

A Virtual Interactive Community Platform Supporting Education for Long-Term Sick Children

Pieter Jorissen¹, Fabian Di Fiore¹, Gert Vansichem², and Wim Lamotte¹

¹ Hasselt University
Expertise Centre for Digital Media - IBBT
Transnationale Universiteit Limburg
Universitaire Campus, BE-3590 Diepenbeek (Belgium)
{pieter.jorissen, fabian.difiore, wim.lamotte}@uhasselt.be
<http://www.edm.uhasselt.be>
² Androme NV
Wetenschapspark 4, BE-3590 Diepenbeek (Belgium)
gert.vansichem@androme.be, <http://www.androme.com>

Abstract. Analysis of existing ICT-based solutions for the education of long-term ill children reveal several weaknesses with respect to social and cooperative involvement, cost of development, and mobility issues. We present a scalable and affordable solution that supports both the social needs and learning process of these children. An educational platform was created based on the concepts of 3D networked virtual environments and Virtual Interactive Communities. In this work we describe the system architecture, network setup and protocols, and how we implemented the educational support. Our solution incorporates innovative hardware, software and connectivity features, set in a user friendly user interface based on networking and 3D technologies. It helps to establish high quality involvement of the long-term sick children in a communication based scenario between the place where the child stays/has been moved and their original classroom learning setting.

Key words: Cooperative Learning Environment, Edutainment, E-Health, E-Learning, Virtual Interactive Communities, Social Networks

1 Introduction and Motivation

Health care is becoming less hospital-centric; the hospitalization periods are much briefer and treatments are increasingly carried out at home. This shift has mediated medical, social and economic reasons and consequences. Furthermore, in the case of long-term and chronically ill children, this shift has a major impact on their education. The responsibility to provide education, is transferred from the hospital to the school which the children attended before their absence. Regular schools, however, are hardly able to set up high quality instruction for

their home-based pupils. As a result, the socialization opportunities offered by schools are no longer available for them and friendships between the ill child and fellow pupils become more and more disintegrated [1, 2].

In this work, we present a cooperative, community-based, E-Learning platform that aims at re-establishing the communication link between the place where the child is staying and the original school setting in view of supporting high-quality instructional scenarios. Analysis of existing ICT-based solutions, for educating long-term sick children ([3–6]), reveal weaknesses including the absence of social involvement, high development costs and mobility issues. Furthermore, although scientific publications have been written about several ICT-solutions, most of them focus on the results rather than giving a clear insight into the development process or the technical details.

The solution we propose, is based on the concepts of 3D networked virtual environments (NVEs) and Virtual Interactive Communities (VICs). It has to be stressed that we do not aim on the development of a new electronic learning environment (ELE) such as there are: Blackboard, WebCT, Anywise, etc. or a content management system (CMS). In contrast, we concentrate on a solution that can be linked to or used in cooperation with existing ELEs and CMS applications. The platform we present, does not focus on developing content, is based on concrete user needs, is educationally sound and relevant, and offers a scalable and affordable solution. The communication provisions build on audio and video (A/V) links, using standardized protocols, and help to support educational scenarios that support learning processes. Regarding the entire process, five interactive steps were undertaken: (i) analysis of user needs, user characteristics and context factors, (ii) design, (iii) development, (iv) evaluation, and (v) delivery. The focus of this work is on the more technical aspects of the design and development processes. More specifically we will give an insight on system architecture, network setup and protocols, consistency maintenance and visualization.

In the following section we elaborate on related research in the fields of E-learning solutions and VICs. Thereafter, we will give a brief overview of the required functionality. Section 4 discusses the details of how this functionality was implemented. In the fifth section, we will give some preliminary results. Finally, we give some concluding remarks and future research directions.

2 Related Work

2.1 ICT solutions for E-Learning

To date, several ICT-based solutions have been proposed in order to support education for children with health impairments. The most remarkable probably is the PEBBLES project (Providing Education By Bringing Learning Environments to Students). It comprises an advanced prototype solution developed in the USA and Canada [3, 6]. It was launched as the worlds first fully functional telepresence application: a social and technological solution that virtually places

a child within the classroom by putting a robot which replaces the sick child in the regular classroom. The robot is connected through a high-speed Internet line to the tools in the hospital. Major advantages are stated to be the remote control opportunities and the possibility to create an authentic learning setting because of two-way video, audio and document transmission possibilities. The PEBBLES solution enables a strong synchronous and authentic presence of the sick pupil in the original classroom setting. An advantage for the school/teacher perspective is the scant investment of extra time and energy to develop alternative solutions for these children. In addition to these advantages, it is possible to determine some problems of the PEBBLES-solution. For example, no documentation can be found on what happens when the sick pupil is unable to follow lessons synchronously due to health related problems such as tiredness, lack of concentration or therapies scheduled during the lessons. As no information can be found in articles or on the Internet, we assume that this solution lacks backing-up functionalities for missed classes. Furthermore, there is the issue of mobility of the ICT-based solution. Moving to another hospital room or classroom, moving back and away from hospital or home may cause interruptions or even an impossibility to continue the instructional experience. Another critical issue, hardly mentioned in the literature, is the high cost of developing and maintaining the PEBBLES-provision.

Another solution is STARBRIGHT World (SBW), a safe and secure online community where these children can connect to each other. Children on SBW can chat, read and post to bulletin boards, email, search for friends with similar illnesses, participate in fun events and contests, surf pre-screen Web sites and play games [7]. Further analysis of the available papers on the use of SBW, however, pointed out that the communicative possibilities were seldomly used by the children: only 3% to 15% of the time was spent on communication [5]. Another downside of this project is that it does not support educational scenarios.

In Flanders, the Dutch speaking part of Belgium in which our research project is carried out, a type of video phone is already in use to support long-term sick children to stay in touch with their family and peers at school [4]. However, hospital personnel has experienced some problems with these tools: there is an asynchronous delivery of sound and images, they offer basic video connection capabilities of rather low quality and if it is often used, it is a rather expensive solution because of the payment per minute of talking. In addition, each use of a video phone involves extra costs. For these reasons, the video phone has ended up in a closet in most of the hospitals we visited. Besides the use of this video phone device, ELEs through which the school, parents and pupils get in touch more regularly, are increasingly promoted. However, these tools build heavily on text-based input and communication and are therefore less suited for pupils of elementary school age. Also, they create a less authentic setting and hardly support the active and interactive involvement of the sick children.

2.2 Virtual Interactive Communities

We define Virtual Interactive Communities (VICs) as 3D virtual environments (VEs) in which (groups of) people sharing the same interests or ideas can easily communicate and interact with each other. Over the past decade, VICs have evolved tremendously. Undoubtedly the most popular VICs, nowadays, are the Massive Multiplayer Online Role-Playing Games (MMORPGs) with millions of subscriptions worldwide. Second Life [8], World of Warcraft [9] and Knight Online [10] are some examples. What these VICs have in common, is the fact that they build on high-end PCs with permanent broadband access to the Internet. Also, most of them rely on text chat as their primary means of communication (although some also supports voice communication, usually with added costs). In most VICs the users are represented by virtual animated characters, usually referred to as *avatars*. These represent the location and the actions of the user in the virtual world. In [11, 12] we proposed an extension to these avatars. We presented a framework for networked 3D VE applications that incorporates real-time video communication between avatars. We strongly believe that the combination of 3D worlds and live video streaming is perfectly suitable for this work, since it results in more explorative and fun learning environments than present-day solutions. Moreover, our aim is to work with children of elementary school age, which are assumed to prefer non-abstract environments with a limited amount of textual cues.

3 Requirements and User Needs

Our goal is to re-establish both the communication link between the sick child and its classroom and allow it to follow the class activities with fellow pupils. By providing the involved parties (child, teacher and fellow pupils) with access to a VIC that allows for A/V communication, we can already solve some aspects of our goal, however it is far from sufficient. From our observations and functional analysis ([13, 14]) we learned that in order to fully achieve our goal, our learning platform should support:

1. *Synchronous lessons*, including two way A/V stream so both sides can see and hear each other, a high resolution view the chalkboard and the possibility for the child to draw the class' attention.
2. Digital versions of the *diary and timetable*.
3. *Asynchronous lesson possibilities*, as the sick child will not be able to follow all classes due to tiredness, treatments or sickness.
4. Exchanging *homework, exercises, tests and markings* both for synchronous and asynchronous lessons.
5. *Social functionalities*, since the involvement in informal activities at school are at least as important as being involved in formal activities. In particular, children indicated they miss the socialization opportunities offered at school.

In the next section we discuss how our platform supports these functionalities.

4 Approach

4.1 The VIC Learning Platform

The virtual learning environment that we propose is based on our earlier experiences with VICs [11, 12]. This includes support for various synchronous communication components into the 3D VE such as audio conversation, video conferencing, text chat and video-based avatars. We have extended this work with community support functions like buddy lists, authentication and authorization functionalities, rich presence (showing who is on line, which mood, what kind of activities one is pursuing), media publishing (diary, timetable, exercises), and file exchange.

Using this platform, we built a VIC application implementing the required functionality to achieve our goals. A logical overview of our platform is shown in Fig. 1. Here, we also see that our solution consists of several components, which all play an important role: (i) the virtual school environment, its functionality and how it is displayed, (ii) the underlying network supporting it and (iii) the hardware. The platform is programmed in C++ and runs on most Windows based systems. This was chosen so most home computers will be able to run the software.

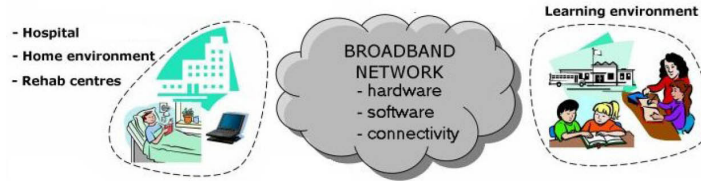


Fig. 1. Overview of the Virtual Learning Platform approach

4.2 The Virtual School Environment

The VE of our test application is a 3D modeled environment that is rendered using the Ogre 3D open source graphics engine [15]. Taking into account that the community is children-oriented, all its objects are created in cartoon style and interaction is kept as intuitive and natural as possible (by clicking easily understandable icons). Our system, however, is not restricted to a specific 3D environment. Instead, each school is free to choose the look of the VE. Navigating in this virtual world happens by means of an animated avatar that is controlled through the arrows on the keyboard. This corresponds to navigating in present-day games and on-line communities like Ice Age 2 and (Teen) Second Life. The looks of the avatars is also adjustable by the participants (type of character, colors of clothing, hair,...).

The virtual playground (Fig. 2 (a)) is the starting position of the user in the VE. Therein, the child can navigate, meet its friends and teachers and communicate through video avatars (Fig. 2 (b)). When it is time for class, the pupil can enter the virtual classroom, which is a 3D scene containing the most important elements of the real classroom environment: a live video A/V stream of the class (top, left), a high resolution captured image of the chalkboard (middle), the pupils school desk containing his diary, homework book and school timetable (bottom), and the pupils personal bookshelf (top, right). A view of the virtual classroom is shown in Fig. 2 (c). In the real classroom, the teacher and fellow students, have a similar view. Only here, the video stream of the child is displayed instead of the class' video. Furthermore, several function buttons allow the users (child, teacher, fellow student) to quickly launch specific tasks such as: scanning and transmitting documents (with automatic printing), capturing the chalkboard and drawing/granting attention. Finally, the user is also able to switch to a close-up view of classroom elements by simply clicking them to get a better look, and get a view of its own video. Virtual books (Fig. 2 (d)) can be used to store documents, scans (of homework, tests,...) and pictures and can be accessed at any time. Furthermore, teachers can add recorded lessons (webcam feed and slideshow of the chalkboard images), in case the sick pupil could not attend a lesson. This way, he can view these at later.

Before we discuss the details on how the functionality works, we take a closer look into the network architecture.



Fig. 2. Views of the school environment: (a) the virtual playground, (b) video avatars, (c) the virtual classroom, (d) a close-up of the class diary with scanned pages.

4.3 The Network Architecture

The network architecture that supports our virtual community consists of several servers, each with their own responsibilities. In order to be as general and interoperable as possible, we used standardized protocols as much as possible. The division into the different servers is not necessary physical; in practice several servers can be run on a single machine. Due to the desire to be compatible with existing standards and the dependency on communication services a substantial part of the architecture is inspired on the IP Multimedia Subsystem (IMS [16]). With the Session Initiation Protocol (SIP [17]) being the key technology behind

the IMS, the client application incorporates the Intellivic SDK [18], a toolkit for integration SIP-based video telephony into an application. On the network side this choice also implies the presence of elements from the IMS architecture like session management and communication services.

The first server used, is the *Session Management Server*. It takes care of and authenticating users and logging users on and off. In addition, it is also being used to initiate A/V communication sessions. In order to communicate and manage these sessions, SIP is being used. A second server is used for communication. Although peer-to-peer communication is possible when only two parties are involved (e.g. sick child and classroom). When more parties become involved, it is desirable to use the *Communication Server*, since the setup of peer-to-peer connections between multiple parties is impossible due to bandwidth limitations. The solution is to send a single stream to the communication server, which relays it to the different parties. This way every participant only has to send a single A/V stream instead of one for every participant. It is the Session Server that is responsibility to indicate clients which communication server to use (in case more than one is available). Due to the real-time nature of the A/V streams the Real-Time Transport Protocol (RTP [19]) is the obvious choice for the underlying protocol. Other possibilities for even more efficient A/V stream transmission would be to use multicasting as was done in [11, 12]. However, since multicasting is not yet fully available in practice, it could not be used during our field trials. In order to not exceed the 128 kbps (or 256 kbps) upload bandwidth limitation, that is available in most Flemish homes, we need to compress our A/V streams. Currently we use mpeg4 80kbit QCIF compression for video and speex_narrowband 15kbit compression for the audio. These parameters can be remotely configured to provide the desired quality and exploiting the available bandwidth optimally. Managing synchronization and persistence of the 3D VE is the responsibility of the *VIC Server*. This involves transmitting the world itself, synchronization of object and user states, and managing and storing a persistent world state. Apart from location, user states also include information on its mood and actions. File transfers are done using the File Transfer Protocol (FTP [20]) and update messages for objects and clients are sent using ENet [21], a thin, simple and robust network communication layer on top of UDP. Finally, The *Data Server* is used for two categories of data. On the one hand, it stores configuration files of the users containing personal information such as A/V parameters and address lists of the community. On the other hand, the server makes it possible to transmit and store (shared) media and other data files such as shared virtual books. Underlying protocols used are FTP and Hypertext Transfer Protocol (HTTP [22]). An overview of the network architecture and the used protocols is given in Fig. 3.

4.4 Providing Education

Since we are working with children and teachers who usually have little computer knowledge, we want to hide as much of the technical details as possible and provide intuitive interaction methods. All user data and configuration settings

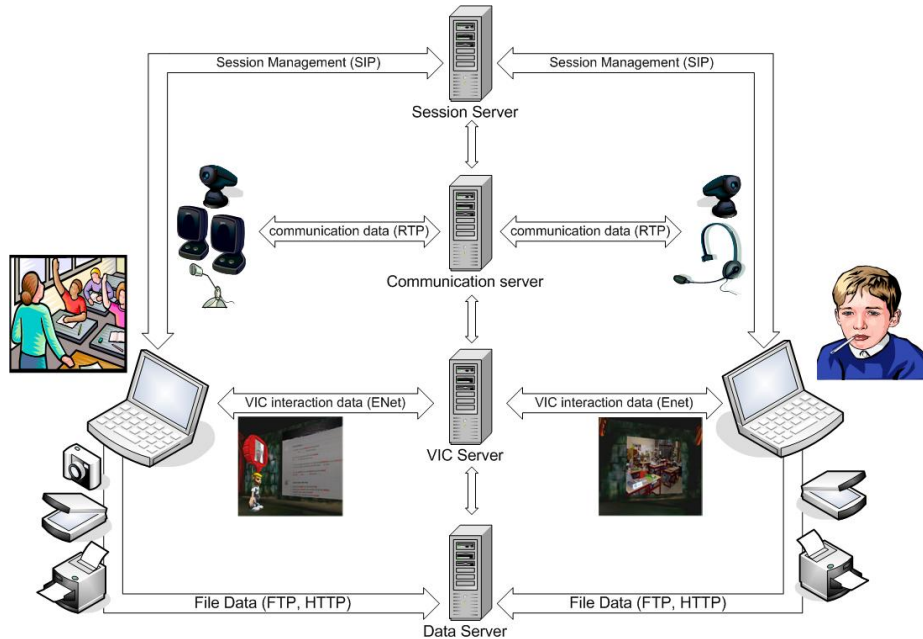


Fig. 3. Overview of the network architecture and the necessary client hardware.

are set up in advance and saved in an XML file that is parsed on startup of the application, and backed up on the data server. Logging on and off via SIP is done completely transparently on startup. The user data, webcam and audio settings are read from file and used to log on to the session management server. Immediately, the user can see which other participants are in the environment and in what mood they are in. When other users are encountered on the playground, an A/V session is set up automatically. Also, when both the sick child and classroom are logged in and have entered the virtual classroom (which is done simply by clicking the door of the classroom in the virtual school environment) an A/V session is started automatically. The video images are incorporated in the 3D classroom using video textures.

Apart from synchronous A/V communication, the most important functionality is the easy *exchange of documents* (drawings, tasks, exercises or tests). We have also kept this task as simple as possible for the users. The different steps of synchronous exchange of a scanned document are shown in Fig. 4. Note that the user only has to perform step 1 and 2, the rest is done completely automatically. An example of the use of this functionality is when a teacher hands out a page of exercises to his students. He can put the sick pupil's page in the scanner and send it to him with a single button click. On the other side the document is automatically downloaded and printed. The child can then fill in the exercises and send his results in the exact same manner for correcting and/or grading. Taking *pictures of the chalkboard* happens in a similar way. The only difference

is the final step. Instead of automatically download and printing, the picture is integrated in the virtual 3D scene. In order to take these pictures, the classroom must be equipped with a digital camera that can be controlled by software. More details on the hardware are given in the next section. The *virtual books* also use the same principle and can, besides the diary and time table, be used for storing recorded lessons, pictures from field trips, drawings that fellow pupils made or just digital notes. Currently a virtual book page can contain: a scanned image (text, photo, drawing,...), a video, a recorded sound or a link to a file or web page. Books can be opened from the virtual library, and virtual pages can easily be added and removed. The data server contains a list of all books and each book (with a description of the pages) is stored in a separate file that is only downloaded when it is opened. The actual page data (images, videos,...) is only downloaded when the page is viewed for the first time, in order to keep the network load as low as possible. A local copy is then kept on the client.

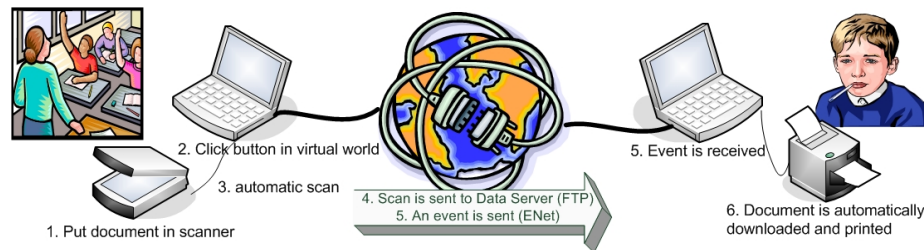


Fig. 4. The different steps of a synchronous document exchange in the learning environment.

4.5 Hardware and Network Requirements

One of our initial objectives was to develop an affordable solution. Therefore, we tried to exploit commodity hardware as much as possible. The hardware that is necessary at the pupil's and the classroom site can be seen in Fig. 3. At first, we have the PC/laptop that runs the software, this can be a commercial off-the-shelf machine; it only needs to have a 3D accelerator graphics card in order to fluently visualize the 3D environment. Most Flemish households have a home PC that fits these requirements, so only the software installation is required to run our E-learning platform. In our current setup we use Pentium 4, 2.4 GHz PCs with 1 GB of RAM in the classroom and Pentium 4, 3.06 GHz portables with 512 MB of RAM for the pupil's setting. Furthermore, both sides need A/V hardware. Tests pointed out that a resolution of 320 x 240 suffices to have a decent view and frame rate (25fps), so off the shelf webcams can be used. Concerning audio, the pupils can make use of a headset or a microphone that is integrated in most available webcams. The classroom on the other hand is equipped with a wireless microphone together with a set of speakers. This requires the need for acoustic

echo cancellation (AEC), which may be integrated in the microphone or in the software. Furthermore, both sides need a printer and a scanner. Since we use Windows Image Architecture (WIA) for scanning and printing to the default printer, most available printers/scanners can be used. Finally, in the classroom, a digital camera is employed in order to take snapshots of the chalkboard. During the test phase, results indicated that we should use a minimal resolution of 1600 x 1200 pixels (i.e. 2.1 Mpixel) for readability, (ii) to avoid using the flash, (iii) that colors look better on a green chalkboard than on a marker whiteboard. In our current setting, we use Canon Powershot A620 cameras controlled in our software using the Canon PowerShot Remote Capture SDK [23]. This allows to control the camera's settings (resolution, zoom, flash,...) and permits taking pictures which are automatically loaded onto the PC.

Regarding bandwidth, the real bottleneck is the streaming live audio and video. In the Flemish setting, most people and schools own a xDSL or cable connection (download speed: 4.4Mb, upload: 128 to 256kb). These connections can easily handle the streams of the resolution and frame rate mentioned here, using the compression techniques mentioned in section 4.3. Even far better quality can be used when 256 kbps upload is available.

5 Results

In order to evaluate our system, several field trials have been set up with a realistic target audience (i.e. sick children and their schools) and within authentic settings (i.e. at home or at a hospital). The authentic setting required a diversity of preparatory work: methodology, installation, establishing connectivity, introduction of the package, planning front-office support, data analysis and reporting. Although at the time of writing not all field trials have finished, current tentative data are significant enough to conclude that our solution provides the functionality that we proposed and that the current hardware requirements are able to support it. With regards to usability, we can already conclude that the users, especially the children, are really pleased with the approach of a VIC for both the learning and social aspects of attending class. All users state that the social bond between the sick child, his fellow pupils and friends and the teacher has dramatically improved. To completely evaluate the fulfilling of concrete user needs as indicated by the needs analysis, the data of the individual case studies will need to be combined. These complete results of the user needs are not yet available however. In order to give a more complete image of our learning platform, a video of a lab test and some field trials can be found on <http://research.edm.uhasselt.be/pjorissen/CDVE2007/>.

6 Conclusions and Future Work

This work presents a cooperative, community-based, E-Learning platform based on the concepts of 3D NVEs and supporting VIC functionalities. While our goal was not to develop an ELE or CMS, we showed how, based on concrete user

needs, our platform can be used to support educational scenarios that support learning processes. We described the underlying network architecture and the employed protocols, that support the communication provisions through which the communication link between the sick child and its original classroom and friends was restored. Our system offers a scalable and affordable solution, it utilizes standard hardware and is adjustable to most home network setups.

Tentative results show that our solution functions well and is technically sound. Currently more in-the-field trials are being conducted. The investigation of how our platform will socially and educationally influence long-term ill children will be subject of our future work. More specifically, it is recommended to investigate the effects of using user-based ICT-solutions on several relevant outcomes such as the experiences of children when reentering their school, the academic achievement and psychosocial outcomes (e.g., well being, coping and social adjustment).

Acknowledgments. This study was conducted in view of the ASCIT-project (Again at my School by fostering Communication through Interactive Technologies for long term sick children) financed by the Flemish Interdisciplinary institute for Broadband Technology (IBBT) and following partners: Androme, Alcatel, Artec-Electronics, Televic, Vlaams Patiëntenplatform, Hospital School Gasthuisberg (Leuven), Bednet vzw, and Vlaamse Liga tegen Kanker. We gratefully express our gratitude to the European Fund for Regional Development (ERDF) and the Flemish Government which are kindly funding part of the research at the Expertise Centre for Digital Media. Many thanks go also to Xemi Morales for his artistic contribution.

References

1. F.F. Prevatt, R.W. Heffer, and P.A. Lowe. A review of school reintegration programs for children with cancer. *Journal of School Psychology*, pages 447-467, 2000.
2. D.I. Fels, B. Shrimpton, and M. Robertson. Kids in hospital, kids in school. In *Proceedings of EdMedia*, pages 2358-2363, 2003.
3. P.L.Weiss, C.P. Whiteley, J. Treviranus, and D.I. Fels. PEBBLES: A personal technology for meeting educational, social and emotional needs of hospitalized children. In: *Personal and Ubiquitous Computing (2001)*, pages 157-168.
4. Jonge Kamera v.z.w. Jonge kamera. <http://www.jongekamera.be>.
5. H.B. Battles and L.S. Wiener. Starbright World: Effects of an electronic network on the social environment of children with life-threatening illnesses. *Childrens Health Care (2002)*, pages 47-68.
6. The PEBBLES project. <http://www.ryerson.ca/pebbles/>.
7. Starlight Starbright childrens foundation. Starbright World. <http://www.starbrightworld.org>.
8. Second Life. <http://secondlife.com>.
9. World of Warcraft. <http://www.worldofwarcraft.com>.
10. Knight Online. <http://www.knightonlineworld.com>.

11. P. Quax, C. Flerackers, T. Jhaes, and W. Lamotte. Scalable transmission of avatar video streams in virtual environments. In: Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME 2004), pages 631–634.
12. P. Quax, T. Jhaes, P. Jorissen, W. Lamotte, A Multi-User Framework Supporting Video-Based Avatars. Proceedings of the 2nd workshop on Network and system support for games. NETGAMES 2003. California.
13. E. Lombaert, P. Veevate, D. Schuurmans, L. Hauttekeete, M. Valcke, A special tool for special children: creating an ICT tool to fulfill the educational and social needs of long-term sick children. In Proceedings of the International Conference on Multimedia and Information and Communication Technologies in Education (m-ICTE), 2006.
14. E. Lombaert, M. Valcke, Education for Long-Term Sick Children: Towards an Integrated IT-Solution, To appear in Computer Assisted Learning conference (CAL '07), 2007.
15. Ogre 3D. <http://www.ogre3d.org>.
16. SIP Centre on IMS. <http://www.sipcentre.com/sip.nsf/html/IMS+IP+Multimedia+Subsystem>
17. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol, RFC 3261. Technical report, 2002.
18. Androme Intelliviv SDK, <http://www.intelliviv.com/>.
19. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications, RFC 3550. Technical report, 2003.
20. J. Postel and J. Reynolds. FTP: File transfer protocol, RFC 959. Technical report, 1985.
21. ENet. <http://enet.bespin.org>
22. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. HTTP/1.1: Hypertext transfer protocol, RFC 2616. Technical report.
23. Canon's European Developer Programmes. <http://www.developers.canon-europa.com>.