Vrije Universiteit Brussel

FACULTEIT WETENSCHAPPEN EN BIO-INGENIEURSWETENSCHAPPEN
Departement Computerwetenschappen
Web & Information Systems Engineering Laboratory

Audience-driven Presentations based on the MindXpres Presentation Tool

Proefschrift ingediend met het oog op het behalen van de titel Master in de Ingenieurswetenschappen: Computerwetenschappen, door:

Christophe Vermeylen

Promotor: Prof. Dr. Beat Signer
Begeleider: Reinout Roels

AUGUSTUS 2013

Vrije Universiteit Brussel

FACULTY OF SCIENCE AND BIO-ENGINEERING SCIENCES
Department of Computer Science
Web & Information Systems Engineering Laboratory

Audience-driven Presentations based on the MindXpres Presentation Tool

Dissertation submitted in partial fulfillment of the requirements for the degree of
Master in Engineering: Computer Science, by:

Christophe Vermeylen

Promoter: Prof. Dr. Beat Signer
Advisor: Reinout Roels

VRIJE UNIVERSITEIT BRUSSEL • SCIENTIA VINCERE TENEBRAS •

AUGUST 2013

Samenvatting

Met ruim 30 miljoen PowerPoint presentaties per dag, kunnen we het belang van presentaties niet negeren of onderschatten in onze moderne samenleving. Van onderwijs tot bedrijfsleven, overal zijn presentaties aanwezig en helpen ze om kennis en informatie door te geven en te verspreiden. Op de dag van vandaag kunnen we kiezen uit verscheidene presentatietools. Deze bestaan al sinds de jaren tachtig, maar tot nu toe is er bitter weinig veranderd aan hun kernconcepten. Eén van de meest elementaire concepten zijn slides, de basis van de zogenaamde "slideware" tools. Deze vinden hun oorsprong bij analoge toestellen zoals een overhead- of diaprojector. Slideware ondersteunt de presentator in een eenzijdige communicatiestroom van presentator naar publiek. Terecht kunnen we stellen dat slideware presentatorgericht is.

We kunnen met zekerheid zeggen dat huidige presentatietools geen ondersteuning bieden aan meerzijdige communicatiestromen. Iemand uit het publiek kan met een vraag zitten over de presentatie en deze willen stellen aan de presentator. In deze situatie kan men geen beroep op een technologische eigenschap van de presentatietool. Bovendien kennen presentaties vaak een tweede levenscyclus. Een presentatie kan worden gebruikt door het publiek nadat de effectieve presentatie heeft plaatsgevonden (e.g. het gebruik van hand-outs dat als studie materiaal aangeboden wordt). We zijn er van overtuigd dat deze ongelijkheid van ondersteuning in evenwicht moet gebracht worden en geloven dat het publiek minstens even belangrijk is (zo niet belangrijker). Presentatietools moeten het publiek veel meer betrekken, zowel tijdens als na de effectieve presentatie. Technische innovaties, zoals de introductie van smartphones en tablets, creëren hierbij nieuwe opportuniteiten.

In deze thesis werpen we een blik op bestaande oplossingen voor de vermelde ongelijkheid in ondersteuning. Verbeteringen aan presentatietools op dit vlak vallen onder de noemer van Audience Response Systems (ARS). Vooral in het onderwijs maken ARS programma's hun intrede en dus komt veel literatuur uit het educatieve gebied. Met een uitgebreide literatuurstudie en kritisch denkwerk identificeren we een aantal problematieken die inherent zijn aan de beschikbare ARS. Dit breiden we uit met een eigen onderzoek naar de huidige praktijk bij gebruikers van presentaties en hun percepties, opinies, en aspiraties naar publiekgerichte functies in een presentatietool. Op basis van

onze conclusies, identificeren we nuttige functies in het realiseren van multidirectionele communicatiestromen tijdens en na een presentatie. Dit wordt dan gebruikt om onze eigen ARS te maken in de MindXpres presentatie tool.

MindXpres is een nieuwe kijk op presentie tools, en baseert zich op recent onderzoek in de gebieden van hypermedia, spatial hypertext, en zoomable user interfaces. Het bevat een uitbreidbaar plug-in systeem, wat een interessante eigenschap is. MindXpres biedt immers de mogelijkheid om op een crowdsourcing-manier een ARS tool te construeren. Bovendien zal een gebruiker van MindXpres zelf zijn ARS functionaliteiten kunnen kiezen.

Wat volgt is een uitbreiding van de MindXpres-kern met een communicatiemodule die bovenvermelde conclusies en ideeën concretiseert. De gewenste ARS functies worden ontwikkeld als plug-ins voor MindXpres en maken gebruik van deze communicatiemodule. Hiermee wordt het mogelijk om alle communicatiestromen te ondersteunen tussen het publiek en/of de presentator tijdens en na een presentatie. Aangezien MindXpres gebruik maakt van HTML5, faciliteert dit de compatibiliteit en distributie van de presentie over een heterogene groep van toestellen. Als laatste, maar niet minder belangrijke bijdrage, ontwikkelen we een presentatie-box die de mogelijkheid biedt om alle gebruikers met elkaar te connecteren en hen te laten interageren via een WiFi-hotspot.

Abstract

Presentations are omnipresent in our daily life, in education as well as in business and in personal leisure time. Presentation tools are a great way to share information and knowledge with other people. From their first introduction in the early 1980s, the core ideas of presentation tools still remain unchanged. The most quintessential idea in present day presentation tools is the format of slides and hence these tools are referred to as "slideware" solutions. With slides, presenters are able to deliver their information and knowledge to present it to the listening audience. Slideware only supports one-directional communication flow from the presenter towards the audience. Therefore, we state that slideware presentation tools are purely presenter-oriented.

We can safely state that current presentation tools' features do not support other communication flows. For instance, an audience member can have a question concerning the presentation and ask it to the presenter. This oral communication flow from the audience towards the presenter is not really supported or enhanced by a presenter-oriented presentation tool. Additionally, presentations often have a second life and will be with the audience after the presentation was given (e.g. lecture handouts given as study material). We believe that this inequality of support by the presentation tool is incorrect and that the audience is equally important (if not more important). Tools should involve the audience more often, during and after the actual presentation. Technical innovations, such as the introduction of smartphones and tablets, create new opportunities in this respect.

In this thesis, we extensively cover existing solutions that try to solve this inequality. These enhancements to presentation tools are generally addressed as Audience Response Systems (ARS). Particularly in the field of education, ARS keep gaining popularity to this day and much of our research comes from this field. We studied the outcome of several literature studies to pinpoint advantages and disadvantages of actual available ARS. Through critical thinking and a conduction of our own preliminary study, we identify some features that can prove their usefulness in realising a complete, multi-directional communication flow during and after a presentation. These conclusions are used for the creation of our own ARS for the MindXpres presentation tool.

MindXpres is a new presentation tool that introduces some radical changes to create, share and deliver presentations. It differentiates itself technically from existing tools by providing innovative features such as a plug-in architecture. This becomes interesting for our thesis because it allows a croudsourcing-based way of creating an ARS. Furthermore, it also allows the user to select only the ARS feature they desire.

What we present next is the extension of MindXpres with a communication module that takes our earlier defined ideas in consideration. All desired ARS features can then be created as plug-ins that use this module to communicate with all the devices that are actively listening to the presentation. The presentation is visualised in HTML5 which facilitates the portability and distribution over heterogeneous devices. Last but not least, we also built a dedicated presentation-box which enables users to connect and interact by means of a WiFi hotspot.

Acknowledgements

This Master thesis would have never even existed without the help and guidance of several people. First of all, I would like to thank my promoter Prof. Dr. Beat Signer, for giving me the opportunity in realising this thesis and all the guidance in the process. The same gratitude goes towards my advisor Reinout Roels who was also always there to help, for all the proofreading (over and over again) of my thesis document, and for his time and effort. Also Sandra Trullemans receives my gratitude for the help she provided in my research.

Next, I would like to thank my family for their support. First to my parents, for giving me this amazing opportunity of studying. A special thanks goes to my father for the extra proofreading of my document and giving me some extra insights on how to improve it. Without his interest and support in my study and giving me extra lessons in Mathematics, I would have never been able to start this Master. My brother Patrick did some of the artwork in this thesis and I thank him for the beautiful illustrations. Philip gave me useful information about classroom management systems. Of course, sister Veronique is not to be missed out in my gratitude.

Finally, a big thanks goes to all my friends. I am not going to name any because of the fear I leave someone out. However, if you read this you can be sure you are one of them. Thank you for the support in studies, sharing my interests, or letting me blow off some steam once in a while.

To all these people, thank you!

Contents

1
Introduction

With more than 30 million presentations [34] that are created every single day, we are all familiar with modern presentation tools such as Microsoft's PowerPoint[1], Apple's Keynote[2], or OpenOffice Impress[3]. These tools are used in various ways ranging from education to business to making a presentation in your leisure time.

Currently, most of the features in the modern presentation tools are very much presenter-oriented. The features are there to help the presenter in creating and modifying their slides. Additionally, in some cases presentations have a second life and will be used by the audience long after the presentation was given (e.g. class lecture slides that are also used as study material). We believe that the audience is equally important (if not more important) and that tools should involve the audience more often, both during and after the actual presentation.

The presenter-oriented features of the presentation tool are sometimes sufficient for the presenter's needs. Think in the context of a lecture course where the professor explains different concepts with the help of their presentation slides. In this case, the professor does not require audience input since it is a one-directional communication going from the professor towards the audience. But what if this same professor would like to test their students after every concept that was explained. We quickly reach the limits of the offered features in current presentation tools. One way to solve this is without the use of any technical tool. The professor could do this by asking a question to the audience and expect the audience members to

[1] http://office.microsoft.com/en-us/powerpoint

[2] http://www.apple.com/iwork/keynote

[3] http://www.openoffice.org/product/impress.html

react in some way like raising their hands or answer the question. However, the audience members are very exposed and are often too shy to answer the question. Or the audience follows the rest of the members in order to avoid embarrassment of giving an incorrect answer.

To solve this issue, we look at Audience Response Systems (ARS). ARS is a technology that focusses on including the audience in any sort of presentations. It creates interactivity between a presenter and the audience of a presentation and it gives us insight of some audience-oriented features. Throughout this document, we adopt the term ARS. Depending in which setting you are in, this term is often called differently (e.g. in educational settings the term Student Response System (SRS) is often used). We believe that ARS is the most general term.

To extend the available literature we also conducted a study of our own. We made a qualitative questionnaire that we distributed to a wide range of people. In the findings of our study, we discuss predefined hypotheses and determine perceptions, opinions and aspirations about ARS of people. Based on both our own findings and those derived from literature, we are able to identify what audience-oriented features are desirable for both presenter and audience.

The presentation tool that is used as a base for this Master thesis is MindXpres. It is a presentation tool that takes a radically new approach to create, share and deliver presentations. It provides a LaTeX -like language for the easy creation of content. This content is then automatically visualised by an HTML5 visualisation layer. MindXpres differentiates itself from existing slideware by providing some innovative features such as the plug-in architecture, extreme portability by the use of HTML5, support for multimodal input, semantic linking and navigation of information, non-linear traversal, a zoomable user interface, transclusion, interactivity, innovative ways of visualising specific types of information and much more.

Our goal in this thesis is not to create an anecdotal ARS in a presentation tool. We first familiarise ourself with the state-of-the-art solutions and conduct our own qualitative research. We wish to find out how people use presentations to this day. After we concluded our findings, we can start to create a general communication module that functions as an Abstract Programming Interface (API) for MindXpres. This API can be used by programmers to build their ARS plug-ins that require communication between multiple devices. This way, we can build a extensible and modifiable ARS tool. The programs can use this API to extend the ARS features offered at any time. As a proof of concept, we already implement some of the features that ARS offers today.

In order to build this communication module, we first have to ask the question "*What are the communication flows during a presentation?* ". After identifying the communication flows that exist during a presentation, we used these as a base for the communication module.

The work presented in this thesis fits in the context of making presentations more audience-oriented. It is therefore important to explore the current state of audience-oriented features in current presentation tool, these fall mostly under the umbrella of ARS. We extensively examine different ARS tools in Chapter 2, both academic and commercial, we identify and argument all the benefits, issues, and opportunities of those systems.

In Chapter 3, we present the findings of our own qualitative research that we conducted. We find out how people use presentations to this day. It also includes finding out some perception, opinions, and aspirations of people about audience-oriented presentation features.

Chapter 4 introduces us to the MindXpres presentation tool. We explain the information model that is used as an underlying base and describe the plug-in architecture. This plug-in architecture is important to support us in creating a extensible ARS tool.

Chapter 5 is a description of more technical aspects of this Master thesis. We describe different architectures that supports our intent. An investigation of different network protocols is also presented in this chapter together with a comparison of overhead and latency. The goal of this comparison is to select the best possible communication protocol for our implementation.

Chapter 6 brings all the previous work together and the implementation of our extensible ARS tool is explained. The description of how our communication module is built and the communication flows that we used as a base, are presented. We also explain the integration of the MindXpres presentation tool and how plug-ins with the desired audience-oriented features should be developed. This is followed by Chapter 7 where we describe a use case of our work. Finally, we conclude this Master thesis with Chapter 8 where we discuss our contributions and future work.

2 Existing ARS and their Limitations

In this chapter, we take a closer look at some of the existing ARS. All the tools have in common that they try to create interactivity between audience and presenter [23], but they differ in terms of technology and features. We discuss only a subset of all available tools because there are too many to discuss and many of them do not differ that much from each other. For every ARS tool that we discuss, we give a short description and what the exact differences are compared to other systems. Once we explained the existing ARS tools, we discuss the positive, pedagogical effects of using such a tool. We base ourself mainly on other studies. The chapter ends by discussing some issues of such systems and where the future opportunities are.

Many research studies about ARS have been done in the field of education. Therefore, the related work chapter contains a lot of research studies that focus on this field. We also mix the academic and commercial solutions with each other since many academic research has been conducted with commercial solutions. Looking at existing ARS systems gives us insights of what has been proven and what are still considered to be challenges. The goal is that our ARS framework continues where other studies stop, in order to go beyond what the existing presentation tools currently offer.

2.1 History

The idea of including the responses of your audience dates back to the early 1950s. The American military started to lay down the basic development that academical researcher could use as a starting point [7, 16]. One of the first real implementations of this system was developed by Stanford

University in 1966. This system was very expensive, difficult to use and did not function well [1]. Much of the history presented in this chapter is based on articles [1, 21].
One of the first real successful implementation of ARS was on television in the show “Who wants to be a millionaire?”. The idea of this television show was that the audience can participate during the quiz between the host and participator. Each audience member was handed a small handheld device where they could choose options to answer the question. Based on the result of that round, the audience member with the highest score could then play for the money.

2.2 Existing ARS

In this section, we discuss all different technologies used to create an ARS. We go from simple voting machines to a more advanced system where audience members have specialised devices that can be managed by the presenter (e.g.: the presenter is able to turn off all devices when they want the audience's focus).
For the first part of discussing existing ARS tools, we base ourselves on the following well-described literature studies [15, 20, 22]. While some of the studies are somewhat older, they remain relevant to this day.

2.2.1 Clickers

Clickers were the first kind of ARS and were introduced by Stanford University in 1966. In the literature, other terms are used like keypads, handsets, or zappers. We adopt the term clickers because it is the most common in academic research and commercial solutions. Clickers are little transmitters that can be held in one hand. They contain some buttons which represent possible answers to the question that has been asked by the presenter. The audience can answer the question by transmitting their choice to the central access point (computer). In the early implementations, there were a lot of drawbacks regarding the cost, reliability and usability that are solved nowadays. Today, there is a wide range of commercial implementations of these kind of systems like Turning Technologies[1], Iclicker[2], Meridia[3], IML[4], and many others. Clickers are currently the most used ARS. Iclicker alone claims to have sold more than 2 million of its clicking devices in 2011. In Figure 2-1 we present such a clicker device.

[1] www.turningtechnologies.com
[2] www.iclicker.com
[3] www.meridiaars.com
[4] www.imlworldwide.com

Existing ARS

Figure 2-1: Clicker device

The idea of having a small hand-held device did not change over the years. However, technology does not stand still and so these devices have evolved accordingly. In the beginning these devices were still wired to a computer, which made the implementation of such a system very static in its setting. Now one has a wide range of wireless technologies available which removes this static setting. It started by making the devices equipped with Bluetooth or Infrared (IR). More recently, clickers with Radio Frequency (RF) came in production. The benefits of using RF transmitters is the reliability and that the signal can be unique. This uniqueness can be used to identify the clicker if desired by the presenter. Many of the commercial clicker devices offer an integration with the most common presentation tools like Microsoft Powerpoint or Apple Keynote.
In table 2-1, we sum up some of the most important advantages and disadvantages of using clickers based on the following studies [15, 20, 22].

Advantages	Disadvantages
• Low equipment cost	• Limited two way communication from audience to presenter
• Most systems are easy to use	• Requires a complete installation
• No permanent installation with wireless	• Restricted to only voting
• Fast response time	• Time to distribute and collect the clickers
• Anonymity in answering	

Table 2-1: Advantages and disadvantages of the Clicker solution

2.2.2 Cellphone

Since sending small messages (SMS) via a phone is very familiar to people, it is a good medium candidate to use for ARS. An SMS-based ARS requires the utilisation of the audience's cell phones. Nowadays, these devices are nearly omnipresent in our daily life which makes the medium scalable to the size of audience. There is no need to acquire a designated device like a clicker. It provides a low-threshold application with a gentle learning curve. In the brief literature study in [31] about using SMS systems to create an ARS, it is reported that 80% of the students send a daily text message. Hence, special training is not needed. However, a dedicated SMS management system is required to be able to receive, process and display the messages. We can rely on some commercial systems for this like UR voting[1], Poll Everywhere[2], or SMSpoll[3], but as with the clickers systems, many other solutions can be found.

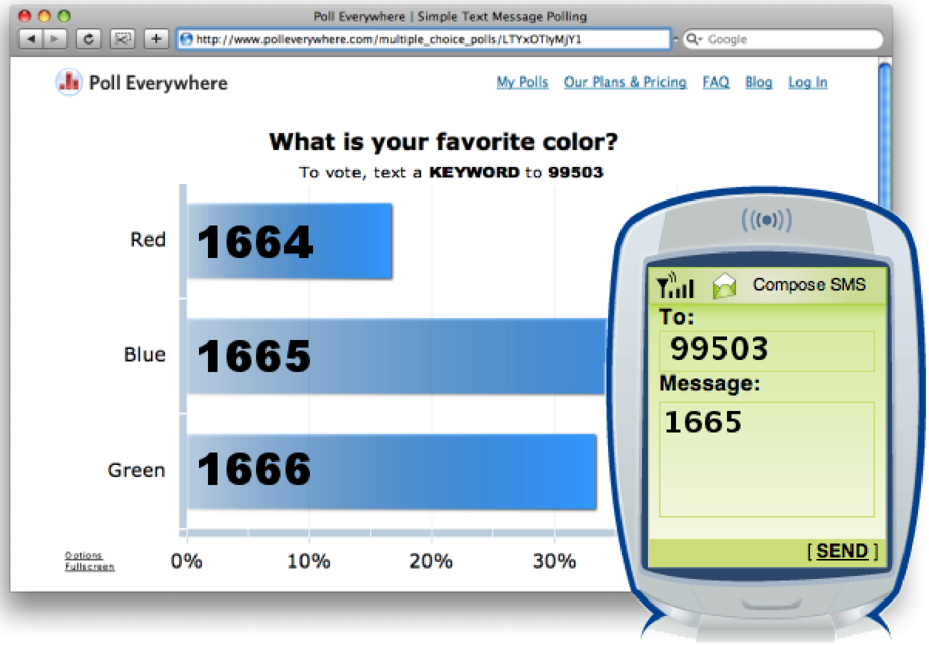
This technology still has some significant drawbacks compared to others. The main drawback is the cost of sending a text message. While it is reported in the studies that some students have free text messages, most of the commercial solutions charge an extra fee. This is why such a system is not feasible when one likes to use it extensively during a presentation. Other drawbacks were the lack of anonymity because everyone was using their personal number, possible time delays between sending and the display of information, and the need for the SMS management system.

With these drawbacks we can conclude that it is not a good candidate during a presentation. However, these systems are widely used in television shows or newspapers to participate in a game. The audience here are the viewers and readers accordingly. In such a setting, the cost of sending an SMS to the dedicated SMS management system becomes desirable for the organisation due to the profit of receiving such a SMS, other drawbacks become less important.

[1] www.urvoting.com

[2] www.urvoting.com

[3] www.smspoll.net



Poll Everywhere | Simple Text Message Polling
http://www.polleverywhere.com/multiple_choice_polls/LTYxOTIyMjY1
Google
Poll Everywhere
My Polls
Our Plans & Pricing
FAQ
Blog
Log In
What is your favorite color?
To vote, text a KEYWORD to 99503
Red
1664
Blue
1665
Green
1666
Options
Fullscreen
0%
10%
20%
30%
Compose SMS
[SEND]

To:
99503
Message:
1665

Figure 2-2: Screenshot of Poll Everywhere

2.2.3 Web Applications

Web applications provide a good alternative for the technologies described above, since this can handle a great amount of audience members at once. There have been several implementations like Votapedia [2] that is well explained in the following study [17]. Votapedia is a web application that allow presenters to pose a question. The audience members can access the website and join the correct virtual room with an ID. This virtual room is a representation of all the participating audience members. It also allows you to use SMS or call to a number to vote. Votapedia is open-source and mainly restricts itself to traditional voting machines features. Viewing it in the bigger picture, this is a clicker system but the clickers are now replaced by another device like laptop, mobile phones, or tablets. We believe that this shift opens a lot of perspectives since it is shown that such devices are penetrating in our daily life [18]. Other systems are often based on the Votapedia idea [2, 19, 30]. The main problem is that in these systems, there is no integration with a presentation tool. However, some offer an XML-based output which can be processed by a plug-in in Microsoft PowerPoint. If one uses a search engine again, we find a wide range of web applications like todaysmeet[1], nextup[2], Infuse Learning[3], and many more. Often those websites created by a single person without any integration with a presentation tool.

Yawnbuster[4] is another commercial implementation. This is a plug-in for Microsoft Powerpoint that will put your presentation on-line and make it accessible. Audience members can then connect to your presentation given the correct password and start with following and give feedback like a traditional ARS. We tested this product and found it quite convenient and it does what you expect it should do. Disadvantages that we experienced were that the installation required an old version of PowerPoint and there was a delay in the feedback system.

Web applications have the benefits of both aforementioned technologies, such as the simplicity and ease of use from the SMS systems or the speed and anonymity from the clicker systems. Another important advantage is that with a web application, we are able to merge other ARS systems together. Everything, whether it is clicker or SMS system, is then processed on a central server. The web applications do not limit themselves by the 4 walls of a room as in a clicker system. It will be accessible in various ways. External audience members (think of a live stream of a presentation like TED conferences) will be able to join the ARS. The real requirement is that both presenter and audience should be and able to connect to a wireless network with a device. These devices can be mobile phones like in [24, 28]. In both of these studies, the technical difficulties are still reported to be high. Many students were unaware on how to connect their mobile phone to a wireless network.

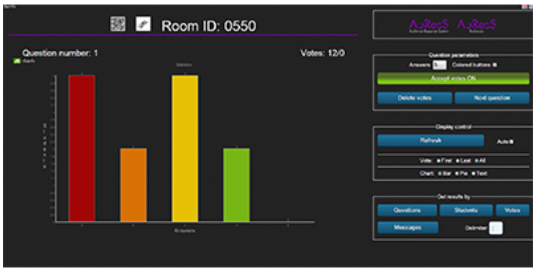


Figure 2-3: Screenshot of Auress (presenter POV) [19]

[1] www.todaysmeet.com
[2] www.nextup.com
[3] http://www.infuselearning.com
[4] www.yawnbuster.com

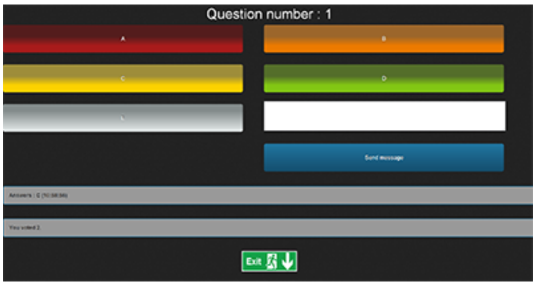


Figure 2-4: Screenshot of Auress (audience POV) [19]

To conclude, web applications are an interesting ARS approach that solves many of the previous disadvantages while keeping the advantages. The cost is still relatively high if one wants the audience to use their own mobile device to connect to the ARS. However, WiFi coverage is growing more and more and according to a study about smartphones [18], the ownership of smartphones is already high. It is also predicted that this will only grow in the future. This fact is a promising advantage for the future.

2.2.4 Advanced Systems

In the previous systems we kept ourself to the traditional ARS. In the web applications, we saw that it gave us more possibilities to create a more advanced system by merging it into a hybrid application. We discuss these expansion of traditional ARS to advanced systems in this section. Some of the possibilities which are desirable are:

- Let the audience share information
- Use the modalities available in a room (e.g. handheld devices, projector, or interactive whiteboard)
- Easy analysis of the results of feedback (e.g. voting, quiz)
- The use of roles
- Control of all the audience's devices

These advanced ARS are often called classroom management systems in commercial implementation for education. The last few years, many big companies like Texas Instruments, Microsoft, Apple, Acer, Intel, and many more have come up with their solution for a classroom management system. So it is safe to say that this market is growing and still in its early stages. The ARS in these systems is always integrated in a proprietary software package, most of the time specifically designed for education.

2.2.4.1 TI-Navigator

Texas Instruments (TI) came as one of the first on the market with its TI-Navigator product and is relevant for us in the architecture of the system. A visual representation of this product can be seen in Figure 2-5. It uses small WiFi devices to which students have to connect their calculator to with a wire or via an application on a laptop. All these small WiFi devices connect to the main device that runs the software and where a beamer is connected for providing visual output. The architecture TI uses is a client-server architecture. Since the devices only support calculators or an application on a laptop that mimics a calculator, this systems is only available for mathematics and science classes. A teacher can then give assignments to the students. The solutions of the students are shared with the teacher, who is able to share it with other students or display it with a beamer or screen.

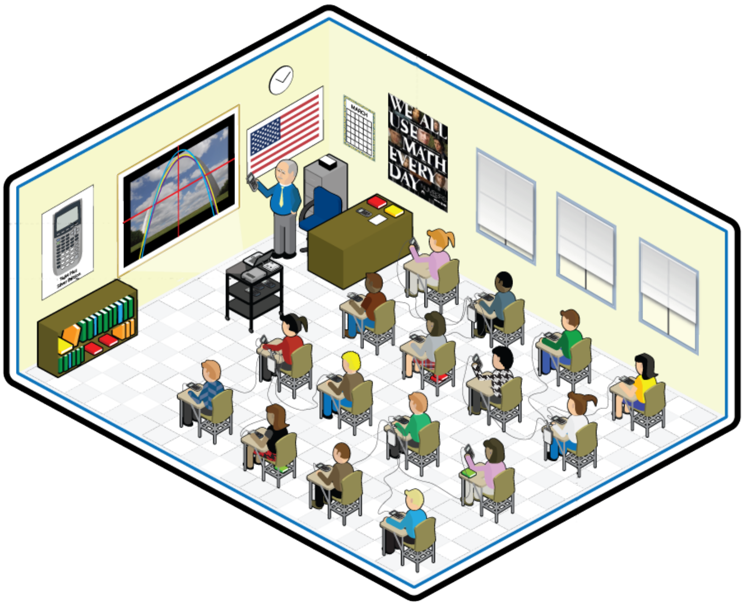


Figure 2-5: TI navigator. Source: TI-Navigator[1]

The TI research division[2], did research on their own product. This technology gives some positive feedback from both teachers and students such as:

- The effective use of TI-Navigator in Algebra teaching improves achievement an average of 14% more. [12]
- TI-Navigator is shown to increase student achievement in academic maths classes [39].

[1] http://matthewrea.com/04.05.08/TI-Navigator:-a-visual-explanation

[2] http://education.ti.com/research

The main drawbacks are that you can only use TI calculators or an application on a laptop that mimics a TI calculator. Implementation of these systems in classrooms have proven to be buggy and in need of a lot of assistance from the TI support group, especially in the early versions of TI-navigator. These experiences were found on various fora and Google groups. We also gained these insights by communicating with companies that sell these products.

2.2.4.2 NetSupport

NetSupport[1] is a company that has 22 years of experience in solutions to aid the management of desktop computers and their users and won a lot of awards for most innovative product awarded by organisations like CeBit, PC Pro, or Bett. They currently support over 10 million desktops, servers and mobile devices. The *NetSupport School* product, as seen in Figure 2-6, is what interest us the most. It is a software package that goes beyond the scope of presentations. However, it includes interesting features where we can gain some insights from.

Figure 2-6: NetSupport School. Source: NetSupport School[2]

While testing out this software, we quickly discovered that the focus lies at the K12[3] education. Many of the features that they pre-define are there to support this K12. I.e. Quizzes with country names or memory games. It provides teachers with the ability to instruct and visually monitor students' devices. Teacher can interact with the students individually or as a pre-defined group or to the whole class.
The system also works with a client/server architecture. While the software covers all major OSs, students still need to install a dedicated application for their platform to connect to the server. We consider this as a drawback

[1] http://www.netsupportsoftware.com

[2] http://www.netsupportschool.com

[3] The expression is a shortening of kindergarten (K) for 4- to 6-year-olds through twelfth grade (12) for 18- to 19-year-olds.

because they have to be up-to-date on every platform which increases the cost. The iOS and Linux environment are lacking some features compared to the Windows environment.
The features that interested us the most are: poll, quiz, students able to work in group on a task and later share this with the class, individual tasks for the students, and complete control of the students devices (i.e. teacher has the option of opening the browser with an URL or even shutting down the devices). This software is often integrated in other packages like Acer Classroom Manager[1]. With this, we have a vendor lock-in from Acer and Windows.

2.2.4.3 Other

We discussed two of the pioneers in classroom management systems. Nowadays, one can choose from a wide range of products. Below one can find a short list of the products of companies. Basically, most of the systems are the similar and only differ in small functionalities. The main idea is that we have a device (mostly tablets or smartphones) which connects to a server (that includes a WiFi hotspot) that runs the software. The client/server architecture is popular in these technologies. The following information has been found by doing some research on the Internet and by going through the brochures of the products. We see that all the products are still in early stages and have not yet introduced themselves in the common classroom.

- Prowise Proconnect[2]: brings interaction with a client-server architecture. This system allows us to 'bring our own device' meaning that your are independent from any vendor. The software they deliver is mainly to aid teachers for teaching children and no integration for presentation tools.

- Bic education[3]: ARS is completely designed for K6 education. It contains quizzes, educational exercises, following the tablets of children, and many more features. They mainly restrict themselves to the interaction between teacher and student. Not student with each other.

2.2.5 Out of the box

The following ARS are made in the mindset to create something different than previously discussed systems. We discuss an ARS where the attempt is to reduce the cost to the minimum and ARS where Augmented Reality (AR) is added.

[1] http://www.acer.fi/foreducation
[2] http://www.prowise.com/nl/proconnect
[3] http://www.bic-education.com/

Existing ARS

In a recent implementation of Microsoft (MS) [10], the idea was to come up with a cheap solution for interactivity in a classroom. In this study, it is shown that having the latest technology is not always needed in order to have a workable ARS. MS introduced a system where you only require a computer, a web cam and paper. The paper is used to print QR codes. These QR codes represent answering options in analogy with the clickers. When a teacher poses a question, student have to raise the paper corresponding with their answer. The QR codes are registered with an ordinary web cam. Since every QR code is unique, the technology identifies every student's answer. The learning curve of this system is reported to be quite gentle. Furthermore, this solution is about 15 times cheaper than a regular clicker system.

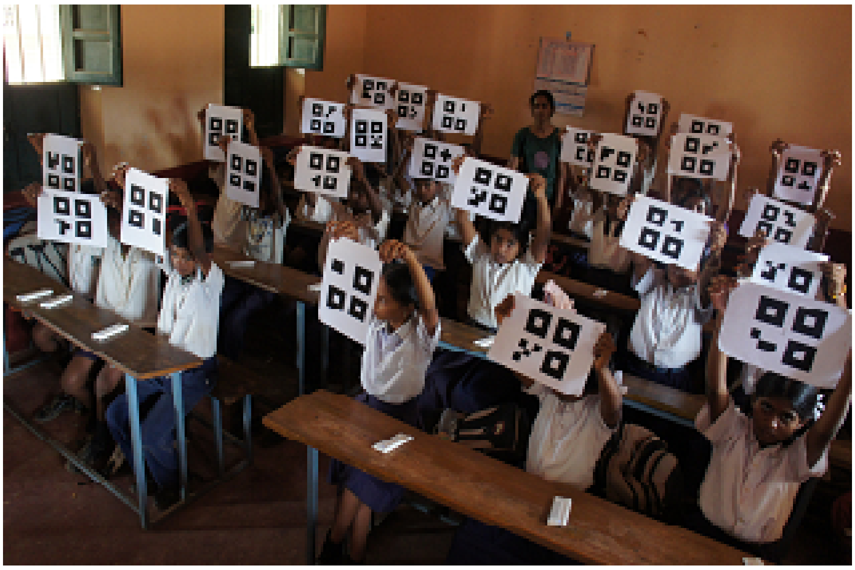


Figure 2-7: Microsoft QR solution. Source: [10]

An ARS was also developed with AR. AR adds some elements to the real world as we see it, in real-time. For instance a scoreboard on television during a broadcast of a live football match on the television. In [40], an AR based ARS is introduced. The presenter has to wear AR goggles to display students' feedback and presentation notes. A Microsoft Kinect is used to register the presenter's gestures to activate the ARS. Students vote with a mobile device if the presenter asks something. The goggles then augments the responses (green, orange, red colour) over the students. After testing, the teacher's overall opinion of this system was positive while for the students this is more unclear. The system is still in it's early stages and both teacher and students reported that more technical fine tuning is necessary. The overall results of the experiment are promising.

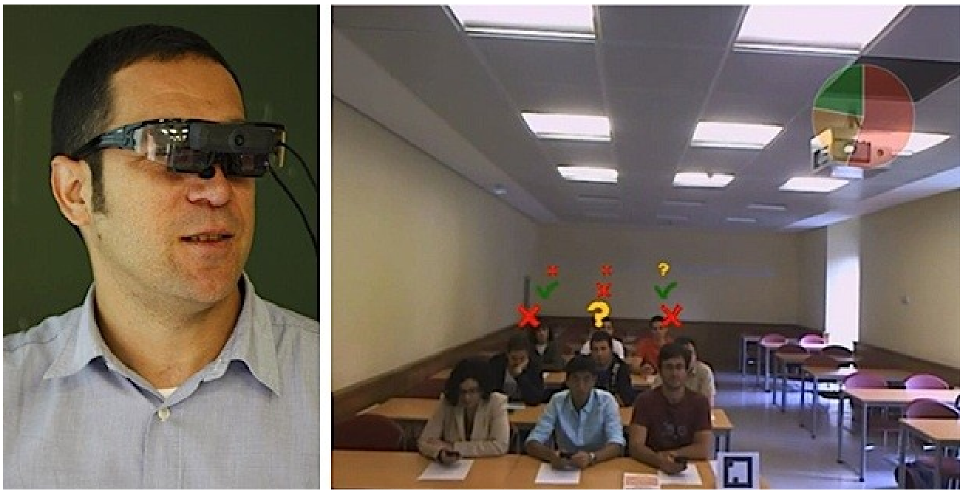


Figure 2-8: Augmented Reality. Source: Engadget[1]

2.3 ARS Benefits

The ARS tools have been extensively tested in the educational environment. This also implies that the benefits are mainly focussed on the pedagogic effects of using such a system for teaching. We cite two extensive literature studies to identify these benefits of ARS [9,11,20,22]. The ARS tools tested in the studies are mainly clickers. But as we have seen in previous sections, the core idea is always the same. Independent from which technology you use. What follows is a summary of the benefits reported in the two literature studies. For a complete list, we kindly refer to those studies. [22] contains a study of 67 peer-reviewed papers from 2000 until 2007, [11] uses 16 peer-reviewed papers from 2003 until 2009, and lastly [9] and [20] are studies on their own that also contain a good literature study where we based ourselves on for some parts.

The overall attitude of both presenter and audience members towards ARS is reported to be positive which is supported by quantitative and qualitative evidence. Both of the literature studies use the same three divisions of grouping the benefits. What comes next is a short description of the benefits who are a bit more specific. Interactivity and participation, satisfaction and learning benefits, and assessment benefits.

[1] www.engadget.com/2013/06/17/spanish-augmented-reality-smart-glasses-education

ARS Benefits

Interactivity and participation Learning engagement that ARS brings is the element that is most common in all the studies used in both literature studies. The *attendance* of students for a course that uses ARS increased up to 15%. Even for the courses that did not link grades to the use of ARS became more popular. The attention span of students increased during the presentations. A tactic that has proven to be efficient is to ask ARS questions every 20 minutes to restrengthen the focus. *Anonymity* is highly recommended in order to get feedback from your audience. While it is not yet fully understood why this is, it is often reported that students can answer now without being judged by others or having fear of giving a wrong answer. In addition of the previous benefits, students were more *engaged* to class and started to participate more with other audience members in order to solve given problems.

Satisfaction and learning benefits The *peer interaction* between students to discuss ideas is strengthened with the use of ARS. Students are able to probe more questions, increase of the active learning, and focus on student needs. A majority of students reported that they liked the immediacy of feedback.

In some cases it is reported that the ARS increased the quantity and quality of *discussions.* Both the learning performance and quality of learning is increased as a result of using ARS. It appears, according to the studies, that ARS emphasize the depth of student understanding. However, it does not help with handling the amount of study material. In the education environment, ARS helps the teachers and students. Teachers are able to modify instructions or re-explain concepts based on feedback from the students.

Assessment benefits This includes that there is a regular *feedback* given to the presenter and audience. ARS helps to increase a better feedback process because without using ARS, students tend to copy others in their behaviour or are too shy to raise their hands. ARS also makes the student think before answering a question. There is also evidence that students like to compare their understanding with collages.

2.4 Current Issues

In the literature studies mentioned above, we already identified some issues with ARS. In this section, we classify the issues into four categories which are technology, presenter, audience, and research based.

Technology-based challenges In some cases, the students had the responsibility of keeping the dedicated remote devices which led to the fact that students forgot or lost these devices. Some more critical problems were that the ARS did not work, the setup was too difficult or some remote devices failed to work. Technical issues can be overcome by extensive evaluation before releasing the product on the market. We also learn that it is not good that the students keep the dedicated device for ARS. The cost of clicker systems are still reported to be high. Since most studies are already a few years old, we believe the cost has dropped a lot because of the abundance of commercial solutions. Another way to solve this problem is by implementing the ARS with the idea of "bring your own device" like Prowise Proconnect and others do. Another issue that should take into account is the fact that the devices are battery based and the battery needs to be to be charged before starting a presentation. In [3], they propose a strategy of encouraging audience members to bring a charger with them.

Presenter-oriented challenges For the presenter, the ARS should be easy to use. It is reported that developing the questions often was too cumbersome. Using feedback systems also means that the way of presenting changes a bit. Some presenters had some difficulties to find a good way to include feedback in their presentation.
There is also time you spend in order to include the audience, this time has to be subtracted by the amount of time you spend on doing the presentation. Meaning, you have less time to cover the same presentation. We do not see this issue to be solved quickly but a presenter should know this when making their presentation.

Audience-oriented challenges A first challenge is that the audience is somewhat sceptical about ARS. It is something new they have to work with which can lead to stress, frustration, and resistance in the beginning. Testing and making your systems easy to use can help with this challenge. Still, an extra effort by the presenter is required to explain everything to the audience. Audience also do not like to feeling of being permanently monitored (having the "big brother" effect). The key problems that leads to this feeling is holding a summative assessment, eliminating anonymity, and a bad reaction from others if you give negative feedback.

Conclusion

Methodology for investigating ARS Literature study [22] reports that the current research has some key problems: a lack of systematic research (lacking of reliability and validity analysis), a bias towards using anecdotal and qualitative data, excessive focus on attitudes as opposed to learning and cognitive processes, and samples derived from limited educational settings. More detailed research is an opportunity for the future and will lead to better understanding of some of the previous mentioned benefits. The research should also include the more advanced systems. Currently, a lot of the research is based on Clickers.

2.5 Conclusion

In this chapter we have introduced different ARS tools. We see that there are many solutions available going from a traditional voting machine to a more advanced classroom management system where the ARS is integrated with a complete educational software program. For all the different technologies, we have discussed some commercial solutions and looked at research done with these. Many more solution exists but due to the fact that many of them do not differ we omitted them and focussed on a subset.

The literature studies are used to substantiate the benefits and issues of current ARS. We do see that future research is required to identify and explain more of these benefits and issues but the overall attitude of using ARS is positive. Allow us to summarise the main benefits according to the three categories again:

- Interactivity and participation: includes anonymity, increased attendance, attention, participation and engagement levels
- Satisfaction and learning benefits: reflects in interaction, discussion, contingent teaching, quality of learning, learning performance
- Assessment benefits: creates feedback, formative assessment and students being able to compare responses with peers

3
Perceptions, Opinions and Aspirations about ARS

The study of literature about ARS, as described in the previous chapter, gave us an extensive look at the currently available solutions with their potential benefits and weaknesses. However, we feel it necessary to conduct our own research on how the present day practice of giving and attending a presentation is experienced. We want to know how people act and react as a presenter and as a member of an audience, and what their needs and aspirations are. We believe it is an added value not to restrict ourselves to our own assumptions or those found in literature but also to include the feedback of other presentation professionals.

3.1 Survey

We invited people that we assumed to give or attend presentations regularly, to answer an on-line questionnaire. We used the tool LimeSurvey[1] hosted by the VUB. We tried to distribute this survey as widely as possible. We also mailed this to our social contacts.

People that accept the invitation to answer the questionnaire were given the choice to answer in the capacity of someone giving a presentation (the presenter) or to answer in the capacity of someone sitting in on a presentation (the audience). For each capacity we created an appropriate set of questions, addressing the same topics and issues, but still differentiated from a presenter's or an audience's point of view. We like to point out

[1] https://www.limesurvey.org

that it is a preliminary study done in order to test our hypotheses and propositions of features. We believe this is an added value because we do not restrict ourself to our own assumptions but also include the perceptions and opinions of others in order to determine requirements and features for our own implementation.

3.1.1 Method

This study employed an embedded case study with two sub-units. It provides us with the means of integrating qualitative methods into a single research study. The case of our study is “presentations”. We created the two sub-units based on the point-of-view of a user towards a presentation. These are presenter and audience. The research method is by means of an on-line questionnaire.

3.1.1.1 Participants

The participants were divided into two groups, an audience group and presenter group. For each of the groups we created a different questionnaire with questions that addressed the same topic. A total of 157 responses to our survey were recorded and Table 3-1 provides the characteristics of the participants.

	Number of part.	Male	Female	Avg. age
Presenter	91	34 %	66 %	39
Audience	66	47 %	53 %	33

Table 3-1: Participants characteristics

The familiarity of the respondents with giving/attending presentations was quite high as shown in Table 3-2.

Presenter giving a presentation		Audience attending a presentation	
Daily	6,59%	Daily	5,88%
Weekly	9,89%	Weekly	5,88%
Bi-weekly	39,56%	Bi-weekly	32,35%
Monthly	21,98%	Monthly	36,76%
Less than monthly	17,58%	Less than monthly	17,65%
Empty	4,40%	Empty	1,47%

Table 3-2: Frequency of presentations

The context in which the respondents give/attend presentation is presented in Table 3-3. Note the participants could select multiple contexts.

Survey

Presenter giving a presentation		Audience attending a presentation	
As a teacher	27,5 %		
As a student	24,2 %	As a student	41,2 %
In a professional environment	57,1 %	In a professional environment	58,8 %
In leisure time	5,5 %	In leisure time	7,4 %
Other context	5,5 %	Other context	2,9 %

Table 3-3: Context of presentation

3.1.2 The Present Day Practice of Presentations

The intent of this section is to identify and test how presentations are used in today's practice. More specifically, we are interested in the perceptions of people about hand-outs of the presentation and about the accessibility of the presenter by the audience during a presentation. To calculated the P-value, we used the Mann-Withney U test [29] because of the two independent groups and our data is non-parametric. This test is used for the upcoming statistical testing for the two groups (presenter and audience).

3.1.2.1 Handouts

Our survey shows that the handouts of a presentation are important to have for the audience. This importance is reflected when we asked how audience members take annotations, 54 respondents (81 %) told us that they take annotations on printed handouts. These handouts are moderately used by our audience respondents after the presentation. We stated that handouts are heavily used in an educational setting. Both presenter and audience groups have the perception that they strongly agree with our statement. The perception of both groups are not significantly different from each other (p=0,1473).

We indicated the importance of handouts, but what about the availability of the slides? We wonder if the handouts are always available for the audience members in advance of the presentation. It appears that the slides are only moderately available to the audience. This is indicated by both groups who are not significant different from each other (p=0,4226). The high degree of importance and only a moderately degree of availability of handouts indicate a problem. We hypothesise that this lack of availability is due to distribution. It can be challenging to distribute the up-to-date handouts to all the audience members in time.

3.1.2.2 Communication

During a presentation, there is a communication flow going from the presenter to the audience. When an audience member has a remark or question, this flow goes in the reverse direction. In our study, we identified that there is also a communication flow going between the audience members. The communication flow between the presenter and audience is the flow we are mostly interested in. We questioned how accessible the presenter is during the presentation. The perception of our respondents is that the presenter is rather highly accessible for remarks or questions. With a median of 4 (pretty useful) for both presenters and audience point of views. However, we do see a difference between the perception between the two groups. The two groups are significant different (p=0,000004) from each other compared to our starting hypothesis: "The audience members have the opportunity to communicate with the presenter without ARS".

Survey

In the studies [11, 22], it is mentioned that the communication between the presenter and audience is facilitated by the use of ARS (that the presenter receives the answer quicker) and in study [26], it is shown that ARS foster the interactivity between the presenter and audience. However, in the survey we asked what the current perception is of people about responding to the presenter when a question has been asked (without ARS). The medians of the presenter is at a rather high degree while the audience give a moderately degree. The difference of the perception between the groups is also significantly different (p=0,0002). We can safely say that there is a mismatch between the perceptions of the presenter' and audience'.

3.1.2.3 Conclusion

Handouts are heavily used in an educational setting and many audience respondents (81 %) use them to write annotations. We conclude that the handouts of the presentation are of high importance. This makes it crucial that the audience has the handouts available before the presentation starts. However, we concluded that the handouts are only moderately available so a mismatch is found. Now that we know that there is such a mismatch, we can try to solve this during the implementation of our ARS.

To conclude the part about the communication, in the literature is said that the interactivity is increased (and so the communication) between the presenter and audience. We do however see that there is a significant difference between the perceptions of the presenter and audience point of views about this communication without using ARS. Further research is needed to investigate this significant difference.

3.1.3 Aspirations of a Future ARS

In order to determine whether or not our proposed features would be useful in an ARS tool, we wrote down scenarios to explain each feature in our questionnaire. We assign some features to the audience point of view while other are at the presenter point of view. Some features are explained to both the audience and presenter point of view. What follows is a description of our features including the key findings. We divided the feedback in 3 categories. Feedback can be positive, negative, or it can be a concern that we include while implementing. We asked in our questionnaire to give a rating from 1 to 5 (1 = useless, 5 = very useful) followed by an explanation why. We value some of the opinions of a feature by people higher than others. The reason for this is that some respondents are from a specific domain and can give some more useful insights of a feature for that domain. E.g. a teacher that holds a presentation every week can give us more insights than a person who only does one presentation a month in his leisure time. Allow us to summarise the features:

In the audience point of view features:

1. Piggyback: the idea to follow the presentation in real-time on your own device
2. Add your own annotations and share them with others

In the presenter point of view features:

1. Polling: ask a multiple choice question to the audience
2. Add annotations ad-hoc to the presentation if you which to clarify something to the audience

Both the audience and presenter point of view features:

1. Question list: audience can add questions to a list, the presenter can look at those questions when it suits him
2. Taking over the presentation: allow an audience member to take over the presentation path to go to a slide where he has a question

3.1.3.1 Piggyback Feature

The idea of this features is that every audience member has its own device. This device can be a smart-phone, laptop, or more. As long it is able to connect to a WiFi hotspot and render HTML5 in a web browser, it is possible. When the presenter starts his presentation, all the audience members connect to the hotspot and are forwarded to the correct IP address where the presentation is located. When the event occurs that the presenter

goes from slide X to Y, it will be shared with all the connected audience members so that it also changes on their device. The audience members can choose whether or not to follow these events. When not following, it is possible to navigate backwards or forward in the presentation.
We wish to add this because we believe that this helps the audience attention when giving a presentation in a large room. This way, the audience's members in the back are able to completely follow the presentation. It also allows the possibility for audience members who are not in the room to follow this presentation in real-time. An other reason is that it is useful because audience members will have a copy of the presentation for themselves.
Let us look at the feedback given in our questionnaire. When we look at the rating given in Figure 3-1 to give us a first indication of the overall appreciation of this feature, we see that it is a bit skewed but more people think this is a good feature.

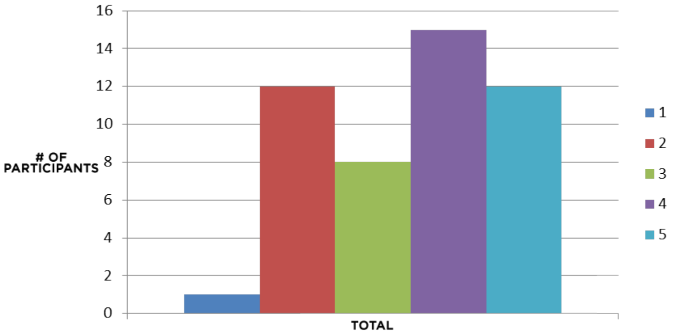


Figure 3-1: Indicative histogram - Piggyback

In the following table, we categorised the key findings in the three categories. We order these key findings based on how important the feedback is of the person. We determine this based on how many times they are in a presentation and why they do this. A person who participates in a presentation every week in a professional environment has a bigger weight than a person who participates once a month.

Positive	Negative	Concerns
• Useful to have the slides at the time when the presentation is given	• Less attention towards the presenter	• Feature requires training for audience
• In educational setting it gives an added value	• Overall, people think the focus of audience members will be lost	• Could interfere with didactical methods where the presenter asks questions as a thinking exercise
• Navigation is useful for the audience	• With their own device, chance on distraction is high	• Going forward in the presentation may not be that good
• Distribution of the presentation becomes easier	• Gives the impression to the presenter that nobody listens to him	• Only possible if you have a device
• It saves work and paper if you can follow the presentation on your own device		

Table 3-4: Feedback piggyback

Conclusion piggyback We conclude that this Piggyback feature seems to be useful. The negative feedback (especially the distraction of the presentation) is something to keep in mind when we evaluate this feature in real-time. Other key concerns that are listed above are aspects that we should consider while implementing. This feature is worthwhile taking in consideration when we start with the implementation.

3.1.3.2 Annotation Feature

Making annotations during a presentation is widely done. Every respondent of audience point of view told us that he at least takes notes on a moderate bases. The majority of respondents selected that they often take annotations on printed slides when available. Only 6 people out of 68 said that they never make annotations on printed slides. Annotations made in a notebook are also relatively frequent. The least popular ways of annotating the slides is on a digital device. This is due the lack of such a device or the lack of the digital version of the slides on their device. We conclude that the respondents make annotations on their slides.

The intention of this feature is that audience members will be able to make digital annotation on the presentation in real-time. We also believe that sharing the annotations with each other would be a nice feature to add. Sharing information has the implication of privacy issues. To consider the

privacy of the user, the annotations made by this user only belongs to them alone. Unless this user likes to share it, then the user has to option on selecting an other audience member. When this happens, the annotations are now also belonging to the selected user. This way of sharing requires the explicit consent of a user to share information.
Let us look at the first indication on how useful this feature is based on the rating of the respondents in figure 3-2.

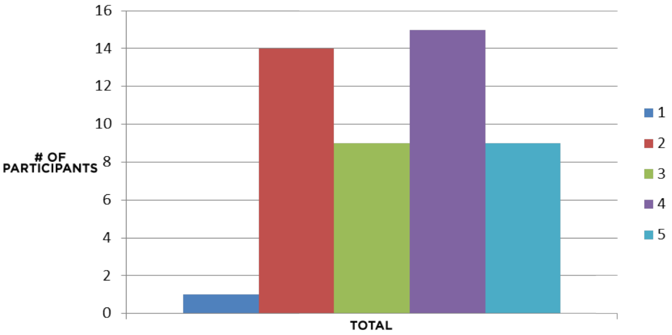


Figure 3-2: Indicative histogram - Annotations

Like the previous feature, we see that there are two peaks. One peak around the 2 value (not so useful) and the 4 (useful). When we look at the feedback the respondents give us, we do see that some of them do not fully understand our intention of this feature. In table 3-5, we see the key findings of the annotation feature.

Positive	Negative	Concerns
• Overall detailed feedback is useful	• Attention to presenter is lost while making annotations	• It becomes too chaotic if you have too many annotations
• Handy and saves time	• Some do not see the advantage. However, those people hardly take annotations.	• Users can annotate wrongly
• One person believes that it would foster the interactivity between audience members		
• Learn from someone others annotations		
• Ability to select the users you wish to share annotations		

Table 3-5: Feedback annotations

Conclusion annotations After the investigation of the feedback of this feature, we believe it is desirable for audience members. The overall feedback tended to be positive. This feature is worthwhile taking in consideration when we start with the implementation.

3.1.3.3 Poll Feature

This is a feature that is proposed in the presenter point of view of the questionnaire. The idea of this feature is to give the ability to the presenter to ask a multiple choice question to the audience. The audience, who is following the presentation on their own device, sees the possible options of the question. After an amount of time, this possibility to answer stops and the results are displayed in a graph. These results are shown on all devices that are following the presentation and may be saved for later use. This feature that is already standard in most commercial solutions, so we can add key conclusions of other studies in ours. Basically, this feature is the earlier mentioned description of a clicking device. We distinguish ourselves from clickers by not just limiting ourselves to simple buttons. E.g. It would be possible to enrich the questions with images. The other distinction is that the audience members will see the results on their own device.
In figure 3-3, we give the general appreciation of this feature among our respondents. We base ourselves on the rating the respondents gave us to determine the indication. It indicates that the feature is considered as useful. We clearly see that most people tend to give a high rating.

Survey

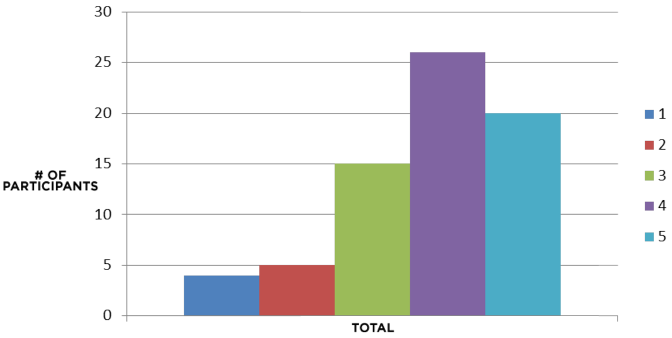


Figure 3-3: Indicative histogram - Polling

Let us take a detailed look at the key findings that the respondents shared with us.

Positive	Negative	Concerns
• In educational setting, it is useful to see if the presentation is clear for the audience	• Prefabricated answers limits the audience some believe	• Some people worry about the complexity
• Increases the attention of audience members	• Losing valuable time to present	• Difficult to create a discussion between presenter and audience
• Few people mentioned that the interactivity increases between presenter and audience	• Is too complicated for small groups	
• Anonymity is highly appreciated		

Table 3-6: Feedback polling

Most of the key findings given by our respondents also stroke with some of the key findings of [11]. In order to complete this picture or key findings, we add the most important ones in the following table. Note that these findings are based on clicking systems but they are relevant since the main idea behind the feature is the same.

Positive	Negative	Concerns
• Small learning curve	• Technical difficulties	• Useful for large groups
• Improved understanding of presentation	• Potential overuse	• Time consuming
• Students came more often to class (note that in some classes it was mandatory to participate)	• Poorly written questions	
• Increased interactivity	• Usefulness lies heavily at the capability of working with the tool of the presenter	
• Overall pleasant way of interaction		

Table 3-7: Feedback polling study [11]

Conclusion polling We conclude that the polling feature we described above is highly desirable by the presenters. It could also be possible to allow arbitrary answers from the audience instead of limiting them to a set of answers defined by the presenter. However, we do fear that if this ability goes to the audience, the amount of answers will be too high when working in a large group.

3.1.3.4 Add Global Annotations Feature

This feature is a variation of the previously explained annotation feature. We make it possible for the presenter to add annotations to slides during the presentation. The annotations can be shown on a new slide or on the slide itself. In addition to seeing the annotation they are also shared with all the audience members who are watching on their own device. We believe that this feature could prove powerful in case that some audience member(s) do not fully understand the slide. It gives the presenter the ability to add extra information about the explained concept. Now-a-days, a blackboard is often used to do this explanation. After the presentation, this information goes away if the audience member did not copy this for later use. This issue dissolves too with our proposed feature since that they can save the annotation for later use.
Let us look at the following indication histogram. We see that the presenters hesitate about whether or not this is a useful feature. The results tend towards the useful side of the graph.

Survey

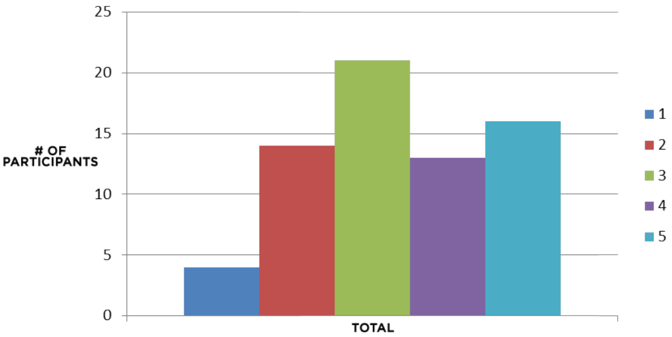


Figure 3-4: Indicative histogram - Global annotations

Positive	Negative	Concerns
• Extra annotations enriches the presentation	• Quite a few mentioned that they do not see the benefit of such a feature	• Risk of losing your message and spend to much time on details
• If slides need to be changed later, it can be used for the presenter as reminder	• Time consuming and maybe irrelevant for the majority of audience members	• risk of making it too complicated
• One person mentioned it would help him understanding something in case he did not		
• Will make concepts more clear		

Table 3-8: Feedback global annotations

Conclusion global annotations The opinions of this feature are diverse. We believe that both sides (positive and negative) have some valid points. We believe that this feature could really enhance the presentation for the audience but it will require some training for the presenter. It is a feature worth considering while implementing our ARS prototype. To test the problems mentioned above, further qualitative research is needed.

3.1.3.5 Question List Feature

This feature introduces a way for the audience members to ask questions about the presentation because sometimes it is hard to speak up in a big group (e.g. due to shyness or the fear to ask a silly question). Our proposal is that audience members can see a list of questions of their colleagues. It

is possible that a colleague has the same question and already submitted it, then this question can be up-voted (meaning that the score of the question is +1). If the question does not occur in the list, the user can submit their question. It is also possible for the audience members to down-vote a question (meaning the score of the question if -1). On the presenter side, it is possible to see the list when he desires. He is then able to delete a question out of the list once it is handled. To get an opinion of this feature, we asked this in both audience as presenter questionnaire.
On the following two histograms, we have our results again of the rating of our respondents. We clearly see that the indication of both point of views are towards useful. Especially in the presenter point of view, we see that many respondents give a rather high rating to this feature.

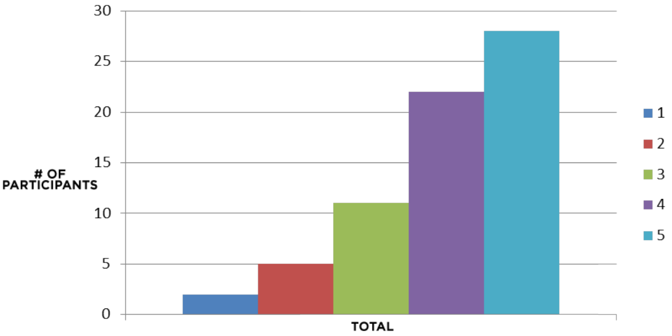


Figure 3-5: Indicative histogram - Question list (Presenter)

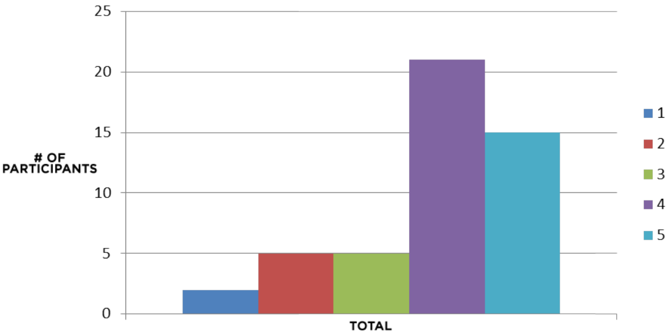


Figure 3-6: Indicative histogram - Question list (Audience)

To take a closer look at the feedback given by our respondents. In Table 3-9, we find the feedback given by our respondents in the presenter point of view. Table 3-10 is the feedback given by our audience respondents point of view.

Survey

Positive	Negative	Concerns
• Great way to handle questions in big groups	• Oral interaction is more suitable	• With many questions you need some kind of assistant
• Enhancing the interaction with audience without interrupting the presentation	• Distraction	• There may be no technical issues if you use this
• Audience members who are scared on asking their question can do it more anonymously	• Seems very time-consuming	• Questions might be hard to interpret if they are poorly worded
• Presenter can choose when to answer		

Table 3-9: Feedback question list - presenter

Positive	Negative	Concerns
• Presenter does not get interrupted	• Some find it a distraction (2)	• In big groups, the question list will be big
• Good way to include everyone of the audience (not only those who are not shy)	• Loss of attention from the audience members	• Handling the questions that ask the same but are differently constructed
• You remove the issue that you do not understand what the audience member is saying	• Multi-tasking from the presenter	
• You do not forget your question		

Table 3-10: Feedback question list - audience

Conclusion question list From the feedback given by our respondents on both point of view, we conclude that they are overall positive of this feature. It is reported many times that it would minimise the bridge between the presenter and audience. Also the fact that the presenter can see the questions when he desires is something that appeals. One respondent told us that he already used such a system and that it worked nicely, except that they experienced technical issues with the WiFi. One thing we have to keep in mind when implementing such a feature is handling questions that are equivalent. Someone mentioned it would be nice if a stack-overflow principle could be implemented as well. This means that when you type your questions, some suggestions come up of previous asked questions. Another respondent suggested to use Twitter. While the idea is nice, we believe it will bring some limitations such as filtering. When the issues occurs that a question is poorly written down, the presenter has still the option on asking orally what the person meant. If nobody answers, the question can be removed. Also if this question is noticed by the audience before, they may down-vote it to the bottom. We believe this down-voting system is useful but there might be a chance of misuse by the audience.

One of the respondents suggested to create a slide for every question at the end of the presentation instead of going through the list. This might be an interesting idea to investigate during the implementation.

3.1.3.6 Taking over Presentation Navigation Feature

We consider the following feature as a rather controversial, yet interesting feature. The idea is that all the audience members are following the presentation in real-time on their own device. If the presenter goes to the next part of the presentation, this gets updated on the audience's devices. It happens that an audience member has a question on some specific previous shown information. This person asks the presenter permission to navigate to this information on his own device rather than that the presenter has to find this information. The presenter can accept or deny the request of the audience member. If the request is accepted, the navigation of the audience member is broadcasted to presenter and other audience members. The permission can also be revoked by the presenter. If the user found the wanted information, it would be possible to add some annotations to explain why he is confused.

In Figures 3-7 and 3-8, we see a slight indication of appreciation from our respondents. We notice that there is a difference between the audience and presenter. While it seems that the audience can appreciate the proposed feature, the presenter results are more towards the useless side. Tables 3-11 and 3-12 give us more understanding on why there is a difference.

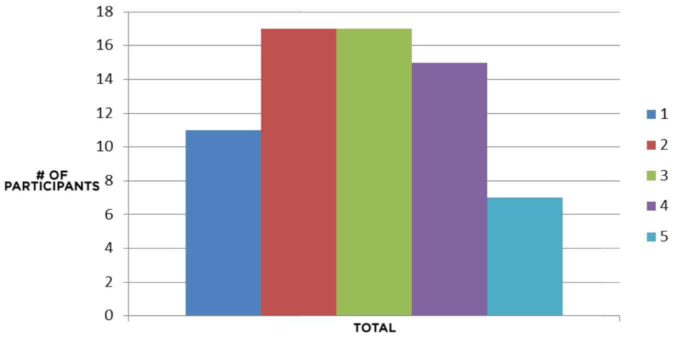


Figure 3-7: Indicative histogram - Taking over navigation (Presenter)

Survey

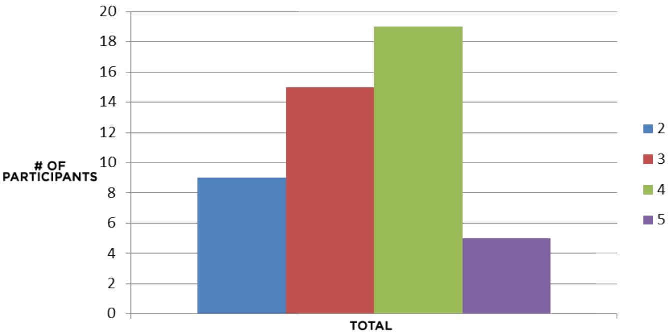


Figure 3-8: Indicative histogram - Taking over navigation (Audience)

Positive	Negative	Concerns
• Interesting evolution of holding a presentation	• You lose the control	• Chance on making it very chaotic
• Would increase the communication between presenter and audience	• Process seems very time consuming	• It sounds useful but fear that in practice it will not be used that much
• Saves time if the audience member knows where the information is	• Disrupt the flow of the presentation	• How handling conflicts of many audience members?
	• The average audience is not ready for such a feature	• Only useful in certain domains like in small groups. This feature should be disabled for big groups
	• The audience may not alter the presentation with annotations	

Table 3-11: Feedback taking over navigation - presenter

Conclusion taking over presentation As mentioned above, it is a feature that may sound scary for the presenters because it is very innovative. It does not exists in current presentation tools or other ARS. This is also what is reflected in the feedback given by our respondents. In the Tables 3-11 and 3-12, we see that the feedback consists of mostly negative comments or concerns.

The idea that the presenter loses the control of the presentation is the biggest concern. The respondents do not believe that the audience should have the

Positive	Negative	Concerns
• Could save time (presenter does not have to seek the information)	• Misuse of the system	• Restrict itself to a professional environment
• Hold positive aspects if used correctly	• Not so efficient for the presenter	• As long the presenter holds the control
• Help you with explaining your problem	• Complicates the presentation	• Fear of the "I-am-the-smartest" discussions

Table 3-12: Feedback taking over navigation - audience

ability to control the navigation. However, if the presenter is still capable of revoking this ability that was granted (meaning that the control remains at the presenter). This feature could then prove its usefulness in some circumstances. These circumstances do not involve big groups because it is feared that in big groups, misuse of the feature will happen. This misuse will cause a waste of time for the presenter.

Adding extra annotations by the audience member to the presentation raises some more concerns. This ability should not be possible according to most of our respondents because they fear that the presentation will get chaotic. We conclude that the respondents are fearful of this feature with some correct concerns. However, we believe (like some of our respondents) that this would benefit the communication between presenter and audience and that it could prove its usefulness if used correctly. Concerns are rightfully there to warn us for misuse, especially in big groups.

3.1.3.7 Other Features

Most of our respondents never heard of ARS, and so most features sound very innovative. Since they had to think about these innovative features, it could happen that they would come up with new ideas themselves. We asked this very open question at the very end, whether or not they have such ideas.

Survey

As expected, a handful of respondents gave some new ideas that could be useful to implement in our ARS tool. Let us list these ideas:

- Give the audience the possibility to indicate that they can not hear the presenter. Optionally, the location on where they are in the room
- A speed-up or speed-down graph to indicate that the presenter goes too slow or slow respectively.
- A kind of forum that the audience can use to hold a discussion.
- Something to support brainstorm sessions. Dividing the audience in groups, the them work together on their devices and share these with everyone later on the projector or something equivalent.
- A feature to hold a quiz
- Follow your presentation from a different place
- The ability to get the audio files from the presentation synced with the presentation

3.1.4 Conclusions of the Features

In this section, we investigated some of our proposed features. The intent was to get preliminary feedback so we can get extra insights of how people think of it. The features that we proposed were:

- Piggyback
- Add annotations and share these (audience side)
- Polling
- Add annotation and broadcast these (presenter side)
- Question list
- Taking over the presentation navigation

Most of the features that we proposed were appreciated by the respondents. While concerns were mentioned, there was always feedback that the feature could be used in educational or professional settings.

3.2 Conclusion

In this chapter, we wrote down our findings of our conducted research. We started out with finding out how our participants use presentations today. We concluded that handouts are very often used by the audience because they like to use it for writing down annotations. This makes it important. They are also heavily used in an educational setting. However, the availability of these handouts seems to be a problems sometimes. Secondly, we looked for aspirations of some of our audience-oriented features that we thought are useful. For example our feature "Piggyback" which can overcome the problem of availability of slides. The intent of this feature is to give users a mirror during the presentation of the presentation on their own device. An other proposed feature to take and share annotation is to support those who are already taking those during the presentation.
We like to point out that this survey is a preliminary study. Further research is needed to evaluate our features. All the features should be tested out with a diverse audience group to get more insights.

4
MindXpres

In previous chapters, we focussed on the development and implementations of ARS followed by how the ideal ARS should look like. In this chapter, we take a step back and introduce ourself to some presentation tools. More specific, we talk about the MindXpres presentation tool.

MindXpres [35] is a new presentation tool clears the way of working with presentations for a more flexible and semantically enhanced future of presentation tools. It introduces a radically new presentation format that solves the lack of some features introduced by the common tools like Microsoft PowerPoint, Apple Keynote, or OpenOffice Impress. Some features that lack in current tools are for example the enforcement of a linear traversal of the slides without the possibility to easily navigate between slides. It is also not possible to display multiple slides at once because the complete space is always used for one single slide, this can be solved by introducing a zooming feature. Last and most important for us is that the presentations are presenter-oriented. Most of the features offered by the common slide-ware are there to increase the ease of use for the presenter. These features, like all the creating the content and the whole aesthetic part of it, publishing and giving your presentation are all designed for the presenter. MindXpres gives a solution for these problems without any loss of ease of use offered by the common slide-ware and gives us the opportunity to make presentations more audience-oriented. This opportunity is made possible because MindXpres is developed with eye on extensibility.

MindXpres separates the content and visualisation similar to a LaTeX document. A graphical editor handles the creation of a domain-specific language that focuses on the content, the visualisation is then done by the tool (compiler) based on a chosen template and outputted in HTML5 which makes everything very portable and distributable over heterogeneous devices. The MindXpres tool also offers components to facilitate specific types of information and has a plug-in mechanism that can add new future media types. Some more advanced features of MindXpres are non-linear traversal of the presentation, hyperlinks, transclusion, semantic linking and navigation of information, multimodal input, dynamic interaction with the content, the import of external presentations.

In the following sections, we take a closer look at how MindXpres is built. We discuss the hypermedia model that is used as a base, the domain-specific language, a short overview of the architecture and what features make it so interesting for us to use it.

4.1 The RSL Model

The Resource-Selector-Link (RSL) model [38] is a general metamodel for hypermedia that is based on the semantic, object-oriented data model OM [33]. Hypermedia is the extension of hypertext with multimedia facilities like sound or video. This RSL model is used as the base for MindXpres. Features of hypermedia like semantic linking, navigational links, structural links, transclusion, annotation and context awareness can be handled by the RSL model.

The resource, selector and link are seen as the three base components of the model and is defined as *the Link Meta model*.

- Resource: a resource is a piece of data that becomes concrete in a domain-specific context. E.g. in MindXpres, this can be text, video, images, or more.

- Selector: a selector is an abstract concept that selects a part of the resource. We can have many selectors on one Entity. E.g. we select a time-range of a video data file.

- Link: the linking concept allows us to link resources directly or via a selector. This is most fundamental in a hypermedia model. The RSL model also supports the concept of multi-source links.

Additional features are introduced by two other concepts in the core model on top of the three previous defined concepts.

- Property: a property is a parameter that is associated with an Entity. An Entity can be any of the previous defined concepts.

- Context Resolver: Each entity can be linked with a set of Context Resolvers. The correct Entity will be displayed based on a calculation by the Context Resolver.

The RSL Model

The previous concepts are forming the base for the RSL model. There are other features built around these concepts. The following features (user model, layers, and structural links) lay down the bases as how MindXpres does this.

4.1.1 The User Model

The user model describes the ownership and access of information. This explicit feature often lacks in other hypermedia models. It is possible to create rights for a group of users or even an individual user to access a piece of information. This feature is also important in our context of ARS when we share information with other people. It can happen that you wish to share some information with one person of the audience but not all.
The context resolvers can also be used to help you define the rights in certain context. For example, the presenter can share some extra notes to the audience members while giving his presentation. When the same presentation is published on-line, it may be the choice of the presenter that only the audience members have the right to access his extra information. Other people who obtain the presentation that did not participate will not be able to see this information.

4.1.2 Layers

Layers are the next feature that the RSL model provides. Previously, we defined the core concept *Selector* and saw that it is possible to define multiple selectors for one Entity which can cause overlapping. In the example that we kindly borrowed from [35], we see a conflict in the anchor tag of HTML. If we click on the link, the browser will not know which anchor tag to select. The layer feature of RSL model provides a solution for this. The solution is that for every resource, you can define a layer. Since many selectors can be defined for the same recourse, the restriction is that you can define a selector for a resource for a given layer. This makes a hierarchy of selectors. If a conflict occurs with the selectors, the one from the top layer is selected.

```html
<a href="target1">
  This is some text that links to target1, but
  <a href="target2">
    this text links to target2!
  </a>
</a>
```

4.1.3 Structural Links

Structural links are a subtype of the first class Link object in the RSL model. These specified links bind information into a structure. E.g. a document has a structure, it contains chapters with sections.
The RSL model also has a navigational subtype of the Link object that specifies the way of navigation. The concrete form depends on the domain.

4.1.4 Conclusion

We have introduced the RSL model that is used as a base model for MindXpres. For completeness, the complete model is given in figure 4-1. It is a solid model for hypermedia. In our context, the RSL model is important because it provides a well defined way of sharing information between everyone who is involved with a presentation and we can use the context-awareness.

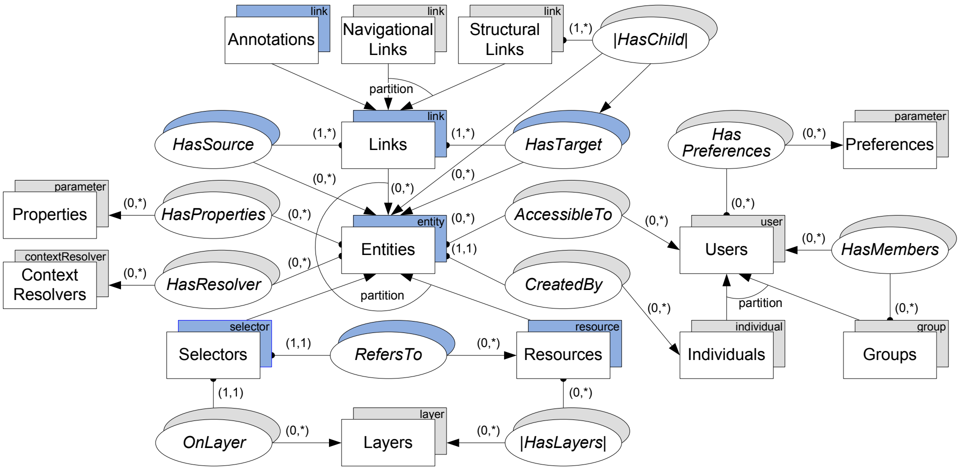


Figure 4-1: Complete RSL model (Source: Signer and Norrie [38])

4.2 MindXpres Building Blocks

In the following paragraphs, we discuss the how MindXpres is build without going into details. For all the details, we kindly refer to [35]. We first discuss the XML Authoring language that will be outputted into HTML5 and the visualisation library. In figure 4-2 you can see the general architecture of MindXpres.

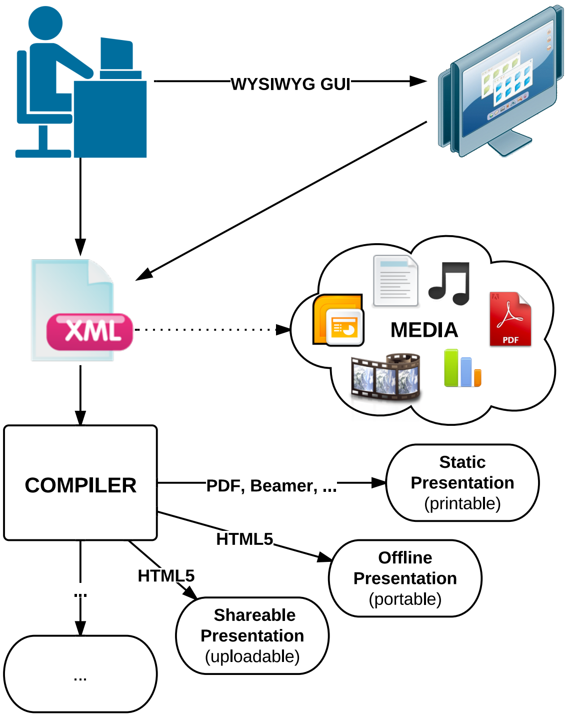

WYSIWYG GUI
XML
MEDIA
PDF
COMPILER
PDF, Beamer, ...
Static Presentation (printable)
HTML5
Offline Presentation (portable)
HTML5
Shareable Presentation (uploadable)
...
...

Figure 4-2: MindXpres Architecture (Source: R. Roels, S. Beat [35])

We see that a user has two options to create his presentation. First, opting for writing your XML language yourself or the second way is to use a GUI that does this for you. This domain specific XML language includes all kinds of media. The XML document is then processed by the compiler that first checks if it is in the correct format. If everything is correct, the compiler outputs the XML into the desired format.

4.2.0.1 XML Authoring language

We see that MindXpres uses XML[1] as a base mark-up language. The XML document is validated by the compiler as seen above with the use of an XML schema. This comes in handy when the user creates his presentation and forgets a few attributes. If this happens, the compiler adds them.
In the following box, we see an example of the language used to compile into the desired HTML5. It will generate a presentation that contains one slide. This slide displays a title and has an image with two item bullets underneath it.

[1] http://www.w3.org/XML/

```
<presentation theme="vub">
  <slide title="foobar">
    <image src="foobar.jpg">
    <bulletlist>
       <item>item1</item>
       <item>item2</item>
     </bulletlist>
  </slide
</presentation>
```

4.2.0.2 Compiler

The compiler of MindXpres is written in Java and has three main goals. The first goal of the compiler is to validate the XML document made by the user based on the XML schema. This validated document is then used in a SAX parser that traverse the document in order to transform it into valid HTML5. The previous given XML is compiled to the following HTML code:

```
<div data-type="presentation" data-theme="vub">
  <div id="element_1" data-type="slide" data-title="foobar">
    <img width="300px" src="foobar.jpg">
    <ul>
     <li>item1</li>
     <li>item2</li>
    </ul>
  </div>
<div>
```

4.3 Plug-in Mechanism

As mentioned above, MindXpres has some interesting presentation tool features. It separates the content and visualisation as in a LaTeX document. Due to HTML5, which is accepted in all major browsers, portability and distribution over heterogeneous devices is possible. The MindXpres tool also offers components to facilitate specific types of information and has a plug-in mechanism that can add new future media types. Non-linear traversal of the presentation, hyperlinks, transclusion, semantic linking and navigation of information, multimodal input, dynamic interaction with the content, the import of external presentations are also features why MindXpres becomes powerful to use.

MindXpres does not have a core with hardcoded components and aesthtics. This is possible due to the plug-in framework of MindXpres. The presentation tool has no core with hardcoded components or aesthetics. All these features are plug-ins which allows thirds parties to replace, modify or even add functionality to the tool. This is why it makes it so interesting for us to use this presentation tool. This mechanism creates expandability in a presentation tool, which is often lacking or fairly limited in other existing presentation tools.

Plug-in Mechanism

We will have to expand MindXpres with a communication module that can be used by users to create ARS plug-ins. These will in essence be MindXpres plug-ins that make use of the communication module. If an ARS feature is not present or it lacks some functionality, it is possible to add it yourself or change it to your needs. This mechanism is not only for the programmers that wish to create plug-ins. It also allows other users to only download the plug-ins that they desire. In MindXpres, there are three major types of plug-ins that we discuss in the following subsections. These plug-ins focus more on the functionality and implementation details rather than aesthetics. The aesthetics of plug-ins can be changed by templates.

4.3.1 Components

Components are the group of plug-ins that provides visualisations and functionality for a specific content type for different content containers. E.g. a bullet list, images, video, and more. It decides how relevant content is displayed, and defines the interaction with it. Note that everything is a plug-in, including the content containers to the simplest content types. An example of where it would make sence to create a new component plug-in is video. We could create a plug-in component that provides video that uses the HTML5 component. However, one can prefer the use a flash player for videos and so an other plug-in component can be created.

4.3.2 Containers

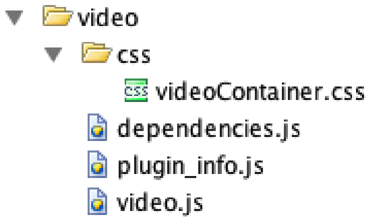
Containers are the elements that group or organise components visually. The most obvious container would be a slide. Each slide contains different content and can have other content that reoccurs. E.g. a slide title, footnote with author's name or slide number. These reoccurring content can then be abstracted to a higher level of container which facilitates the work of the user. It is also possible that the container provides functionality to help the user with the layout by defining presets or allowing easy definitions of layouts.

4.3.3 Structures

Structures are the elements that lay out components on a larger scale. For example displaying element in a grid so it can be used to have a Zoomable User Interface. The difference with the containers is that structures have ties to the XML language to define presentations. This information the XML language has is often necessary for the plug-in, because it needs to know the legal ways of laying out the components. This is mainly needed for complex visualisations.

4.3.4 How to use

Every plug-in is contained into individual folders that are placed in a specific folder. The plug-in folder contains all the Javascript, CSS, and other resources that are relevant for that plug-in. The mechanism to place the plug-in in a specific folder, makes it easy in order to add or remove certain plug-ins. A specific naming convention allows the plug-in to be found by MindXpres. Every plug-in must have a file called plugin_info.js that is loaded as first. This file shares the plug-in-specific information to MindXpres such as the tags that it provides for the XML authoring language. The information that is shared also contains the name of the main plug-in object, which is initiated by calling its `init` method. In figure 4-3, we see an example of a video plug-in folder.



```
▼ video
    ▼ css
          videoContainer.css
      dependencies.js
      plugin_info.js
      video.js
```

Figure 4-3: The plug-in structure Source: R. Roels [35]

4.4 Conclusion

In this chapter, we have introduced MindXpres. It is a presentation tool that will help us creating the next-generation presentation. We have seen that it is highly portable due to HTML5. This high portability is needed for our ARS if we wish to include all the audience members. The visualisation layer of MindXpres is heavily inspired by Zoomable User Interfaces and spatial hypertext. We have seen that MindXpres offers a plug-in mechanism through a specific naming convention. The plug-in mechanism offers third parties the possibility to add, modify, and remove features in an easy and clean way. This flexibility is a killer-feature for us when we wish to create ARS features. We have to extend the MindXpres with a communication module so that plug-ins can use it as an API.

5 Architecture and Communication Technologies

In order to build our ARS for MindXpres, we need to identify some critical aspects of creating an application. These aspects will influence what our communication backbone will look like. We start by identifying candidate architecture models. This architecture should satisfy our needs. These are: able to handle a flexible growth of clients, provide cross-platforms, and separation of managing and processing information. After an extensive look at some candidate architectures, we start with identifying protocols and techniques so that devices can communicate with each other. The needs for our protocol is that it provides a real-time and bi-directional communication with a minimum latency and overhead over the communication line.

5.1 Architecture

For this section, we discuss the candidate architectural models that we can use for our ARS. What we try to achieve in this section is to find a solid solution for communication between many devices. In order to find this solution, it is necessary to see the benefits and drawbacks from an architecture since it provides the backbone of the application. Based on what we discuss here, we choose an architecture that satisfies our needs considering its properties. Based on [3], a good architecture for an ARS tool should address the following characteristics and components:

- **Portable:** Audience response unit should be small and wireless to provide students with mobility and ease of use.
- **Cost Effective:** No customised hardware should be use since this improve the system's cost effectiveness. Mobile phones are preferred.
- **Privacy:** Audience should feel confident to participate and this participation should be able to do anonymously.
- **Prompt:** The time needed to access the ARS tool should be minimal and easy.
- **Scalable:** Adding and removing audience members should be able without any interrupt.
- **Support Heterogeneous Systems:** Given that audience members uses their mobile phone, the ARS tool should be independent from a platform.
- **Content Adaptability:** To help the audience members understanding the concepts of a presentation, changing the content should be possible. This helps the audience gain a better understanding of the concepts.
- **Battery Power:** Since audience response units are battery based, there should be a strategy to recharging depleted phone systems to minimise any interruptions. Encouraging to take a charger can be an option.

What follows now is a description of client-server and a Peer-to-Peer network architecture. We omit all other architectures (e.g. N-tier models with N different from 2) since we consider them to be overkill for our needs and it only makes the application more complex.

5.1.1 Client-Server

The term client-server, often also referred in the literature as 2-tier, is a popular model to design a web application. It started to gain popularity the moment the Personal Computer (PC) was released. In Figure 5-1 we see a representation of this model.

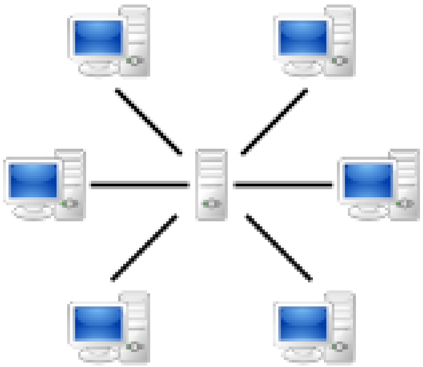


Figure 5-1: Client-Server model

It is an upgrade from the 1-tier or mainframe model where you move some responsibility from the mainframe towards the client. Often, this is the presentation layer from the application that gets separated from the logic and resource layer. This presentation layer is not just the interface of the application, it can also be responsible for processing data that will be presented on the client or send back to the server. We can divide the clients into thin and fat clients, depending on how much responsibility they have. We give an overview in Table 5-1.

Thin client	**Fat client**
minimal functionality, typically restricted to the presentation layer	Extended functionality, includes parts of application logic
Advantages: small code base, easy to update, complexity is left to the server	Advantages: Client is powerful and does not load the server, added value to the client, client processing can be tailored to different users
Disadvantages: does not use capacity of clients, functionality of clients is limited, needs to load the server often	Disadvantages: Larger code base, maintenance and upgrade cost, difficult to port across platforms

Table 5-1: Differences between thin and fat client

The client-server model creates the notion of services. Services offers us the use of an Abstract Programming Interface (API) that every client can use. There have been many attempts to agree on a standard for every type of server. Another interesting aspect about using the client/server architecture, is that it offers a central point for integrating other modalities. If in the future we wish to include other modalities (e.g. digital pen or Kinect), we can do so on the same machine where the server is running.
Allow us to sum up some of the characteristics of client-server that are important for our needs:

- Use the computing power of the client
- Central control (also useful for future extensions of modalities)
- Scalability is limited since there is a limit that one server can handle
- Creates the concept of an API

5.1.2 Peer-to-peer

A peer-to-peer network (P2P) can be defined as follows:

> A distributed network architecture may be called a peer-to-peer network, if the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers). These shared resources are necessary to provide the Service and content offered by the network (e.g. file sharing or shared workspaces for collaboration). They are accessible by other peers. [37]

The idea is that the distributed systems consist of nodes that are connected with each other. They are able to organize themselves into a network with the intent to share resources. This should all be possible without the need of any centralised server or authority. Basically, every peer can take the role of a client or server. If they receive some information request from another peer, they responds like a server does. It is then also possible that this peer asks for information to another peer, like a client does in the client-server model. In a peer-to-peer network, we put a lot of responsibility at the peers. Using this kind of network also has a lot of social impacts which are described in the P2P manifesto [4]. This can become relevant if the network is getting big. Since we will keep it relatively small scale, the social impacts are less relevant. We can see a representation of such a network in Figure 5-2.

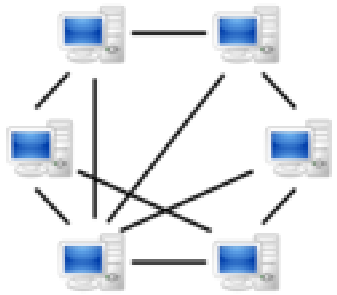


Figure 5-2: Peer-to-peer model

Allow us to sum up some of the characteristics of P2P:

- Peers are free to come and go, this makes the network very flexible
- Decentralised network, meaning that there is no central point that manages the network communication.
- It is cheaper than client-server because of the lack of the central server
- Relatively simple to set up
- Responsibility for providing or consuming information lies at the peers
- All peers run on the same protocol and software to communicate with each other

A popular implementation of this protocol for file sharing would be BitTorrent.

5.1.3 Conclusion

In this section we discussed two candidate models that we can use in the creation of our ARS extension. We discussed the client-server model and made a separation of thin and fat clients. After this, we look at the second candidate model which is peer-to-peer. We omit other models because we consider these as the most appropriate ones for our needs.
After some investigation and discussion on the architectures and based on their strengths and weaknesses, we have chosen to work with a client-server architectural model for our application. Allow us to justify why we take client-server architecture:

1. A central point where we can put the logic and management. Extensibility with other modalities (e.g. digital pen) is possible.
2. Put less responsibility to the clients: we also wish to support devices with less calculation capacity. In order to realise this, we have to put some more responsibility towards the server like the management of all the active clients. In P2P, they have to act as a server and client.
3. With the eye on the future, we could enable remote access to the server for further logic and features.
4. We can rely on an optimised server rather than unknown and maybe less capable devices in a P2P.
5. It satisfies the design components described in [3].

5.2 Communication Technologies

In order to create our application, we need some way of communicating with all the devices. There is a wide range of communication protocols that we can use in order to realise this communication. In this part, we will take a closer look the some of the communication technologies who are candidate for our application. The first requirement that this communication technology needs to have is to create a real-time communication that is bi-directional or emulates this. We divide these technologies into two groups based on the protocol, HTTP and Web Socket. HTTP is the most used protocol in the World Wide Web which makes it a natural candidate. Web Sockets are a candidate because it is recently added as a standard protocol of HTML5. We will look into detail what these technologies include, the advantages and disadvantages.

5.2.1 Comet Technologies

Comet programming [36] is a model that achieves communication from the server to the client without the browser of the client explicitly requesting it. This can be done for example with the help of Javascript. Comet is an umbrella term for a range technologies, mostly with the use of the HyperText Transfer Protocol (HTTP). All these technologies are trying to achieve the same interaction between the client and server and overcome some of the major drawbacks of HTTP. What follows is an introduction to HTTP (including the drawbacks) and then we discuss some interesting Comet technologies. We will describe Ajax long polling, HTTP streaming, and finish with Web Sockets.

5.2.1.1 HTTP

The HyperText Transfer Protocol (HTTP) is the foundation of data communication in the World Wide Web that is commonly implemented on top of the Transmission Control Protocol (TCP). We consider this protocol as a valid candidate because when using a web solution, HTTP is one of the most common protocol used. The term HyperText is introduced by Ted Nelson [32] who was inspired by V. Bush [8] with his vision about the Memex. When the World Wide Web project was announced by Tim Bernards-Lee, et al [5], they developed the first version of HTTP. This protocol has changed over the years since it was first introduced and became HTTP v1.0 in 1996 [6]. Improvements where then made, and currently we are using the HTTP v1.1 that was released in 1999 [14]. We will go into some more technical features of HTTP now.

HTTP works with a request/response system. Below, we see a standard HTTP request and response for the website of VUB. The HTTP protocol consists of a message type, message headers, and a message body. We go through the most important aspects of these parts. We start with the Request.

A request is always initiated by the client towards the server, this server will then respond to the client accordingly. There are different kinds of methods that the client can use which are always placed in the beginning of the request and defines the message type. Allow us to describe the most important ones with their semantic.

- GET: using this method, we only expect to receive data from the server and nothing else.
- POST: a request that the server accepts data send by the client.
- HEAD: we only request the header of the response, without the body.
- PUT: Requests that the data be stored under the supplied URI.
- DELETE, TRACE, OPTIONS, CONNECT, and PATCH are omitted since they are hardly used in daily practise.

The header lines will follow after the method and can be seen as meta data. They provide information about the request (or response), or about the object sent in the message body. In the example below, we see that the request is made by a Windows machine using Firefox as a browser. This is defined by the User-Agent header. In the Accept header, we define which are the appropriate media types which are acceptable for us in the response.

```
GET /?lang=nl HTTP/1.1
Host  www.vub.ac.be
User-Agent  Mozilla/5.0 (Windows NT 6.1; WOW64; rv:22.0) Gecko
    /20100101 Firefox/22.0
```

```
Accept  text/html,application/xhtml+xml,application/xml;q=0.9;q
    =0.8
Accept-Language en-US,en;q=0.5
Accept-Encoding gzip, deflate
Cookie  lang=nl; __utma
    =1.1760003613.1373271543.1373271543.1373271543.1; __utmb
    =1.2.10.1373271543; __utmz=1.1373271543.1.1.utmcsr=(direct)|
    utmccn=(direct)|utmcmd=(none); has_js=1; __utmc=1
Connection  keep-alive
Cache-Control max-age=0
```

After receiving a request from a client, the server responds with an appropriate message. The Response of the previously shown Request can be seen below. A response always starts with the version followed by a response code. This code gives information about the message whether it was successful or if an error occurred. All the codes have their specific semantic. E.g. code 200 stands for a standard response of a successful HTTP request. Just like in the Request, there is a list of header lines. These header lines gives us again some information about the message or the machine where the message comes from. E.g. we see in the server header that it runs Apache 2.2.14. And that the website of the VUB uses a Drupal 7 system. A more important header is the Content-type. This describes the format of the message body, in this case it is text/html. Other content types are possible like Javascript or css for example. The encoding of the message is done by some kind of charset (UTF8 is currently a widely used encoding).

```
HTTP/1.1 200 OK
Date  Mon, 01 Jul 2013 08:20:31 GMT
Server  Apache/2.2.14 (Ubuntu)
X-Powered-By  PHP/5.3.2-1ubuntu4.15
X-Drupal-Cache  HIT
Etag  1373270134-1
Content-Language  nl
X-Generator Drupal 7 (www.drupal.org)
Cache-Control public, max-age=60
Expires Sun, 19 Nov 1978 05:00:00 GMT
Vary  Cookie,Accept-Encoding
Content-Encoding  gzip
Last-Modified Mon, 01 Jul 2013 07:55:34 GMT
Content-Type  text/html; charset=utf-8
Set-Cookie  lang=nl; path=/; domain=.vub.ac.be; expires=Fri, 23-
    Jun-2073 08:20:31 GMT
Keep-Alive  timeout=60, max=999
Connection  Keep-Alive
Transfer-Encoding chunked
```

For every update or file of a website, there is a new HTTP Request and Response. This means for connecting to the home page of the VUB, there are 56 request and responds needed (Tested on 6th of July 2013). Many of those are for CSS and Javascript. As mentioned earlier, it is important to

know that whatever other technology we use, the request must be initiated by the client. Over the years, solutions to this limitation have been created that we will discuss later in this chapter. These group of solutions were made possible by Javascript and are called Asynchronous Javascript and XML (AJAX). This mechanism is called short-polling.
Short polling is considered as the traditional technique. The idea is that the client regularly sends a message to the server to see if there is an update (pull) of the desired information. If there is no update available, the server responds by sending an empty message. After a given amount of time, the client initiates this process again. If there is information available, the server will send this to the client. The client can then use this information in its application to update its current state. In HTML, this can be updating the Domain-Object Model (DOM) tree. The drawback is that it consumes a lot of resources from the server and network to an unacceptable level and that a server response has to be initiated by the client.
To overcome these drawbacks, server-push techniques have been developed. These techniques are under the umbrella term Comet as defined above. What follows now is an introduction of the three implementations of Comet: long polling, streaming, and Web Socket. For the first two, we base ourself of [25, 36]

5.2.1.2 Long Polling

In long polling, we try to minimise the mentioned drawbacks of short polling. The server only responds to a request on an event, status or time out. The communication line between client and server remains open until the server responds. Most of the time, the client makes a new request right after receiving a long poll response. As a result, the server will only respond to the client if new information is there to send to the client in an asynchronous way. It is possible to make the HTTP connection persistent to avoid extra overhead of establishing a new TCP/IP connection. The life cycle of long polling is:

1. Initial request is made by the client and waits for a response.
2. The server waits with the response to a poll request until an update is available, or until a particular event occurs.
3. Whenever an event occurs, the server sends it back to the client as the response.
4. The client can send a new long polling request after it receives the response. This starts a new life cycle.

Issues

- Header overhead: Every request and response is a HTTP message which creates an overhead in communicating.
- Maximal latency: If the client sends a long poll request to the server, it has to wait for the response before it can do a new long poll request. This implies that the maximal latency of the long polling takes 3 network transits (long poll response from server, long poll request from client, long poll response from server). Since HTTP is on top of the TCP/IP, it is possible to lose a package in which the maximum latency will take more than 3 transits. Overall, it gives good latency if the frequency of messages is low. But it can create high latency is there is always a quick update from the server back to the client.
- Allocated resources: To every TCP/IP connection and HTTP request, resources have to be allocated. The long polling requires that these connections remain open for a given time. The allocation of those connections, especially the HTTP requests, can be high which can create an overflow on intermediates.
- Timeouts: The long polling needs to wait until the server has new information to send to the client. This pending can create a time out.
- Caching: this mechanism that is implemented in most applications can interfere with long polling.

Conclusion We see that long polling technique removes some of the overhead of the traditional or short polling technique. It is an interesting technique that qualifies itself as a candidate for our need. However, it has some issues that we should consider. Some of the issues are possible to avoid while others remain.

5.2.1.3 HTTP Streaming

HTTP Streaming is a comet technology that turns the regular HTTP principle into a stream. This also allows us to create an event driven communication. The idea behind this mechanism is that the request remains open for a period of time or even indefinitely. The server can push data to the clients and the connection between them remains open. This significantly reduces the network latency because the connection does not need to be re-initiated for every message going between them. The server needs to be able to send small parts of the information to the client. The life-cycle is as follows:

1. Initial request is made by the client and waits for a response.
2. The server waits with the response to a poll request until an update is available, or until a particular event occurs.
3. Whenever an event occurs, the server sends it back to the client as a part of the response.
4. The data is send by the server but does not terminate the request or the connection.

Issues

- Network intermediaries: The standard HTTP protocol gives the opportunity to work with intermediaries. These intermediaries can be proxies, firewalls, gateways, and more. These intermediaries can do some logic on the data, buffer it, or it can happen that the intermediary has a big workload. This implies that every intermediary decide themselves when to send the data. This is not desirable for streaming because then the client can become out of synchronisation. So we can not work with the standard network intermediaries.
- Maximal latency: the maximum latency of the HTTP Streaming protocol can become high because of limitations of the network and browser. Since the HTTP is implemented on top of the TCP/IP protocol, packet losses can occur which means we need to retransmit packages. The limitation of the browser arise because of the limited memory. Every event given by the server increases the memory use at the client side. This means that the client has to terminate the streaming occasionally.
- Client buffering: In HTTP specification [14], it is not a requirement to make the data from a partial HTTP response available to the client application. The application can wait until it receives the full response which can cause a buffer overflow.
- Framing techniques: Separating the response stream into multiple application messages is performed at the application level instead of the HTTP level since it is not possible to use the HTTP chunks as delimiters of those messages. This is not possible because the intermediate proxies may re-chunk the message stream. In long-polling this does not occur, it is possible to send for each application message a new HTTP response.

Conclusion Next to the long-polling technique, HTTP Stream also makes a good candidate. The overhead is even less that in long polling but it comes with a price that is discussed in the issues.

5.2.1.4 Web Sockets

Web sockets protocol is a relatively new technology (finalised in December 2011) that is also build on top of the TCP. It enables a bi-directional communication between the client and server and allows us to communicate real-time with each other. The protocol starts with an opening handshake which is followed by basic message framing. The intent of this protocol is to provide a way for applications that need a real-time bi-directional communication with the server and do not rely on the HTTP protocol like the two previous discussed technologies. Web games like Quake live[1] are a good example of these applications that need such a communication.

The Web Sockets have become more important with the rise of HTML5. The protocol is adopted as a standard [13] in the HTML5 specifications and it provides a dramatic improvement from the earlier discussed Comet technologies. A quote from Google engineer Ian Hickson that shows how significant using Web Sockets can be:

> *Reducing kilobytes of data to 2 bytes and reducing latency from 150ms to 50ms is far more than marginal. In fact, these two factors alone are enough to make Web Sockets seriously interesting to Google.*

Let us take a closer look at the web socket frames. There is a difference between a sending and receiving frame. The sending frame consists of:

1. one byte that contains the type of data (and some additional info which is out of scope for a trivial server)
2. one byte that contains the length of the frame
3. two or eight bytes if the length does not fit in the second byte. In this case, the second byte is a code to say how many bytes are there to follow to describe the length
4. the actual data

This takes a minimum total of 2 bytes and maximum 10 bytes of overhead for the frame to be sent.

The receiving frame consists of the following bytes

1. one byte that contains the type of data
2. one byte that contains the length of the frame
3. two or eight bytes if the length does not fit in the second byte. In this case, the second byte is a code to say how many bytes are there to follow to describe the length
4. four bytes that are the decoding keys
5. the actual data

[1] http://www.quakelive.com/

This takes a minimum total of 6 bytes and maximum 14 bytes of overhead for the frame to be received.
So the protocol gives a good reduction of the overhead in the communication and the network latency, especially compared to earlier discussed communication ways. We will go into detail in the comparison section of this chapter when we compare the mentioned techniques. It does require a initial HTTP request in order to connect to the desired site. From there on, it will be upgraded to a Web Socket connection. The life cycle of a Web Socket is as follows:

1. Client attaches itself to a series of events he wishes to know from the Web Socket.
2. Initial request is made by the client to a Web Socket endpoint (upgrading the HTTP request to a Web Socket during initial handshake).
3. Once the connection is established, data can be sent from client to server or via an event-based way.
4. Client can close the connection when he is done working in the application.

Issues

- Network intermediaries: This is not such a big issue any more but some intermediaries do not understand the Web Socket protocol yet. Also, the connection can be open very long, it is up to the intermediaries to decide when it is too long. If this occur and the connection is closed, a new connection has to be established.
- Supportability: the Web Sockets are only supported by the latest version of browsers. Because of this, we wish to make a web application and wish to support older versions of browser, we will have to rely on other communication technologies.

To end this section about Web Sockets, we like to mention that Web Sockets are also considered part of the Comet model (using HTTP is not a requirement for Comet). As we saw earlier, the Comet model is a paradigm to create communication between client and server without the client explicitly asking for an update. Since this is also the case for Web Sockets, it should be considered as a third group of Comet technologies that differentiates itself from the rest because of the bi-directional communication between client and server without any problems.

5.2.2 Comparison

Allow us to compare the discussed protocols with each other. We focus on the latency and throughput produced by each protocol.
Let us first start with the initial handshake. Here, the complexity of both protocols are the same since a Web Socket uses the HTTP handshake to start with and then upgrades its protocol. The size of the HTTP handshake are always different since in some requests, the header can be larger due to extra fields like cookies. More important to know is that the handshake always happens again in HTTP. This is not the case for Web Sockets. Additionally, the Web Socket protocol uses only one TCP socket connection per user while for HTTP, a new connection is needed for every request.
So how much is this overhead? In [27], they use 871 bytes of header data for a HTTP request. Since we believe this is not a good representation (it is from a single HTTP request & response) of the real average HTTP header data we conducted a small research of our own. We adjusted a web crawler[1] that crawled the complete website of the VUB. With a restriction of a url level of 5. With this we calculated the average HTTP header data (total HTTP message size minus the HTTP body size) and we came on the average of 1084 bytes of header data (overhead). Depending on the day you crawl, this average can change. For Web Sockets, once the connection is established, the overhead is only 2 to 14 bytes as seen before.
We continue with the comparison from [27] where they are going to look at what this overhead of normal polling means for a large amount of users in 3 cases. We do change the amount of bytes with ours.

> **Case 1**: We have 1000 clients polling every second: Network throughput is (1084 x 1,000) = 1,084,000 bytes = 8,672,000 bits per second (8.27 Mbps)
> **Case 2**: We have 10.000 clients polling every second: Network throughput is (1084 x 10,000) = 10,840,000 bytes = 86,720,000 bits per second (10.34 Mbps)
> **Case 3**: We have 100.000 clients polling every second: Network throughput is (1084 x 100,000) = 108,400,000 bytes = 867,200,000 bits per second (103.38 Mbps)

As we can see, this is a huge amount of bits just for overhead. So let us do the same for Web Sockets. In the study they use the minimal amount of overhead in a frame in their calculation. Again, we believe this is not that representative and somewhat biased. We take the average of the possible overhead of the frames, which is 8 bytes. We conduct the same calculations:

[1] https://code.google.com/p/crawler4j/

Case 1: We have 1000 clients polling every second: Network throughput is (8 x 1,000) = 8,000 bytes = 64,000 bits per second (0.064 Mbps)
Case 2: We have 10.000 clients polling every second: Network throughput is (8 x 10,000) = 80,000 bytes = 640,000 bits per second (0.64 Mbps)
Case 3: We have 100.000 clients polling every second: Network throughput is (8 x 100,000) = 800,000 bytes = 6,400,000 bits per second (6.4 Mbps)

Let us plot the results in Graph 5-3. We clearly see how significant Web Sockets reduce the size of overhead used. Web sockets clearly come out as the winner in this comparison. If we compare this with HTTP Streaming, there is no difference with Web Socket. This is due to the fact that in HTTP Streaming, the connection also remains open from server to client. The server sends messages to the client the moment they are ready without the header data. We do lose the bi-directional communication from Web Socket in the HTTP Streaming.

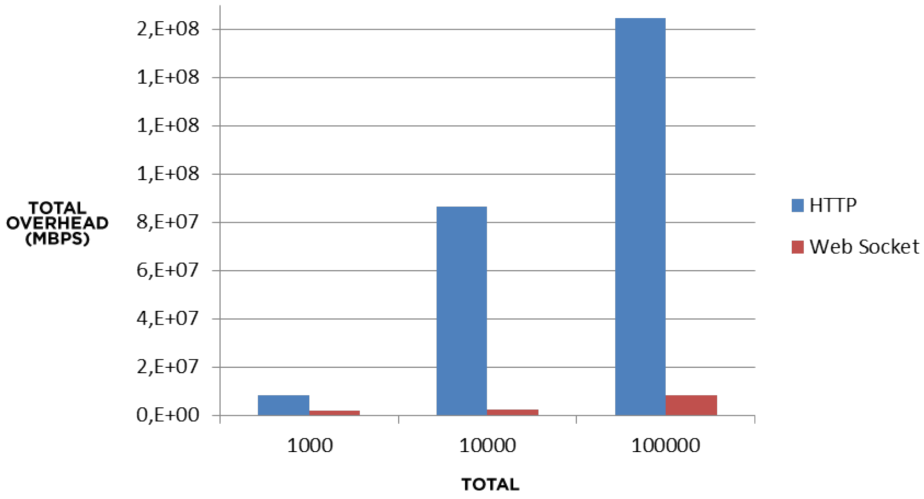


Figure 5-3: HTTP vs Web Socket - overhead

To investigate the latency reduction, we use Figure 5-4. The figure represent the communication between the client and server. The top half is using a polling technique with HTTP and the bottom half is with an upgraded Web Socket connection. The moment the server has a new message ready, it will send this to the browser. In this example, we take that it takes 50ms to travel from server to the client. For the polling method, we keep have to send a request to ask if there is a new message available. This stands in

contrast with the Web Socket protocol. The connection remains open and so whenever a new message is available, the server sends it to the client. This is a reduction of 3:1!
For the HTTP Streaming again, there is no fundamental difference since the connection also remains open. For long-polling it depends on the frequency of success message. If there is a very quick successor of the request, there is no difference from the short-polling as we see in Figure 5-4.

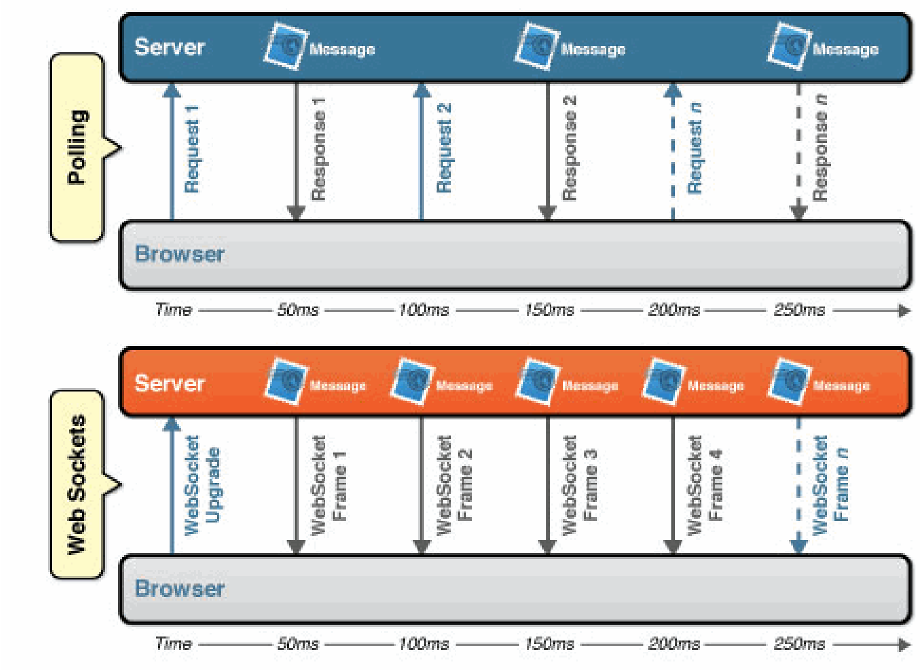


Figure 5-4: HTTP vs Web Socket - latency [27]

Conclusion The main requirements are that our communication protocol can create a bi-directional and real-time communication line. As an engineering point of view, we should search for the best solution. This includes to find a protocol with a minimum of overhead and latency.
With this said, we opted to use Web Sockets. In the previous chapter, we made a comparison between HTTP solutions and Web Sockets. In these tests, the Web Socket came out as a winner with a latency reduction of 3:1 and header data overhead reduction of about 500:1. Allow us to summarise why we opt for this protocol.

1. Gives a bi-directional communication line
2. It is real-time
3. Standard of HTML5
4. Comes out as best protocol in our comparison with HTTP solutions

5.2.3 Frameworks for our ARS

So to recapitulate of the previous conclusions, we choose to work with a Client-Server architecture and use WebSockets as the communication protocol on top of TCP.
In order to work with this protocol, the best thing to do is to work with a framework. A framework encapsulates a lot of features so that we, as programmers, do not have to worry about underlying details. We believe that a good investigation of the possible frameworks will enlighten us with future work since choosing a framework is part of the solution of the problem. What follows are some frameworks that allow us to create server side WebSocket applications. The frameworks are designed to handle a large number of open WebSocket connections. This handling requires an architectures that receives high concurrency with a minimal performance cost. Threading or non-blocking IO, are the base for designing these architectures. Most of the frameworks are focussed on solving the C10k problem. This problem refers to optimizing the server to handle a large amount of concurrency at the same time.

Javascript Javascript is an event-driven programming language that gained a lot of popularity with the rise of the internet over the last 15 years. However, Javascript does not limit itself to the browser, it has many other applications. For example, creating a server-side platform for writing internet application. This is where we are interested in.
Node.js[1] is such a platform and is defined by the creators as:

> *A platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.*

This platform gives us the opportunity to create the back-end that we require for building our ARS. While it is relatively new and still under development, it seems it is already here to stay. Many big vendors like Microsoft, Yahoo, LinkedIn, and more are speaking highly of this platform.
We can use this platform with some other libraries such as Socket.IO[2], WebSocket-Node[3], and WS[4]. These libraries creates an abstraction of the Node.js platform for both server and client-side. It makes is easier to create a data-intensive real-time application.

[1] http://www.nodejs.org

[2] http://socket.io

[3] https://github.com/Worlize/WebSocket-Node

[4] https://github.com/einaros/ws

Java Java is a very popular programming language and has many web application architectures that have proven themselves as solid solutions for defined problems. Jetty[1], is a server side solution for creating the back-end for our application. It is a Java-based HTTP server and Java Servlet container that is integrated as a component of the Eclipse IDE. Jetty provides Web services that supports Web Sockets. It can be embedded with existing frameworks like Java Spring framework.
Like Node.js, it is still under heavy development but has some stables releases. Companies like Google, Yahoo and others are using Jetty to power some of their applications. Such as Active MQ, a popular Message Broker.

Others There are many other solutions in many other programming languages. Such as Python, .Net, Ruby, Erlang,... We discuss two more frameworks who are gaining popularity on the internet among programmers. Tornado[2] is a framework that is written in Python that was used to develop FriendFeed. This application was a aggregator of many social media websites that showed what your friend were doing. In 2009, Facebook bought it and integrated this.
EventMachine-WebSocket[3] is framework that is written in Ruby. It does not differ much from previous talked frameworks, other than the programming language.

5.3 Conclusion

In this chapter, we talked about our application architectures and communication protocol that we are going to use as building blocks for our application. We opted for a client/server architecture and use Web Sockets as our communication protocol. Once we decided this, we looked at some candidate frameworks that helps us in creating this. When we chose a framework, the main points we looked at was the ease of setup and the knowledge of the programming language. We first tried the Node.js platform in combination with the Socket.IO libraries. We were amazed by the ease of setup and how quick everything worked. While the platform and libraries are not well documented, we could easily found our way to create a small chat application. After this we used Jetty in combination with a Spring framework. Coming from the Node.js platform, we found this setup rather annoying. We also encountered issues with Eclipse that took a long time to solve. After a few hours of work, we were successful with setting everything up.
Because Node.js was so easy to setup and yet very powerful, we chose to continue our work on this platform in combination with the Socket.IO libraries.

[1] http://www.eclipse.org/jetty

[2] https://github.com/facebook/tornado

[3] https://github.com/igrigorik/em-websocket

6
Implementation

In chapter 3, we have detailed some perceptions, opinions, and aspirations about ARS. These were used to supplement the findings from the related work of chapter 2. Later, in chapter 4, we argued why MindXpres is an excellent presentation tool to build our ARS prototype, particularly due to its extensibility. This is followed by an investigation of the candidate architectures and network protocols for building our prototype. Allow us to recapture the key points of this investigation.

1. The prototype uses a client-server architecture
2. Web Socket is the network protocol we use
3. Node.JS is used as a platform in combination with Socket.IO libraries, all in Javascript

6.1 Goals

In this chapter, we go more into the technical aspects that are important for building our prototype of ARS. We formulate the goals that list up what is needed. Then we clarify how our goals are implemented, we give some proof of concepts of our communication module and show how it is implemented on an event-based manner.

6.1.1 Server

1. Define expectations and requirements of the server
2. Create an event-based server

6.1.2 Communication Module

1. Define and abstract the communication flows during a presentation into an API
2. Create the API with all methods

6.1.3 Integration and Use

1. Integrate the communication module into MindXpres' core
2. Use the communication module with MindXpres' core

6.2 Server

In the context of giving a presentation easily and without many technical issues, we came up with the idea of having a presentation box. This box is specifically made to give presentations including a complete ARS. Easy portability should be mandatory, like a laptop device that is currently widely used to give a presentation. However, making this presentation box, it also gives the opportunity to make a fixed unit in every room where presentations are given. The idea of having a fixed unit would benefit giving a presentation in many cases. Currently, it is often the case that when giving a presentation, it takes time in order to connect it to the beamer and configure all the correct settings. With this fixed unit, everything can be set beforehand and reduce the technical time lost of giving a presentation.
The presentation box creates a hotspot network that allows connecting other devices (the clients). To make this hotspot, we use an external WiFi router because the one made with the presenter box' network card was not strong enough to support a large amount of users. We have a complete network with a central point, which acts as the server. The clients can navigate to the correct path where the presentation is given. This way, every client has its own copy of the presentation and is able to interact with the presenter and other audience members.

6.2.1 Intel Next Unit of Computing

With the idea of the presentation box in mind, an Intel Next Unit of Computing (NUC) is purchased to realise this idea. The NUC is a small device (only 9x9 cm) with a relatively powerful Intel i3 processor. It has an excellent overall performance. It has all the options of a modern laptop or desktop with dual HDMI outputs and a gigabit network card.

Server

Figure 6-1: Intel Next Unit of Computing

Upon receiving this device, it still had to be installed. This include installing the hardware parts (memory, Solid State Drive, Wireless adaptor). After the installation of the hardware parts, we encountered a problem with the BIOS of the device. The BIOS was not yet flashed to the up-to-date version which had the implications that it did not recognise any of the other hardware including a screen. After flashing it to the up-to-date version, everything worked fine. The fact that Node.JS makes the software portable across all major OS, makes our choice of OS not matter that much. Eventually, we chose for Linux (Ubuntu) with the reason that the Linux version of Node.JS is more up-to-date compared to other OSs.

6.2.2 The Publish Subscribe Pattern

Patterns have become much accepted in the industry, especially after the publication of the book Design Patterns: Elements of Reusable Object-Oriented Software by the Gang of Four in 1995. This started a complete movement on documenting reusable and flexible systems into patterns.
We explored the design patterns that we can use for the manner of communicating between objects and allows event driven development. This is where the Publish and Subscribe (pub/sub) pattern comes in. This pattern is also referred as the Observer pattern as described in the software design patterns book of the Gang of Four. Before explaining the pub/sub pattern, we look how the Observer pattern is defined by the Gang of Four.

> *One or more observers are interested in the state of a subject and register their interest with the subject by attaching themselves. When something changes in our subject that the observer may be interested in, a notify message is sent which calls the update method in each observer. When the observer is no longer interested in the subject's state, they can simply detach themselves.*

The intention of this pattern is to promote the decoupling of objects (or also referred as loose coupling in other literature). Objects can subscribe themselves to a change of state (e.g.: specific event or activity) of another object. When such change occurs in an object, it is the responsibility of that object to publish it to the collection of subscribers. The objects who are subscribed to it get a message. The objects that subscribe themselves are called the Observers in the Observer Pattern or Subject for the publishing objects. An object that is subscribed to another object can also unsubscribe itself from it. This results in the publisher removing the subscriber from its collection.
Let us take a closer look at how we use this pattern in our application. We use this with a central messaging server as shown on figure 6-2. In our context the publishers and subscribers would be the clients. A client can be a presenter or an audience member. The server maintains channels that can be created by the clients. This channel has a name and the option to be protected with a password so that only clients with the password are allowed to publish in this channel. Every client can subscribe itself to a channel for certain events. With these channels we introduce the notion of namespaces. It is possible for every plug-in to have its own channel. It is not necessary to use a channel to publish an event. The channels are used to broadcast to a group. When it is needed to target an individual user, it is possible to send an event to that user.

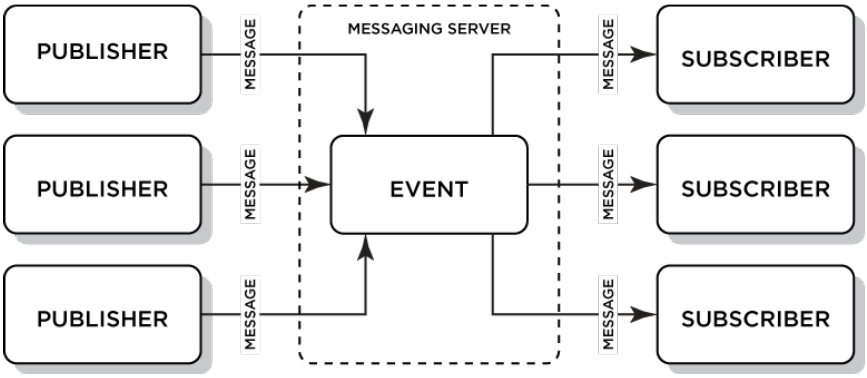


Figure 6-2: Publish Subscribe Pattern. Source[1]

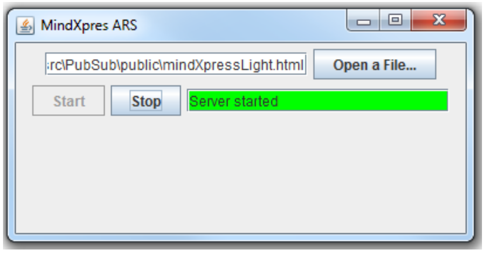
Advantages As mentioned above, one of the main advantages is that we have loose coupling between the objects (clients in our case). If an event occurs at a client, he can publish this in the correct channel but it does not matter for him who is subscribed to this event in that channel. The server has the responsibility to look at who is subscribed to the channel and forward this event. Flexibility is another advantage. We can have many publishers or subscribers to a channel. They can come and go at run time.

[1] http://gsraj.tripod.com/jms/jms.html

Disadvantages The loose coupling that is described in the advantages can also become a disadvantage when you wish for a complete guarantee of functioning. It is possible for a publisher to assume everyone is listening to him but due to an error of any kind, the subscribers can be not listening. The publisher will keep publishing new events to the channel because it does not know the subscribers problems. We solved this problem in our system by the use of a synchronise method in the communication module. However, more on this later in this chapter.

6.2.3 GUI

In addition to command-line interaction, we provide a simple Swing GUI in order to make it more accessible for the users that prefer a graphical interface. As shown in Figure 6-3, the GUI itself is fairly simple. A button to select a presentation is provided (HTML file). Once selected, one can start the presentation by clicking on the button with "Start" and feedback is given in a small text field. Stopping the presentation is done by clicking on the button with "Stop". As Swing is a built-in part of the Java VM, this GUI works on all platforms that run Java.



-rc\PubSub\public\mindXpressLight.html

Server started

Figure 6-3: Server GUI

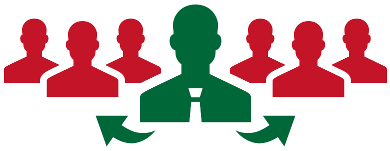
6.3 Communication Module

As we established earlier, we need to expand MindXpres with a communication module that is an API so that programmer can use this easily if they desire communication in their plug-in. In order to determine which methods our API should have, we started by identifying the kinds of communication flows that exists in a presentation. What follows in this section is a description of these communication flows, followed by how we implemented these in our module. We also give a short description of all the methods one can use in our module.

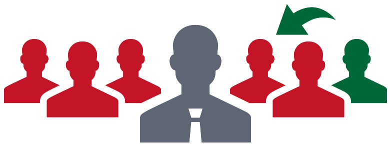
6.3.1 Communication Flows

Before implementing the communication module, we first have to identify the communication flows (CF) that can occur in an interactive presentation. We describe seven CFs and classify them in three different broadcasting-groups. The broadcasting-groups are: *broadcasting to a group*, *broadcasting to a presenter*, and *broadcasting to a single audience member*. This classification gives us a generic model for the implementation. Other CFs can occur in other settings (e.g.: including social media or multiple presenters), but these remain within our generic model of the broadcasting-groups. What follows is a description of these broadcasting-groups and the communication flows that we identified.

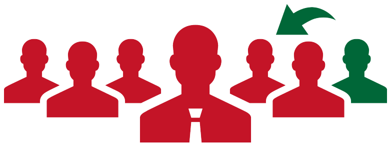
6.3.1.1 Broadcasting to a group



CF from presenter to audience
This is the classical way of communication during a presentation, also known as *ex cathedra.*

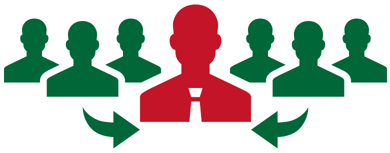


CF from audience member to audience
For instance when you make a group assignment, an audience member can broadcast his ideas to only a selected group.

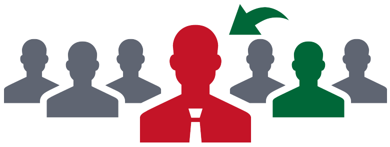


CF from audience member to audience and presenter
For instance, during a discussion an audience member can broadcast an argument to everyone

6.3.1.2 Broadcasting to a presenter



CF from audience to presenter
When the presenter wants feedback from the whole audience



CF from audience member to presenter
For instance when the information is unclear and the audience member asks a question

6.3.1.3 Broadcasting to a single audience member

CF from presenter to audience member
Can occurs for instance when the presenter poses a question to a person

CF from audience member to audience member
For instance, sharing annotations with each other

6.3.2 Methods

In previous sections, we discussed what features our communication module should cover. These features result in the API methods that a programmer can use to create the desired plug-in. What follows now is a description of those methods.

6.3.2.1 Publish Subscribe methods

To create an event-based server, we used the publish-subscribe pattern as previously described. We created an object `Channel` where a client can register or un-register itself from. The methods one can use are the following:

subscribe A plug-in can `subscribe` itself to a channel based on the channel name. If the channel name does not exists, the server creates this channel and register that plug-in to it. Every event from that channel will be forwarded to the client. On the client-size, the plug-in registers for the interested events of that channel.

unsubscribe This is the opposite from subscribing to a channel. When the plug-in calls this method, the server will remove the client from the channel which results in having no more updates from that channel.

6.3.2.2 Communication methods

To create the methods for the communication, we based ourselves on the communication flows as described in previous sub-section. As seen, these resulted in three different categories. The methods available in the module are the base for creating communication between all the attendants of the presentation. For every category, we created a method that can be used by the plug-in.

publish A plug-in can publish data to a channel. It has to specify the channel and event together with its data. The server will publish the data in the correct channel with the event to all the subscribed clients in that channel. This method is equivalent to the "broadcast to a group" category of the communication flows.

publishToClient A plug-in can publish data to a client based on the its ID and event. The server will forward this data to the correct client. This method is equivalent to the "communicate with one person" category of the communication flows.

publishToPresenter A plug-in can publish data to the presenter on an event-based way. The server will forward this data to the presenter with an event that has been defined by the senders. This method is equivalent to the "communicate with presenter" category of the communication flows.

6.3.2.3 Plug-in methods

It is required for plug-ins to give some extra functionality to the presenter. Therefore, the plug-ins are able to subscribe themselves to the communication module on load. When a presenter identifies himself by claiming the ownership of the presentation by giving the correct password, all the modules that are registered will be called so that the extra functionality is executed.

registerPlugin The plug-in is registered in the communication module.

claimOwnership This method can be called by the presenter. The presenter is identified by a password that is given to the presentation during the creation. When the presenter is successfully authenticated, all the plug-ins that are registered in the communication module are called to execute their presenter functionality. For instance, in a polling plug-in, the presenter is now able to open and close polls while this is not possible for non-authenticated users.

6.4 Integration in MindXpres

In this section, we describe how the integration of our ARS tool is done in MindXpres. In Section 4.3, we described how the plug-in mechanism of MindXpres works. We now use this mechanism to integrate our files into MindXpres in order to develop plug-ins with audience-oriented features.
The first task is to integrate our libraries that we use. This are the Socket.IO client side libraries. This is done by adding a folder in the `lib` folder of MindXpres with the name `SocketIO`. In this folder, we place our

Javascript files. We now have to define in MindXpres to include these files on initialisation. This is done by registering our files to the dependency injector as seen in Listing 6.1.

```
DI.register("socket.io", "socket.io/socket.io.js");
```

Listing 6.1: Dependency injection of Socket.IO (register)

After registering our library, we have to include it. This is done by calling the `include` function that has an optional parameter with a callback. In our case, we leave it empty.

```
DI.register("socket.io", function (){ });
```

Listing 6.2: Dependency injection of Socket.IO (include)

Our communication module that we wish to include, can be done exactly the same way. We create a new folder `communication` and add our `communication.js` file. We register it with the dependency injector of MindXpres and then include this as seen in Listing 6.3.

```
DI.register("communication", "communication/communication.js");
DI.register("communication", function (){ });
```

Listing 6.3: Dependency injection of Communication (register & include)

What we have now is an integration of our two libraries as plug-ins. Since these are integrated upon initialisation of the presentation, the Javascript objects will be available for the other plug-ins. Every plug-in that we wish to implement is done in the following way.

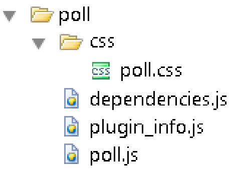
1. Create a new folder in the `compontents` folder.

2. Add the Javascript files that contain the plug-in information and the plug-in object following the defined naming convention of MindXpres.

3. Optionally, register and include the dependencies of the plug-in. (e.g. a CSS file) and place this in `dependencies.js`.

4. Register and include the plug-in with the dependency injector like we have seen with the libraries.

What we can do next is apply these steps for a `poll` plugin. We create the folder and place all the files in it. MindXpres will include this plug-in when it encounters a `poll` tag in the HTML document. We can see this in Figure 6-4.



```
poll
  css
    poll.css
  dependencies.js
  plugin_info.js
  poll.js
```

Figure 6-4: Poll folder as a plug-in

6.5 Implemented Plug-ins

As a proof of concept for our integrated communication module we implement some plug-ins for MindXpres. We select a few from the insights we gained in Chapter 3. The features we implemented are the Poll, Quiz, Piggyback, and Question list features. We describe how the plug-ins use the communication module with the publish/subscribe pattern and what the XML authoring language looks like.

6.5.1 Poll

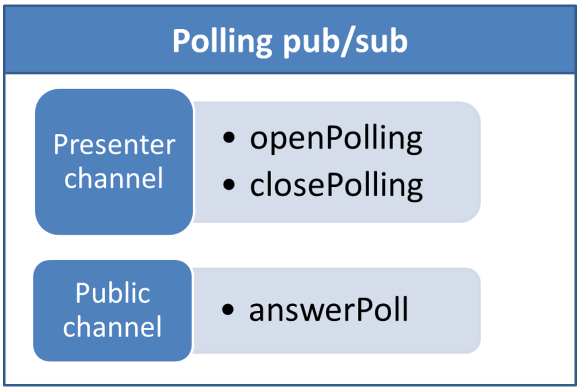


Figure 6-5: Polling pub/sub

The polling feature is one of the basic features in an ARS system. The idea behind the polling feature is that the presenter will ask a question to the audience. The audience will see the possible options to answer the question. Whenever the presenter opens the poll, the audience will have the option to make their choice. The poll will close after a certain amount of time. When the poll is closed, the results of the audience will be presented in a graph. There are many setups where we can use such a features. For instance if the presenter would like to opinion of the audience during a presentation, presenter would like to sample if the audience understood what he just said, or more.

We created a `Poll` object in Javascript that will `register` itself upon initialisation at the communication module.

Since this is one of the basic elements of giving an interactive presentation, ask a question to the audience and get an answer, it is a must for every ARS. We use the publish subscribe system as seen in Figure 6-5. The plug-in subscribes itself to two different channels, one for the presenter and one public. The presenter channel is protected by the presentation password so only authenticated presenter can publish in this channel but the audience can subscribe themselves to this channel. When the presenter is authenticated, extra options are available to open and close the poll. The open event will get published in the presenter's channel. This event is broadcast to everyone by the server. All the audience members then have the time to select an answer. This answer gets broadcast in the public channel to everyone. To give a more complete picture of the implementation, we provide the XML representation and how it is outputted in HTML5 for this feature.

The XML representation The XML document is quite simple, it has the parent tags <poll> that will contain all the information about a poll. It has two attributes that will indicate how many seconds the poll will be open and if the audience should be authenticated before selecting an option. By default, the authentication is on false and it is highly recommended to leave this on false. The XML that described a poll is translated into HTML5 by the MindXpres compiler.

```xml
<poll openSeconds="5" authentication="false">
  <question>What architectural model is used in this ARS tool?</question>
  <options>
    <option correct="false">
      <text>Peer-to-Peer model</text>
    </option>
    <option correct="true">
      <text>Client/Server model</text>
    </option>
    option correct="false">
      <text>N-tier model</text>
    </option>
  </options>
</poll>
```

Listing 6.4: XML for a Poll

6.5.2 Quiz

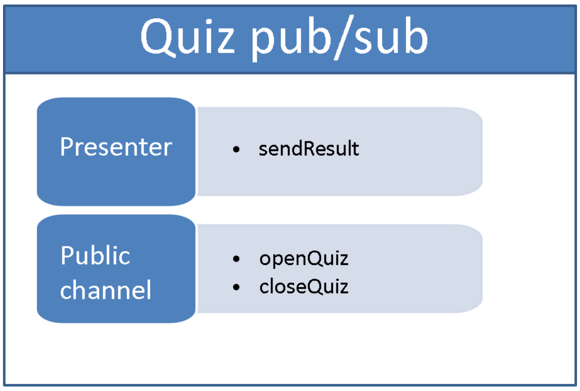


Figure 6-6: Quiz pub/sub

A few of our respondents mentioned that a quiz would be a desirable audience-oriented feature to include in a presentation tool. Therefore, we implemented this feature in our ARS prototype. When the presenter starts the quiz, an event `openQuiz` is fired from the presenter's device to the server. All the audience

members receive this event and their presentation focusses on the container (slide in this case) with the quiz.
We create a Quiz Javascript file that we place in a quiz folder. This folder is then placed under the component plug-in folder. All the dependencies (CSS) are placed in the dependencies.js and are included upon initialisation. The quiz_plug_in object, that is located in the Quiz, uses our communication module as follows. It initialise a public channel where the presenter publishes events like openQuiz and closeQuiz. When an audience member is done with the quiz, the result is sent to the presenter as seen in Figure 6-6.

6.5.3 Piggyback

The context of this feature is that every audience member has their own device. When the presenter starts his presentation, all the audience members connect to the hotspot. When the event occurs that the presenter goes from slide X to Y, it will be shared with all the connected audience members so that it also changes on their device. The audience members can choose whether or not to follow these events. When not following, it is possible to navigate backwards or forward in the presentation.

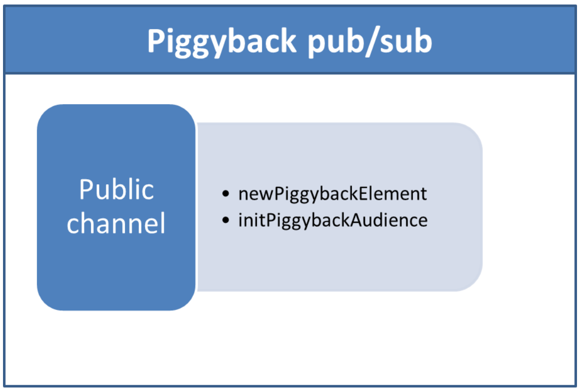


Figure 6-7: Piggyback pub/sub

This feature is implemented in the Piggyback plug-in. Upon initialisation, the plug-in creates a public channel. In this channel, the presenter publishes the event when traversing the presentation. The value behind the hash in the URL (container identification) will be shared to everyone listening device whenever this value is changed at the presenter side. When this event is received at other devices, the MindXpres core API is called to update to the new container id. Once the presenter is authenticated, the option of publishing this information or not is available. The publish/subscribe pattern is used as seen in Figure 6-7.

6.5.4 Question list

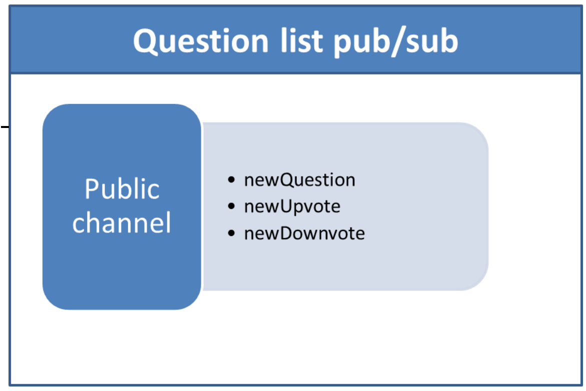


Figure 6-8: Question list pub/sub

This feature introduces a way for the audience members to ask questions about the presentation because sometimes it is hard to speak up in a big group (e.g.: due to shyness or the fear to ask a silly question). Our proposal is that audience members can see a list of questions of their colleagues. It is possible that a colleague has the same question and already submitted it, then this question can be up-voted (meaning that the score of the question is +1). If the question does not occur in the list, the user can submit their question. It is also possible for the audience members to down-vote a question (meaning the score of the question is -1). On the presenter side, it is possible to see the list when desired. The presenter is able to delete a question out of the list once it is handled.

The plug-in uses one channel where multiple events are published. The possible events that are sent by the plug-in are `newQuestion`, `upvote`, and `downvote`. When the question list is opened, the presenter can locally delete some questions. This is not broadcasted in the channel.

6.5.5 Presentation takeover

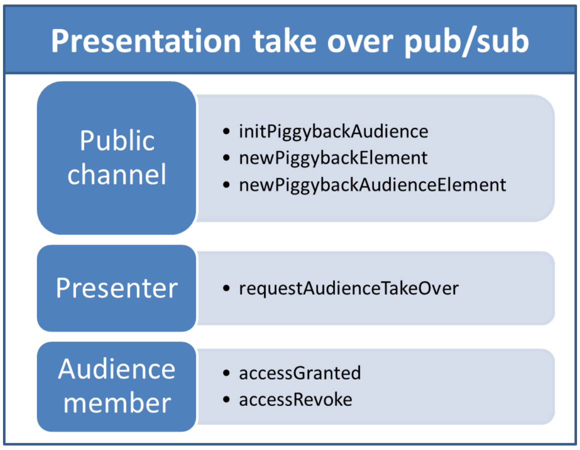


Figure 6-9: Presentation take over

This feature is basically an extension of the previously seen Piggyback plug-in. The idea is that the audience member can ask the presenter to take over the navigation of the presentation. This feature also works with the channel that is used in the piggyback plug-in. Once the presenter is authenticated, an event is published in the channel that gives the audience the option of asking to take over the presentation. When an audience member asks this, the event is send to only the presenter device. Here, a notification is given and has the option on accepting or rejecting this request. When the presenter accepts such a request, an event is sent back to the audience member (`publishToClient`). It is then allowed that this audience member publishes the value behind the hash of the URL (current information container).

6.6 Conclusion

In this chapter, we took a closer look at how our prototype is implemented. We discuss how our server uses a WiFi hotspot and a publish/subscribe pattern to support all the devices in a room during a presentation. For our communication module, we first identified the communication flows that exist in a room during a presentation. We made an abstraction of these flows so we can support them all and are included in our communication module. This modules was integrated in the MindXpres core. Lastly in this chapter, we discussed which audience-oriented features we choose from our research that is presented in Chapter 3. These are Poll, Quiz, Piggyback (with the ability to take over navigation), and Question list.

7 Use Case

In Chapter 4 we introduced the MindXpres presentation tool followed by an investigation of technical aspects to create our ARS tool. Together with all our insights that we gained from Chapter 2 and 3, we started with the implementation which is explained in Chapter 6. The time restriction implied by this thesis did not allow us to conduct another research to evaluate our implemented solution by means of user trials and such. In this chapter we walk through a scenario that spans the audience-oriented presentation process, we wish to highlight the benefits of our solution. Additionally, this chapter provides insights on how the tool works from a practical standpoint. By documenting the scenario, the reader will get a feeling of how an audience-oriented presentation in MindXpres works, and what it looks like.

7.1 The Scenario

The scenario that we define here is used during the rest of this chapter. This illustrates the goals and needs to create an audience-oriented presentation better. For the scenario, we use a situation of defending this Master thesis because it makes sense in the context. To explain some of the audience-oriented features, we add some hypothetical aspects to this situation. We envision that this audience-driven presentation will be used during the final defence of this thesis. We also include how the `Communication` object is used in a plug-in. We hope to illustrate the ease of use and extensibility of the ARS plug-ins.

7.2 Creating an Audience-Oriented Presentation

7.2.1 Creating Presentation in MindXpres

Before we start by explaining how we should add and use audience-oriented features to a presentation, we like to give a brief explanation about how we create a presentation in MindXpres. We skip possibilities that offered by MindXpres to create a presentation. The intent here is give the reader the bigger picture. For a more detailed version of possibilities, we kindly refer to [35]. As earlier established, MindXpres translates an XML language into HTML5. Every presentation has the root node called `presentation` and we start with that element in Listing 7.1. The result of these two lines is shown in Figure 7-1.

```xml
<presentation>
</presentation>
```

Listing 7.1: An empty presentation

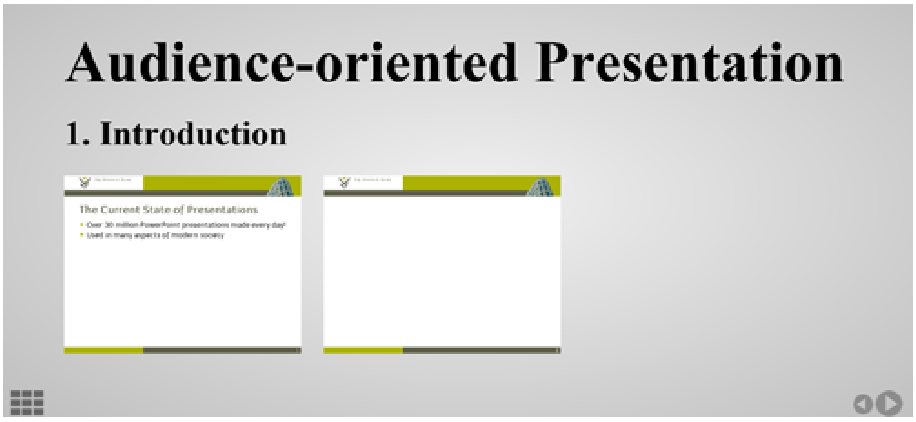
Figure 7-1: An empty presentation

As expected, this is an empty presentation without any content what so ever. Based on this empty presentation, we can start adding more content. When we wish to present this Master thesis, we may want to adopt the same structure that we used in this document. The document is divided in multiple sections, where every section addresses a new aspect of this thesis. The

content of every aspect, can be then placed in multiple containers (e.g. slides). MindXpres offers a way to change the layout by the use of themes. This way, we can modify the layout of the container with the house style of the VUB. The way it is done, is by placing a CSS file to style the container (e.g. slide.css) into the folder themes.
Placing everything we described above together results into the following XML code as seen in 7.2. The result of XML is shown in Figure 7-2.

```
<presentation>
<structured title="Audience-oriented Presentation">
    <section title="Introduction">
      <slide title="The Current State of Presentations">
        <bulletlist>
          <item> Over 30 million PowerPoint presentations made
              every day! </item>
          <item> Used in many aspects of modern society< </item>
        </bulletlist>
      </slide>
      <slide></slide>
    </section>
</presentation>
```

Listing 7.2: Presentation with content



Audience-oriented Presentation

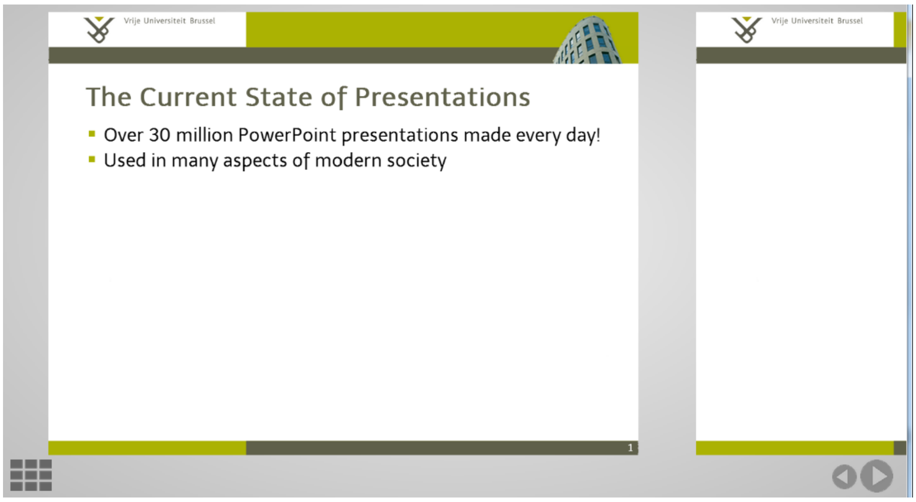
1. Introduction

The Current State of Presentations

- Over 30 million PowerPoint presentations made every day!
- Used in many aspects of modern society

Figure 7-2: Presentation with content

With the use of the ZUI of MindXpres, we can zoom in to more detail when we click on a slide. The visualisation layer of MindXpres automatically zooms in on it.



Vrije Universiteit Brussel

The Current State of Presentations

- Over 30 million PowerPoint presentations made every day!
- Used in many aspects of modern society

Figure 7-3: Zoomed in on slide

We hope that the very brief description of how to create a presentation in MindXpres, gives the reader an idea of what the possibilities are and how the MindXpres presentation tool works and looks. What follows now is how one uses the audience-oriented features with the use of our communication module. We first discuss the features and how to use them. After we established this, we discuss how the presenter should initiate the presentation to gain all the controls.

7.2.2 Use of Communication Module

The use of our communication module is quite simple. In Section 6.4, we discussed how our module is integrated in MindXpres. Imagine that we wish to add an ARS plug-in in MindXpres, for example a poll.
We start by creating a new folder in the `components` folder, which we call `poll`. To specify which tags that are offered by the plug-in, we need to add a `plugin_info.js` file. In our case, we only wish for the single tag `poll` as seen in Listing 7.3.

```
var poll_plugin_info = {
    types: ["poll"],
    path: ""
};
```

Listing 7.3: XML for a poll

Next, we add another file called `poll.js`. In this file, we define the main plug-in object. This object has to have two methods, `init` and `process`. The `init` function can be used when we wish to instantiate local plug-in resources. In our case, a counter can be declared to make all the polls have an unique id. The `process` is used to handle elements that represent information about the poll. In this function, the visualisation and initialisation of events of the poll is done. In Listing 7.4, we see a snippet of the code. The rest of the poll_plugin contains logic the create the poll. This also includes logic that has to use the Communication module. Since the object Communication is initialised in the beginning of presentation. Plug-ins can use this object as seen in Listing 7.4 where we see an initialisation of Channel and the plug-in that subscribes itself to that Channel.

The use and result of this plug-in is discussed in the following section.

```
var poll_plugin = new function () {
  this.init = function () { var pollAmount = 1; };

  this.process = function (type, elList) {
      elList.each(function (index, el) {
          //data variables
          var dataAuthenticated = $(el).data("authenticated");
          var dataQuestion = $(el).data("question");
          var dataCountdown = $(el).data("countdown");
          ...
      });
  };
  // Other poll logic
  // Subscribing to channels
  var presentationChannel = new Channel("presenterChannel", true);
  var responseChannel = new Channel("responsesPolling", false);
  Communication.subscribe(presentationChannel);
  Communication.subscribe(responseChannel);
  ...
}
```

Listing 7.4: Use of the Communication object

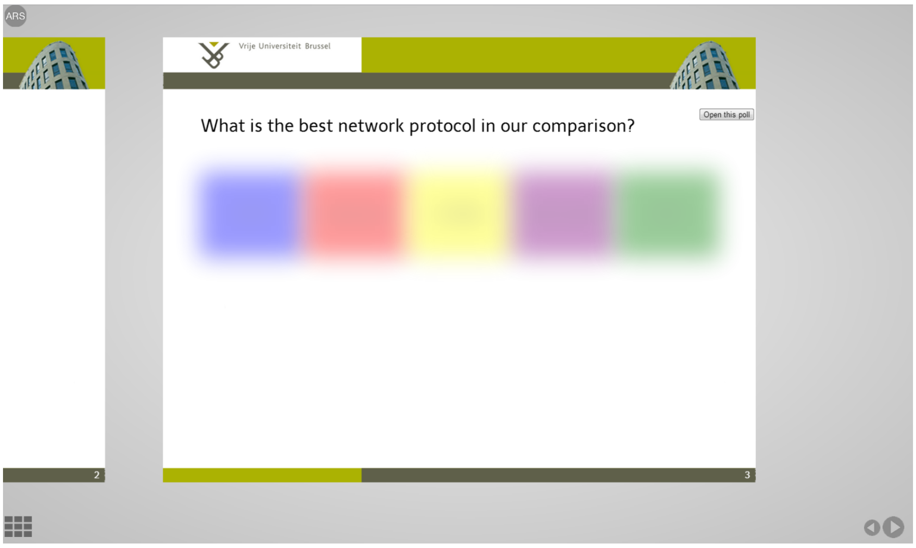
7.2.3 Use of Audience-Oriented features

7.2.3.1 Poll

During the presentation of our defence, we may want to poll our audience to see if are still understand the concepts that we are explaining or restrengthen the focus. An example of such a poll is shown in Listing 7.5. We can determine whether audience members have to be authenticated and how many seconds they have to answer the poll. The result of the code is shown in Figure 7-6. In Figure 7-4, we see the poll that is not yet opened by the presenter. Followed by Figure 7-5, where we see the opened poll.

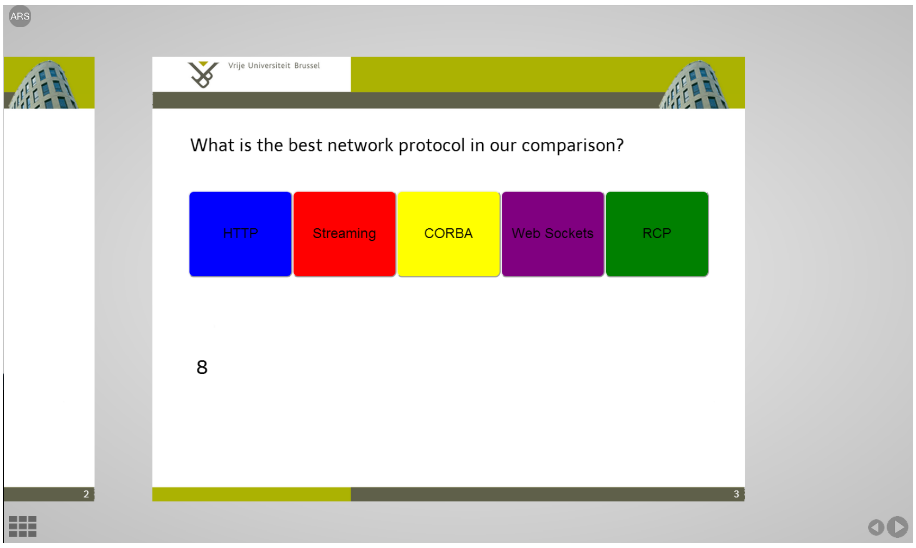
```xml
<poll openSeconds="5" authentication="false">
  <question>What architectural model is used in this ARS tool?</question>
  <options>
    <option correct="false">
      <text>HTTP</text>
    </option>
    <option correct="false">
      <text>Streaming</text>
    </option>
    <option correct="false">
      <text>Corba</text>
    </option>
    <option correct="true">
      <text>Web Sockets</text>
    </option>
    <option correct="false">
      <text>RPC</text>
    </option>
  </options>
</poll>
```

Listing 7.5: XML for a poll



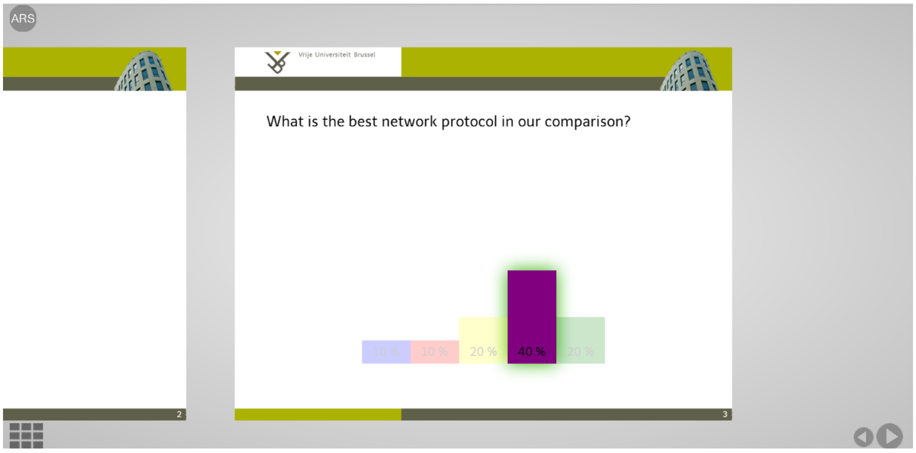
What is the best network protocol in our comparison?

Figure 7-4: Poll to open (presenter side)



What is the best network protocol in our comparison?

Figure 7-5: Audience poll



What is the best network protocol in our comparison?

Figure 7-6: Poll results

7.2.3.2 Quiz

At the end of our presentation, we might want to have a quiz with the audience. This way, we can see what the audience remembers from the presentation for instance. In the Listing 7.6, we see the XML language that generates our quiz. Note that next to text, it is also possible to give images as options. If the quiz is not yet opened by the presenter, the questions are blurred as seen in Figure 7-7. The moment a presenter opens the quiz, the audience members are able to answer and get feedback if the question is answered correctly. When the audience member is done with the quiz, the results are presented and sent to the presenter as seen in Figure 7-8.

```
<quiz authentication="false">
  <question>
    <text>What is our architecture?</text>
    <options>
      <option correct="true">
         <text>Client/server</text>
      </option>
      <option  correct="false">
        <text>P2P</text>
      </option>
      option  correct="false">
        <text>3 Tier</text>
      </option>
  </options>
  </question>
  <question>
    <text>Which of the following is a Clicker?</text>
    <options>
      <option correct="true">
         <img>images/clicker.jpg</img>
      </option>
      <option correct="false">
        <img>images/smartphone.jpg</img>
      </option>
      option correct="false">
        <img>images/tablet.jpg</img>
      </option>
  </options>
  </question>
  ...
</poll>
```

Listing 7.6: XML for a quiz

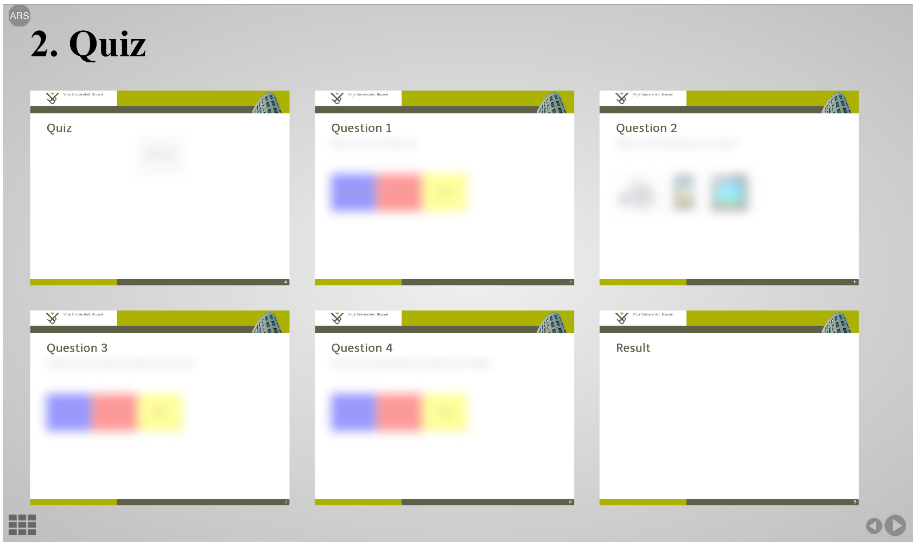


Figure 7-7: Blurred version of the quiz (not open yet)

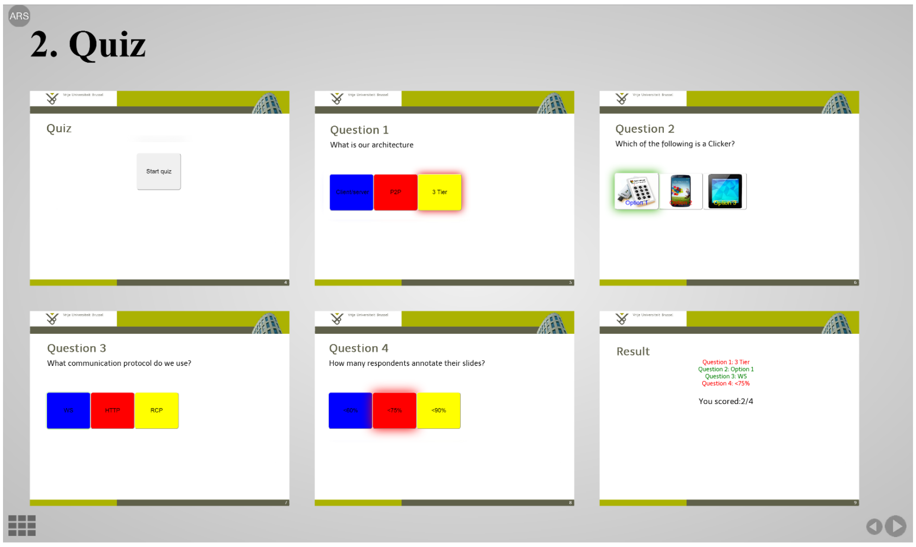


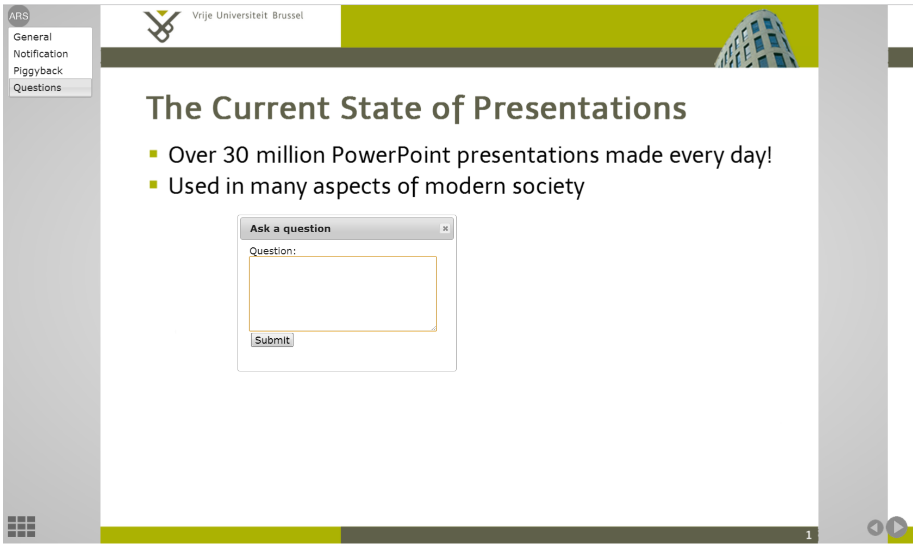
Figure 7-8: After the quiz is filled in

7.2.3.3 Question list

Imagine that during our presentation, our audience has some questions but do not wish to interrupt the presenter. With this feature, the audience can submit question that the presenter can handle when they desire. Adding this feature in our presentation is quite simple and is shown in Listing 7.7. Because of the empty nodes, the compiler now includes the question list plug-in in the presentation. It adds options in the ARS menu that the presenter and audience can use. One can now view, up-vote, or down-vote questions as seen in Figure 7-10. When an audience member would like to ask a new question, one can do this as seen in Figure 7-9.

```xml
<questionlist>true</questionlist>
```

Listing 7.7: XML for a questionlist



The Current State of Presentations

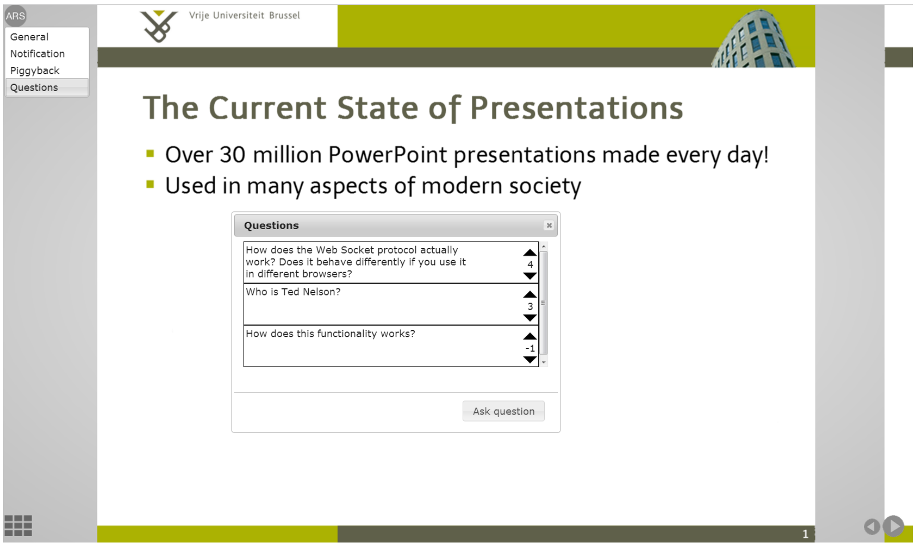
- Over 30 million PowerPoint presentations made every day!
- Used in many aspects of modern society

Ask a question

Question:

Submit

Figure 7-9: Ask a question



The Current State of Presentations

- Over 30 million PowerPoint presentations made every day!
- Used in many aspects of modern society

Questions

How does the Web Socket protocol actually work? Does it behave differently if you use it in different browsers? 4

Who is Ted Nelson? 3

How does this functionality works? -1

Ask question

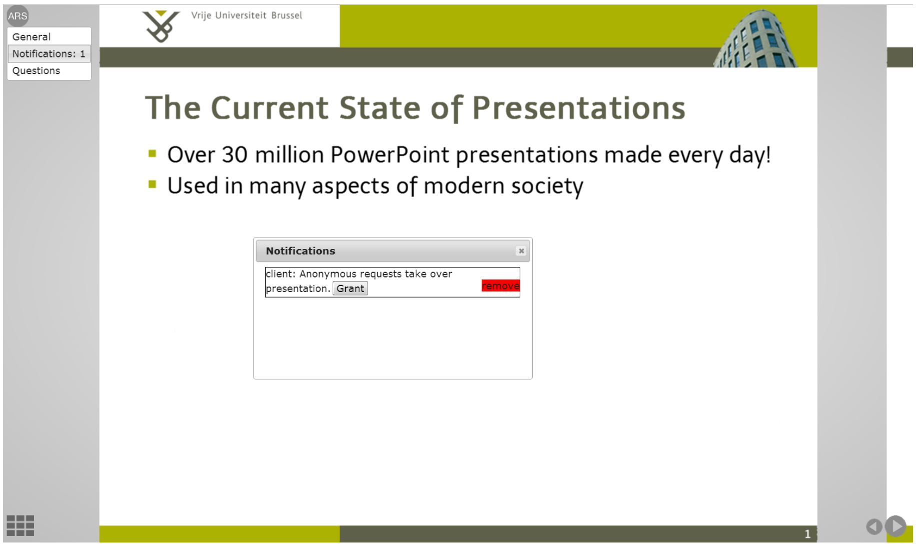
Figure 7-10: View, up-vote, down-vote questions

7.2.3.4 Piggyback

During the presentation, the presenter may wish that the audience gets a mirror of the current slide on their device. This is done with the piggyback feature. Moreover, in the case that an audience member does not understand something one may request the right to navigate the main presentation in order to illustrate his question. It is possible for the presenter to find this specific slide and lose time doing so. With piggyback, the audience may request the navigation of the presentation. When the permission is granted by the presenter, they are able to navigate to the specific slide which is updated on all listening devices. The presenter can revoke this permission whenever they desire.
In Listing 7.8, we show how we can enable this feature. The compiler will add the plug-in in the presentation that offers these described options. In Figure 7-11 and 7-12, we see what these features look like.

```xml
<piggyback>true</piggyback>
```

Listing 7.8: XML for a piggyback



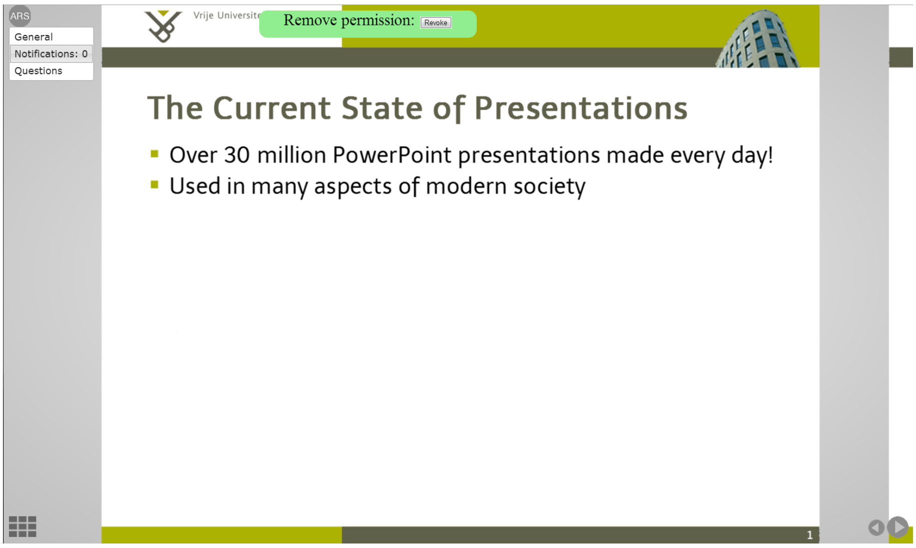
The Current State of Presentations

- Over 30 million PowerPoint presentations made every day!
- Used in many aspects of modern society

Notifications

client: Anonymous requests take over presentation.

Figure 7-11: Presenter received a request and is shown in notifications



The Current State of Presentations

- Over 30 million PowerPoint presentations made every day!
- Used in many aspects of modern society

Figure 7-12: Presenter can revoke the permission to navigate the presentation

7.2.4 Initialisation of the Presentation

In the features described above, we see that there is a separation of responsibilities. The presenter has more options than the audience. These options are granted only when a connected device is authenticated by logging in. Note that the password is also encrypted so audience members can not just read this in the HTML code. Adding the possibility to log in is also a plug-in. This is added when we add the code from Listing 7.9 in our presentation.

This plug-in is also responsible for creating the ARS menu and adding notifications as seen on Figure 7-13.

```xml
<ars>
  <password>myPresentation</password>
</ars>
```

Listing 7.9: XML for ARS initialisation

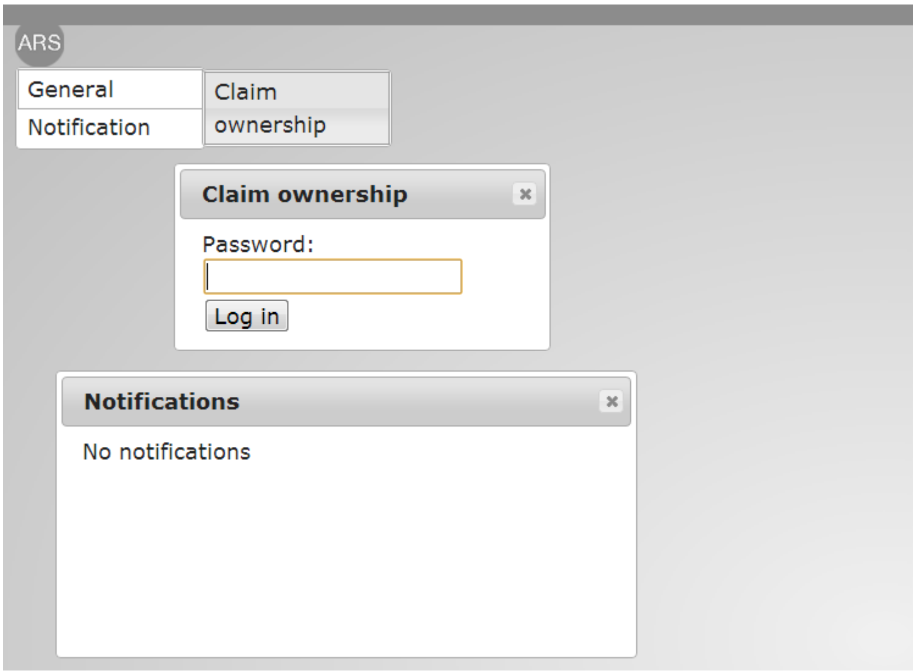


Figure 7-13: Basic options of the ARS tool upon initialisation

7.2.5 Conclusion

In this chapter, we demonstrated the usage of our extensible ARS tool in MindXpres by means of a simple scenario. We have shown how a plug-in can be made that uses the `Communication` module. We have shown how easy it is for the presenter to make use of our plug-ins. The reader must also note that what we demonstrate in this use case represents just a limited showcase of what is possible in our ARS tool.

8 Conclusion

Allow us to recapitulate a number of contributions that have been made in this thesis. Additionally, we outline and discuss some future directions for our extensible ARS tool in the MindXpres presentation tool.

In this thesis, we take the position that the audience is equally important (if not more) in a presentation. This position is consistent with many student-oriented pedagogical and didactic theories. However, in the common presentation tools, no features are present to involve the audience members. Our solution invites and encourages all members of the audience to participate in a dialogue with the presenter and with other audience members during and after the presentation. Participation implies communication between everyone involved in a presentation. In order to improve upon the common presentation tools with audience-oriented features, we identified and characterised the communication flows that can or should occur during and after a presentation. Our model of communication flows and broadcasting groups became our conceptual framework to realise a communication module. This module can be employed as a base to implement audience-oriented features in a presentation tool.

Our initial literature study has familiarised us with existing state-of-the-art ARS solutions and allowed us to learn about their strengths and weaknesses. To supplement this literature study, we also conducted our own preliminary survey to gather and identify people's perceptions, opinions and aspirations about ARSs. We concluded that most professional presenters and audience members favour audience participation but are still quite sceptical about some innovative features. This scepticism mainly originates from people's unfamiliarity with ARSs due to the dominance of common presenter-oriented slideware.

We integrated our communication module into MindXpres, an innovative, non-sequential presentation tool. We appreciate in particular MindXpres' asset to have a plug-in architecture. Furthermore, MindXpres compiles a LaTeX -like language into HTML5 which eases the way of distribution among a myriad of devices with the most divers operating systems. We chose our communication module to work with a central server and Web Sockets communication carrier. The client/server architecture allowed us to centralise all the logic behind the server including the management of all participants. It also facilitates the use of more modalities such as digital pens or other input devices. After investigating various communication protocols we opted for Web Sockets due to their low overhead and latency.
Finally, we applied all our acquired insights on the construction of an API to develop audience-oriented features with an eye on extensibility. As a proof of concept, we implemented some of those features as plug-ins for MindXpres.

8.1 Contributions

The contributions of this dissertation are threefold.

- A literature study was performed in order to identify problematic areas in ARS tools that offer audience-oriented features to a presentation. Additionally, we describe some more advanced educational ARS that are not yet discussed in detail in the available literature.
- In our preliminary qualitative research, we gathered insights in people's perceptions, opinions and aspirations about ARS in greater detail than what was found during our literature study.
- We constructed a conceptual framework that describes and structures all possible communication flows during and after a presentation.
- Based on this conceptual framework, our main contribution is the extension of MindXpres with a communication module that can be used to develop audience-oriented features. This conceptual framework ensures that programmers can use this module for all future development. It's potential was demonstrated by implementing plug-ins for a variety of audience-oriented features.

8.2 Future Work

In this section we suggest and discuss possible expansions and improvements of the results presented in this thesis.

8.2.1 Server

At the moment, the server implementation is fairly basic. We implemented a publish/subscribe pattern to realise event-driven communication. Currently, all the sessions are stored at run time. When our system is used more extensively with many more clients, it becomes desirable to ease the server's workload by transferring it to a more dedicated application. A Redis database[1] would be a candidate key-value store to mimic the current behaviour of the server.

8.2.2 Plug-ins

From literature study and from our own qualitative study, we know that many more plug-ins are found worthwhile to be implemented. Due to time restrictions implied by this thesis, we focused on finding a broad and solid foundation for developing audience-oriented features in presentations.

8.2.3 Evaluation

As mentioned earlier, the study we conducted to find perceptions, opinions and aspirations about ARS was preliminary to the development of audience-oriented features in presentation tools. For each of the implemented features, we should conduct a new qualitative research to test and asses it with a real audience. This way, we can gain additional insights into the necessities of a complete audience-oriented presentation experience.

[1] http://redis.io

Bibliography

[1] L. Abrahamson. A Brief History of Networked Classrooms: Effects, Cases, Pedagogy, and Implications. *Audience response systems in higher education: Applications and cases*, 7:1–25, October 2006.

[2] M. Ayu, K. Taylor, and T. Mantoro. Active learning: Engaging students in the classroom using mobile phones. In *Proceedings of Symposium on Industrial Electronics and Applications*, pages 711 – 715, Kuala Lumpur, Malaysia, January 2009.

[3] M. Baqer and A. Kamal. ACRS: An Advanced Classroom Response System Architecture for Learning Enhancement. In *Proceedings of Second International Conference on Developments in eSystems Engineering (DESE)*, pages 287–291, Washington, DC, USA, December 2009.

[4] M. Bauwens. Peer to peer manifesto. http://p2pfoundation.net/Manifesto, 2005. retrieved at 2013-06-10.

[5] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret. The world-wide web. *Communications of the ACM*, 37(8):76–82, August 1994.

[6] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext transfer protocol–HTTP/1.0. Internet RFC n°1945, May 1996.

[7] W. C. Bessler. The Effectiveness of an Electronic Student Response System in Teaching Biology to the Non-Major Utilizing Nine Group-Paced, Linear Programs. Unpublished doctoral dissertation, 1969.

[8] V. Bush. As We May Think. *Atlantic Monthly*, 176(1):101–108, July 1945.

[9] J. E. Caldwell. Clickers in the large classroom: Current research and best-practice tips. *CBE-Life Sciences Education*, 6(1):9–20, March 2007.

[10] A. Cross, E. Cutrell, and W. Thies. Low-cost Audience Polling Using Computer Vision. In *Proceedings of UIST '12*, pages 45–54, New York, NY, USA, October 2012.

[11] J. C. De Gagne. The impact of clickers in nursing education: A review of literature. *Nurse Education Today*, 31(8):e34–e40, December 2011.

[12] e. a. Douglas T. Owens. The Connected Algebra Classroom: A Randomized Control Trial. In *Proceedings of International Congress on Mathematical Education (ICME)*, pages 24–38, Monterrey, Mexico, July 2008.

[13] I. Fette and A. Melnikov. The websocket protocol. Internet RFC n°6455, Dec 2011.

[14] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol–HTTP/1.1. Internet RFC n°2616, June 1999.

[15] C. Fies and J. Marshall. Classroom response systems: A review of the literature. *Journal of Science Education and Technology*, 15(1):101–109, March 2006.

[16] H. P. Froehlich. What about classroom Communicators? *Audiovisual communication review*, 11(3):19–26, May 1963.

[17] C. Habel. VotApedia for student engagement in academic integrity education. *Education Research Group of Adelaide*, 2(1):15–25, February 2011.

[18] Ipsos MediaCT Germany. Mobile Internet & Smartphone Adoption. http://services.google.com/fh/files/blogs/FinalMobileInternetSmartphoneAdoptionInsights2011v3.pdf, 2012. retrieved at 2013-03-08.

[19] M. Jagar, J. Petrovic, and P. Pale. AuResS: The audience response system. In *Proceedings of Electronical System and Information Process*, pages 171–174, Zagreb, Croatia, September 2012.

[20] M. A. A. Jonathan Mamngkiang Mantikayan. Exploring Audience Response System: Impacts in Learning Institutions. In *Proceedings of ICT4M*, pages A13 – A17, Jakarta, Indonesia, December 2010.

[21] E. Judson et al. Learning from past and present: Electronic response systems in college lecture halls. *Journal of Computers in Mathematics and Science Teaching*, 21(2):167–181, April 2002.

[22] R. H. Kay and A. LeSage. Examining the benefits and challenges of using audience response systems: A review of the literature. *Computers & Education*, 53(3):819–827, November 2009.

[23] F. F.-H. N. Keng Siau, Hong Sheng. Use of a Classroom Response System to Enhance Classroom Interactivity. *IEEE Transactions on Education*, 49.3(26):398–403, August 2006.

[24] S. L. Lam, K. Wong, J. Mohan, D. Xu, and P. Lam. Classroom communication on mobile phones–first experiences with web-based clicker system. In *Proceedings of Ascilite 2011 Hobart*, pages 763–777, Hobart Tasmania, Australia, December 2011.

[25] S. Loreto, P. Saint-Andre, S. Salsano, and G. Wilkins. Known issues and best practices for the use of long polling and streaming in bidirectional http. Internet RFC n°6202, October 2011.

[26] P. Lowry, N. Romano, and R. Guthrie. Explaining and Predicting Outcomes of Large Classrooms Using Audience Response Systems. In *Proceedings of Hawaii International Conference on System Sciences*, volume 1, pages 4c–4c, Hawaii, USA, December 2006.

[27] P. Lubbers and F. Greco. Html5 web sockets: A quantum leap in scalability for the web. url = http://websocket.org/quantum.html, 2010.

[28] H. R. Maier. Student participation in lectures using mobile phones. In *Proceedings of the 20th Australian Association for Engineering Education Conference*, volume 1, pages 43–48, Adelaide, Australia, December 2009.

[29] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 18:50–60, 1947.

[30] T. Mantoro, M. Ayu, E. Habul, and A. Khasanah. Survnvote: A free web based audience response system to support interactivity in the classroom. In *Proceedings of IEEE Conference on Open Systems (ICOS)*, pages 34–39, Kuala Lumpur, Malaysia, December 2010.

[31] C. Markett, I. A. Sánchez, S. Weber, and B. Tangney. Using short message service to encourage interactivity in the classroom. *Computers & Education*, 46(3):280–293, 2006.

[32] T. Nelson. *Getting it out of our system. In Information Retrieval: A Critical Review. G. Schechter.* Thompson Books, Washington, 1967.

[33] M. C. Norrie. An Extended Entity-Relationship Approach to Data Management in Object-Oriented Systems. In *Proceedings of ER '93, 12th International Conference on the Entity-Relationship Approach*, pages 390–401, Arlington, USA, December 1993.

[34] I. Parker. Absolute PowerPoint: Can a Software Package Edit Our Thoughts? *The New Yorker*, pages 76–87, May 2001.

[35] R. Roels and B. Signer. MindXpres - An Extensible Content-driven Cross-Media Presentation Tool. In *Proceedings of the 27th BCS Conference on Human Computer Interaction (HCI 2013)*, London, United Kingdom, 2013.

[36] A. Russell. Comet: Low latency data for browsers. http://infrequently.org/wp-content/LowLatencyData.pdf, 2006. retrieved at 2013-06-20.

[37] R. Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Proceedings of First International Conference on Peer-to-Peer Computing*, pages 101–102, Munich, Germany, 2001.

[38] B. Signer and M. C. Norrie. As We May Link: A General Metamodel for Hypermedia Systems. In *Proceedings of ER 2007, 26th International Conference on Conceptual Modeling*, pages 359–374, Auckland, New Zealand, November 2007.

[39] M. Sinclair, H. Wideman, R. Owston, M. Fioroni, and L. Kallish. TI-Navigator Study Final Report. http://education.ti.com/sites/UK/downloads/pdf/References/TI-Nspire/Sinclair_TINavigator_study_final_report_Oct09.pdf, 2009. retrieved at 2013-06-04.

[40] T. Zarraonandia, I. Aedo, P. Díaz, and A. Montero. An augmented lecture feedback system to support learner and teacher communication. *British Journal of Educational Technology*, 44(4):616–628, July 2013.