might have used RIVALE before so they can skip the tutorial or make it in the introduction area. After this the students need to find team members in the team building area. He will go then to the company office and get the story and the tasks. The task can then be done alone or as team. If there is a team then they will use the collaborative chat system and the group work components to communicate and share the information they find. Then the student will move to the NPCs and interview them. The interviews will be saved via the log system and can be opened again with the logbook component. The student might then share the outcome and finish the task.

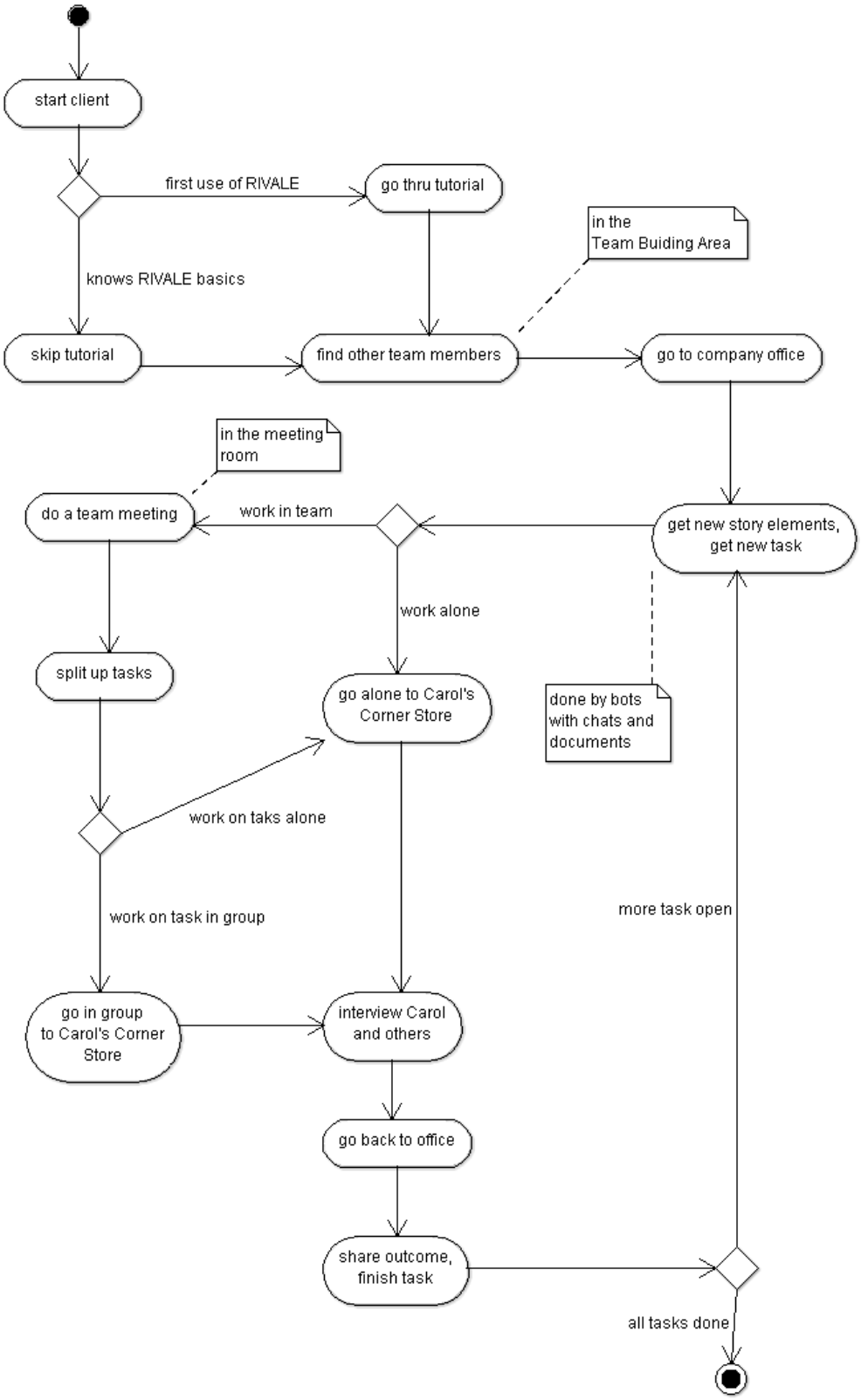

Figure 14: Shows how RIAVLE should work for students

3.4 Design decisions for the first Prototype

Based on the literature review, the requirements and the conceptual architecture the following design decisions were made (a more detail discussion progress is outlined in 3.5 – 3,6):

- Open Wonderland has been selected as virtual 3D environment
- A.L.I.C.E. as one of the most advance systems is taken as chatbot base and Pandorabots is chosen as hoster because of the easy access, good support and because it is free
- Three worlds must be developed
    - A start world which introduce the user to the system and allows the user to build teams
    - A office world where the user get task information, a story and extra information on requirement elicitation techniques
    - And area where the students use the techniques
- Three modules must be developed to extend the OWL functions
    - A chatbot module to simulate conversations
    - A story telling module
    - A module to present logs and other information to the user

The next chapters will explain in detail the reasons for the design decisions and will give more information on used techniques.

3.5 Open Wonderland

A main point of the design and implementation of RIVALE is to set up all functionality in a 3D virtual world environment. Especially such environments which have already a certain level of support for educational approaches are for interest. The chapter 2.2.3 reviewed and compared some of these environments such as Simple Life, OpenSim and Open Wonderland. The open source project Open Wonderland provides already a lot of collaborative tools like Whiteboards, Sticky Notes or chat functions what fit perfect with the requirements.

Open Wonderland History and basic technology has been descript in the topic 2.2.3.2. This topic will do a deeper look into OWL, it features, modules and capabilities in general way as well as in developing point of view.

The following topics point the OWL features into the deep. It will pick up main design states of OWL, explain them a bit more and explain afterwards how they will be used and extended in the project. The chapters introduce also a list of available modules and capabilities which are planned to be used in the project and will describe in which context.

3.5.1 OWL Features

The following list gives an extracted overview on the most significant Open Wonderland features for the research, provided at the Open Wonderland Webpage (OpenWonderland, 2012a). This does not mean that every installation has those features. They are depending on the installed applications and utilities on the server. This features can be extend by additional modules shown in following topics.

Basic Features

- Access worlds via URL
  - Java Web Start technology used to start Wonderland client
  - Links can go to specific location within a world
- 2D Application Sharing
  - Multi-user Java applications written using the Open Wonderland shared application framework. Examples include a whiteboard, PDF Viewer, Sticky Notes, and others.
- Communication
  - Group and private text chat
  - Private voice chat with one or more users
- Avatars
  - In-world avatar configuration
  - Low fidelity avatars for use on lower-end systems
- Placemarks and Portals
  - Associate placemarks with any object for easy navigation
  - Create portals to locations on same server or on different servers

World Assembly & Content Creation

- Support for industry-standard modeling tools that export to COLLADA
  - Content creation can be with Google SketchUp, Autodesk Maya, Blender, and others
  - Drag and drop import of .kmz (Google Earth format) and .dae files
- Arrange objects using in-world tools
  - Move, resize
- Drag-and-drop existing content
  - Images (.gif, .png., .jpg)
  - Documents (.pdf, .svg)
  - Multi-media (.mp4, .mov, .wmv, .ogg)
  - 3D models (.kmz, .dae)
- Add "capabilities" to any object
  - Includes security, container, placemark, clickable link, audio, and others

Extensibility

- 100% Java toolkit enables modifying or extending Wonderland client or server
  - Extensive developer documentation and tutorials
- Modules (akin to "plug-ins")
  - Mechanism for packaging and sharing Wonderland extensions
  - Most extensions can be accomplished using modules

System Administration and Management

- In-world administrator tools
  - Force mute
  - Force exit
  - Become invisible
- Web-based server administration console
  - Manage servers, groups, users
  - Monitor servers
  - Install / remove modules
  - Save and restore "snapshots" (saved worlds)

3.5.2 Open Wonderland Architecture

It is important for the success of the project to understand the design and how to develop in OWL. The main OWL architecture was described in the literature chapter. An OWL scene can be represented by a scene graph. A scene graph has nodes. In OWL nodes are called Cell. A Cell stores data about a visible object in the scene, such as a Sticky Note or a user avatar. These objects can have 3D or 2D visualisation as well as other features, such as interactive behaviour and the ability to communicate with other parts of the OWL system. These Features are called components. It needs also rendering functions to give cells positions and so on. This rendering code for the cell is written in a cell renderer class that we have to create. The cells, components and renders are all created with code. But Wonderland can use also other sources like audio, video, images animations and so on, inside the world (Finnigan, n.d.). A cell has two parts, a client piece and a server piece shown in Figure 15.

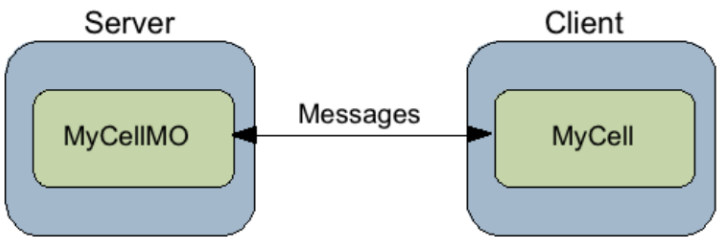

Figure 15: Cell Architecture (OpenWonderlandWiki, Open Wonderland Documentation Wiki, n.d.)

The server side makes sure that the cell has the same state all over the server, if needed. The basic for this is communication architecture in OWL. But not only for this has the OWL team developed a extensible communication architecture. This architecture has three main layers as shown in the Figure 16. The lowest layer is the protocol layer, which is used by all clients to select how the communicate with the server. The second layer is the session layer, which organizes Java-bases clients. The top layer is the connection layer, used by application to send applications specific data.

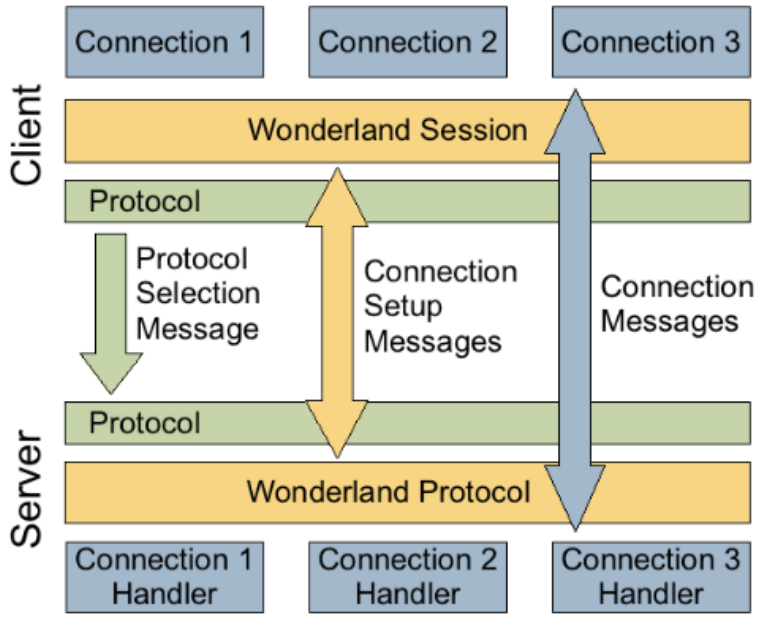

Figure 16: Layers of communications architecture (OpenWonderlandWiki, n.d.)

3.5.3 Modules

Kaplan and Yankelovich (2011) stat, they main target was to build an extensible toolkit rather than a fixed-feature environment. They aimed to let developers build quickly a highly customized world with task-specific applications. Therefore the OWL team was looking for a solution enable this broad range of extension. The out coming is a modular architecture based on open Java Components, developed by the team as well as by the community. The Project is structured with a small set of core services that mange the 3D world, including authentication, networking, content management and client rendering. All other features are built in modules. Because of the implementation of many core features in modules such as avatars, audio and shared applications, meant that they need to implement a comprehensive set of extension points to make this possible.

They core service layer provides the features that the Wonderland modules use. Extended core services, such as ability to load modules, enforce security and calculate real physics, are layered on top of the core as modules.

3.5.3.1 Module System

"*Wonderland modules are the mechanism for packaging extensions, including objects, capabilities, plug-ins, connections, and Web applications.*" (Kaplan & Yankelovich, 2011)

A module is a specially formatted Java archive (JAR). There are extra information and files added to the basic JAR attributes, such as metadata including the module name, version number and dependencies on other modules. The big part of a module is the data. The module data is divided by type, with each type represented as a top-level directory within the module. Each type is handed by the module system with a deployer. This deployer is in charge to unpacking the data and making it available to the correct system. (Kaplan & Yankelovich, 2011) give as example for a deployer "artwork", which is unpacked into a directory in the Web server where clients can download it. Other examples are: "*client code, which is also made available to clients via the Web server; server code, which is installed in the Darkstar server; and Web administration modules, which are deployed to the Web server using standard Java EE mechanisms."* Even the set of deployers themselves is extensible. Developers can use a new deployer contained in a module to custom content into other modules and so on.

There are much modular architecture advantages, including extensibility and manageability. But there are also downsides. Developing in this modular fashion way needs much more fragmentation to the code than normal. It can be difficult to figure out which module implements which feature. Furthermore, administrators can update, add, remove and actually make what every they want with a modules individually, module dependencies and versions become a management challenge. Despite this complexity, the module architecture allows to build an ecosystem of extensions around the Wonderland toolkit. OWL provides a Module Warehouse where developers can share their modules with others and they also host module repositories to share code (Kaplan & Yankelovich, 2011). The warehouse is split in different categories.

3.5.3.2 Manage Modules

To see all modules installed on a server, the server admin page has to be opened and "Manage Modules" on the left has to be selected. It will bring up a list with modules as shown in the Figure 17. There are four sections with functions:

- "Install a new Module" provides the ability to select a module JAR file to upload and install at the server at the next restart from the server.
- "Installed Modules" shows all modules currently installed on the server, their version, description and a checkbox to mark them for deletion.
- "Pending modules" gives a table of modules that are waiting to be installed. They will be installed during the next restart of the server.
- "Removed Modules" gives a table of modules waiting to be uninstalled, again they are uninstalled when the server is restarted or when a module that requires it is uninstalled.

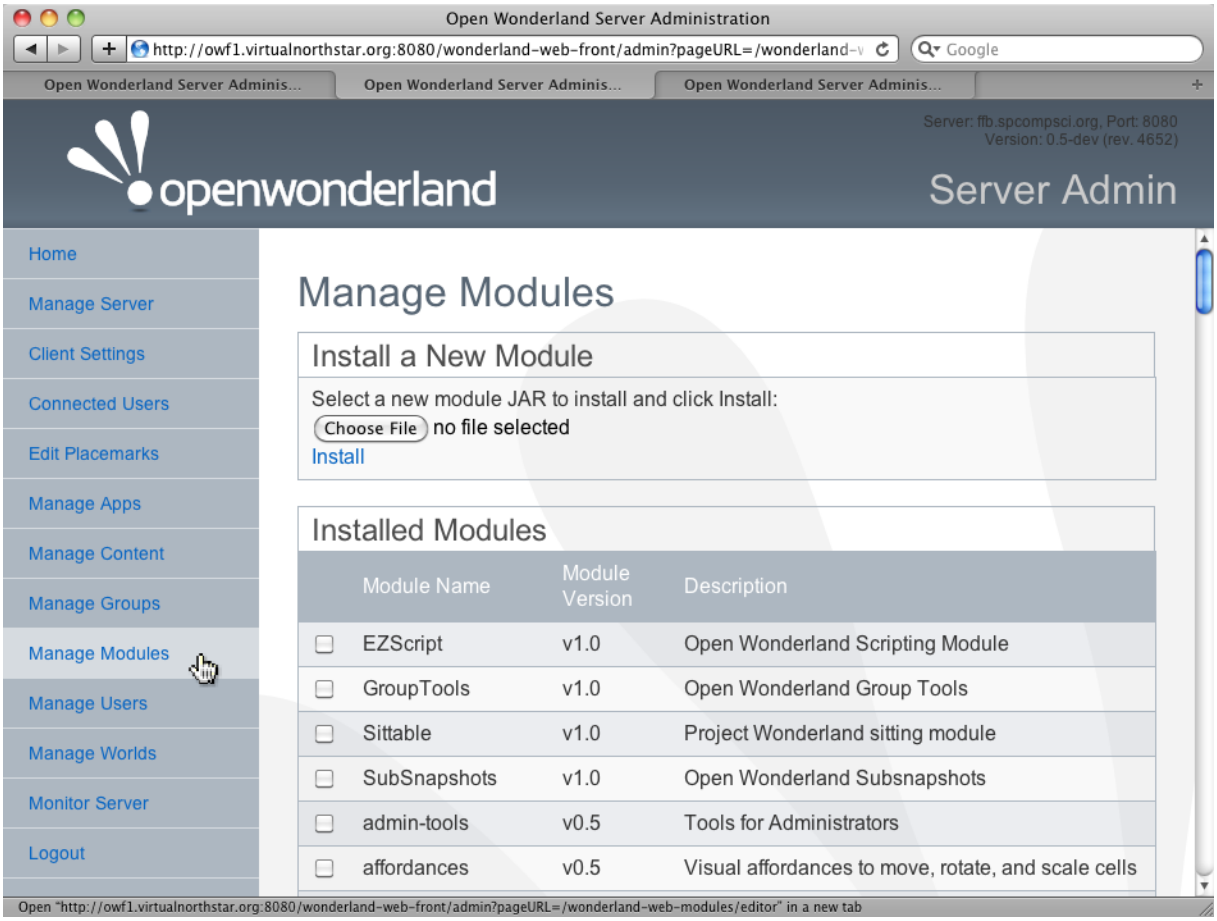

Figure 17: Module Manage (OpenWonderlandWiki, OpenWonderland Community Wiki, 2013)

3.5.3.3 Provided Modules and Community Modules

There are many available modules for OpenWonderland. Many are available in the module Warehouse (OWLWarehouse, 2012a), developed by the project developers as well as by community members. Another way to get community modules is the community SVN server. There are stable and unstable modules provided from version 0.4 and 0.5 This Topic will give an overview on modules planed to used or used in the implemented project. This will help to understand the design parts of RIVALE.

Collaborative Text Editor

The collaborative text editor (v0.9) was released 28/11/2012 under, like all OWL sources, GNU v2 license. It is a multi-user collaborative text editor and allows multiple users to update the same text. It supports only plain text modification and no formatting. The module can handle files with the extensions txt, log, html and xml. Dropping one of this into the world will create a new cell and open the file. Then it is possible to modify the file. This allows great teamwork and would be useful to collect information found from different team members.

NPC

Figure 18: Default NPC look (OWLWarehouse, 2012a)

This module allows to places NPCs (Non-Player-Character, illustrated in Figure 18: Default NPC look Figure 18) in the world. They get placed like the most other objects over the insert object menu. It is possible to move the NPC to other locations with a control panel. It is also possible to us a scripting component to move it and let it look more alive. There is a set of looks provided for the avatar. The default look is like shown in the figure above. This is the main module for a more visual simulation of chatting and talking with a human for the project.

Group Tools (v0.1)

This module adds a group tab to the Users list window. This will list all groups that have been set up using the edit-> groups in world or using the group section on the server admin console. It allow to see all groups and there members and give the possibility to send them a message. This is very good for instructors to give quick instructions to teams in the project (Illustrated in Figure 19).

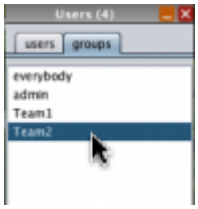

Figure 19: Group Tools (OWLWarehouse, 2012a)

Default World - Almost Empty (v1.0)

Figure 20: Almost empty World (OWLWarehouse, 2012a)

This is the module where all the other modules are based on. It provides the 3D module used to create the “Almost empty” default world that is included with the Open Wonderland v0.5 Preview 5 release. It contains a floor texture and a sky texture shown in Figure 20.

Poster

Figure 21: Poster in Open Wonderland (OWLWarehouse, 2012a)

The “Poster” (illustrated in Figure 21) module allow to place posters in Open Wonderland by using the Insert Object dialog. This will place a default poster with “Hello World” at the inserting position. It can be modified by right click and using the “Set Text…” context menu point. The upcoming window shows the poster text. It is possible to use HTML (restricted by the limits of Java Swing Components). This allows more complex posters.

StickyNote

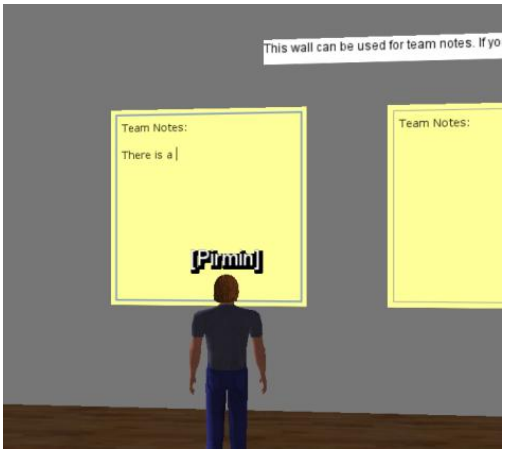

Figure 22: Sticky Notes

The Sticky Note (illustrated in Figure 22) is a module to share short information in world. They are also inserted with the Insert Object dialog and are placed in front of the user. It can be modified in different ways like representation color and basic stuff like size and position. It is a nice module to give user quick information or to select key words. Many code parts from Sticky Note are adopted from the Whiteboard module. There are further some project based on Sticky Note, extending the

functions, such as “iSocial Styled Sticky Notes (v1.0). This module adds two new types to the existing sticky note, a Styled Sticky Note and the Section Sticky Note.

Whiteboard

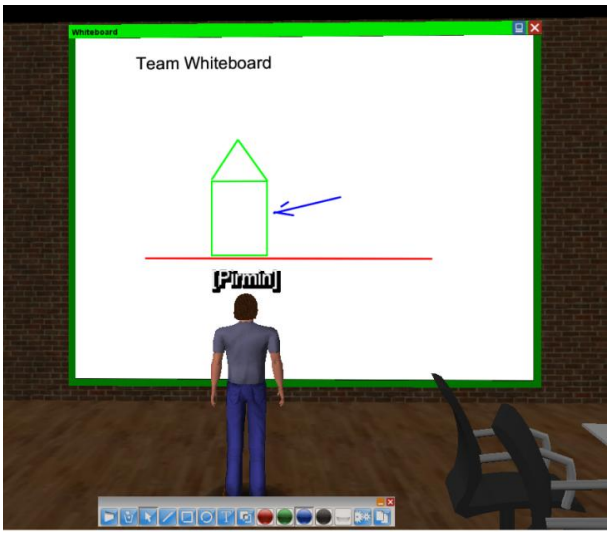

Figure 23: Collaborative Whiteboard

Whiteboard (illustrated in Figure 23) is a multi-user collaborative tool for editing text and painting together. It is also inserted with the Insert Object dialog and it is placed, as the Sticky Note, in front of the user. It is also for the project a nice tool to share ideas and findings with team members.

PDF Viewer

The PDF Viewer (illustrated in Figure 24) allows users to drag and drop PDFs into the world, show them and interact with them. It is possible to load, zoom in and out, go to other pages and so on. This is a good tool to give users additional information on a topic provided by instructors.

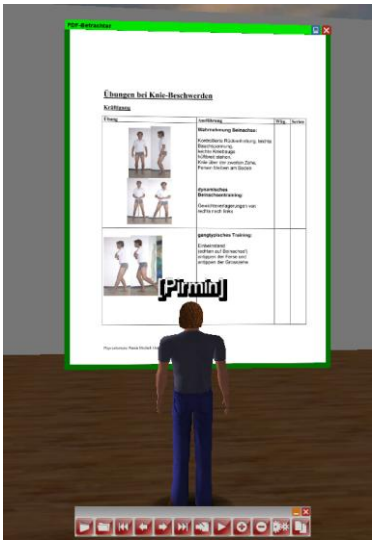

Figure 24: PDF Viewer, open and shows PDFs

3.5.4 Capabilities

Capabilities are features or actions that can be applied to any OWL object. They have to be installed on the server and then can be simple attached to any object by using the properties menu. There are a set of common capabilities installed with every OpenWonderland server installation. Others are available to add from the Wonderland Module Warehouse (OpenWonderlandWiki, 2011a). Capabilities are very important for the project. Some developed modules are implemented as capabilities. More details on that in the implementation chapter. A lot of the common as well as other capabilities have been used to build up the project and to make the use case as authentic as possible.

There are three capabilities what are automatically added to every new Wonderland object, as well as a default Model Component capability what is added to all 3D objects, provides options like “Lightning Enabled”, “Picking Enabled” and to set the deployed module URL. Also all 2D objects have a default Best View capability, descript in the next section. The three basic capabilities are “Basic”, used to set the object name and some properties like “this object can be selected”. Then “Position” allows the user to modify the location of the selected object in world and “Performance” displaying the performance characteristics of the selected object.

Capabilities are called Component on the programmatic level. Therefore developing a capability is “developing a new cell component”. As said before, a capability can be added to every object. An object is a cell in OWL.

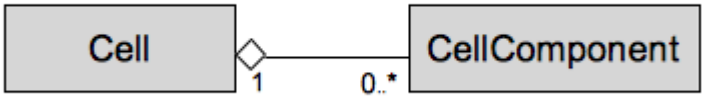

Figure 25: Cell - CellComponent relation (Slott, 2012)

Every cell can have zero or more cell components attached (see Figure 25), once attached they add the functionality of the cell. A cell component has a similar structure as a cell, consisting out of three parts, the server part, the common part and the client part. The following table, provided by the Author (Slott, 2012) illustrates the objects and interfaces in the parts, for the cell component mechanism and their analogous cell objects and interfaces, shown in Table 7.

Package	Cell class	CellComponent class
Client	Cell	CellComponent
Common	CellClientState	CellComponentClientState
Common	CellServerState	CellComponentServerState
Server	CellMO	CellComponentMO

Table 7: Cell Component Classes (Slott, 2012)

The Components can make use of server and client plugins and can have associated CellComponentFactorySPI classes. It is also possible to add a properties panel to be used with the object editor and might have an associated display class, what was also done in the project.

Slott (2012) describes more details about the implementation part in his tutorial "DevelopingNewCellComponent" as following. *"On the server, a CellComponentMO can either be "live" or "not live", similar to the state of CellMO. A live CellComponentMO indicates that the CellMO in which it is attached is present in the world. When a CellMO transitions from the "not live" to "live" state (or vice versa), it iterates through all of its attached CellComponentMO objects and sets their "live" state accordingly."* If the object changes his state, a function is called and can be used to call other methods, for example to reload a connection to a Pandorabot.

"Every CellMO can have at most one CellComponentMO of a particular type, as determined by the java.lang.Class of the CellComponentMO object. A list of (Darkstar references to) all components may be obtained via the CellMO.getAllComponentRefs() method; an individual component is obtained via the getComponent() method by giving a java.lang.Class. The CellMO object also supports listeners on changes to the attached components on it via the ComponentChangeListenerSrv interface."

3.5.4.1 Most Common Additional Provided Capabilities

This section provides a list with description of the most common provided capabilities where are the most used in this project. First a short description how to attach this function to objects.

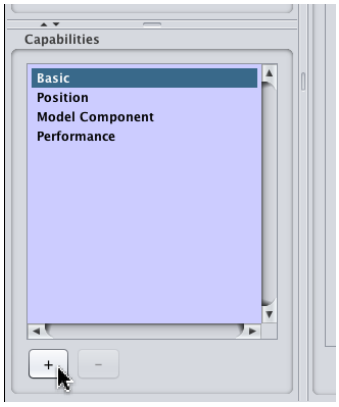

Figure 26: add new capability (OpenWonderlandWiki, 2011a)

The properties window lists all attached capabilities of a selected object in the left bottom corner (see Figure 26). The figure shows the initial state with the three automatic added capabilities, Basic, Position, Performance, and the default "Model Component". To add a new capability simple press "+" and the following window appears:

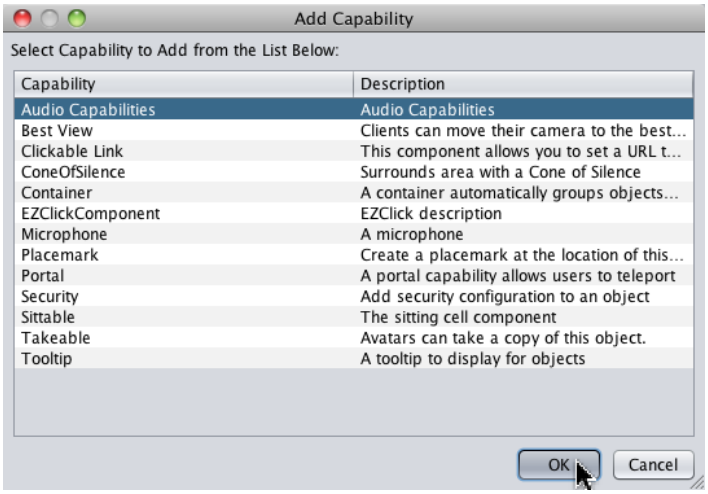

Figure 27: Add new Capability window (OpenWonderlandWiki, 2011a)

In the "Add Capability" window (see Figure 27) a list of all available and not yet added capabilities of the selected object is provided.

Best View Capability

Figure 28: Best View Capability (OpenWonderlandWiki, 2011a)

All 2D objects, including images, PDFs, whiteboards, sticky notes, and X11 shared apps, automatically have the best view capability applied. It can be added to all other cells additionally. This capability has no property sheet. It adds a "Best View" object to the context menu as shown in Figure 28. Selecting this will show the object on the full screen. This is a important function to read content better on objects and is used in the project implementation for a better usability. It is possible to use the scroll wheel on the mouse to zoom in and out to adjust the object presentation. By simple using the "Esc" key on the keyboard, it will return to the default 3rd person camera (OpenWonderlandWiki, 2011a).

Audio Capability

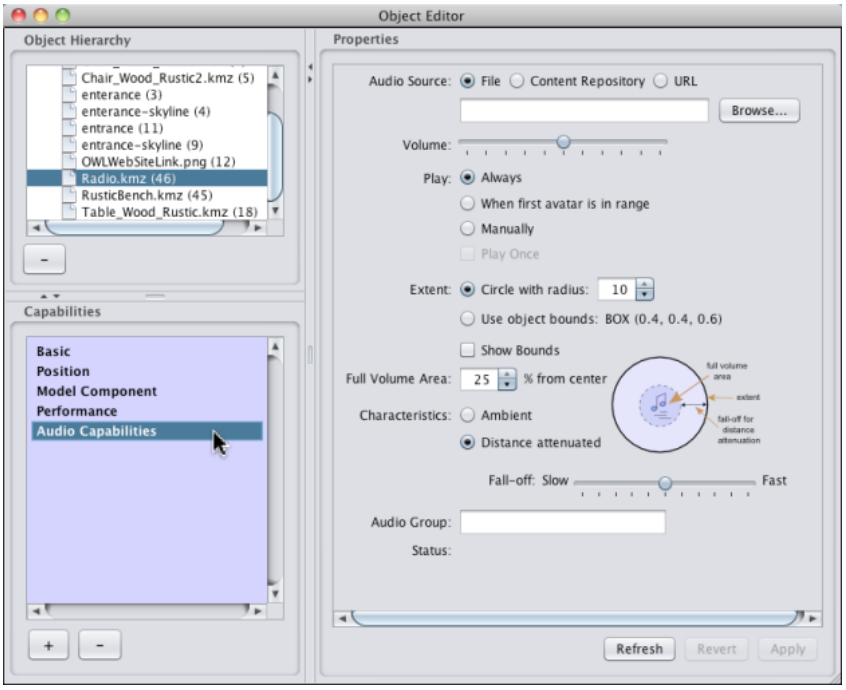

Figure 29: Audio Capability (OpenWonderlandWiki, 2011a)

The Audio capability allows to either add audio to any object in the world or to stream it (see Figure 29). Audio can come from three different sources, a file on the computer, a file stored on the server in the Content Repository or a file on a server accessible on the Internet via a URL. When a source from the local computer is chosen, the file will be uploaded to the OWL server and available for everyone. There are a bunch of other options such as volume, volume area and other audio characteristics (OpenWonderlandWiki, 2011a). Audio can and will be used in further project version to get a more realistic world.

Clickable Link Capability

Figure 30: Quick Link Capability (OpenWonderlandWiki, 2011a)

Quick Link (see Figure 30) is a simple capability that allows adding a URL to an object, what can be called and followed by clicking on the object. It will open another window asking to open an external browser and to navigate to the linked source (OpenWonderlandWiki, 2011a). This is used to provide additional info in the project.

Cone of Silence Capability

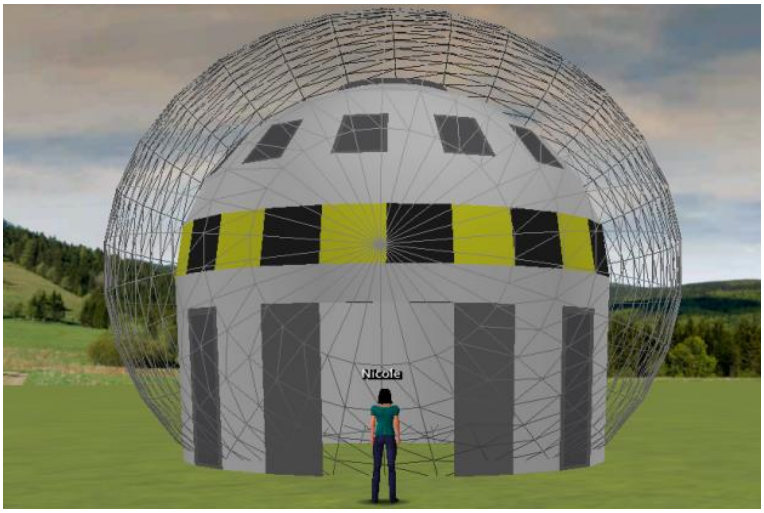

Figure 31: Cone of Silence Capability (OpenWonderlandWiki, 2011a)

Cone of silence (see Figure 31) is a sound proved area. Avatars in the area where this capability is applied cannot be heard outside the bounds of the "cone". There are options to allow sound to come in or to make it entirely sound-proof. There is an option in the properties sheets allowing to "show bounds". This helps to modify the cone area as wanted, as illustrated in the picture above. There are different options to set the bounds, make the bound fit to the object boarders or to set a radius or box as border. When now avatars come into the area, the names change from white to gray, proving a visual indication of who is inside the bounds of the cone (OpenWonderlandWiki, 2011a). In the project, cones of silence are used where different teams use the same area to separate them and make a private team discussion possible.

Placemarks Capability

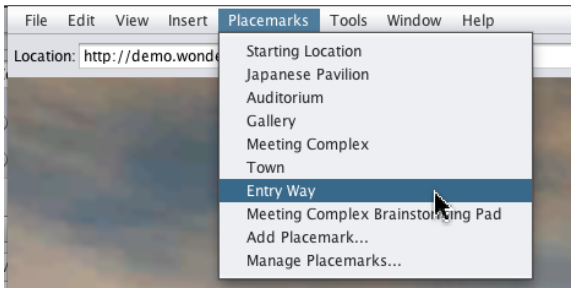

Figure 32: Placemarks Capability (OpenWonderlandWiki, 2011a)

Placemarks (see Figure 32) allow fast traveling thru the maps and places on a server. They can be added to any object like the most other capabilities. After adding the placemarks capability to an object, the name entered will be show in the placemarks menu as show in the picture above. All users on the server can see the placemarks. If selected, the avatar get teleported to the object. When moving the object, the placemarks is moved too, to the new coordinates of the object on the server. This is manly used from instructors and teachers assisting the students in world.

Portal Capability

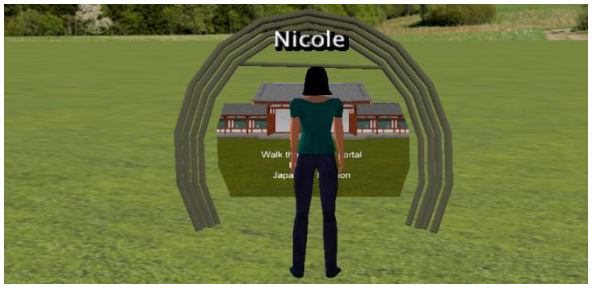

Figure 33: Portal Capability (OpenWonderlandWiki, 2011a)

Portals (see Figure 33) are like placemarkers a method to teleport from one position to another. The difference is that they are not explicit show on a menu and that they basically work the other way round. The capability is added to an object and as soon an avatar run into the object, he get teleported. The place where the person is teleported to is specified in the properties sheet. It is possible to teleport to a different server, also set in the properties sheet. The target place can bet set direct with X,Y,Z coordinates as well as choosing an predefined placemarks. There can be also set the look direction and a audio source for the sound made when using the portal. In the project, this will be used to allow users to get from one area to another such as the different team areas or tutorial world.

Security Capability

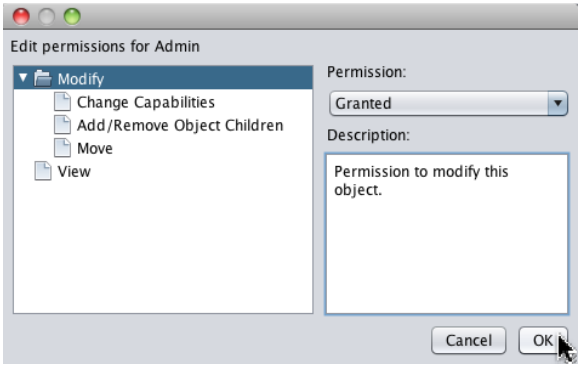

Figure 34: Security Capability (OpenWonderlandWiki, 2011a)

This capability (see Figure 34) allows adding security to any object. Once added there is the function to allow users or groups to interact with the object depending on their permission given. The users and groups added with special permissions to an object, are hierarchical ordered. The top user can grand or remove permission from all person under him. The second user can has power on everyone but not from the person above him and so on. This capability is important for the project. With it users can not “destroy” the world or make stuff they should not do. The capability is easier to use with predefined groups of users. There are two ways to make groups (see Figure 35). From inside of

Wonderland, it is possible to select “groups” out of the Edit menu. This will pop up a window helping creating, editing and removing groups, shown in the following figure.

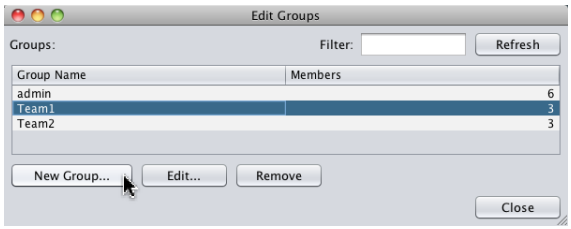

Figure 35: Adding Groups Capability (OpenWonderlandWiki, 2011a)

Sittable Capability

Figure 36: Sittable Capability (OpenWonderlandWiki, 2011a)

Sitting capability allows adding a sitting animation to an object (see Figure 36). The sitting position can be adjusted using the properties sheet. It is possible to define the sitting direction with a heading value as well as the sit offset, what is the amount the avatar is sitting forward or backward in relation to the sittable object. To let the avatar sit on a chair, simple us the “Sit Here” command seen by right clicking on an object with the sitting capability. If this is clicked, the avatar walks automatic to the chair and sits down. There is a tip on the OWL wiki for it: “*For fun, walk a short distance away from your chair and select the “Chase Camera” from the View menu before you sit. There’s no special path following implemented yet, so be sure there are no obstacles in your avatar’s way before issuing a sit command.“* (OpenWonderlandWiki, 2011a)

3.5.4.2 RIVALE Modules

The research on OWL showed what additional functions have to be developed to please the project requirements. The following list gives the modules which have to be developed and a short description.

- **Chatbot**
  A chatbot module is needed to provide an interface for communicating with the educational artificial intelligent agent.
- **Story and Task Component**
  A module to add story and task to the environment has to be developed to serve the student with information.
- **Log Book Component**
  A kind of a logbook function is need to collect the story and task information as well as the chat log information and present them in a proper way.

The decision to develop the following three modules results from the used case study for the story described in 3.6.

- **Store**
  Carol's Corner Store is a module representing a world where the student will do his interviews and other requirement elicitation techniques.
- **Instruction Area**
  Entry world is a world where people start. It will contain the tutorials as well as first RIVALE instructions .
- **Office**
  The area where the students will get more tasks, help from bots and the possibility to learn more and get tips about requirement elicitation.

3.5.5 A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) and AIML

As already mention in the literature review in 2.3.3., ALICE is one of the most advanced chatbots, developed 1995 by Dr. Richard S. Wallace. There are still a lot of improvements in ALICE and just recently, 19th of April 2013 (news post at alicebot.org), ALICE 2.0 was published, designed specifically for mobile devices such as handsets and tablets. Regarding to the ALICE community (AI Foundation, Inc., 2013a), ALICE 2.0 repairs the shortcomings of the original, because it can obtain factual information of external services and other bots, providing up to date resources. On the other side, the AIML categories required are reduced. This makes it even more important to use web based and provided ALICE bots. The Figure 37 illustrates the conceptual A.L.I.C.E. architecture. The Interface can be from any kind like web based, Java or others.

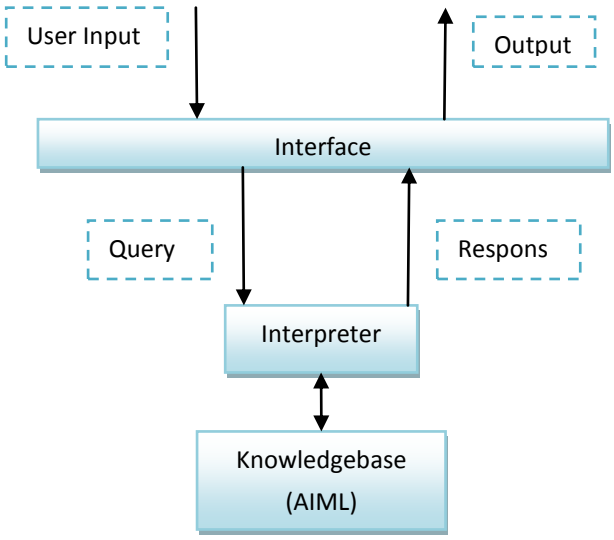

Figure 37: Conceptual A.L.I.C.E. Architecture

ALICE is based on *Artificial Intelligence Markup Language (AIML)*, using simular style like XML language does. It was released und the GNU GPL licence and so many chatbots have been released with this technology. AIML consist of data object called AIML objects, which are containing topics and categories, surrounded with the “<aiml>” start and end tag, showing end and start of the document. It can contain parsed or unparsed data. The topic is an optimal top-level element and contains a set of categories related to that topic. The category (tagged as “<category>”, see Listing 1) marks the “unit of knowledge” in Alicebot’s knowledge base. Each category provides the rules for the matching and answer of each node, consists of the pattern (tagged as: “<pattern>”) and the template (tagged as: “<template>”), containing the respond to a user input. These are the most important tags. There are about 20 additional more tags often used in AIML files and it is possible to create own “custom predicates”. Listing 1 and 2 show the basic structure for an AIML knowledge base entry.

```
<category>
  <pattern>WHAT ARE YOU</pattern>
  <template>
    <think><set name="topic">Me</set></think>
    I am the latest result in artificial intelligence,
    which can reproduce the capabilities of the human brain
    with greater speed and accuracy.
  </template>
</category>
```

Listing 1 AIML basic tags example (AI Foundation, Inc., 2013a)

The ALICE community give the following description to Listing 1: *“In any case, if this category is called, it will produce the response "I am the latest result in artificial intelligence..." shown above. In addition, it will do something else interesting. Using the <think> tag, which causes Alicebot to perform whatever it contains but hide the result from the user, the Alicebot engine will set the "topic"*

in its memory to "Me". This allows any categories elsewhere with an explicit "topic" value of "ME" to match better than categories with the same patterns that are not given an explicit topic. This illustrates one mechanism whereby a botmaster can exercise precise control over a conversational flow.“ (AI Foundation, Inc., 2013a)

```
<aiml version="1.0">
   <topic name="the topic">
      <category>
         <pattern>PATTERN</pattern>
         <that>THAT</that>
         <template>Template</template>
       </category>
..
..
   </topic>
</aiml>
```

Listing 2: AIML example with <that> tag (Shawar & Atwell, 2007)

The “<that>” (see Listing 2) tag is optional and means that the current pattern depends on the previous chatbot response.

The ALICE pattern matching algorithm by his self starts with the normalization of the input. First the punctuation is removed, if appropriate the input is split in two or more sentences. After this the produced phrases set to uppercase. Then the interpreter tries to match word by word to find the longest pattern matching the input (Shawar & Atwell, 2007). Then the result is returned.

ALICE is open source and published under GNU GPL license. There are different implementations for different coding languages. An abstract is shown in the Table 8. There are more versions based on programs in the table available.

Name	Language
Program O	PHP/MySQL
Program E	PHP/MySQL
Program R	Ruby
Program P	Pascal
Program Y	Python
Program #	C#
Program Q	C++
Program J	C++
Program D	Java
ChatterBean	Java
Program W	Java

Program AB	Java
Program Z	Common Lisp

Table 8: Table of Alice Implementations

Open wonderland project is written in Java, therefore we are taking a deeper look into Java based ALICE and AIML versions and on code nondependent chatbot solutions.

3.5.5.1 Alice in Java

The following programs are Alice implementations based on Java code.

Chatterbean

Chatterbean is as ALICE and AIML licensed under GNU General Public License as published by the Free Software Foundation. It is written in pure Java and is fully compliant to the AIML standard 1.0.1. It is made to be self-contained and should be possible to run on every machine supporting Java, like all normal Java bases application, with no restriction to platforms and no depending on anything not provided. The program relies on a Java API as much as possible, providing a highly controllable application framework, easy to understand and supported by Javadoc code documentation, as well as code comments at important places. There is only limited interface available, this is open to implement for the developer. There are also JUnit tests included.

There are different samples implementation available like Ifurita, including features like the ability to learn new responses from users, can open a web page for the user and the bots pictures can change as a part of her response.

Program D

Program D is another Java implementation and seems to be on a good developed level. It allows multiple bots and connections in a single server instant, supporting J2EE as well as GUI applications. There are Drop-in listeners available for IRC, AIM and Yahoo. It provided a JUnit testing suite and JavaDoc. The project wiki says that program D is most widely used open source AIMPL bot platform in the world. This cannot be taken serious. The latest release is 4.6. of the 12th of March 2006, pointing out that 4.7, 5.0 is in development. The last modification of the wiki page was at the 18 July of 2009. It looks like there will not be further updates soon, or ever. However it is a running available system used in various projects.

Program W

Program W is still in the Alpha and is explained as following by Bush (n.d): "*Program W is an AIML interpreter written in Java. It extends Program D technology with new AIML tags that allow chatbots to query the WordNet lexical dictionary. Chatbots can use information about lexical terms (gloss synsets lemma hyponyms hyperyms holonyms meronyms antonyms) and evaluate existing relations between words*", published under GNU General Public License version 2.0.

Program AB

Program AB first released in January 2013 is developed by Richard Wallace, the developer of the first ALICE and is the newest Java chatbot using AIML 2.0. Wallace follow the naming tradition after letters of the alphabet (Program B, D, N), the name AB intended the fresh start with AIML 2.0. Program AB is mainly for creating chatbots and mobile virtual assistants.

Charliebot

Charliebot is a extension of program D, ALICE and ANNA v7.0, developed by Josip Almasi. It works on MAC OS x or any Java VM. It simulates a 58 year old computer enthusiast.

3.5.5.2 Alice Platform Independent

The following implementation is an ALICE implementation which is platform independent and can be used with every code.

Program Z – Pandorabots

Programs like Program D is build in Java and so only people with extensive computer Java skills could host or modify these programs. *Pandorabots*, also called Program Z, got available at 2002 and is an ALICE version anyone can use, offering free hosting, modifying and developing of chatbots. Another advantage is that there are free knowledge bases available, so a botmaster does not need to start from zero. Now a day thousands of bots made and published on the page. The letter Z is taken because the developer thought that this might be the last version anyone ever will need (Wallace, Tomabechi, & Aimless, 2003).

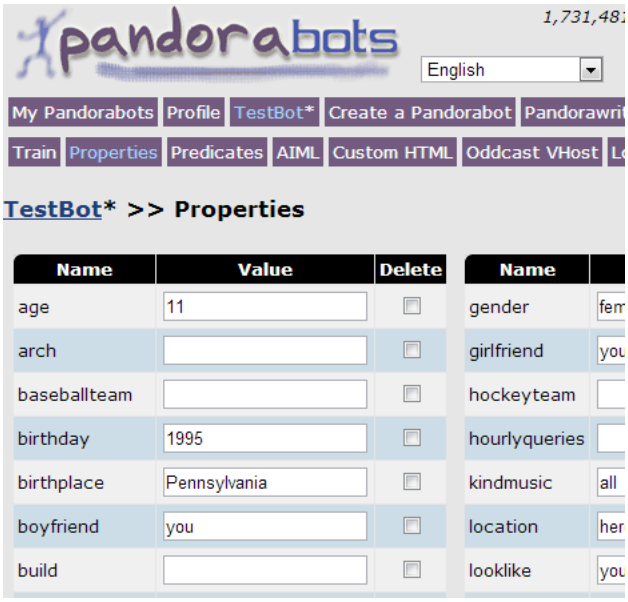

Figure 38: Pandorabot admin page

The botmaster are only using the browser to train and deploy there bots. The bots can simple be called over a HTTP call and because the most common programming languages have HTTP library, it can be used everywhere, it needs only an active Internet connection.

Pandorabots provides facilities for creating and storing knowledge in AIML format, shown in the Figure 38. The site provides different input methods for the different AIML features, such as a properties site or a predicates site.

Pandorabot Implementation and Architecture

Pandorabot is coded in *Common Lisp* and is running on a Linux based system. Common Lisp was taken, because Wallace and his team estimate that the same system in e.g. Java would take about 10 times of the resources to host the same amount of bots. Also, as Wallace et al. descript, because of the following reasons (Wallace et al., 2003):

> "1. *The ability to change the system while it runs,*
> *2. low-cost hosting systems, and*
> *3. very low-cost software development and deployment costs."*

Pandorabots has a multilingual interface and supports bots in almost every language. But this also shows limitations and problems of the program. E.g. there are different Japanese writings for the same thing. The system cannot recognize this and therefore the version has to be entered separately. It is also not possible to simple translate one language in another, because the meaning can be lost or there are sometimes simple no translations for the word available. This arises of the structure of the data which is in a fixed form. Pandorabots tries to overcome some of these issues with extra AIML tags (see Table 9).

Pandorabots Extension	Purpose	Remark
<condition name="X" value="*">	Branch with undefined value	One shot branch
<li name="X" value="*">	Branch with undefined value	used by <condition>
<li value="*">	Branch with undefined value	used by <condition>
<date locale="X" timezone="Y" format="Z"/>	date and time	Unix strftime format
<that index="M,*"/>	previous bot utterances	multi-sentence
<request index="N"/>	input request	multi-sentence
<response index="N"/>	output response	multi-sentence
<learn>	save AIML category	non standard
<eval>	AIML evaluation	expression inside <learn>

Table 9: Pandora AIML Tags (AI Foundation, Inc., 2013b)

3.6 The used Case Study

The used case study used for the prototype is an adoption from a case study developed by John Venable (2012b) (see appendix for source). The corner store owner Carol Chan has problems with her shop, mainly money because of efficiency lacks. Therefore her daughter Susan has forced her to introduce a new retail system. This is the point where the students get in action. The student will be a new trainee at a retail company. Because of different issues with the employees, the student has to work on Carols case. Therefore three different areas are needed. One is the introduction area to the RIVALE program. Then an area representing the retail company where the student get the tasks as well the possibility to gather more knowledge about RE. And last but not least and store area representing the customers work place where the student used the RE techniques.

3.7 Summary

The success of a project is heavily connected to a good design. Too create a good design, all requirements must be identified and analyzed. Followed by technology decisions and finally an architecture. The projects simulate the fictional case study in a way that student will practices and learn the most important requirement elicitation techniques. These are interviews and document reviewing. Main requirements are interview simulation, document review function, time, cost and labour saving environment, learning in collaborative way, interview realism, availability and support and different assessment functions to name only a few. All this need to be set in a 3DVW which supports collaboration and interaction for multiple users as well as visualization functions, must be extensible and performance and user friendly. And finally it should motivate the user with game based approaches. The conceptual architecture gives an introduction to components needed in the first prototype such as chatbot module, story module and notebook. Therefore OpenWonderland was elected as framework to develop the project in. OWL already comes with a lot of features and a module bases architecture which allows to simple add or remove functions on the server with only a few clicks. A deeper look into OWL was necessary to guarantee the success of the project. Further existing modules were checked to get an optimal supported virtual world. OWL already provides modules for collaborative text editing, NPCs, default worlds, posters, sticky notes, whiteboard, PDF viewer, and much more. And extra are the capabilities which can be added to every in world object and extend in this way there functionalities such as a best view, audio, clickable links, place markers or portal functions. Therefore three modules and three worlds are necessary to develop. Chatbot, Story Component, LogBook/Notebook and an entry world with a tutorial, an office world and Carol's corner store.

Finally ALICE was chosen as chatbots system and analyzed into deep. ALICE is one of the most advanced chatbots systems now a day. It is based on AIML files making up the knowledge base. AIML provides quit simple rule pattern and there are a bunch of projects in different languages providing an AIML translator. One of the simplest ways is the platform independent Pandorabot project were

user can host for free unlimited ALICE bots. The homepage also provides tools to edit the AIML files. All the project need is an active Internet connection.

4 Implementation Details

This chapter outlines the implementation of the different modules according to the requirements. First the system requirements for RIVALE will be outlined. Then the used existing modules will be listed and explained how they are used in the first prototype of RIVALE, followed by a description o the developed modules.

4.1 RIVALE System Requirements

The most requirements for RIVALE are the same as the requirements for Open Wonderland. Extra is that an Internet connection is required because of the used chatbot system which has his knowledge base on an extern server. In general the used system should be at least double about the minimum requirements. The reason for this is that OWL is not developed for worlds using many 3D models. Since RIVALE has gaming approaches, the world has more than usual models. This means, a better system will be a guarantee for a lag free study and practicing experience. The system requirements are shown in Table 10.

	Requirements
Computer Processor	1,5 GHz: recommended 3 GHz
Computer Memory	1 GB: recommended 2 GB
Graphic Card	Recommended Nvidia driver, 128 MB video memory, recommended 256 MB
Internet Connection	Required, DSL or higher

Table 10: RIVALE System Requirements

4.2 RIVALE Architecture

The Figure 39 shows the architecture in which the first prototype of RIVALE is made. Orange coloured fields are made for RIAVLE and blue are pre existing parts. It shows that there was the need to develop six modules for the client side, which are three worlds and three functional components. The OWL server provides the story source, the logs and the tasks which are new. The chatbot further communicate with Pandorabots which is using an ALICE system. For this system the bot specific AIML sources had to be developed.

The next topics will go in detail into pre existing used modules, the developed modules and all other developed components for RIVALE.

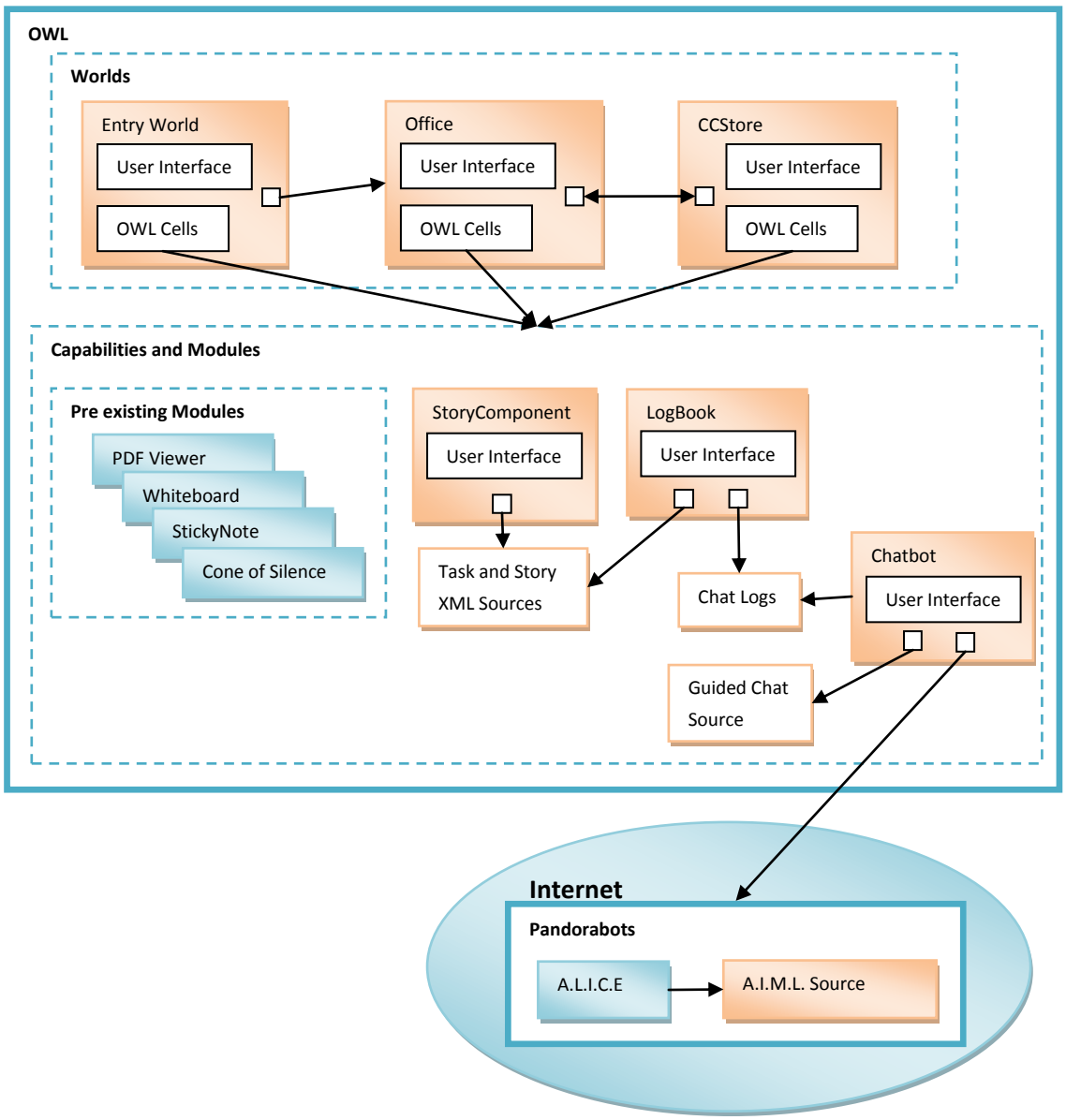

Figure 39: RIVALE Architecture

4.3 Open Wonderland Modules

The general module system and the basic implementation are shown in the literature review and the Requirements and Design chapter. The Table 11 gives an overview on pre existing used modules and how they are used in the first prototype of RIVALE.

Module Name	Usage
Sittable Capability	To make the world more realistic and provide different places with chairs where avatars can sit on.
Portal Capability	Used to connect the worlds and areas. In RIVALE

	portals are represented by a bus station.
Cone of Silence	In the team search area where people need to get a group.
Placemarker Capability	Used to identify the places the portals lead to and to make short travel possible.
Best View Capability	Used at all objects which can be read.
PDF Viewer	Used in the library to present extra information about requirement elicitation techniques to the students.
StickyNote	Used in the office to inform the students about his tasks. Also used in the meeting room to note things down for other team members.
Poster	Used at different places to give students information.
Whiteboard	Used in the team meeting room to collaborate together.
Default World	Used as base module for all worlds.
NPC	Used as representation form of the most bots in world.
Group tools	Used to make group access possible for administrators.

Table 11: pre existing used modules in the first prototype

4.3.1 General Developed Modules

As defined in the design chapter, three general useable modules, StoryComponent, LogBook and Chatbot have been developed. They are all designed and implemented with respect to performance, stability, usability and for user friendly behaviour. Those three modules will be explained next. Every module has a descriptions outlined with pictures followed by a class description implementation information.

4.3.1.1 Chatbot Module

The Chatbot module is the most important module in the RIVALE which was developed. It supports the chatbot functionalities used to simulate interviews done by a virtual agent. The chatbot module in the first version used by the first prototype only supports single user interviewing. This means that every user can only interview a bot alone and need to share the information after. The reason for this is to simplify the first version because the module does not need a simultaneous state with other students or supervisors and therefore the server side implementation was not necessary. The module has two different chat available which are working both with their own system.

The first type supports an ALICE implementation provided via Internet connected hosted on Pandorabots.com. This is the most convenient way to use ALICE, because there is no need to implement the ALICE system on the server and therefore no extra work for OWL hosts. The connection is made via simple HTTP calls done by the module. The other side is that the chatbots can be hold up to date in real time. Often systems have to be shut down to change content and normally especially extending databases needs a certain level of technique knowledge (an expert). With Pandorabots a supervisor can, without special knowledge, edit the AIML files while the students are working with the system.

The second chat type is managed and gives the user a list of questions to choose from by entering numbers. This type is used to tell a story in an easy way to the students. The questions are set with an extensible markup language (XML) file.

The chatbot system is implemented as capabilities. Every inworld cell can attach a chat. This was done to have the alternative to add the chat to e.g. a computer and simulate in this way a kind of a chat via a computer system. The Figure 40 shows the chatbot properties field. The name is important to set because after chatting the log will be saved and then connected to a user identified by the name of the bot. Here also the choice of Pandorabot or managed chatbot is set. The XML for a managed chat has to be placed in a user folder on the server and can be address with the standard OWL database path pattern e.g. “wlcontent://users/Pirmin/game/tutorial_chat.xml”. The “BotID” is provided by Pandortabots.com and set by publishing a bot on the site. Every object can only have one connected chatbot.

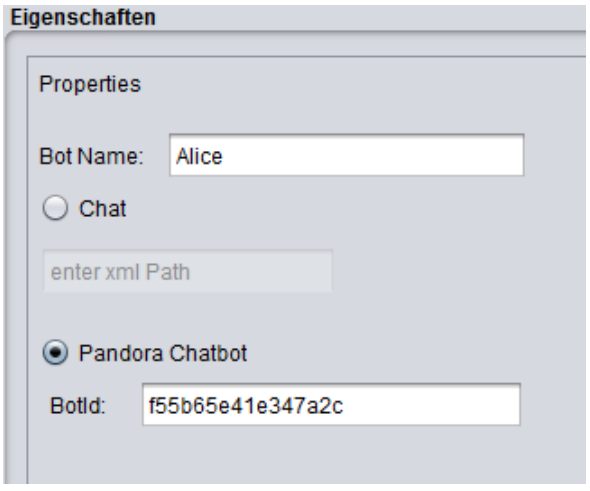

Figure 40: Chatbot properties field

The chatbot system is ALICE using AIML files as knowledge source. The system is provided by Pandorabots. Pandorabots provides free basic AIML knowledge bases with over 60,000 entries (AI Foundation, Inc., 2013a). RIVALE uses two approaches to improve the chatbots brain. The two

developed “characters” Carol and Susan have both separate and not connected sources. At the developing process from Carol an empty base was used and has now more or less only content related knowledge available. Susan on the other side is implemented with the example bot knowledge base from Pandorabots. She got, as said before about 60,000 entries, which are obviously not all related to the topic. The idea is to find out which developing approach is the better. Expected is that Susan might seem to be clever because she can talk about a lot of different topics and not only about the store. On the other side she might often respond unexpected and incorrect because it is impossible to update all 60,000 entries in a way the researchers would like it. More about this will show the evaluation. There are also five chatbots using the guided chat. They are manly telling the story and give some tips and information on tasks.

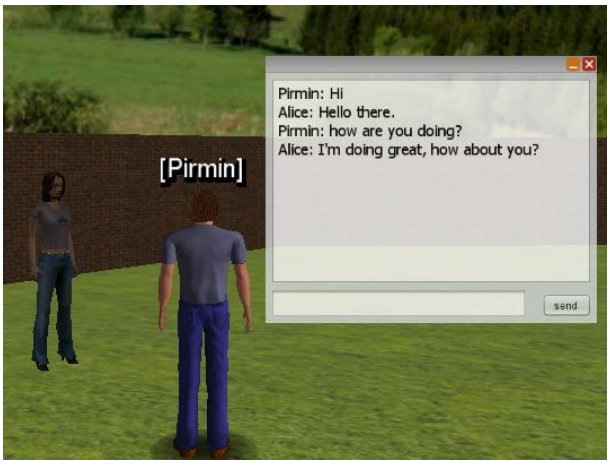

Figure 41: pandora chat window

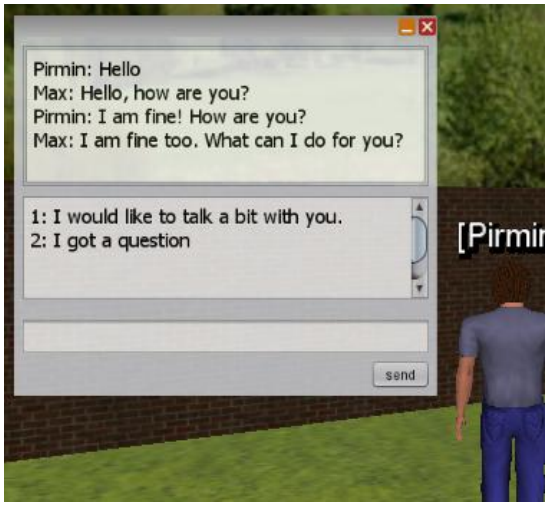

Figure 42: manged chat window

A cell which has a chatbot capability has always a “talk” item in the context menu as shown in Figure 45. This can be opened by right clicking on the object representation in the world. The two chat windows are made in the style of the normal OWL chat. The difference, as showed in the Figure 41 and Figure 42, is the extra space with the provided questions in a managed chat window.

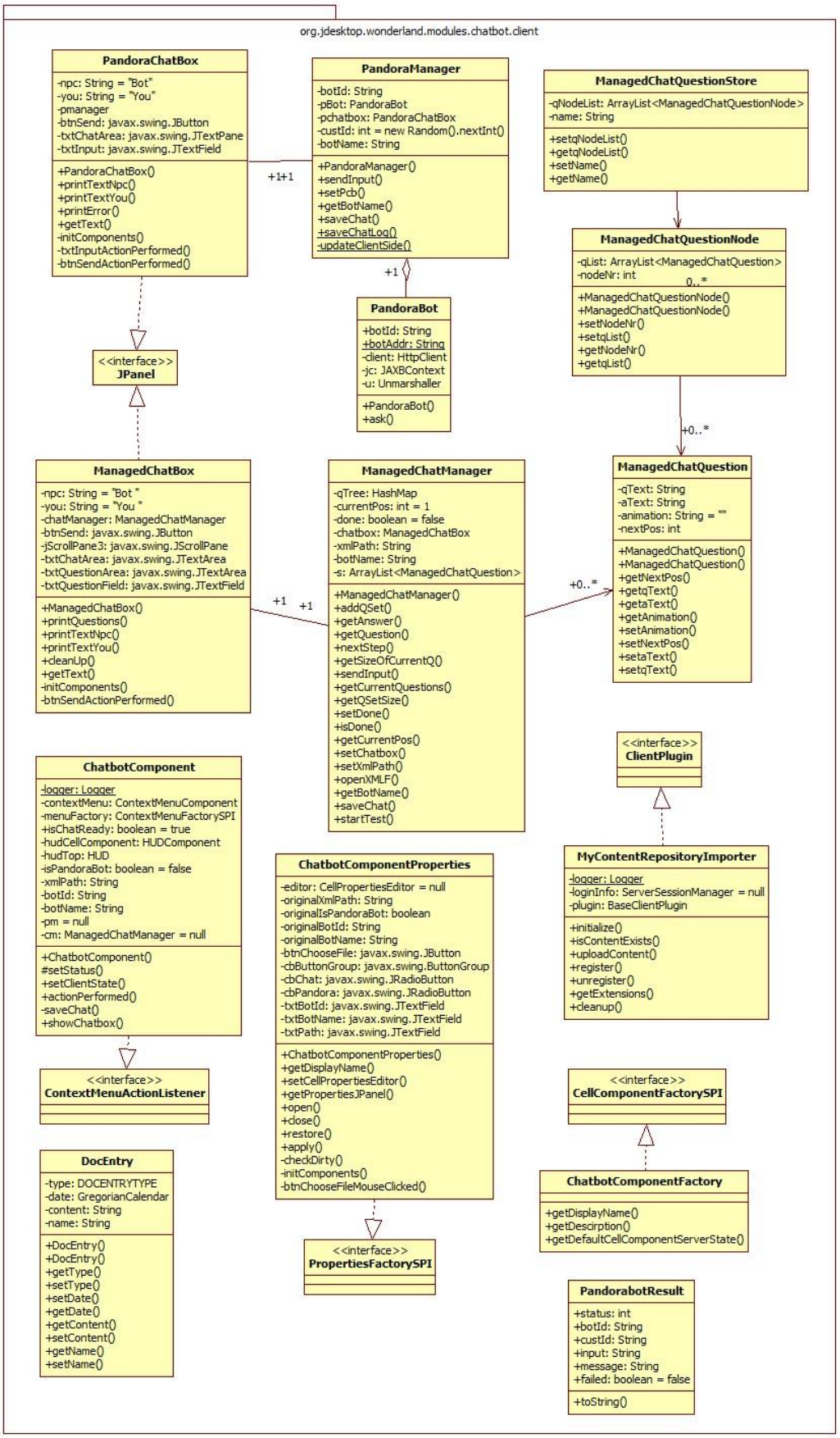

Figure 43: simplified client side class diagram from chatbot module

Class Description

The chatbot module has only a few classes in the common and server packages. This classes sharing are for storing the properties information. The main part is done on the client side. The module is like OWL defines a .jar file. The common and the server side classes are done in the normal OWL way with only small changes, thus this explanation will focus on the client side architecture of the chatbot module.

Figure 43 shows the client side class diagram. There are three parts in which the client site can be split. The general part is handling the module functions with the *ChatbotComponent* class as module entry point, the ChatbotComponentFactory class is responsible for naming the module and initialization the server state object and the *ChatbotComponentProperties* representing the properties sheet discussed before. Also important is the *MyContentRepositoryImporter* handling the upload from the chat logs. Every chat log is sent to the server and saved. This has two reasons, first, the user can then later log in from another computer and would still have the same log files in his logbook and second, the files are stored on the server for the supervisors to control the student's inputs as well as to learn what the chatbot has answered wrong and has to be updated.

The two different chatbots have their own classes. Depending on the properties settings is always one of the chats loaded. The managed chatbot is handled by the *ManagedChatManger* providing the answers for the client window *ManagedChatBox.* As explained before, the managed chat has an XML file defining the content for the chat. This is defined by the *DocEntry* class, loaded with the *ManagedChatQuestionsStore* and rapped into *ManagedChatQuestions*. The Pandorabot chat is handled by the *PandroaManager* class. This class handles the Pandorabot instant and all related actions to communicate with an online Pandorabot.

4.3.1.2 StoryComponent

RIVALE provide the student with a certain amount of tasks, system information and lines telling a story to give a feeling of moving in a real world. The different information types are again visualized by the Logbook described next. The *StoryComponent* is also implemented as a capability and can be attached to every in world cell (objects). The texts are saved in a XML format and mapped after to a *DocEntry* class. This is the same type and file as used from the log files, therefore only one file type is necessary for the StoryComponent, chatbot module and LogBook. The Listing 3 shows the XML structure. The type is provided, a date stamp, the content and the name how the file should be listed in the Logbook.

```
<DocEntry>
  <type>TASK</type>
  <date>2013-05-14T11:11:21.243+08:00</date>
  <content>- Find Team members.

Make a group, go to Edit-> Group, make your group and add your team members. (only one of the
```

```
group)

- Go to your office and talk to Ray Barnes.
        </content>
   <name>Starting...</name>
</DocEntry>
```

Listing 3: DocEntry file structure

The Figure 44 displays the properties sheet of the StoryComponent. It only provides a field for an XML path leading to the source file on the server. Again every object can only have one attached story file. The Figure 45 shows the context menu from a cell object having a chat capability and a story capability. After selecting “Add to Notebook” the content file will be downloaded.

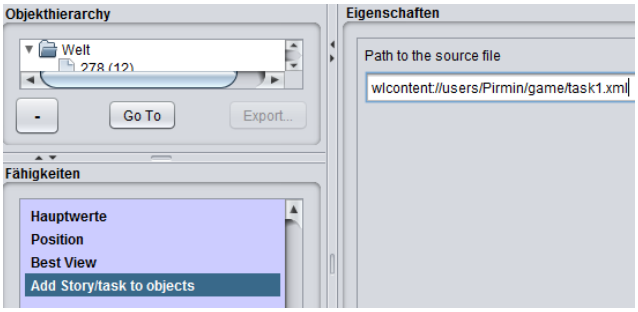

Figure 44: Illustrates the StoryComponent properties sheet

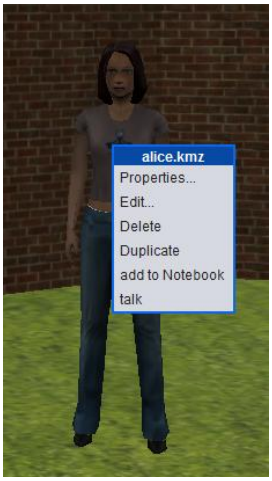

Figure 45: Showing a bot with attached chat and story capability

Class Description

The Figure 46 shows again only the client side from the StoryComponent because the common and server side are quite usual and only at the source path of the file to the cell XML. The *StoryComponentFactory* does again the initialization of the server state. The StoryComponentProperties is handling the properties sheet shown at Figure 44. The main class is the *StoryComponent* class with the function *getXMLFromServer*(). This function opens a connection to the repository, tries to connect to the user directory and then downloads the file. The file is then stored in the private wonderland user folder on the client machine.

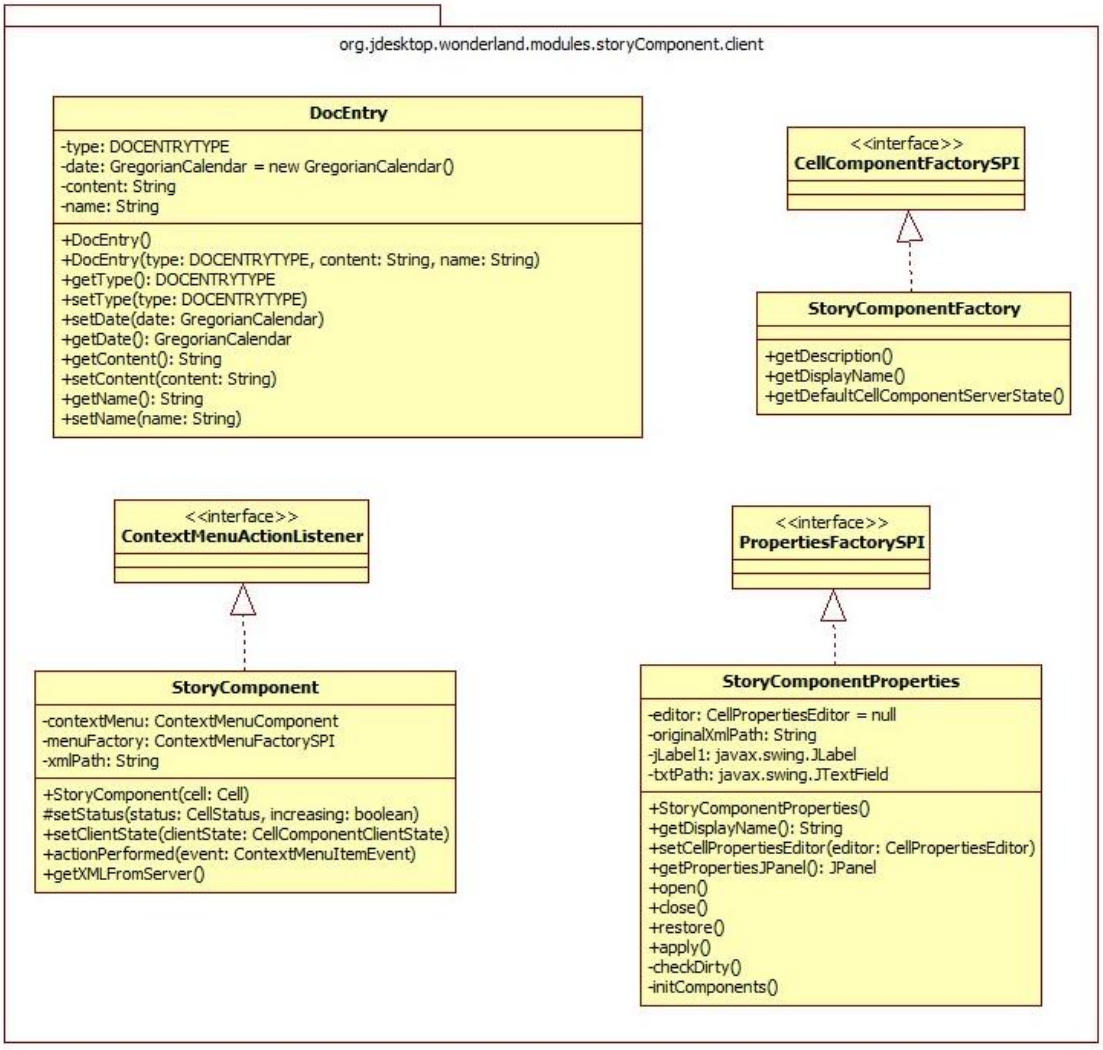

Figure 46: client side class diagramm from StoryComponent

4.3.1.3 LogBook

The *LogBook* is the module bringing the log entries, the story, the tasks and extra information together and represents it to the user. This module is important for the experience for the student. It is not only the tool to know the way to go, it is also the tool to know what the student has done by e.g. locking up the chat logs. It simplifies different aspects of practicing which would be time consuming in realty such as taking notes while interviewing. This time is saved and can be used for

practicing instead. The LogBook also has a “Note” tab which allows students to insert notes and save them. This can be used for e.g. “saving” the information a student got from a team mate without using an extra program or a paper notebook. The LogBook is opened over the top menu. In the category “tools” is an extra menu object called “LogBook”. On opening the tool the first automatic the “Logs” tab is selected as shown in Figure 47. It provides the students with a list of all chats done before and at what date. The “Tasks” tab as shown in Figure 48 is built the same like the “Story” tab. It gives a list where students can select what they would like to read. Task often also provide some extra information how to handle the task. The story tab (see Figure 49) presents entries telling story or give extra information on objects or thinks which have been read from the student. The “Notes”, as shown in Figure 50, tab give as described above a system to take notes in game.

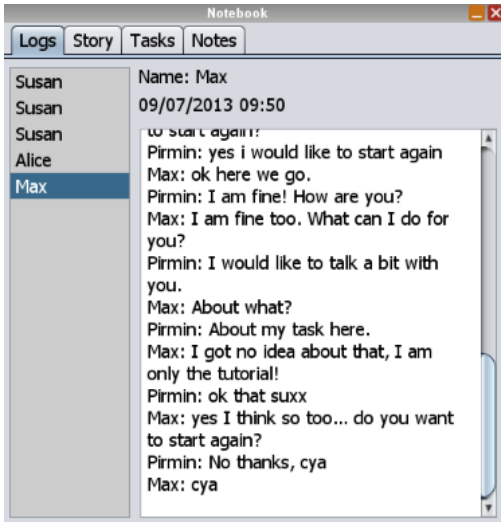

Figure 47: LogBook showing Logs tab

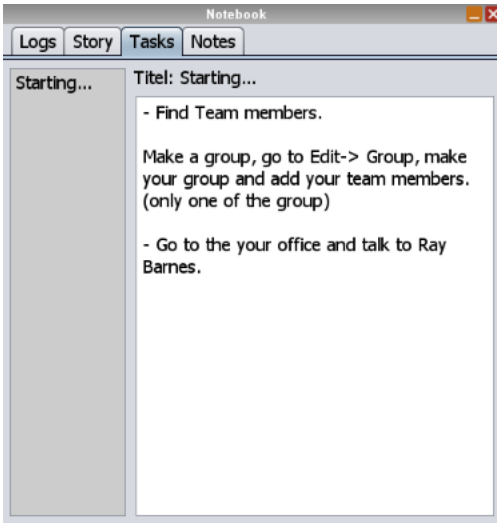

Figure 48: LogBook showing Tasks tab

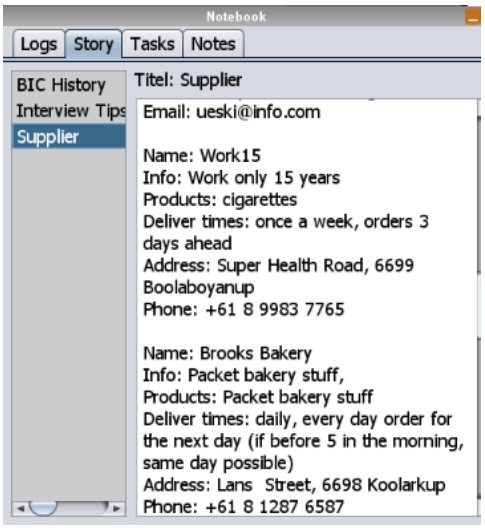

Figure 49: LogBook showing Story tab

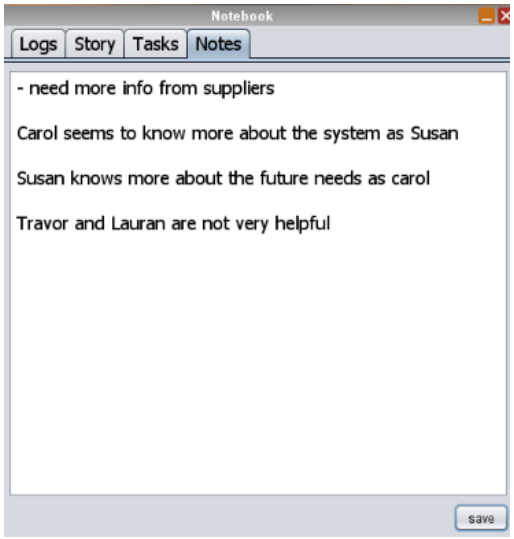

Figure 50: LogBook showing Notes tab

Architecture

The Logbook module is implemented as plugin like OWL defines. The example HUD and the existing tutorial were used as code base (OpenWonderlandWiki, 2012). The LogBook only has a client package. The Figure 51 shows the class diagram for the module. The main class is the *LogBookClientPlugin* responsible for setting up the new HUD windows and initialization the *LogBookManager*. The *LookBookManager* loads the content lists and fills up the lists afterwards. The Tabs are hold by a container class called *LogBook*. Every tab has his content represented in a JTextPane panel. This makes restricted HTML possible.

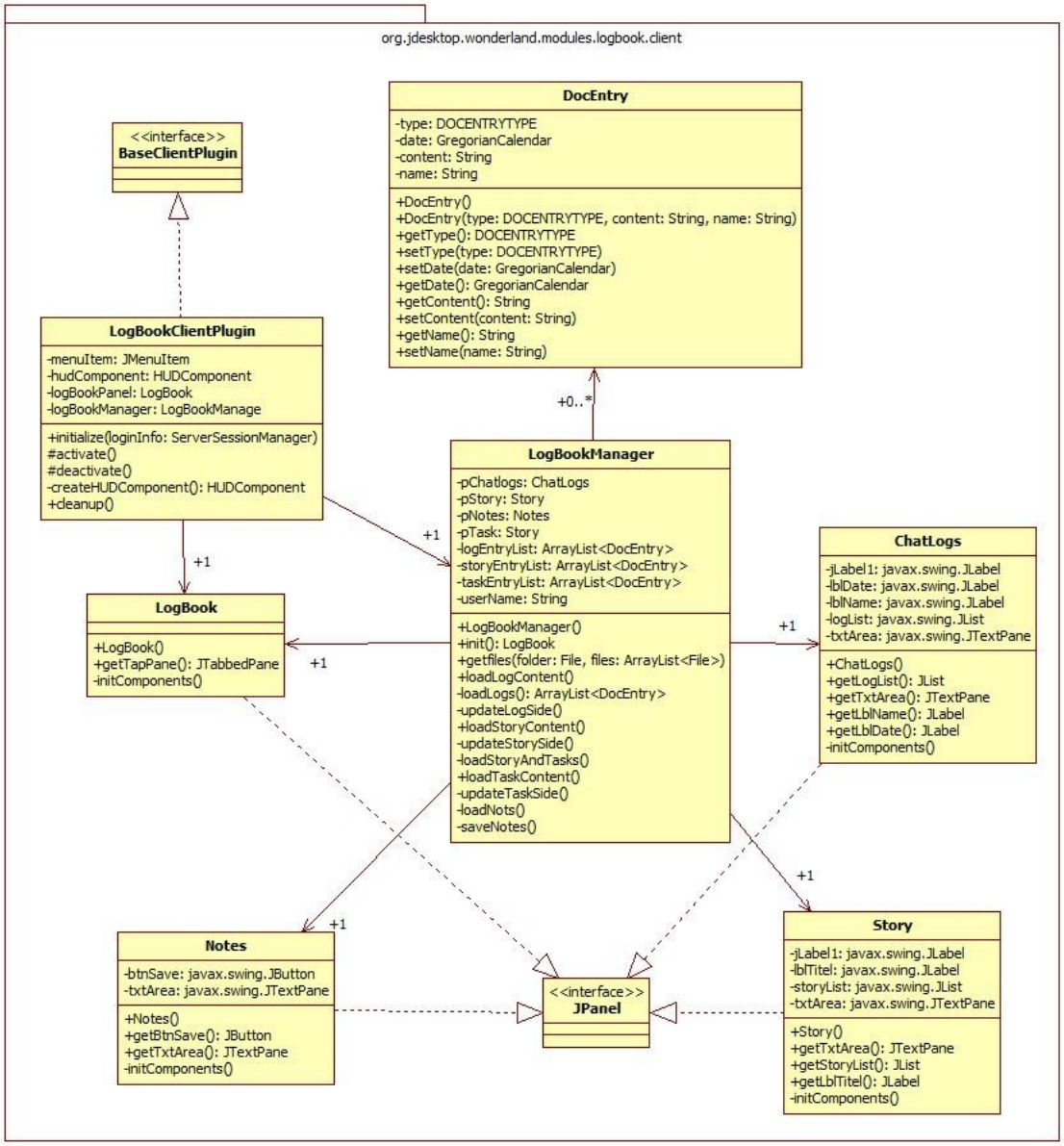

Figure 51: class diagram from LogBook

4.3.2 Developed World Modules

Following the requirements and the story three different worlds modules had to be developed. The worlds serve different purposes. All of these worlds have been designed in a way to simulate a world which can be explored but still has his restriction for moving around made by different objects such as houses, walls, road work blocks or cars. This should give a feeling of an open world which is simple not explorable from this way. This also should take the feeling of being trapped in the world. The most used models are from Google Warehouse and published from the community for free use. Neverless there had to be some buildings designed or existing models modified for the worlds.

Examples are Carol's Corner Store (Figure 55), the Library (Figure 53), the flower beds, the used walls and the office.

4.3.2.1 Entry World

The entry world is the world where students and supervisors, connecting to the world, are placed first. The Figure 52 shows the world from a top camera view. The world basically gives an introduction to the program mechanics and the story. The module his self holds all configuration files of all objects in world. It includes not only the files from the entry world it has also the office and store files for every used object. These are about 1,250 object configuration files for the entry world and the three team worlds.

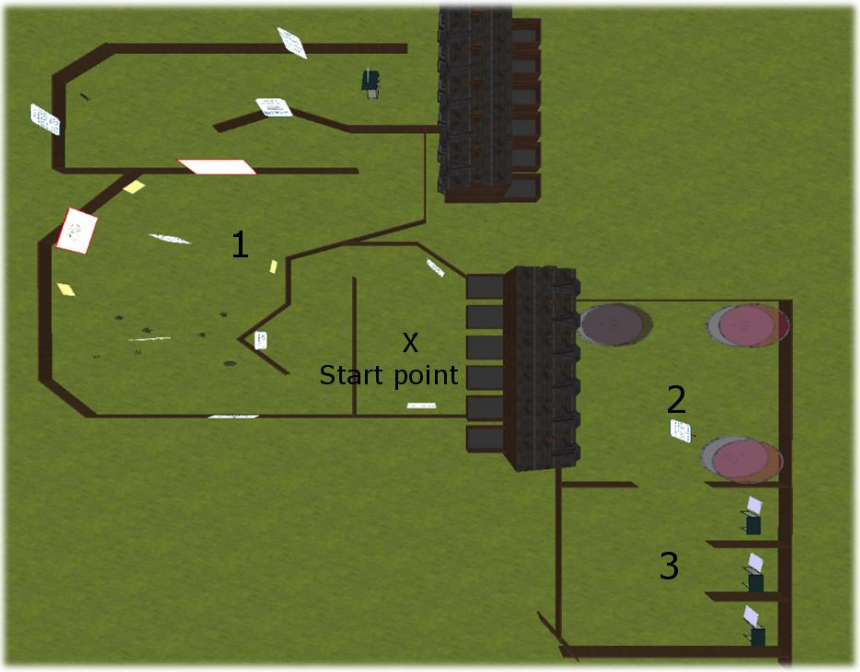

Figure 52: Entry World top view

The X marks the start point where everyone is places connecting to the world (see Figure 52). Nr. 1 is the tutorial area where students can learn the basic interaction with the game world. It explains almost every detail of the game such as which keys need to be used to move, the basic interaction with objects, the developed plugins like the LogBook or the chatbot plugin and the travel system. Nr. 2 is the team area. The area has cones of silence for private talks. The students get here the first task, like building teams in the size of supervisors order and to go to Ray Burns to get introduced to the game tasks. Nr. 3 marks the teleportation point to the team's office worlds. Every team has his own instants from the office and store world. The first prototype provides therefore space for three teams working with RIVALE at the same time.

4.3.2.2 Office World

The office world represents an area for the student where he gets the tasks and story information. He can meet Ray Burns, the company owner and the secretary Aaron Rogers who introduces the student to the world and provides the first tasks. The module includes the art for the objects and all configuration files for every used object. The world contains about 110 objects. It also includes self made objects like the office building and the library (see Figure 53).

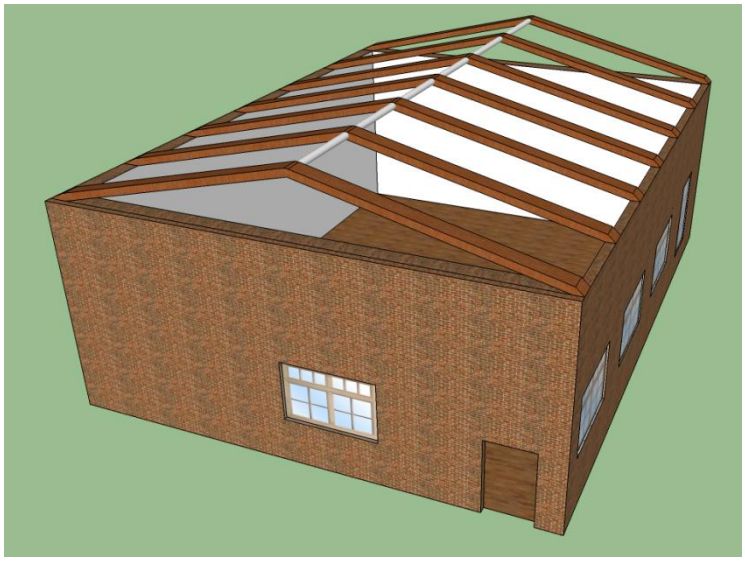

Figure 53: Model of the Library

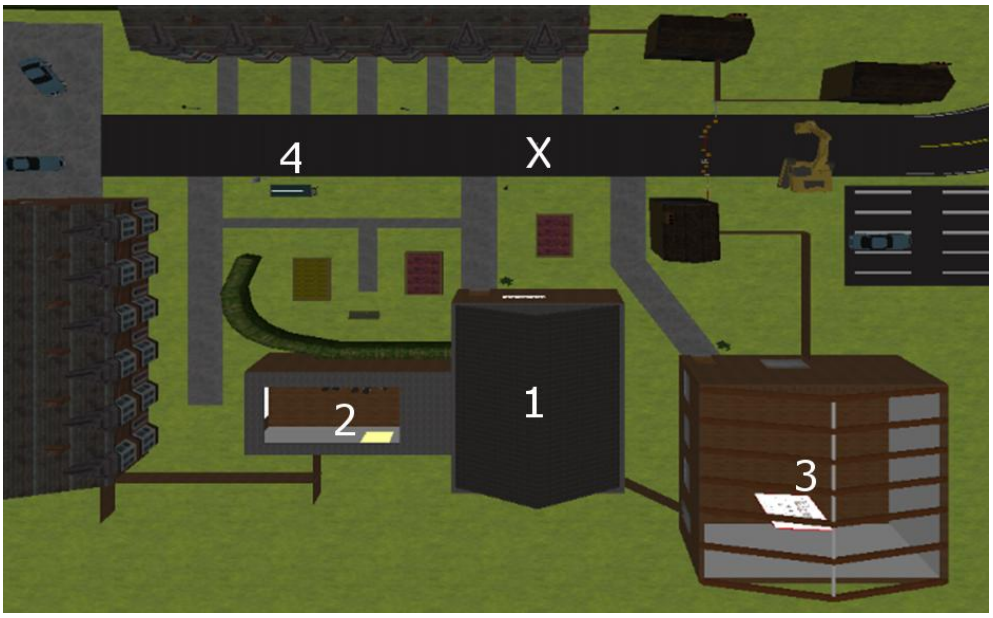

Figure 54: Office World top view

X marks the point students and supervisors get placed when they us the travel functions (Figure 54). Nr. 1 is the Barnes & Ignoble Consulting office building. In the office is a task board giving the main tasks which have to be done by the students. It also has computer workplaces where students can

access story and task related documents. In the back of the office is Ray Barnes office. Nr. 2 is the team meeting room. Students can use it to share findings and discuss with other team members on further steps. The room has a whiteboard and sticky notes for free use. Nr. 3 marks the library. Students will find useful information about the program and the tasks in the room. It provides also map information. Nr. 4 shows the bus station. The bus station is used to travel to Carol's Corner Store.

4.3.2.3 Carol's Corner Store World

Carol's Corner Store World contains the objects where the students have to use their requirement elicitation techniques to elicit information manly from the chatbots by interviewing. There are four bots placed on the map. The module by contains about 90 objects. It also has self made models like the corner story building (see Figure 55)

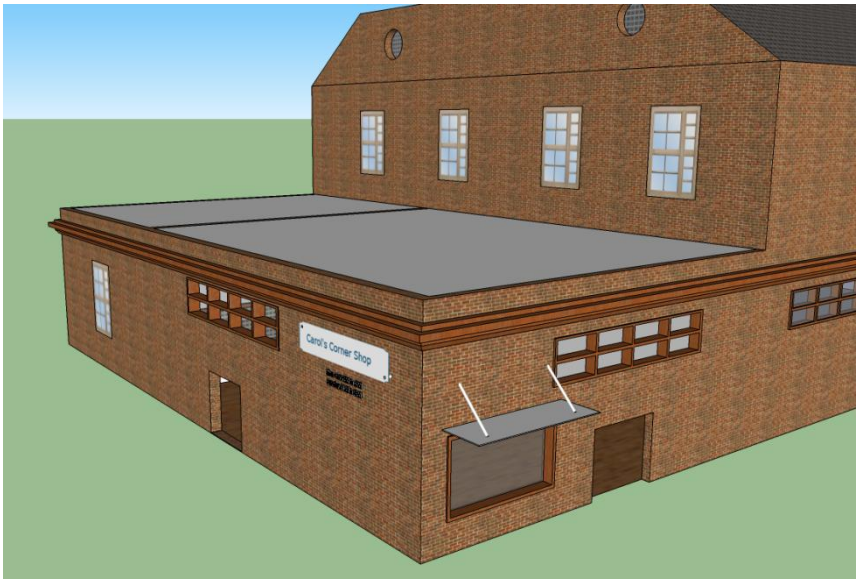

Figure 55: Model of Carol's Corner Store

X marks the point students and supervisors get placed when they us the travel functions (see Figure 55). Nr. 1 marks the front store. Here is Susan and other Lauran which provide the students with a chat. Nr. 2 is Carol's office and back store. Here is Carol placed. Nr. 3 is the place where suppliers deliver their goods and the bot Traver is located. Nr. 4 shows the bus station, where students can travel back to the office.

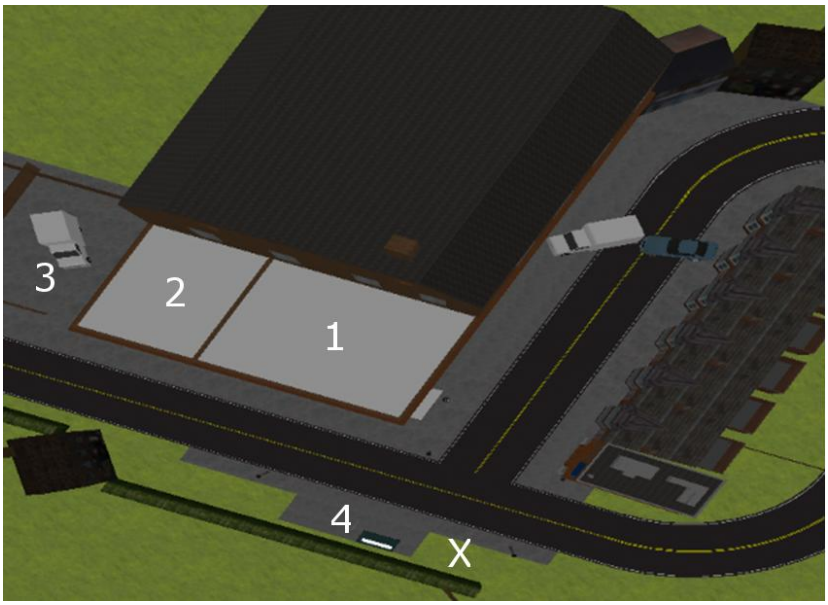

Figure 56: Carol's Corner Store top view

4.4 Summary

RIVALE first prototype supports three worlds, the entry world, the store world and the office world. All these worlds are connected to each other and have different purposes. All worlds are set up in a way that they simulate the real world as good as possible. They try to make an illusion of an open world, but still restrict students moving areas to a quite small place. All these worlds are set up with different models, the most are from Google's warehouse. However still a lot modelling was done and included in the worlds. The user starts in the entry world where he has to make a tutorial about the system and the program. It provides also space to find team members and to get first tasks for the further progress. The second world is the office world representing the company for which the users "works". It has a library for extra information on the task and requirement elicitations techniques, a meeting room and bots which give new task and provide a story. The third world is Carol's Corner Store. It contains the objects where the students have to use their requirement elicitations techniques, like interviewing.

Open Wonderland already provides a lot of modules which were used such as Sticky Notes, Whiteboards and Portals. These modules are chosen and combined to fulfil as much requirements as possible. Still new modules had to be developed to satisfy the project requirements. The module "Chatbot" implements the functions to call Pandorabot via Open Wonderland. It offers free input chat window as well as an automated saving of the chat logs. As second chat function, managed chat is provided allowing in RPG manner to communicate with a bot. Pandorabot was chosen to overcome issues regarding availability and management of the system. Supervisors can update, edit and extend chat answers on the fly without having expert (coding) knowledge. It is implement in a way that it can be attached to every object in world, means it is possible to simulate a conversation with a human (bot) direct or with e.g. a computer. This is also the module with the most work in it. The big

part is on the client side since there was no need in the first version for implementing a lot on the server side. The second developed module is the StoryComponent module. This allows an administrator to add story, information and task content to an OWL cell. If a user selects the object and activates the story functions, the file is taken from the server and sent to the user. All these information, as well as the chatlogs are presented to the user via a window offered by the LogBook module. This is the third developed function module. This module brings all together and is the connector to the game based approaches of the research. It provides the story information, the tasks and the logs in a logical way. It also provides space for notes where users can add their own information without leaving the program.

5 Usage Viewpoint and Evaluation

The actual teaching session and approach, in which RIVALE is planned to be used, takes several learning hours and session to get an optimal learning output. In a first step only experts were ask for their evaluation in a two hour session in which all functions were used. This means that not all aspects could be tested in deep but it gives a first point of view from professionals. This chapter will explain the evaluation methods and outcome, but first starting with the usage viewpoint. The usage viewpoint will present figures taken in the evaluation and are therefore representative how future program situations will look. The scenes show different in world situations starting from the tutorial world, over the office to the last world Carol's Corner Store.

5.1 Usage Viewpoint

The usage viewpoint gives the reader an understanding how the program looks in use, supported with game screenshots for a better understanding of the current prototype of RIVALE. The program is split in three worlds shown in the following subsections.

5.1.1 EntryWorld

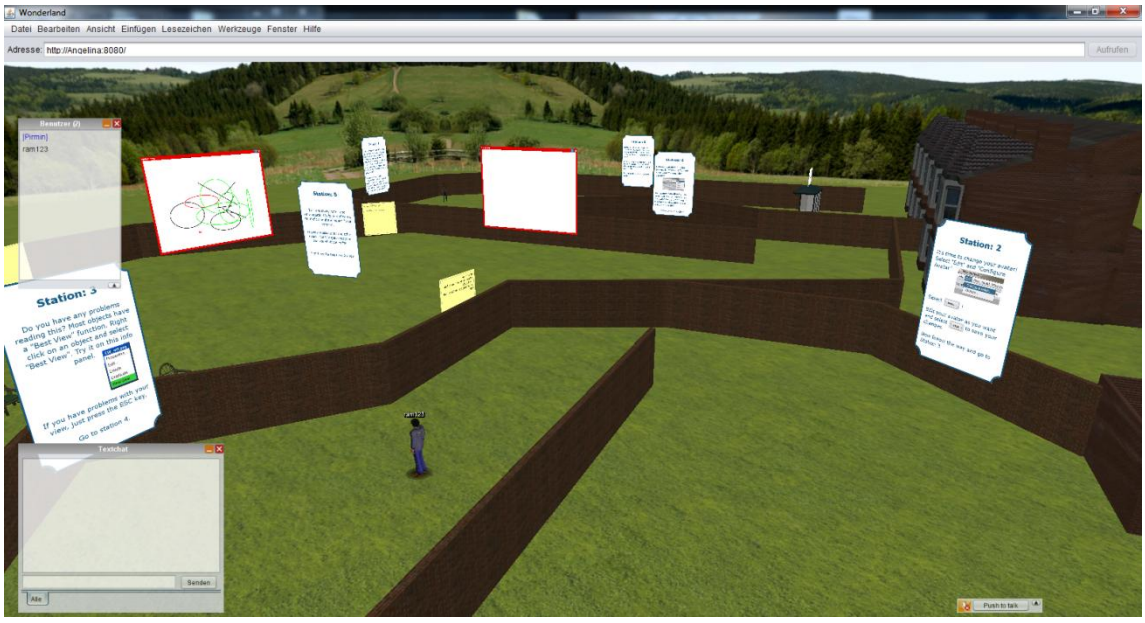

Figure 57: Example view of the tutorial area

The EntryWorld has two main parts, the tutorial area and an area to find team members and first story information. Every user connecting to the world is placed at the same position in the tutorial area. It can be skipped by using the fast travel system (waypoints). New user should go truth the tutorial (see Figure 57). The tutorial is set up with different stations set up with the white information

boards. The Figure 57 shows also the maze style in which the stations are set. Therefore users should not be able to pass any station accidently. The efficiency of the tutorial is important to guarantee that the students will not spend afterwards a lot of time with the system handling, but will be able to focus on the program content, requirement elicitation.

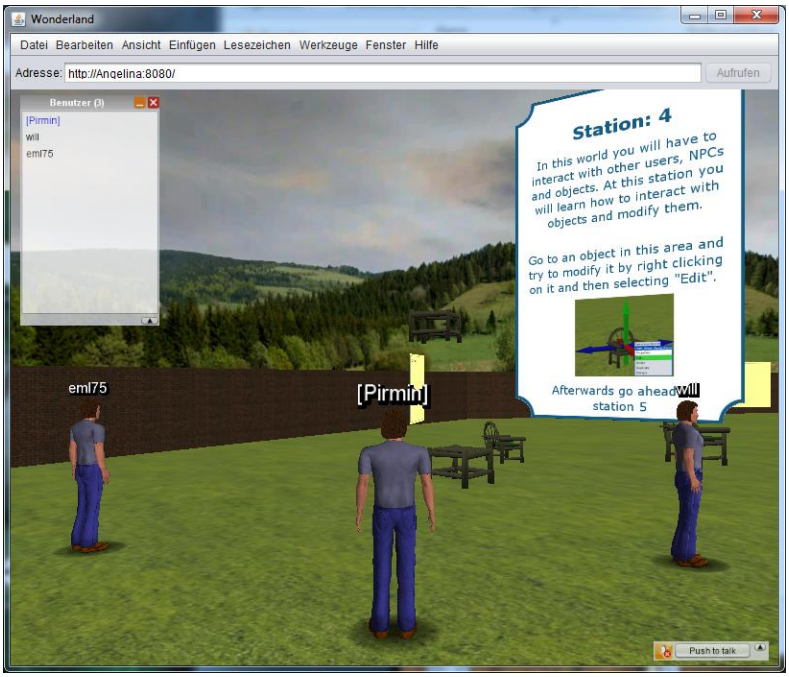

Figure 58: three users working on tutorial station 4, learning how to move objects

Figure 58 illustrates station four and three players currently working on it. In this station students learn how to move, delete and resize objects. They can later use their knowledge to place for example sticky notes. Other stations show how to change the avatar, how to use the “best view” function, how to use the sticky notes, how to interact with the different bots and so on. In the Figure 56 it is possible to see the bus station in the background which leads is used to teleport to the team building area. There students get some first story input and move on to the office world of their team by using another bus stop.

5.1.2 OfficeWorld

After arriving in OfficeWorld students get placed on the street, next to them is the bot Aaron (see Figure 59, Aaron is small visible in the left part of the figure) providing a first guided chat with information on the situation and some story lines. They can see the entry to the library and to the office. The Figure 60 illustrates the inner part of the office in the office world right after the door. It shows the computer with the exclamation over it. The exclamation mark points objects out which provide the user with extra information, task or chats, which might be important for the understanding of the story or simple to get new tasks. In this case gives the computer a first file with information for the customer the student has to take care about. It is possible extract already few

details needed for tasks. In the back is the door leading to the boss and to the right is the entrance to the meeting room.

Figure 59: Office world, view from the street

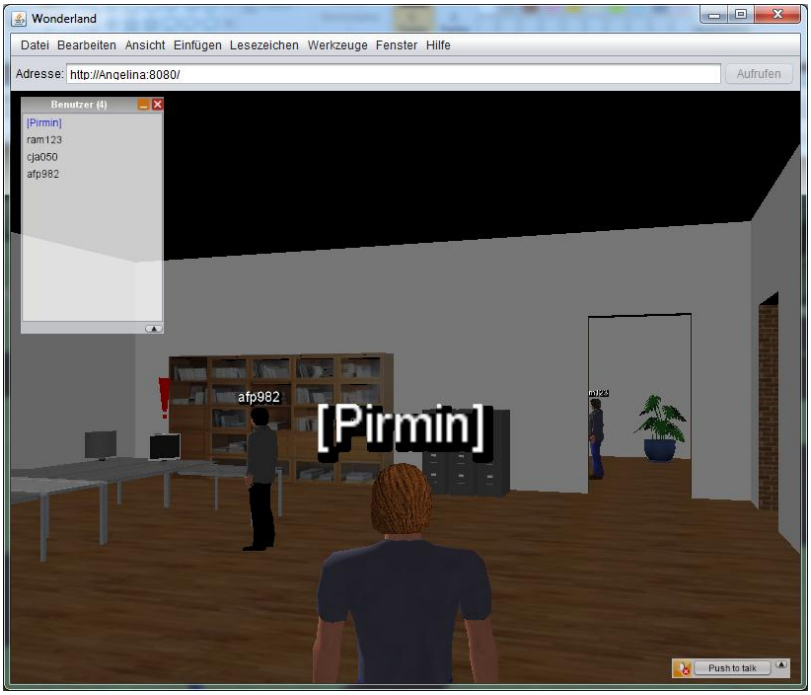

Figure 60: Office entrance with the student's personal computer

Figure 61: Team meeting room in the office

The team Figure 61 shows the team meeting room. It provides different ways to coordinate the learning session in a team. The team members can share information they collected in the interviews or with other ways to improve their own results and team results. Repeating learned questions and methods while telling team mates how and what to do has a positive effect on remembering details and will improve the knowledge the students will keep after the lesson. There is whiteboard and sticky notes to share and keep some information on the wall. The student can further go to the library next to the office building to get some extra information on RE techniques and other stuff like the world map. If they have enough information, the will go to the bus station next to the street and the flowerbeds and will teleport to Carol's Corner Store.

5.1.3 Carol's Corner Store

The Figure 62 shows the entry point for Carol's Corner store with the store building on the right side. Also visible is the bus station on the left which is used to travel back to the office. It is possible to go in every direction so it is possible to enter the store also from the back where the bot Traver is placed next to his delivery truck. It is made in a way that it seems to be a big world with many ways but when students come around a corner, cars might block the way or other methods are used to simulate a "real blockade".

Figure 62: illustrates Carol's Corner Store with the bus station on the left

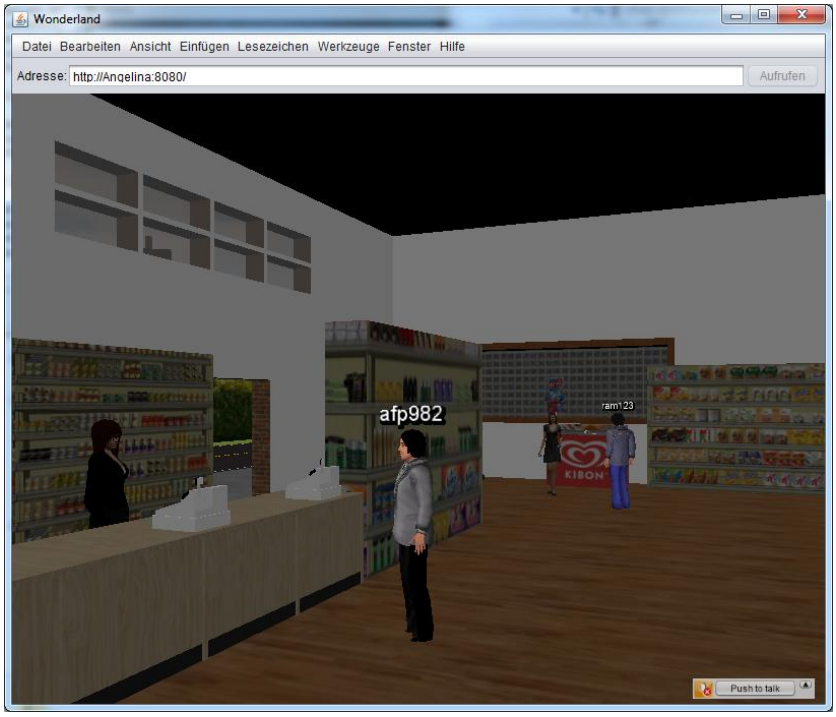

Figure 63: Store front part with students interviewing bots

The Figure 63 shows the front part of Carol's Corner Store. Lauren and Susan are in the picture with two users, "afp982" and "ram123", interviewing them. It is always possible to skip an interview or to do what the student want but then they might miss information needed to complete a task. The Figure 64 shows the back store part which can be reached thru a door in the right corner in the front part of the store. The figure illustrates a chat with Carol, show further the Team chat and the logbook which the tasks tab open. This situation can happen when a student checks again what information

he need to complete a task, then he might not be sure about it and ask his team mates via the normal text chat and then he might chat on with Carol via the window shown on the left in the figure. In the background is a supplier information on the wall which can be inspected as well. Students can go from this room to the front part of the store or in the backyard where Traver is waiting.

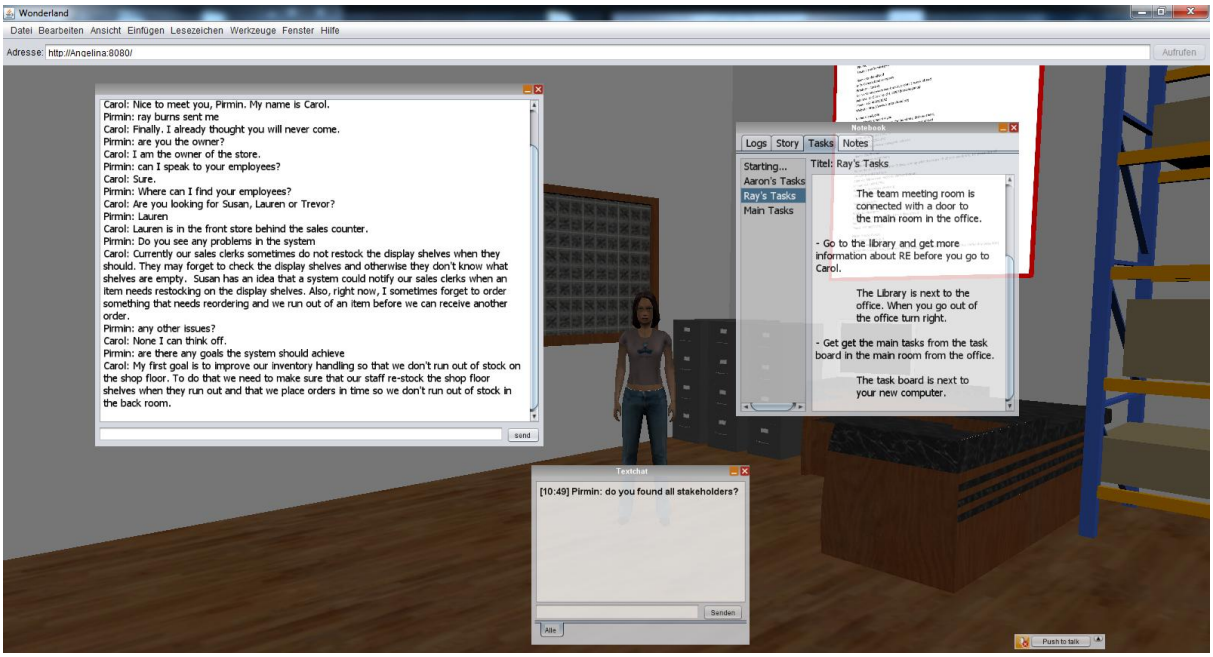

Figure 64: Store back part with Carol, the chat with Carol, a normal Chat and the logbook

5.2 Evaluation

There are different known problems with the used technologies such as a bad knowledge base for the chatbots, lack of acceptance of the system or a not user friendly environment. A further and well known problem is the performance problem, quality and fidelity of rendering, regarding mainly all to the used hardware in classes. Open Wonderland is actually not designed for many polygons and high resolution objects in a single world. Therefore the entire objects are as small as possible. This might make the world “unattractive” or if not small enough it might lag. It is important for the first prototype to know how the acceptance is and if it is possible to work with it. To evaluate the system five focuses were set for the study:

- Is the system first-time-user friendly
- Acceptance of the system
- World component designs and usability like LogBook
- The acceptance and design of the EntryWorld, CCStore and Office
- Bot interaction and it cleverness

5.2.1 Research Methodology

The world was evaluated in a first step by teachers, researchers with virtual world background and students who have previously taken system analysis class. The evaluation was artificial (Venable et al., 2012) in that, that only some of the users are real users. The focus was on evaluating the design aspects from an expert point of view. Important is to find out if the system by itself can be used successfully and if the combination of used methods and functions are fitting together to get an optimal learning output with a least amount of effort for a supervisor. The task was an experimental one rather than real use for teaching and learning elicitation skills, and the system is only a partial (and somewhat rough) prototype (Riedmann et al., 2013). The session was structured with task the participants hat to fulfill. It was further in an informal way, when someone had problems or had any question a session supervisor help direct to overcome the issues.

5.2.2 Procedure

The users where focusing on the functions and usability of the program and less on the RE content since a tow hour lesson is too short for that.

The evaluation was executed with:

1. Pre-questionnaire
2. Using the system
    a. tutorial world
    b. office and store
3. Post-questionnaire
4. Personal interview

After a first pre-questionnaire (1), on paper, were the participants were asked about their experience with 3DVWs. The pre-questionnaire was done without the possibility to ask any question to a supervisor. Then the users were ask to run the RIVALE (2), where the participants were allowed to ask question and got help if needed. In world, they first had to run thru the tutorial area (2.a) which includes changing avatar, explanation of all tools and so on. This took between 30 and 40 minutes. Then they had to make up a team and go to their team office. After this they explored the office (2.b) world where they got more tasks and information and finally went to Carol's Corner Store where they had to interview the bots and to fulfill other tasks. Using the system took about two hours. Afterwards the participants were asked to complete a comprehensive questionnaire (2) about their experiences with the RIAVLE prototype on paper. Finally a personal interview (4) with the supervisor was done were problems in the session were discussed and analysed in detail.

5.2.3 Results

Seven participants evaluated RIAVLE all with a background in VR and RE, 2 tutors, 2 students and 3 teachers. They were asked to evaluate the world from an academic and scientific point of view. On out of seven participants were female. Age range was from 25 to 58.

The test results are a combination of two parts, the questionnaires and an interview with the participations based on the observation done in the test session. The questionnaires are a combination of open-ended questions for further information on what they liked and did not liked and answers where the participants could answer between 1 (No experience/ no knowledge / I strongly disagree) and 5 (expert experience/ expert knowledge / I strongly agree). This two question types were used for every section in the questionnaires. The sections were split in different RIVALE aspects like world in general, office world or chatbot interaction.

5.2.3.1 Pre-Questionnaires

Statement of Question	Average	Std Dev	Min	Max
How much experience do you have with 3D virtual worlds in general (using an avatar to move in a world)?	2,42	0,976	1	4
How much experience do you have with(computer) gaming?	2,72	1,11	1	4
How much knowledge do you have about requirement elicitation?	4,5	0,55	4	5
How much experience do you have with E-Learning?	3,78	0,994	2	5
A virtual world could be a useful environment to teach and learn.	4,22	0,81	3	5
Serious gaming is a useful method to teach and learn.	4,01	0,93	3	5
Collaborative and interactive activities are important for learning.	4,85	0,38	4	5

Table 12: results on the pre-questionnaires

The questionnaire (Table 12) is mainly about the experience with 3DVWs and requirement elicitation. All user stated that they have gaming experience and also especially experience with using an avatar in a 3DVW. Never less was the experience level average with an 2,4 out of 5 relative low since 1 was no experience. On the other side all users had an adequate knowledge base of requirement elicitation with an average of 4,5. The participants also had experience with E-Learning, one stated an average experience level and all other and advanced and expert. Also very high is the believing that a virtual environment can be useful for teaching and learning with an average of 4,2 as well as that serious gaming is a useful method to learn and teach with an average of 4. The highest agreeing was with seven users stated and 5 and one user and 4, at the questions that collaborative and interactive activities are important for learning. The pre-questionnaires had no open-ended questions.

5.2.3.2 Post-Questionnaires and Interviews

The post-questionnaires comprise 77 questions comprised 57 closed questions and 20 open-ended question for further information as well as suggestions for improvement and closed questions. The Table 13 gives a summary on the results. They are separated in *general questions on the prototype*, the *tutorial component*, the *office world*, *Carol's Corner Store world*, the *virtual agent chatbots*, and *other features*. The scales of ratings are from 1 being strongly disagree to 5 being strongly agree. The average midpoint is then 3. Negatively phrased questions are shown in italics (Riedmann et al., 2013).

Statement of Question	Average	Std Dev	Min	Max
I would use this type of program regularly	3,14	1,12	1	4
I found it unnecessarily complex	2,14	0,99	1	4
It was easy to use	3,29	0,70	2	4
The various parts of the tool worked well together	3,71	0,45	3	4
There was too much inconsistency	2,14	0,35	2	3
I think others would find it easy to use	3,86	0,99	2	5
I felt very confident using the tool	3,42	1,18	1	5
I had no problems with the movement of my avatar	3,57	0,73	2	4
I see big potential for such a program in the future	4,42	0,49	4	5
I think the program would be a useful teaching tool	4	0,93	2	5
I had fun	4,28	0,45	4	5
It was easy to learn the basic interaction	4,57	0,49	4	5
The general game look was good	4,28	0,45	4	5
I always knew what I had to do	2,85	0,83	2	4
I think the program is a good real world simulation	3,57	0,73	2	4
I think the scenario can happen like this in reality	4	0,53	3	4
I think the difficulty of the tasks was adequate	4,28	0,45	4	5
The RIVALE prototype was cumbersome to use	3,14	0,83	2	4

Table 13: results on the general question part, adopted and extended from Riedmann et al. (2013)

As Table 13 shows, the results quit positive but mixed with areas with substantial improvement. In general the users found the prototype easy to use, had fund and the difficulty level was adequate. On the other side the users found it cumbersome and complex, but this might be a normal problem for the first time use, never less this might be a part to improve. The participants also thought that the world looks good and the scenario can happen like this in real world what is encouraging since this is an important part. Quite average was the answer to the question if the various parts worked well together. This can be improved to guarantee a better learning experience. This question section had no open-ended questions.

Statement of Question	Average	Std Dev	Min	Max
The tutorial was easy to use	4.00	0,53	3	5
The tutorial had the right amount of information	3.29	0,70	2	4
The tutorial space was useful	4.14	0,64	3	5
I had no more questions after the tutorial	2.86	1,12	1	4

Table 14: result of tutorial component questions, adopted from Riedmann et al. (2013)

The tutorial was easy to use but has parts which need to be improved (see Table 14). One problem was that the notebook was no clearly introduced and some smaller problems where the session administrator had to help. A summary function for the tutorial world might help to solve the most of the smaller problems. It takes about 30 minute to get thru the current tutorial. To completely clarify all it might need to be extended to 45 minutes or even one hour. Some open-ended question answers on the question "what should be improved" are: "*A bit hard to figure out the notebook. At first had Troubles figuring out how to relinquish control from an object. I forgot how to go fast and zoom in/out."* or "*Would be good to add tutorial info to the notebook. Need more info in the Tutorial on the notebook".*

Statement of Question	Average	Std Dev	Min	Max
The office had an acceptable look	3,42	1,40	1	5
The office was a good real world representation	3,57	1,29	1	5
I had no problem to find all the bots (Aaron and Ray)	3,71	1,03	2	5
I had no problem to find the next tasks	3,51	1,18	2	5
The meeting room was useful	3,57	0,43	3	4

Table 15: result of office world questions

In general the users found that the office had an acceptable look and that it was a good real world representation (see Table 15). There is only one user who rated both with one. It might be that this user is use to high resolution programs or games. Some users had more and some less problem to find the next task and the bots. Here comes up the question if a user should always be leaded step by step or if it is a better solution to give some freedom to explore the world by his self. The session war hold in groups of 2-4 people, means quite small groups regarding to an normal class, but still the meeting room was founded as useful even when the whole potential was not used. The observation as well as some interview commentaries showed problems with the camera leading in the building. The open wonderland camera is not made for rooms with "normal" dimension and often gets placed behind walls or other objects. Some open-ended questions answers are "*A bit hard to find ray. Maybe Aaron should be at the desk instead of outside. You could place someone else outside who can tell where to go to BIC.*" or "*where to go? Street direction/map?".* This statements show that it was not always clear where to go. The question for following RIVALE version is "should a student be leaded in every step or give him freedom but accept that he might not find the way always right away?".

Statement of Question	Average	Std Dev	Min	Max
Carol's corner store world had an acceptable look	4,14	0,83	3	5
Carol's corner store world was a good real world representation	3,57	0,90	2	5
I had no problem to find all the bots (Susan, Carol, Trever and Lauran)	3,85	0,99	2	5
I had no problem to find the next tasks	3	1,20	1	5

Table 16: results of Carol's corner store questions

Carol's corner store evaluation looks similar to the office world (see Table 16). The most bots have been found, the world looks ok and it is a good real world representation. Again the question about the guidance comes up and how much a user should get provided. The open-ended answers in this section were mainly about the chatbots which get evaluated next.

Statement of Question	Average	Std Dev	Min	Max
Interacting with chatbots felt like interacting with real humans	2.57	0,90	1	4
Chatbots are important for the RIVALE prototype world	4.29	0,88	3	5
Managed chat (structured/guided chat) was useful	3.43	1,29	1	5
It was easy to start an interaction with a bot	4.12	0,99	2	5
You should remove the managed chat	*1.71*	*1,16*	*1*	*4*
It was easy to get information from Carol	3.14	0,70	2	4
I had fun with Carol	3.71	0,70	3	5
It was easy to get information from Susan	2.29	1,03	1	4
I had fun with Susan	3.43	1,05	2	5
Which bot was more clever - Carol or Susan?	Carol 5-1			
Which bot gave better info - Carol or Susan?	Carol 5-1			
Which bot was more believable as a human?	Carol 5-1			

Table 17: Chabots, adopted and extended (Riedmann et al., 2013)

The chatbots were both considered as useful (see Table 17), the open ended and the guided, but need to be improved. It was with help often hard to get the correct answers, even when in the questions fields where he most rules were placed. Carol was the chatbot build up from zero and Susan was build into a existing knowledge base, but still Carol was consider as more clever and believable, even though both provide almost the same answers. The problem is that often pre existing rules match better then the added rules and return an unexpected answer. Open-ended answers are *"The chats were quite OK although you have to get guided abit"* or *"See above re. Susan. Need to be able to answer more Qs. See questions from my log"* showing that the chatbots still need some work.

Statement of Question	Average	Std Dev	Min	Max
I had no problems using teleport bus stops	4.14	0,99	2	5
The logbook (student notebook) was easy to use	3.50	0,50	3	4
The logbook (student notebook) was useful	4.17	0,69	3	5

Table 18: results on other features, adopted from Riedmann et al. (2013)

Finally the evaluation of other features (see Table 18). The bus stop was easy to use and a believable transportation between the worlds. The notebook was consider as useful but need to be improved and better introduced in the beginning. This section had also some open-ended question asking for input more general. Answers are "*Fun to explore + play. Liked the look. Suprised by the way bots looked. Thought Carol would be older - but this is good*", „*The shop is good. It lookes ? -> so keep up the good authentic look. Make the response for the bots simple*", about the logbook "*positive: keeping track of what need to be done; negative: doesn't tell you which task have been completed*", "*Guide students a bit better in terms of previous knowledge about the topics they can ask about*", "*It should be an engagement (?) tool. It looked good with a first impression and you should continue to motivate the user*".

5.3 Findings

Regarding the main evaluation focuses the following findings were discovered:

- Is the system first-time-user friendly
  With an average of 3,86 out of 5, the question "I Think others would find it easy to use" was rated quite good. The users felt confident while using with an average of 3,42 and it was very easy to learn the basic interaction with an average of 4,57. Including other questions results, this can be rated as "yes" to "is the system first-time-user friendly"
- Acceptance of the system
  The tool was rated as useful teaching tool with average of 4. The users had fun and all in all the acceptance was very good.
- World component designs and usability like LogBook
  The tools were in average rating about on 3 out of 5. In general the look was a bit better rated then the usability.
- The acceptance and design of the EntryWorld, CCStore and Office
  The tutorial world was useful with an average of 4,14 but had a lack of information rated with an average of 3,29 and in general the users had questions left after the tutorial. In the office the most users could find the bots, had only little problems to find the next tasks and rated the meeting room as useful. CCStore had about the same ratings like the office. All worlds were rated as good real world representation and with an acceptable look.
- Bot interaction and it cleverness

The chatbots were rated as important for the prototype, but there is still a lack of there believability. Carol was rated as more clever then Susan and the most users had fun with the bots.

5.4 Summary

The evaluation showed that the first RIVALE prototype has various points which have to be improved but overall it was a success. The participants had fun to use it, got motivated from different aspect of the program and found it a very useful tool and way to teach in the future. The world looked good for the most users and there were only little orientation problems. But the evaluation also showed restrictions of the used systems and methods. A big problem could be wonderland. Wonderland is not made for many objects in a world and the camera leading close to objects is also not optimal. Future project need to keep the objects polygon level low and will have to provide good hardware for the server to guarantee a lag free experience. Another big problem is the chatbot. They chatbot technology is still not on a level where it is easy to script a believable and “failure” free bot. Even with about 800 rules about only a few topics, it was still hard to get all the information the users have been looking for. The new version, ALICE 2.0, might improve the situation a bit but it will not make big steps. There is a lot to research in this area. There is also a need of better tool to work edit the chatbot knowledgebase. It is still almost the best way to handle the xml files with a text editor even with splitting it up in different files, it get very hard to manage the content.
The commands showed that all users are sure that there is a lot of potential in such systems. It will make teaching easier, not as time consuming and more realistic and this not only in the system analyses field.

6 Lessons learned

This chapter illustrate the different experience made during the research in the different periods of the literature study, the design and implementation phases and the usage and evaluation.

6.1 Theory

The literature review had a look at different topics, many in quite new research areas. One of the big growing topics and with a lot of research projects and programs is the E-Learning. Even with a lot of research there is still a drawback in work done with focus on pedagogical and psychological aspects. Often the research projects try to introduce a new way of using the technology to get some specific outputs. It is very uncommon to review then this outcoming critical and compare it with already existing and classic approaches. Other topics like the virtual worlds for education still have drawbacks in the mass of the research projects. The quality is often not to bad but there is simply so much to research and so many ways in which it can be used that it would need much more projects and effort to really research and evaluate the potential of virtual worlds especially 3D virtual environments. Often not even the environments used by the researchers are made for educational reason or supporting specific educational processes such as SimpleLife which gives a very good base but still must be adopted in many ways to get an acceptable outcome. When designing and developing such a system or a system based on it, it is always important to know the boundaries in mind and to be aware of other solutions and their strengths. This saves a lot of time in the implementation. The review showed also that there is still no guideline, tool or other help by developing a program with gaming approaches or a serious game. Basically serious game developers need to rethink or develop new, their used pedagogical approaches for every project. Also a lack of research is in education with chatbots. There are only a few recent projects trying to use their advantages, even when chatbots by their self, need still a lot of research before they can imitate humans believable.

6.2 Development

Starting with a project in an, until that time, unknown framework is always a bit tricky and time consuming in the beginning. Therefore a good documentation is important as well as a good community. It was assumed that the OpenWonderland provides both. To set up the design for the world, like which modules have to be developed and in which way they should interact, was not too difficult. The OpenWonderland warehouse, the blog and the SVN repository offers quite a good amount of modules, even when this not covers all available modules. Therefore it is still a good idea to search on other web pages. When it comes to developing the modules, the problems of Open Wonderland come up. It is not so much the framework, this works the most time good and has not

too much bugs. It is more about the provided resources. At least half of the links on the webpage have not been working any more. A lot of links still targeting to java.net wiki of OWL, which does not exist anymore. There is now a Google code documentation and a new OWL wiki. Both having different versions of tutorials and information what is after all not very helpful. The community seems to be active, but after posting five different questions and getting not a single answer, I think the community is not helpful. Good was that after a very time consuming period of reading into, OWL works quite well and module developing is also easy then. Especially all things which cannot be solved with an approach shown in a tutorial are hard to find out. It was necessary to go deep into the OWL code to get information to solve some issues or to find e.g. out how to save a document on the server side and how to save it on the client side, where the programmer want it. This was very time consuming and sometimes frustrating too. Also important is to check always the compiled module, because errors in the build script can lead to mistakes which are nowhere shown. There was the problem that a module was compiling without errors and sent it automatic to the server. But while testing, the new functions were not working. It did not matter what changing had been done, nothing happen. It was not possible to find the mistake in the code. It took quite a while to realize that the problem was the build script and that the compiled module simple was not including the classes it was working with. Very time consuming is also to develop a chatbot. It can be so hard and take so many rules, added to the knowledge base, before a chatbot is able to talk about a single topic in acceptable manner. A problem can be the language. Non native English speaking persons often uses different sentence structures as a native English speaker would do. For example a try was to simplify the rules with simple returning one out of ten problems always when the input sentence had the word “problem” or “issue” in it. Then people started to enter stuff like “you are the problem” or “that’s not my problem”. Language is so deep and complex that a language scientist is almost essential to get a good chatbot at all.
Also very helpful is to have the supervisor and other person nearby which are interested in the project and work in the study field. It is very time saving to be able to address experts direct.

6.3 Evaluation

It was very interesting to see when people got stuck or had problems with things never expected in this way. This manly comes from the implementation period in which the developer got so used to the prototype that it was not possible to see obvious problems. A good example is the camera leading of Open Wonderland. It felt like this would be no problem for no one but actually it was a problem for almost everyone. This proved how important non project related test persons are. It was also quite unexpected that Susan, the chatbot with a big pre existing knowledge base, was not as successful as Carol, which was build up from zero. The problem here is that many other rules had interferences with the added rules and so a lot of unexpected answers were produced. I also learned that it is important to make performance test earlier. There was never any lag problems in the developing period but already with three users a lot of lag appeared. The test showed that it can be very useful and time saving when already little test are done in early stages. Especially changes which

are part of a sequence can lead to very long updating times. In this project for example the tutorial world. The evaluation showed that some stations could be done better or some topics even need some further information or need to be an own station to be more clear to the user. This leads to the problem that when a station is added, all other stations need to be updated too. The bigger problem is that inserting an extra station leads to an update to the world itself too. Means that every object has to be moved and it might be that there is not even enough space in one direction so everything need to be chanced just for maybe one extra station. So this showed that testing parts and small test are really important.

7 Conclusion and Outlook

The idea and aim of this project was the creation of a collaborative 3D virtual learning environment, focused on virtual intelligent agents supporting a fictive case study where students learn requirement elicitation techniques, especially interviewing. A further idea was to add gaming approaches to gain extra motivation from students. The program should provide a user friendly and immersive learning experience. There are several reasons for developing this application. One is the lack of real world problems in teaching and practicing techniques and tools for requirement elicitation. Since interviewing is one of the main techniques, such a system had to be developed. Therefore a deeper look into chatbots was necessary, involving the question on advantages and disadvantages in general from chatbots used as educational tool. The virtual 3D environments with educational purpose were another big topic needing a lot of research. Such a system must be easy to configure, collaborative, interactive and motivating. This all together combined with game elements is really rare researched.

The literature review gave insight into E-learning, Serious Gaming, chatbots with natural language processing, requirement elicitation and 3D virtual worlds used for education. This was done by analysing existing researches and scientific work in the specific areas. Computer based learning is essential in today's education. Different advantages and disadvantages have been listed and described. The focus was on virtual environments such as Open Wonderland or Second Life and showed that there is still a big lack of frameworks for educational program development. The most frameworks extra made for education are open source and unfortunately having relatively small communities and do not provide the same quality in there documentations as the most other commercial products. Such a framework used with a virtual intelligent agent or chatbot is very poor researched.

The design and implementation of the first RIVALE prototype showed problems as well as advantages of the open source frameworks for 3D virtual environments. The chapters showed in detail what was needed and what was available as well what was done then explained with technique background. It was not possible to implement all requirements as they were defined. Especially the part with serious gaming approaches had a lower priority and was only implemented in a minimal amount. Good success was achieved with the chatbot system and the general acceptance of the program. The tool helps to understand requirement techniques better and to practice them. The implementation showed that such programs do not need to be built up from zero. Environments like Open Wonderland provide a good amount of modules supporting collaborative and very important, adoptable and extendable modules.

The evaluation was administered by means of test the program in two hours session, pre and post-questionaries and interviews. It was in general very successful. The users had a lot of fun with the program and are sure that this is the correct way the program is heading. However, some drawbacks of the used system were also pointed out. A big problem is the camera movement of

OpenWonderland, not made for normal rooms and buildings and also the performance problems came up. Therefore it might be a good idea to reconsider other frameworks more deeply to find alternatives. The second bigger problem was the chatbot system: ALICE is still not on a level where it is easy to make and maintain a chatbot. A lot of research must be done to simplify this learning in virtual worlds has extreme potential and will likely become an integral part of future education.

The project showed a bunch of research areas where more research need to be done. 3D virtual environments for education are still only in a small amount available and especially free once have drawbacks in support and system technique. There is a big need to develop such a framework. This is a big task and would need a big team or an international cooperation to make this possible. Open Wonderland has his plus and minus points and would also need more research to overcome the issues, but has good approaches and potential to become a very good free tool for developing 3D virtual environments. The problem here is still the right mix of social collaboration, interactivity and other social and pedagogical support to develop not only a well working world at technique level, but also in educational. Further chatbots need more research and can be improved a lot. There is so much potential and so less research with chatbots in education. But also the underlying chatbot system need still to make big steps to provide a framework were it is possible to create fast and believable “humans”. Also serious games need to be research further. There are more and more projects but the most out coming project still only teach a very small amount of knowledge and are often never leave the prototype state. Finally the combination, like it was done in this project need to be research much more in a much deeper way. There is so much potential to improve the current ways of teaching. Since the combination of serious gaming, chatbots, 3d virtual worlds and requirement elicitation is quite complex and the lack of frameworks and other issues in all the topics, it will still take years and tons of hours before a great program can be develop and implemented, but if it will be done, it will revolutionary the current way of teaching and this not only in requirement elicitation.

As a guideline for further research the following list of topics can be summarized, based on the research done in this thesis. The list provides also question to be answered in more detail:

- Chatbot systems
  - How can AIML and ALICE be enhanced to improve chatbot performance and reduce incorrect answers?
  - How can a chatbot system be enhanced to make chatbots more believable?
- Chatbots in education
  - How useful is a chatbot in education really?
  - What language methods can be used to improve the chatbot in educational point of view?
- 3D virtual Worlds frameworks with module systems
  - How complex can a 3DVW framework be to make it still easy to use?
- 3D virtual Worlds framework with focus on education
  - What functions should a 3D virtual World framework for education provide?
- Open Wonderland as educational framework

  - How can OWL be enhanced for better teaching.
  - Is it possible to improve OWL in a way that it can be used for serious games with many 3D objects in a scene?
- Serious games
  - What game aspects are useful for education?
  - Is there a “best practice” for serious games?
- Defining and researching pedagogical useful functions and methods in 3D games
  - Is it possible to define a set of function and methods a game should have to be a serious game?
- The combination of chatbots, serious gaming and 3D world in education
  - Is a chatbot really a benefit in a 3D virtual world used in education?
  - What are the specific functions a 3D virtual world must provide to be serious game?

Bibliography

Michigan State University. (2013). *Telecommunication, Information Studies and Media*. Retrieved May 28, 2013, from Games of Entertainment & Learning Lab: http://seriousgames.msu.edu/

Adolphs, P., Benz, A., & Castello, N. B. (2011). *Conversational Agents in a Virtual World.* Berlin: Springer-Verlag.

AI Foundation, Inc. (2013a). *Artificial Linguistic Internet Computer Entity Artificial Intelligence Foundation*. Retrieved May 13, 2013, from ALICEBOT: http://www.alicebot.org/

AI Foundation, Inc. (2013b). *Artificial Linguistic Internet Computer Entity Artificial Intelligence Foundation*. Retrieved Dec 19, 2013, from Pandorabots AIML Tag Set: http://www.alicebot.org/documentation/ptags.html

AI4US Ltd. (2013). *Artificial Intelligence for us*. Retrieved May 21, 2013, from Chatbots: http://www.chatbots.org/

Allison, C., Miller, A., Sturgeon, T., Nicoll, J. R., & Perera, I. (2010). Educationally Enhanced Virtual Worlds. *Frontiers in Education Conference* (pp. T4F-1 - T4F-6). Washington, DC: ASEE/IEEE.

AL-Smadi, M., & Gütl, C. (2008). *Past, Present and Future of e-Assessment: Towards a Flexible e-Assessment System.* Villach: International Conference on Interactive Computer Aided Learning.

Ambient Insight. (2013). *Ambient Insight Reserach*. Retrieved May 28, 2013, from Ambient Insight: http://www.ambientinsight.com/

Athabasca. (n.d). *Athabasca University*. Retrieved May 16, 2013, from Athabasca University: http://www.athabascau.ca/

Babcock, J. (2009, May 28). *Practical Analsyst: Practical Insight for Business Problem Solvers*. Retrieved June 05, 2013, from Practical Analsyst: http://practicalanalyst.com/requirements-elicitation-most-valuable-questions/

Bates, A. W., & Poole, G. (2003). *Effective Teaching with Technology in Higher Education: Foundations for Success.* Indianapolis: Jossey-Bass.

Baumgartner, P., & Payr, S. (2001). *Studieren und Forschen mit dem Internet.* Wien: Studen-Verlag.

Brennan, K. (2009). *A Guide to the Business Analysis Body of Knowledge (BABOK® Guide).* Toronto: International Institute of Business Analysis.

Bush, N. (n.d.). *Getting Started with Program W*. Retrieved June 19, 2013, from Program W info: http://programw.sourceforge.net/

Cantador, I., & Conde, J. M. (2009). EFFECTS OF COMPETITION IN EDUCATION: A CASE STUDY IN AN E-LEARNING ENVIRONMENT. *Education, IEEE Transactions on Volume 52* , 279 - 285.

Chang, V., & Gütl, C. (2010). Generation Y Learning in the 21st Century: Integration of Virtual Worlds and Cloud Computing Services. *Proceedings of Global Learn* (pp. 1888-1897). Penang: Association for the Advancement of Computing in Education.

Chang, V., Gütl, C., Kopeinik, S., & Williams, R. (2009). *Evaluation of Collaborative Learning Settings in 3D Virtual Worlds.* Perth: International Journal of Emerging Technologies in Learning.

Chemuturi, M. (2013). *Requirements Engineering and Management for Software Development Projects.* New York: Springer.

Chowdhury, G. G. (2003). Natural language processing. In C. Harris, *Annual Review of Information Science and Technology, Volume 37* (pp. 51–89). Information Today, Inc.

Dalsgaard, C. (2006). *Social software: E-learning beyond learning management systems.* Retrieved Oct 15, 2013, from eurodl: http://www.eurodl.org/materials/contrib/2006/Christian_Dalsgaard.htm

Devedzic, V. (2004). Education and the Semantic Web. *International Journal of Artificial Intelligence in Education 14* (pp. 39-65). Belgrade: IOS Press.

DFC Intelligence. (2013). *DFC Intelligence*. Retrieved April 30, 2013, from dfcint: http://www.dfcint.com/wp/?p=343

Dieterle, E., & Clarke, J. (2008). Multi-User Virtual Environments for Teaching and Learning. In M. Pagani, *Encyclopedia of Multimedia Technology and Networking, Second Edition (3 Volumes)* (pp. 1033-1041). USA: Information Science Reference.

Dreyfus, H., & Dreyfus, S. (. (1986). *Mind Over Machine: The power of Human Intuition and Expertise in the Era of the Computer.* New York: Free Press.

Elearning-Companion. (2011). *The Disadvantages of Online Learning | Elearning Companion*. Retrieved May 16, 2013, from The Disadvantages of Online Learning | Elearning Companion: http://www.elearning-companion.com/disadvantages-of-online-learning.html

E-LearningWiki. (2009). *E-Learning wiki*. Retrieved May 20, 2013, from http://elearning-wiki.de/

ESA. (2013). *entertainment software association*. Retrieved April 30, 2013, from theesa: http://www.theesa.com/facts/salesandgenre.asp

e-teaching. (2011). *Leibniz-Institut für Wissensmedien*. Retrieved May 21, 2013, from e-teaching: http://www.e-teaching.org/

Finnigan, D. (n.d.). *Google Code, Open Wonderland Wiki.* Retrieved June 19, 2013, from Open Wonderland documentation Wiki: http://openwonderland.googlecode.com/svn/www/wiki/beginner/Wonderland_Tutorial_01.pdf

Fishwick, P. A. (2009). An introduction to OpenSimulator and virtual environment agent-based M&S applications. *Simulation Conference (WSC) 2009* (p. 7). Gainesvlle, USA: The Winter Simulation Conference (WSC).

Friedman, L. W., & Friedman, H. H. (2013, 01). Using Social Media Technologies to Enhance Online Learning. *JEO the Journal of Educators online* .

Gamage, V., Tretiakov, A., & Crump, B. (2009). Educators' perceptions about using MUVE for teaching. *Same places, different spaces. Proceedings ascilite Auckland 2009* (pp. 324-326). Auckland: ascilite conference.

Garrison, D. R. (2004). *E-learning in the 21st century: a framework for research and practice.* London: Routledge.

Google. (2013). *Google Warehouse: Trible 3D-galerie*. Retrieved July 09, 2013, from Google Warehouse: http://sketchup.google.com/3dwarehouse/?hl=de

Gütl, C. (2011). The Support of Virtual 3D Worlds for Enhancing Collaboration in Learning Settings. In Hershey, *In Techniques for Fostering Collaboration in Online Learning Communities: Theoretical and Practical Perspectives* (pp. 278-299). IGI Global.

Harwell, R., Aslaksen, E., Mengot, R., Hooks, I., & Ptack, K. (1993). What Is A Requirement? *Proceedings of the Third International Symposium of the NCOSE.* Marietta: INCOSE.

Hebbel-Seeger, A., Reiners, T., & Schäffer, D. (2013). *Synthetic Worlds.* Perth: Springer.

Hmelo-Silver, C. E. (2006). Analyzing collaborative learning: Multiple approaches to understanding processes and outcomes. *ICLS '06 Proceedings of the 7th international conference on Learning sciences* (pp. 1059-1065). New Brunswick: International Society of the Learning Sciences.

Hrastinski, S. (2008). *Asynchronous & Synchronous E-Learning.* Uppsala: EDUCAUSE.

Johnson, W. L., Vilhjalmsson, H., & Marsella, S. (2005). Serious Games for Language Learning: How much Game, How Much AI? In C.-K. Looi, *Artificial Intelligence in Education: Supporting Learning Through Intelligent and Socially Informed Technology* (pp. 306-313). IOS Press.

Kaplan, J., & Yankelovich, N. (2011). Open Wonderland: An Extensible Virtual World Architecture. *Internet Computing, IEEE* (pp. 38 - 45). IEEE Computer Society.

Kendall, J. E., Kendall, K. E., Barnes, R. J., Baskerville, R. L., & Schmidt, A. (2007). *HyperCase*. Retrieved June 21, 2013, from HyperCase: http://www.prenhall.com/hypercase/

Kerly, A., Hall, P., & Bull, S. (2006). Bringing Chatbots into Education: Towards Natural Language Negotiation of Open Learner Models. *The 26th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence* (pp. 177–185). University of Birmingham: Elzware Ltd.

Liddy, E. D. (2001). *Natural Language Processing.* New York: Marcel Decker, Inc.

LindeLab. (2013). *Second Life: Your World. Your Imagination.* Retrieved May 30, 2013, from Second Life: http://secondlife.com/?lang=en

Loebner. (2013, April). *Home Page of The Loebner Prize in Artificial Intelligence*. Retrieved May 22, 2013, from The Loebner Prize: http://www.loebner.net/Prizef/loebner-prize.html

Looi, C.-K. (2005). *Artificial intelligence in education supporting learning through intelligent and socially informed technology.* Washington, DC: IOS Press.

Maderer, J., Gütl, C., & Al-Smadi, M. (2013). Formative Assessment in Immersive Environments: A Semantic Approach to Automated Evaluation of User Behavior in Open Wonderland. *iED 2013.* Boston: immersive education initiative.

Mauldin, M. L. (1994). *CHATTERBOTS, TINYMUDS, and the Turing Test Entering the Loebner Prize Competition.* Pittsburgh: ASSOCIATION FOR THE ADVANCEMENT OF ARTIFICIAL INTELLIGENCE.

Messinger, P. R., Stroulia, E., Lyons, K., & more, a. (2009, June). Virtual worlds — past, present, and future: New directions in social computing. *ELSEVIER* , pp. 204–228.

Moor, J. H. (2000). Turing Test. In A. Ralston, & E. D. Reilly, *Encyclopedia of Computer Science* (pp. 1801-1802). Chichester UK: John Wiley and Sons LTD.

Moreno, R., & Mayer, R. E. (1999). Cognitive Principles of Multimedia Learning: The Role of Modality and Contiguity. *Journal of Educatinal Psychology Vol 91(2)* , pp. 358-368.

Morgan. (2010, July 07). *Business Analyst Articles: Business Analysis & Systems Analysis*. Retrieved June 05, 2013, from An Overview of Requirements Elicitation: http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/1427/An-Overview-of-Requirements-Elicitation.aspx

NSF. (2013). *National Science Foundation: Where Discoveries Begin*. Retrieved May 22, 2013, from National Science Foundation: http://www.nsf.gov/

OECD. (2005). *E-learning in Tertiary Education: Where Do We Stand?* OECD Publishing.

OpenSimulator. (2013). *OpenSimulator*. Retrieved May 05, 2013, from OpenSimulator: http://opensimulator.org/wiki/Main_Page

OpenSimulatorWiki. (2013a). *OpenSimulator: Dependencies*. Retrieved May 31, 2013, from OpenSimulator: http://opensimulator.org/wiki/Dependencies

OpenSimulatorWiki. (2013b). *OpenSimulator: Performance*. Retrieved May 31, 2013, from OpenSimulator: http://opensimulator.org/wiki/Performance

OpenWonderland. (2012b). *Open Wonderland Foundatio: FAQ*. Retrieved May 30, 2013, from Open Wonderland: http://openwonderland.org/about/faq

OpenWonderland. (2012a). *Open Wonderland Foundation*. Retrieved May 30, 2013, from Open Wonderland: http://openwonderland.org/

OpenWonderlandWiki. (n.d.). *Open Wonderland Documentation Wiki*. Retrieved June 19, 2013, from OpenWonderland Google Code Wiki: https://code.google.com/p/openwonderland/wiki/OpenWonderland?tm=6

OpenWonderlandWiki. (2013, Jannuary 29). *OpenWonderland Community Wiki*. Retrieved June 17, 2013, from OpenWonderland: http://wiki.openwonderland.org/Wiki.jsp?page=Server%20Administration%20Console%20Guide#section-Server+Administration+Console+Guide-ManageModules

OpenWonderlandWiki. (2011a, June 30). *OpenWonderland Community Wiki: Capabilities*. Retrieved June 2013, 06, from OpenWonderland Wiki: http://wiki.openwonderland.org/Wiki.jsp?page=Using%20Capabilities

OpenWonderlandWiki. (2012, Jan 29). *OpenWonderland Comunity Wiki*. Retrieved July 09, 2013, from OpenWonderland Wiki: http://wiki.openwonderland.org/Wiki.jsp?page=Sample%20HUD%20Tutorial

OWLWarehouse. (2012a). *Open Wonderland Warehouse*. Retrieved June 17, 2013, from Open Wonderland: http://openwonderland.org/module-warehouse/module-warehouse

Pazhanisamy, A., & Gopalakrishanan, R. (2011). *VIRTUAL LEARNING IN HIGHER EDUCATION.* Retrieved May 29, 2013, from K.S.MANIAM SEMINAR: http://ksmaniamseminar.files.wordpress.com/2011/03/r-gopalakrishnan.doc

Reiners, T., Wood, L. C., Chang, V., Guetl, C., Herrington, J., Teras, H., et al. (2012). Operationalising gamification in an educational authentic environment. *IADIS International Conference on Internet Technologies & Society* (pp. 93–100). Perth: IADIS Press.

Riedmann, P., Venabl, J. R., Chang, V., Reiners, T., & Gütl, C. (2013). RIVALE: A Prototype Realistic Immersive Virtual Agent-Based Learning Environment Case Study for Learning Requirements Elicitation Skills. *roceedings of the 2013 AIS SIGED: IAIM International Conference on Informatics Education and Research (SIGED 2013).* Milan, Italy: SIGED 2013.

Romero, M., Vizcaino, A., & Piattini, M. (2008). *Using Virtual Agents for the Teaching of Requirements Elicitation in GSD.* Chill´an: Springer-Verlag.

Sabahat, N., Iqbal, F., & Azam, F. M. (2010). An Iterative Approach for Global Requirements Elicitation: A Case Study Analysis. *International Conference on Electronics and Information Engineering* (pp. 361-366). Kyoto: Electronics and Information Engineering (ICEIE).

Sánche, J., Lumbreras, M., & Silva, J. P. (n.d.). *citeseerx.* Retrieved 09 23, 2013, from Developed at and hosted by The College of Information Sciences and Technology: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.6884&rep=rep1&type=pdf

Schank, R. (1997). *Virtual Learning: A Revolutionary Approach to Building a Highly Skilled Workforce.* New York: Mcgraw-Hill.

Schell, J. (2008). *The Art of Game Design: A Book of Lenses.* CRC Press.

Schmitz, B. (2007, July 26). *multimediablog: Informationen zu Multimedia und Gruppenblog der Studenten von der RFH Köln*. Retrieved June 02, 2013, from multimediablog: http://www.bernd-schmitz.net/blog/index.php/2007/07/26/gastveranstaltung-in-second-life/

SecondLifeWiki. (2013b). *Second Life Wiki: Architecture*. Retrieved May 30, 2013, from Second Life: http://wiki.secondlife.com/wiki/Server_architecture

SecondLifeWiki. (2013a). *Second Life wiki: Education*. Retrieved May 30, 2013, from Second Life: http://wiki.secondlife.com/wiki/Second_Life_Education

Serious Game University. (2013). *SGU*. Retrieved May 28, 2013, from Serious Game University: http://www.seriousgameuniversity.com/

Shawar, B. A., & Atwell, E. (2002). *A Comparison Between Alice and Elizabeth Chatbot Systems.* Retrieved Dec 10, 2013, from SCHOOL OF COMPUTING : http://www.engineering.leeds.ac.uk/computing/research/publications/reports/2002/2002_19.pdf

Shawar, B. A., & Atwell, E. (2007). Chatbots: Are they Really Useful? *GLDV-Journal for Computational Linguistics and Language Technology* , pp. 29-49.

Sloan, A. P. (2013). *Alfred P.Sloan Foundation*. Retrieved 22 05, 2013, from Alfred P.Sloan Foundation: http://www.sloan.org/

Slott, J. (2012, January 29). *OpenWonderland on Google Code*. Retrieved June 07, 2013, from OpenWonderland: https://code.google.com/p/openwonderland/wiki/DevelopingNewCellComponent

Susi, T., Johannesson, M., & Backlund, P. (2007). Serious Games – An Overview. Sköde Sweden.

Tavangarian, D., Leypold, M., Nölting, K., & Röser, M. (2004). *Is e-Learning the Solution for Individual Learning?* Rostock.

TeacherGaming. (2013). *Brining Minecraft to the Classroom*. Retrieved 5 28, 2013, from MinecraftEdu: http://minecraftedu.com/

TechMISLLC. (n.d.). *Technolgy - Managment % Innovative Solutions*. Retrieved June 04, 2013, from TechMISLLC: http://www.techmis.com/our-core-skills/requirement-management/what-is-a-requirement

Valve. (2013). *Teach With Portals*. Retrieved 05 28, 2013, from http://www.teachwithportals.com/

Venable, J. (2012a, July 07). A 3DVW Case Study for Teaching Students SAD Requirements Elicitation Techniques. Perth, Western Australia, Australia.

Venable, J. R., & Pries-Heje, J. (2012). A Comprehensive Framework for Evaluation in Design Science Research. *7th International Conference on Design Science Research in Information Systems and Technology* (pp. 45 - 77). 7th International Conference on Design Science Research in Information Systems and Technology.

Venable, J. (2012b). System Analysis and Design 251 Assignment. Perth, Western Australia, Australia.

Venable, J. (2006). *The Role of Theory and Theorising in Design Science Research.* Perth, Western Australia: DESRIST.

Wallace, R. S. (2005). *Be Your Own Botmaster.* Alice A.I. Foundation, Inc.

Wallace, R., Tomabechi, H., & Aimless, D. (2003). Chatterbots Go Native: Considerations for an eco-system fostering the development of artificial life forms in a human world. *Joho Shori Gakkai Zenkoku Taikai Koen Ronbunshu* , pp. 5.451-5.454.

Wesiak, G., AL-Smadi, M., Höfler, M., & Gütl, C. (2013). Assessment for Complex Learning Resources: Development and Validation of an Integrated Model. (p. 8). Graz: International Journal of Emerging Technologies in Learning.

Westera, W. ,. (2008, March 14). Serious games for higher education: a framework for reducing design complexity. *Journal of Computer Assisted Learning* .

Wikipedia. (2013a, 12 5). *Bots*. Retrieved 12 7, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Internet_bot

Wikipedia. (2013b, Nov 25). *NPC*. Retrieved Dec 7, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Non-player_character

Wilkes, I. (2008, Dec 17). *InfoQ*. Retrieved May 30, 2013, from Facilitating the spread of knowledge and innovation in enterprise software development: http://www.infoq.com/news/2008/12/Second-Life-Ian-Wilkes

Zameer, A. (2010). *Virtual Education System (Current Myth & Future Reality in Pakistan).* Pakistan: Virtual University of Pakistan.

Model Sources

The following links are the used 3D models from Google Warehouse (Google, 2013) last visit at 11 July 2013.

http://sketchup.google.com/3dwarehouse/details?mid=f8a23c22e76666a3d4ab0eed66631cfe&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=babca38e80c63f91e6cd1bb80a7890ad&prevstart=48

http://sketchup.google.com/3dwarehouse/details?mid=3a0bb4b6c7245e0435836c728d324152&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=52cfcdc06ac7b38018b14db3b83de9ff&prevstart=72

http://sketchup.google.com/3dwarehouse/details?mid=53c8c94f6076181ae4406b3bf37e6ffe&prevstart=12

http://sketchup.google.com/3dwarehouse/details?mid=9e21af3684387c9aa88426a9ceacc4c5&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=4247fe061be1065d4a92f48a2b62c4d4&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=6f1cecf465a034be73c75f2533c5ba52&prevstart=36

http://sketchup.google.com/3dwarehouse/details?mid=efec2e07aadd4ec48e35db9103756ad5&ct=mdrm&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=716af513c7f4639ecb00d6dcb3027517&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=5bacc00c91fee1d8d3668b04e6042e3e&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=945e668ffa4c70a8570d1eb9ae551581&prevstart=12

http://sketchup.google.com/3dwarehouse/details?mid=16f91ea010e82f7f4924375dc722fed&ct=mdrm&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=e0b889fcc114178cfdf3032c89935a17&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=e4335eaea4bea8b0985f8a701ee12764&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=1e8d0348ccfd812eaa7046b396c6641f&prevstart=12

http://sketchup.google.com/3dwarehouse/details?mid=123a6bf52a9a2b31caf17c12e77aefad&prevstart=12

http://sketchup.google.com/3dwarehouse/details?mid=5f8b0fd0c34832af7485c9619e38cb38&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=8177a01b82f85e62d53f914824e09203&prevstart=12

http://sketchup.google.com/3dwarehouse/details?mid=f78007f79be0a7f8ca6b84308f94076d&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=2e92e1ec1038ab0914513156cf2b8d0d&prevstart=36

http://sketchup.google.com/3dwarehouse/details?mid=1661cf7c472b693b722461155cacc9e3&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=e15c3181452b90f6a8eab5afb9aabc9a&prevstart=96

http://sketchup.google.com/3dwarehouse/details?mid=e3ac2c55a54713e07f5f437395597a36&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=499630aa04f4a2155f37cd009a491b7e&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=a38fd74a5a5900412251f6cc77ae450c&ct=mdrm&prevstart=24

http://sketchup.google.com/3dwarehouse/details?mid=d0699f84627c8d88f4360c2dcb3ca6c&ct=mdrm&prevstart=24

http://sketchup.google.com/3dwarehouse/details?mid=7f2ccb2e3e711a82bbccc8afc43b40b4&prevstart=0

http://sketchup.google.com/3dwarehouse/details?mid=475da7c72bb5dacb952a1a06cc58a0be&prevstart=12

http://sketchup.google.com/3dwarehouse/details?mid=e3e331e062b04b59f559279c88cd9a37&prevstart=0

List of Figures

List of Tables

List of Listings

List of Abbreviations

UDP	User Datagram Protocol
TCP	Transmission Control Protocol
SIP	Session Initiation Protocol
CAGR	Compound annual growth rate
RTP	Real-Time Transport Protocol
AIML	Artificial Intelligence Markup Language
OWL	Open Wonderland
SL	Second Life
NLP	Natural Language Processing
SAS	shared application server
RE	Requirement Elicitation
IMA	integrated Model for e-Assessment
GSD	Global Software Development (GSD)
RPG	Roll-Play-Game

Appendix

Appendix A: CD Content

The attached CD contains the following resources.

- Evaluation papers
- Sources for the modules
  - StoryComponent
  - Chatbot
  - LogBook
  - EntryWorld
  - Office
  - ccStore
- Evaluation forms
- Story document
- Pdf version of this documentation
- Summary of evaluation results
- The evaluation papers
- System Analysis and Design 251 Assignment by John Venable (UseCase)