



TabletPoint: A Pen and Tablet-based Presentation Solution

Graduation thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Applied Sciences and Engineering: Computer Science

Chunxiang Yan

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

Academic year 2015-2016



Abstract

Presentations have become a major tool for presenting ideas, knowledge and solutions. Since the invention of the slide projector, the invention of new presentation tools has never stopped. Among them, Microsoft PowerPoint which is launched in year 1990 has become the most popular presentation tool today, being used by approximately 120 million users. Everyday, nearly 30 million PowerPoint presentations are created and the number keeps growing.

Inherited from the slide projector, Microsoft PowerPoint and many other alternatives, follow the linear sequence of presenting information. Despite the need for making better presentations, very few innovation has been made for over the last two decades. The presenters are often limited by the structure of their presentation, the physical position of the hardware infrustacture and the methods for controlling their presentations. There are innovations such as Prezi and MindXpres which support creating and presenting information in a non-linear way. However, the restrictions during presenting remain. Most of the presenters have to walk back to their computer for navigating or use a simple remote control which often only offers limited features.

The development of PaperPoint, a paper-based presentation and interactive paper prototyping tool, provides a new experience of presenting with a digital pen and prepared printed handouts. With PaperPoint, PowerPoint users are able to freely move around in the room, preview and navigate through slides as well as to annotate in the real time. In the meantime, the rapid development of tablet PC and stylus pen¹ shows a great amount of possibilities. The affordability of light-weight tablets and accurate stylus pen means that we could build a system which keeps the advantages of PaperPoint and improve some of its limitations.

In this thesis, we propose a tool which is based on a newly released Tablet PC and Stylus pen, TabletPoint. Since PowerPoint is such a popular presentation tool, our TabletPoint prototype is designed as an extensible PowerPoint controller. It shows the possibilities of extending the functions of PowerPoint, building up a multi-user system and extendability for a variety of other presentation tools. TabletPoint inherits the majority of advantages from PaperPoint. It addresses several limitations of PaperPoint such as lacking animations and requiring the preparation of the handouts. Furthermore, it

¹A small pen-shaped instrument that is used to input commands to a computer screen, mobile device or graphics tablet. See in [https://en.wikipedia.org/wiki/Stylus_\(computing\)](https://en.wikipedia.org/wiki/Stylus_(computing))

provides several useful functions to support better annotation. For example, it aids the presenter to draw perfect shapes (e.g. straight lines or rectangles) and provides the feature of saving and reusing the drawn annotations. Last but not least, the client-server architecture makes the system very flexible for future extensions.

Declaration of Originality

I hereby declare that this thesis was entirely my own work and that any additional sources of information have been duly cited. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this thesis has not been submitted for a higher degree to any other University or Institution.

Acknowledgements

I would never be able to finish this thesis without the kindness help and support of many people. First, I would express my gratitude to my promoter Prof. Dr. Beat Signer, for giving me the opportunity to work on such an interesting topic and for all the brilliant inspiration. I also want to thank my advisor Reinout Roels for all the proofreading and advices. Last but not least, I am grateful for all the company and support from my family and friends, life could get much less interesting without them being around.

Contents

1 Introduction

1.1	Problem Statement	2
1.1.1	Mobility	3
1.1.2	Slideshow	4
1.1.3	Navigation	4
1.1.4	Preview	5
1.1.5	Annotation	5
1.2	Objects	6
1.3	Structure of the Thesis	7

2 Literature Review

2.1	Classroom 2000	10
2.2	Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses	11
2.3	Digital Ink in Lecture Presentation	12
2.4	Speech, Ink and Slides: The Interaction of Content Channels	14
2.5	A Study of Diagrammatic Ink in Lecture	16
2.6	Ubiquitous Presenter	17
2.7	Using Pen-based Computers Across the Computer Science Curriculum	19
2.8	Livenotes	20
2.9	PaperPoint	21
2.10	Classroom Technology for Active Learning	22
2.11	Mobile Phones for Active Learning	23
2.12	INK-12	25
2.13	Increased Engagement in Code Review Class	26
2.14	Office Remote	27
2.15	MindXpres: An Extensible Content-driven Cross-Media Presentation Platform	28
2.16	Conclusion	29
2.16.1	Classroom Presenter	30

2.16.2	PaperPoint	31
2.16.3	Office Remote	31
2.16.4	TabletPoint	32
3	System Introduction	
3.1	System Requirements	33
3.2	System Functionality	34
3.2.1	Server Side	34
3.2.2	Client Side	34
4	Implementation of TabletPoint	
4.1	Research Training	44
4.1.1	Digital Pen	44
4.1.2	Interface Implementation	45
4.1.3	Decoding Bytes	45
4.2	Server Implementation	46
4.2.1	Bluetooth Functionality	46
4.2.2	Presentation Streaming	47
4.2.3	PowerPoint Interoperability	48
4.2.4	Communication	49
4.3	Client Implementation	50
4.3.1	Threads	50
4.3.2	Streaming View Implementation	51
4.3.3	Stamps List Implementation	52
4.4	Functions	53
4.4.1	Stroke Synchronisation	53
4.4.2	Loading and Saving	53
4.4.3	Special Interaction	54
4.5	Conclusion	54
5	Use Cases and Usability Evaluation	
5.1	Use Cases	55
5.1.1	First Scenario	56
5.1.2	Second Scenario	57
5.2	Usability Evaluation	59
5.2.1	Procedure	59
5.2.2	Items and Scoring	59
5.2.3	Test Results	60
5.2.4	Conclusion	62
6	Conclusion	

6.1	Contributions	64
6.2	Limitations	64
6.3	Future Work	65
6.3.1	Desktop Operation	65
6.3.2	Universal Compatibility	65
6.3.3	Multi-user Support	65

A Appendix

1

Introduction

Presentations have become a major tool for presenting ideas, solutions and knowledge. As the need for better presentation tools grew, many commercial presentation tools have emerged. Among them, some of the most used commercial tools are Microsoft PowerPoint¹, Apple Keynote², Apache OpenOffice³, Prezi⁴ and the relatively new Google Slides⁵. The research area has been active as well. Academic work such as Classroom Presenter (Richard Anderson, 2004), PaperPoint (Signer & Norrie, 2007), and MindXpres (Roels & Signer, 2014) all investigate the use of alternative approaches for interacting with presentations. Although all presentation tools, both academic and commercial, have their advantages, Microsoft PowerPoint is without doubt the most popular choice (Parker, 2001). It is estimated that nearly 95% of the slides are created and presented via Microsoft PowerPoint. The approximate number of PowerPoint users has reached 120 million. Most importantly, it can be easily observed that PowerPoint dominates the educational domain. Even with such a great number of users, Microsoft PowerPoint has not been improved much recently in comparison to other well-known software, and of

¹<https://products.office.com/en-us/powerpoint>

²<http://www.apple.com/mac/keynote/>

³<https://www.openoffice.org/>

⁴<https://prezi.com/>

⁵<https://www.google.com/slides/about/>

course, it has not really matched the expectation of the users (Taylor, 2007). From the the invention of overhead projector to the early era of PCs, presenters have been restricted by their hardware setup both physically and in terms of functionality. Commercial and academic efforts have been made to break free from these restrictions. One example would be a simple wireless remote. Nonetheless, there are still many issues to tackle.

One solution for interacting with presentations, the one this thesis is mostly concerned, is the use of (digital) pen input. The Telautograph is considered the first attempt to translate handwriting into electronic information (Gray, 1888). After a century of evolution, Stylator came into existence as the very first digitizer tablet (Dimond, 1957). Soon such systems were wildly used to conduct heavy digital graphical work due to its accuracy and performance in comparison to the use of a mouse (Kim, Kim, Yoon, Jung, & Hwan, 2015). Nowadays, Samsung⁶ released several Android-based tablets⁷ with S-Pen support. The accuracy and palm rejection feature of the S-Pen stylus, the light weight and performance of the tablet, together with the blooming Android developer community unveils enormous opportunities.

There is several concepts which need to be clarified here. By the word *slideshow*, we refer to the view which the audience sees on the screen. When we discuss the mobility of the presenter, we mean the ability to walk around in the room while interact with the presenting system. The physical constraint which we will mention is that the user has to be around the place where the presenting computer is to interact with the presentation software.

1.1 Problem Statement

As discussed before, PowerPoint has become the most used presentation software despite all its limitations. Therefore, the discussion of this thesis focuses on how presenters interact with PowerPoint during the presentation. Additionally, we would also like to put our attention to educational use since it is one of the most relevant subjects in our daily life. In this section we discuss a list of the major problems of PowerPoint. The other problems such as "linear content structure" are beyond our reach (Parker, 2004).

At this moment, Microsoft PowerPoint still works as a digital version of the slide projector. There are of course many new features, but the core

⁶<http://www.samsung.com/us/>

⁷<https://www.android.com/>

of PowerPoint such as the content structure and the interaction method is never changed. This imitations of the physical world can largely improve the learnability of the system for the users with little computer experience. However, since computers have become an essential equipment of every family (Subrahmanyam, Kraut, Greenfield, & Gross, 2000), people have improved a lot in terms of their ability to use difficult computer software. Therefore, instead of mimicking the physical world to increase the system learnability, we should also consider the possibility of exploiting the advantages of digital media.

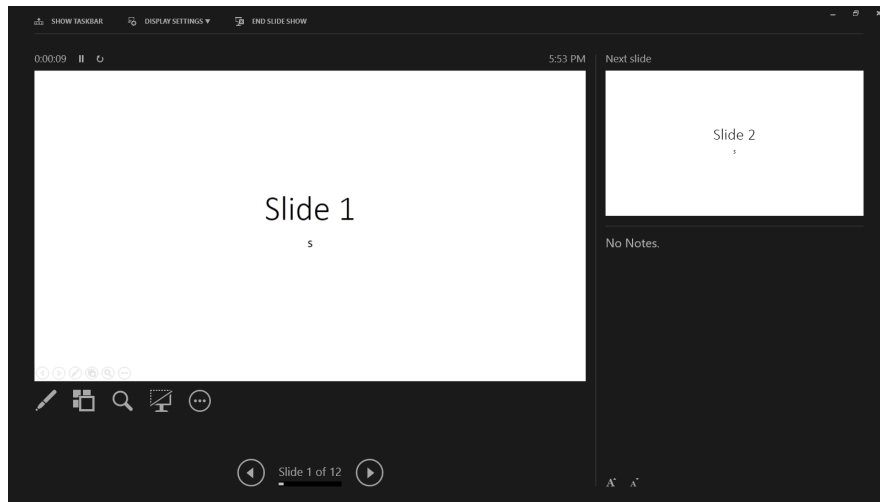


Figure 1.1: The PowerPoint Control Panel

1.1.1 Mobility

In Figure 1.1 we show the PowerPoint control panel, a view the presenter sees while giving a presentation. The PowerPoint control panel offers a great deal of functions. These functions would be considered sufficient for a normal user. The user has the ability to navigate back and forth, annotate with their mouse or touchpad, preview the next slide and see the current slides content on the computer screen. All necessary functions for giving a linear presentation are met. Yet a laptop fails when it comes to mobility due to the physical restrictions. The user has very little choice except standing next to their computer when accessing all functions provided by PowerPoint. The following discussion considers mobility as a priority. We would like to explore what a simple presentation remote control fails to offer while ensuring mobility of the presenter. The majority of the problems which we target at is supported by a previous study (Reuss, Signer, & Norrie, 2008).

1.1.2 Slideshow

During a presentation, the presenter is usually expected to face the audience and make eye contact. However, it is quite common to see the presenters turn away frequently to look at their computer screen or the presenting display. In certain cases it can get even worse if the presenter turns backwards and stares at their slideshow during the entire presentation while completely forgetting to make eye contact with the audience. Such an embarrassing situation might be partially due to a lack of experience. However, it is also *based* on a lack of a second display showing the current slide. We could help the presenters by providing a mobile display showing the current slideshow.

1.1.3 Navigation

Another issue occurs when the user has the need to navigate among all their slides. The issues is obviously observed when the presenter needs to rapidly navigate to one of their slides using a remote. If the presenter navigates per slide, they might have to jump over a few slides unveiling some of the content which is not yet relevant. It is also time consuming when the presenter has to navigate over a relatively large range of slides. Note that the latest version of PowerPoint supports previewing and navigating all the content by clicking one button on the control panel as shown in Figure 1.2. However, let alone the fact that not many users are aware of such function, it is also impossible to access using a simple presentation remote control.

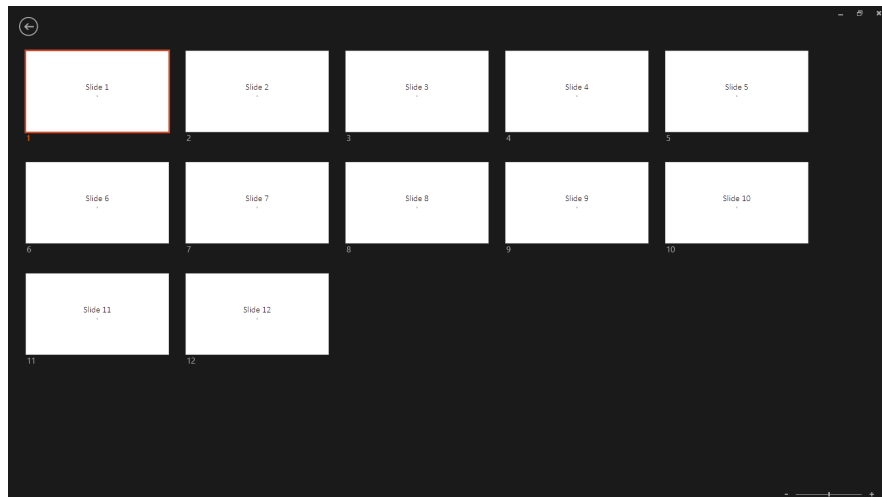


Figure 1.2: The PowerPoint Preview Panel

1.1.4 Preview

In terms of quick navigation, knowing the target slide is an essential requirement. Presenters barely remember the exact content and position of their slides. Thus, providing previews of all slides makes an essential difference. Provided with the content and position of every slides, the user could easily decide which slide is the one they want to navigate to. This could again be achieved by making use of a mobile device's display.

1.1.5 Annotation

The final issue would be the insufficient support for annotations. Users tend to use large chunks of text and often put several key points on the same page (Gaskins, 2007). The issue therefore would be that the audience could easily get lost in the context struggling to find the content the presenter is currently talking about. Currently what people would normally do is pointing to the specific content with a laser pointer. However the laser pointer is powerless when it comes to annotation which is also a very demanding feature.

Research has shown that the use of digital ink can be helpful in various scenarios (Richard Anderson, 2004). However, making annotations is impossible with a simple remote control. Luckily, PowerPoint's default annotation functionality is already quite powerful for us to utilise. It offers laser light, different colours of ink and an eraser. Yet, when considering mobility, it shows its weakness since the presenter has to be at the computer which is used for the presentation. Furthermore, the accuracy of annotations created with a touchpad or a mouse is never guaranteed. It is extremely difficult to produce complex graphics using a touchpad or even a mouse. Such process would never be considered possible in a formal presentation without a great amount of practice (Apte & Kimura, 1993). Most importantly, the ability to exploit the natural advantages of digital ink, namely replicability, remains an interesting challenge. If we could reuse digital ink in an efficient way, it would be a benefit for lecturing.

In the context of teaching practice, the situation where the same question is asked multiple times can be tiring. It is impossible for a teacher to prepare the answers for all possible questions. Meanwhile, creating and modifying content where graphics and formulas are heavily used in PowerPoint can be cumbersome (Shallcross & Harrison, 2007). Having the freedom of making and reproducing certain drawings could come in handy in this case. Imagine the case when a chemistry teacher draws dozens of formulas in one lecture and has the need to reproduce some of them in another lecture. Simply saving all

drawings would produce a mess in their slides. On the other hand, redrawing all formulas would be a waste of energy and time. Most importantly, they will have to make some trivial adjustments when facing different audiences which could never be done easily with PowerPoint only.

1.2 Objects

Despite some of the arguments against using PowerPoint in lectures (Creed, 1998), Microsoft PowerPoint earned its way into the teaching industry in a very short time. The natural advantage of digital presentation material is that it allows careful preparation, easy replication, and high reusability (Bligh, 1998).

In this thesis, we try to solve the problems discussed above by introducing the use of tablet PCs in combination with stylus pen while keeping every bit of the functionality PowerPoint provides. Our system is named TabletPoint in the light of an existing paper-based presentation tool called PaperPoint. We developed a set of software to utilise the mobility of a tablet with a stylus pen and a PC installed with PowerPoint. The client-side software runs on the tablet while the server-side software runs on the PC. Our solution was realised as an extension for the PowerPoint presentation tool. In this way, the user does not have to learn to use a brand new system. They could create their slides with PowerPoint in the method they are used to. All their existing presentation files would also be usable without any extra efforts. Moreover, by utilising the existing functionality provided by PowerPoint, we could focus on developing the features we are particularly interested in.

During the development of our system, special care has been taken to ensure the extensibility of the system. While designing our system, we kept in mind the compatibility for multi-user as well as multi-platform scenarios. The connection between the client and the server is established over Bluetooth. This choice has been made because most university networks have some restrictions on the local network communication. If we choose to use Wi-Fi network, we might have to use the laptop as a hotspot to create our own Wi-Fi network which results in losing the Internet connection. By using Bluetooth we allow the PC to communicate with the mobile device while maintaining its connection to the Internet.

1.3 Structure of the Thesis

In Chapter 2, we summarise and discuss the existing effort in both making better presentation tools and bringing pen-based system into use. By such discussion we could further explore the meaningfulness of our objectives and the possible innovations we could bring.

In Chapter 3, we discuss the design of our system. We will illustrate our system's functionality by presenting several use cases.

In Chapter 4, we first describe the research training we did. Then, we explain our architecture and design decision made on the server and on the client side. The server side is responsible for exploiting PowerPoint functions while providing data to the client side. The client side is responsible for translating the user interactions into combinations of instructions which will be executed on the server side.

In Chapter 5, we firstly go through two working scenarios of the system. Then we discuss the usability evaluations we conducted. The feedback from the users gives us more enlightenment.

Finally, in Chapter 6, we conclude our work and analyse the remaining problems. Then, based on the feedback we gathered from the usability test, we discuss possible future work.

2

Literature Review

In this chapter, we discuss the literature review we have conducted. Since the research of this thesis is largely inspired by PaperPoint (Signer & Norrie, 2007), we discuss this tool and its related work in detail. Furthermore, Microsoft released their own commercial solution, Office Remote¹. It is a well designed software from which we could find useful features but also limitations we could improve. The research efforts to bring stylus together with tablet computers for teaching and active learning are good references for the design of our system and future ideas (Abowd, 1999; Williams & Steinweg, 2009). Thus, to pave our path through all the previous academic work, we reviewed articles of the following types (including several exceptions):

- The introduction of several systems developed with similar purpose: *making better presentation tools with annotation support*.
- The studies based on the tools we reviewed.

After the reviews, we summarise all the work we reviewed. Making a good comparison among them helps in determining the most reasonable directions to explore.

¹<http://research.microsoft.com/en-us/projects/officeremote/>

2.1 Classroom 2000: A System for Capturing and Accessing Multimedia Classroom Experiences

In the article "Classroom 2000: A system for Capturing and Accessing" (Abowd, 1999), Abowd et al. made one of the very first attempts to capture the lecture experience by introducing a whiteboard based ZenPad system² in their research project named Classroom 2000 (Abowd et al., 1997). ZenPad allowed the instructor to present and annotate slides. They have seen a lecture as a collaborative multimedia content authoring session and attempted to capture such sessions with the purpose of generating content which assists students in reviewing the content at a later point in time.

Their physical setup was based on an electronic whiteboard (Liveboard). They developed their Java-based client-server system, ZenPad for the whiteboard. By recording the lecture audio, video, slides, and annotation, they could offer the students later access to the content via a web-based interface. Their interface design is minimal, with two important features which were not available commercially at the time of their writing:

- Producing artefacts which are web accessible.
- A log of important operations with timestamp.

The first feature enables a browse-only mode where the student can read the content via accessing several HTML interfaces. The second feature enables a later integration with recorded digital audio or video. This feature resonates well with the studies we would review later. The potential of combining video, audio and lecture ink helps viewers to analyse the content even without its original context (R. J. Anderson, Hoyer, Wolfman, & Anderson, 2004; R. Anderson et al., 2005).

They provide different interfaces to view the content, but all of them allow jumping from visual content to the related audio content. ZenPad also provides an interactive timeline for a given lecture. The timeline ranged from the beginning to the end of the lecture, marked with two kinds of events: creation of new slide and URL visiting via browser. Moreover, each video in ZenPad is mapped to a series of images which helps navigating more precisely through the video.

²<http://www.cc.gatech.edu/fce/c2000>

To conclude, they stated that the efforts on ZenPad could result in a larger electronic whiteboard. The alternative interfaces like tablet PCs or PDAs could be provided for students to capture notes. They were interested in how speech and vision technology can improve comprehension during content browsing within and across the lectures.

2.2 Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses

In the article "Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses", Anderson et al. presented their design of tools (Richard Anderson, 2004) which enhance interaction among participants in a lecture. Their Classroom Presenter system lays a solid foundation for research towards a pen-based and tablet-based classroom environment. They pointed out two directions to enhance interaction: provide more flexibility to the slide content during a lecture and support new forms of interaction between student and teacher. Classroom Presenter focuses on the first one. It runs on a pen-based tablet PC allowing the presenter to draw on the projected slides. The image of the ink and slide can be further viewed by the students on either a public display or a personal device.

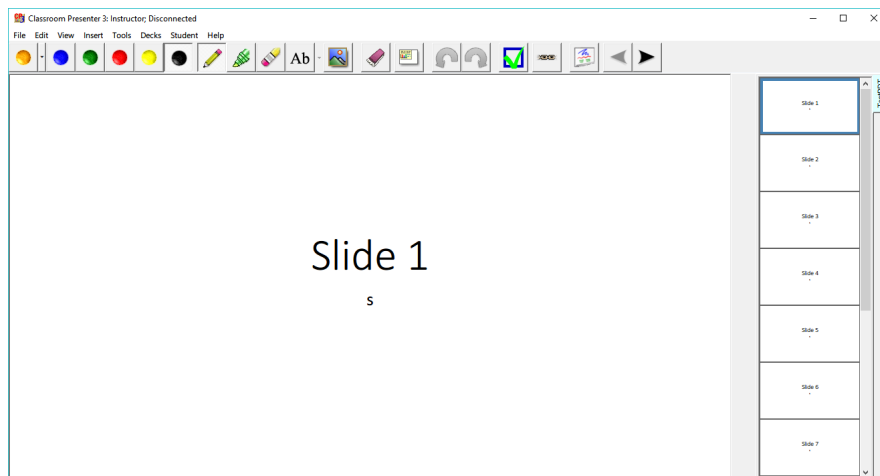


Figure 2.1: Main interface of Presenter

As shown in Figure 2.1, presenter is a slide-based system running on a Tablet PC. The other piece of software runs on a second machine for driving

the image projector. Two devices synchronise with each other via a wireless network. The design of the system is divided into three different focus points: pen-based, distributed and integrated with student devices. The pen-based system provides the new interaction method. The distributed architecture decouples the presenter from the projector. They also conducted experiments where students are able to provide feedback directly on slides and actively developing better functions in future versions.

Classroom Presenter has been deployed in 25 different computer science related courses. Three of them were distance learning courses. The instructors and students were overall enthusiastic about the system. The survey of the students indicated the system has a positive impact as well. Only 10% of the respondents has a negative impression of the system while nearly 55% of the respondents think positively about the effect of the system. The same survey was conducted for the instructors. The overall reaction from the instructors was very positive. They also found something interesting enough that the use of facilities of Presenter varies due to different teaching styles. Several studies they conducted with Classroom Presenter are quite meaningful as a guidance. We will reviewed those articles later.

In the end of the article, the authors took a brief look into several examples of using the system. These examples have been further studied in detail in their later publications.

2.3 Digital Ink in Lecture Presentation

The article "A Study of Digital Ink in Lecture Presentation" (R. J. Anderson et al., 2004) provides support for several ideas we mentioned in the previous chapter. Anderson et al. expressed their concerns when it comes to the flexibility of the presentation content when the presenter interacts with their audiences. The article took a deep look into how presenters use digital ink with the Classroom Presenter software (R. Anderson et al., 2007). The researchers focused on three courses. Although the scope may seem limited, they argued that a small amount of courses helps gaining a deep understanding of the style and context of each course. Moreover, the result, according to article, coincides with the data from other Classroom Presenter deployment. The three courses are taught by three professors separately.

The result of the observations are discussed in three different themes and we quote here: "*uses of ink, which we call **attentional marks**, that are analogous to physical gestures; the tension between ephemeral meaning of*

ink and its persistent representation; and instructors' parsimonious use of system features." We are particularly interested with these themes since they are strongly connected with our plan of work.

First and foremost, they discussed attentional marks. Observation shows that the attentional marks are often arrows, circles and underlines together with some other simple symbols. They noticed that the professors generally used attentional marks as alternatives to hand gestures in order to draw the attention from their students. The surprising high rate of appearance of attentional marks among all ink proves that the attentional marks are a very important component of lecture ink.

In the light of McNeill's theory which classifies the gestures into "*iconics, metaphorics, beats, cohesives and deictics*" (McNeill, 1992), the researchers propose that the attentional marks could be classified with the same standard. The so-called "parallel match" between attentional marks and hand gestures is interesting in the sense that hand gestures are ephemeral while attentional marks could be persistent. The attentional marks could easily outlasts their context and become difficult to interpret without extra information. In such form, ink only preserves ephemeral meaning. In further references to the framework of physical gestures, they conclude that the persistence of meaning for each class of attentional marks could be easily matched to the persistence of its associated class of gestures.

The attentional marks are not the only ones which are meaningless without their context, the authors found that even meaningful content like written text or drawings can easily lose its meaning without spoken context. They attempt to break down such content into different phases. The ephemeral pieces actually convey more information separately since the overlay of the entire drawing cannot preserve the sequence of strokes. This is particular interesting when it comes to reinterpreting or reusing the digital ink. In order to make sense from the combination of the entire saved graph, the professor has to orally indicate the sequence of the drawing or manually indicate the order of its ink by marking with more symbols. Such efforts to break down the entire illustration have been observed commonly during the courses.

The researchers also found it surprising that given all functions the system provided to the users, they did not have much interest to use them. The highlighter and stroke eraser are barely used, the change colour functionality and the page erase are only used by two professors separately. The explanation they proposed is that the professors did not use those functions because all the extra efforts which the system requires to switch between different

functions. Also the authors noticed the fact that although the professors considered colour changing such precious function, they failed to use it in the way the system is designed. Instead of using the highly contrast colour when needed and switch back when finish using, the presenters tend to keep using the high contrast colour. Such use pattern saves one extra interaction which is consider to be the reason for their actions. This sort of observation enlightens our design of the system as well. We need to minimise the unnecessary effort to access the system functionality.

Finally, they conclude their observation and suggest the future digital ink presentation system designer to consider the possibilities of keeping ephemeral information into persistent ink representation. By doing so, it is possible to gain a better understanding of how information is conveyed via presentation. Therefore, it further guides our design of new presentation tools.

2.4 Speech, Ink and Slides: The Interaction of Content Channels

The article "Speech, Ink, and Slides: The Interaction of Content Channels" (R. Anderson et al., 2004), yet another research on how digital ink is normally used in presentations, is based on the deployment of Classroom Presenter. The authors were interested in how ink and speech are used together. They are trying to inform the future development of tools for classroom presentations, distance education and viewing of archived lectures. They also tried to assess the potential of combining speech and handwriting recognition. Therefore they conducted an analysis of how attentional marks were used as well as how the presenters combined ink and speech together. Their analysis results are encouraging for using digital ink. As mentioned previously, Shilman's work (Shilman, Wei, Raghupathy, Simard, & Jones, 2003) on analysing free form ink is particularly relevant. They first focused on a study of ink and speech. They believed by finding a link between the two, we could easily identify different types of ink according to the audio record.

In the article, they found it useful to distinguish three types of digital ink: *textual, diagrammatic and attentional*. They observed that the attentional ink takes up more than 50% of the ink, and the textual content appears majorly as short phases. As for handwriting recognition, they conducted their first analysis, handwriting recognition, to understand to what extent handwriting recognition works in the lecture context. The lecture context presents a potential challenge and a new opportunity as the writing seems

inscrutable to recognise while the combined recognition of speech and ink merit exploration.

The researchers first investigated how well the existing handwriting recognition software works. Doing this would give them an understanding of where the boundary is between recognisable and unrecognisable content. Therefore, they could carefully customise their recogniser to work better. They segment the textual ink from the other types manually then feed it to the recogniser. They coded the result as *exact*, *alternate*, *close* and *none*. Their result shows that the recognition rate is acceptable however the failure still takes up 28% of the cases. The author stated that the factors influencing the recognition are "non-dictionary words, steep angles, bad handwriting and double ink".

In the light of the result from the previous analysis, the authors proposed three methods to join writing and speech recognition: directly combine writing and speech, recognise channel specific term, and merge channels to disambiguate the recognized words. They analysed the correlation between writing and speech. Their analysis suggests that coordinated use of ink and speech could be a promising idea. They also mentioned the technique for multilingual document corpora. Such technique could enhance disambiguation from speech to ink or vice versa.

Later, they turned their focus to attentional marks. As mentioned before, the attentional marks, among all three types, take up a half of the strokes. However, identifying the attentional marks could be quite difficult. The approach proposed by them is to find the attentional marks by the fact that most of the attentional marks contain very few strokes. Afterwards, they attempted to extract the slide content marked by the attentional marks. The result of recognition is coded by four persons by the classes of exact matches, exact matches up to punctuation, close matches and non-matches. According to the result, they concluded that the system works well already for certain attentional marks. However, there are detailed cases when recognition needs further improvement. They also analysed the case how speech, ink and the formula content relate to each other.

Finally, they explored one last example, correction of the slide content by annotation. The correction is quite a special case since it is less frequent but more ambiguous and inaccurate. They developed a software as a proof of concept. The software determines corrections according to the overlap between the ink and the slide content. The system works on certain case as the authors demonstrated but in general not good enough.

They concluded that their work gives a strong basis for using ink and speech together in the analysis of recorded lectures. Their research showed that basic ink analysis itself presents good result. Moreover, the strong connection between speech and ink analysis can be mutually reinforcing each other.

2.5 A Study of Diagrammatic Ink in Lecture

The study "A Study of Diagrammatic Ink in Lecture" (R. Anderson et al., 2005) is conducted as the successive study of the previous one (R. J. Anderson et al., 2004). As previously mentioned, they classified the ink into three types including diagrammatic ink. In this article they focused on diagrammatic ink as the other two types, textual ink and attentional ink, have already been analysed in their previous work. They examined a large number of diagrams that occurred in lectures trying to identifying patterns of using diagrammatic ink. By conducting such an analysis, they further attempted to find the common pattern for extracting diagrams from the other types of ink.

In this article, they first analysed the impact from the lecture environment to the ink diagrams. What they observed is that even the same diagrams drawn by the same instructor can differ from lecture to lecture. The reasons are the following:

- Lecture Dynamics: Diagrams are drawn according to the questions from students without much preparation.
- Physical Settings: In order to maintain eye contact with the students, the presenter might have to draw a simpler diagram.
- Tablet Challenge: Writing on the tablet can be unpleasant when it comes to writing on the edge. Such inconvenience could result in less drawing.

These factors result in imperfect diagrams thus creates difficulty for the extraction of diagrams. Additionally, there are attentional marks which could take up 50% to 70% of the ink strokes according to their previous study. These marks can be extremely difficult to distinguish from the diagrammatic content. They also noticed that structures are often drawn in compound fashion. By compound they mean for example using multiple strokes only to represent a single line. This also adds up difficulty since computer programs would fail to follow simple rules for extraction. With all the challenges above, the authors concluded that it is necessary to make use of extra information, such as speech information and gesture information.

The second observation they made is that diagrammatic inks are often drawn in several phases. Three examples are analysed to illustrate the necessity of identifying phases in order to understand diagrammatic ink. This observation is important to our work since we provided solution for reproducing phased diagrams.

The final observation made is that discussion frequently focuses on part of the diagrams. Discussion of partial diagram can occur when reusing of diagram. Such behaviour could be meaningful within context but makes no sense when viewed entirely afterwards. Therefore, the challenge becomes how to identify the meaningful content of the whole diagram. Despite the difficulty, the authors believe such goal could be achieved by combining information from the other modalities.

From the discussion above, they conclude the three core problems for analysing diagrams being: Ink Classification, Diagram Phasing and Locality of Focus. Though these problems are difficult to overcome, the effort to solve them would help to improve tools for lecturing and reading lecture content.

2.6 Ubiquitous Presenter: Increasing Student Access and Control in a Digital Lecturing Environment

In the light of Classroom Presenter, Wilkerson et al. built Ubiquitous Presenter (Wilkerson, Griswold, & Simon, 2005), a expansion of Classroom Presenter via web technologies. They appreciated the advantages of Classroom Presenter but also found a few limitations to improve. Ubiquitous Presenter (UP), as an improved system of Classroom Presenter, was therefore developed. UP allows internet-enabled devices to access lecture content generated by Classroom Presenter, ink annotation included. It works in a mostly similar sense comparing to Classroom Presenter. Additionally, it provides easy review of the lecture content and a better interaction experience between the student and the instructor. The adaptation of a client-server architecture offers beneficial side effects but also introduces a number of issues.

The issues they noticed with Classroom Presenters are:

- Classroom Presenter uses a broadcast model of communication which shows problems when there are late joiners. Moreover, reliability and availability of multicast networking (Frederick, Jacobson, & Design, 2003) could introduce even more problems.

- Student submission, as a powerful feature, can not be utilised without specific device running Classroom Presenter. Setting up the environment for students before lecture is time consuming.

To overcome the issues above, they developed UP to be web based. Such design enables the Ubiquitous Presenter's resource to be accessed by browsers. It tries to maintain the Classroom Presenter's features while utilising the reliability of the Internet when it comes to broadcasting.

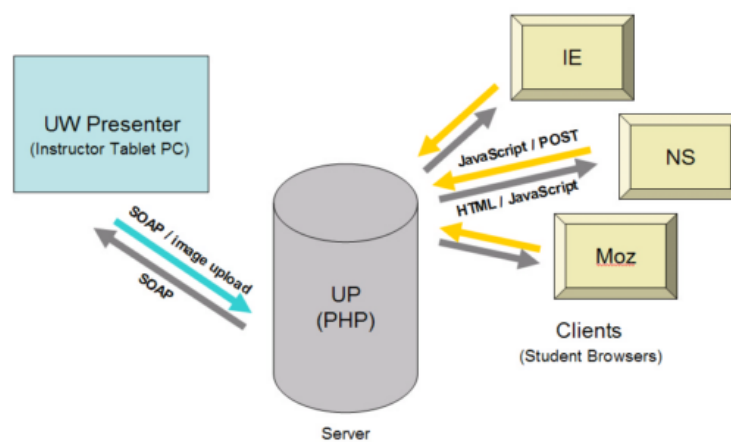


Figure 2.2: Architecture of Ubiquitous Presenter

Source: "Ubiquitous Presenter: Increasing Student Access and Control in a Digital Lecturing Environment" by Wilkerson et al. (2005)

As shown in Figure 2.2 the system is developed in PHP as a client-server application. When students access the server with their browser, the server side generates HTML and limited JavaScript to represent slides and send them to the students' browsers. They utilised separated processes to achieve failure tolerance. The student browser remains synchronised with a delay of less than 3 seconds. The synchronisation relies on short file transfers and SOAP remote procedure class. The change of slides triggers a SOAP call to the server to receive ink and overlay the ink to the original one. This and many other functions are all performed in the same method.

In terms of the system functionality, they kept the Classroom Presenter's functions as much as possible. Thus, on the instructor side, the new features are no more than upload slides for previewing, password protecting slides, starting a feedback request. However, on the student side, there are several

key features available. Accessing system via the browser is already a big advantage. They also have the option to easily switch between synchronous mode and asynchronous mode. Being asynchronous, the student could freely view any page of the slides. Just like previewing the content when the slides are uploaded, the server keeps the slides after the lecture for students to review or even submit more comments. UP enhanced the student submission system of Ubiquitous Presenter by offering full control of the submission to the student. The students could change their text any time before submission and retain full control over the initiation of submission. The system is also considered to be applicable in a remote education setup.

To conclude, UP is an expansion of Classroom Presenter which provides better accessibility for students in the lecture but also extends to before and after the lecture. The web-based architecture brings a good universal solution for introducing student interaction

2.7 Using Pen-based Computers Across the Computer Science Curriculum

In this article (Berque, Bonebright, & Whitesell, 2004), Berque et al. described their experience using their pen-based electronic setup in computer science courses. The research was based on, by that time, the emerging theory of active learning (Center for Science and Mathematics and Engineering Education & Committee on Undergraduate Science Education, 1999). They believed that the transformation from simple observing learning to active interacting learning would essentially help the students. Therefore, they tried to prove that a pen-based system would enhance learning since it allows students to interact with the slide content. The lecture context are complex data structures or algorithms which could be difficult to illustrate by text and oral description.

The physical setup is a 72" rear-projection electronic whiteboard in the front of the classroom and a number of Tablet PCs for all students and teachers. The system is a commercial product named DyKnow³. It shares the teacher's material and annotation to each of the students through both electronic whiteboard and their Tablet PCs. The student could in return submit their private annotation to the teacher's material. Additionally, one or more students could temporarily be authorised to write interactively on the devices of all their fellow students. The setup lays the foundation for highly

³<http://www.dyknow.com/>

interactive teaching environments. The teacher's workspace also stores the entire lecture material for later review. The saved content could be replayed stroke by stroke.

They described three different scenarios they observed during the use of the system, along with an investigation of the cost effectiveness. The two options mentioned here are: desktop machines with video tablets or tablet PCs. The former is much more powerful however expensive while the later is more economic but offers limited performance.

The evaluation of the system comprised of a written survey and a focus group. The survey was designed to measure students' attitudes towards using the DyKnow system. The result revealed several facts. First, students were mostly satisfied with the system. Second, the female students considered the system more helpful than the male students did. Last, students with a Computer Science background had more positive opinions in comparison with the others. During the focus group, the students were asked about the best feature of the system. The answers were mostly "the interaction features". However, several considered interaction to be the interaction with the instructor and their fellow students while the rest referred interaction as the interaction with the instructor's notes. Moreover, students commended the accessibility to the notes as they failed to attend several lectures. As for disadvantages, they concerned the distractions from a computer-based system. The focus group was overall satisfied with the system as well. Their discussion was considered a great guideline for the later improvement of the survey questions.

To conclude, they believed it is necessary to find a proper role for the emerging pen-based devices in classrooms. They also emphasised the need for further investigating the distraction issue. After all, according to the data gathered in this paper, the pen-based system is well accepted and appropriately used.

2.8 Livenotes: A System for Cooperative and Augmented Note-Taking in Lectures

Livenotes (Kam et al., 2005), a shared whiteboard system, aiming to achieve the goal of stimulate interaction within small groups of students during lecture. The system targets minimal changes of physical setup and teaching practice. In comparison with the other system (Ploetzner, Dillenbourg,

Preier, & Traum, 1999) which also supports group interaction, Livenotes features lower financial costs. Multiple iterations have been made in response to the previous studies (Iles, Glaser, Kam, & Canny, 2002; Kam et al., 2005). In this paper, they had the setup of a digital whiteboard wireless connected with a set of Table PCs.

Most of their observation concerns two functions. The first would be the cooperative note-taking. The system enables students to make simultaneous effort recording lecture content. Their annotation are shared among group members synchronously. The cooperative note-taking ensures a more complete content coverage as well as a richer whiteboard activity. They also find that cooperative note-takers tend to make more discussion when possible. The second interesting function is augmented note-taking. The tablet PC interface provides lecture slides as background for students to draw their own augmentation. The note-taking activity, comparing to previous deployment, indicates that augmented note-taking function could improve cooperative learning.

As they conclude from their reported experiment, Livenotes could still be improved in many aspects. Firstly, The system should support bringing instructor into the discussion since most of the time fellow students cannot properly answer the question from another. Moreover, the lack of interaction between the instructor and students are addressed. Thirdly, the group activity breaks down when several members refused to contribute. This is an intriguing problem need to be properly addressed in almost every collaborative tool. Of course the features like augmented note-taking also needs further exploration to better achieve active learning.

2.9 PaperPoint: A Paper-based Presentation and Interactive Paper Prototyping Tool

Utilising the technology of capturing handwriting with digital pen and paper, Signer and Norrie (2007) developed a framework (iPaper) which supports the rapid development of applications with paper-based interfaces. They introduced their framework together with a prototype application they developed to control PowerPoint from printed handouts of slides.

The iPaper framework features the creation of interactive areas on paper documents. An active area is mapped to digital information or services. When a user interacts with an active area, the mapped target will be accessed.

The (x,y) coordinates, document identifier and page number are all encoded into the position. When this information is captured by the digital pen and sent to the iPaper client, the client further send it via HTTP request to the iServer platform (Norrie, Signer, & Weibel, 2006) for further processing. The input devices are based on Anoto's Digital Pen and Paper technology⁴.

The introduced application, PaperPoint, was developed as a add-on interface for PowerPoint. The user could freely choose between PaperPoint and the other control interface. To use PaperPoint, the user first needs to create their presentation with PowerPoint in the way they are used to. However later, they have to print their slides together with the PaperPoint's template on the Anoto papers. While starting the PaperPoint client the user could choose which PowerPoint document they want to use. The handout layout has many active area features such as buttons to support linear or non-linear navigation. Using PaperPoint, the presenter is freed from the physical restriction from the presentation device. Another feature provided by PaperPoint is annotation. The presenter is able to annotate the presentation simply by writing on the slide content area on the handouts. The benefits of having a pen-based annotation interface has been proven in the previously reviewed articles. To fully explore the annotation functionality, PaperPoint also allows the user to create an empty page to draw freely.

The implementation of the system follows several steps. They first define the mapping between the active area and the digital information or services, stored in the iServer database. In PaperPoint, they stored several pieces of Java code. Once the one of the active area gets triggered, the linked Java code gets executed. The Java code, named active components, accesses the functionality of PowerPoint by utilising a Java/Win32 (Jawin Team, 2005) integration project which we will refer to later as Jawin. By using Jawin, the Java code could access the Component Object Model (COM) or Dynamic Link Libraries (DLL). The solution of this architecture could be used for many other Windows applications. PaperPoint itself also has enormous possibility in terms of potential extensions.

2.10 Supporting Active Learning and Example-based Instruction with Classroom Technology

This article (R. Anderson et al., 2007) focuses on a specific teaching activity which uses Classroom Presenter at an Algorithms class. The basic

⁴<http://www.anoto.com/>

deployment is that everybody had access to a Tablet PC running Classroom Presenter. The instructor is teaching with one device and show its slides on a big screen for the entire class. When the activity starts, the presenter sends a description slide to all students. The students then annotate their answer with digital ink and send it back. After receiving all the answers, the instructor has the option to display one to the big screen or continue teaching.

The role of technology was carefully explained because as the authors stated: "all of the activities discussed in this paper could be done as pencil and paper exercises that students work on during class." They stressed four key advantages of using technology:

- Slide distribution to student is almost instant.
- The collection of the answers is very convenient.
- The instructor could privately read the submitted work.
- The instructor could show student work on the shared big screen.

These features are majorly about efficiency and flexibility. There is more to add concerning usability. The students get to retain their copy of answers after submission, the instructor gets to read the first submission while waiting for the others to finish, and the whole class could read the shared information on the big screen resulting in a more focused discussion. Finally the ability to discuss a wrong answer without unveiling the name of the student can be a safe way to address mistakes.

The student reaction to the technology was 18 out of 19 positive. In the scope of all activities 69% of the students made submission. They conclude that technology can be used to support "course-specific and instructor-specific" innovations. The key to successful achieve active learning is, according to the authors, "having a deep pedagogical basis".

2.11 Exploring the Potential of Mobile Phones for Active Learning in the Classroom

Linquist et al. explored the possibility of further extending Ubiquitous Presenter (UP) with mobile phones (Lindquist et al., 2007). Their research is driven by the needs for classroom communication, ubiquitous possession of

mobile device, the low possession rate of Tablet PCs and the insufficient support for laptops in classroom. They stated that the mobile phones can be, at that time, the most promising technology for supporting active learning with minimal effort. In the light of the work from Markett et al. (Markett, Sánchez, Weber, & Tangney, 2006), they realised the limitations of using SMS only when it comes to addressing specific, complex problems. Therefore they introduced the support for images.

The system is based on UP which we already reviewed previously. They utilised the mechanism of sending SMS to an email-address. The PHP proxy server parses the SMS submissions and translate them into an UP compatible submission. For the system to work properly, they decided to identify each message by the sender's phone number and the slide being displayed. Thus, each message targets at a specific classroom and a specific slide. The rendering of the submission message follows a simple set of rules:

1. Message starting with a single letter is considered as answer for multi-choice question.
2. Message of text will be imposed on a copy of the slide and centered, including explanation text in multiple choice submission.
3. The photo in a submission will be rendered on a separate slide, down-sized if necessary.

They conducted an exploratory study where one of the author delivered a experimental lecture. The lecture contains a large amount of active learning exercise cast as formative assessments. The student is free on deciding how to answer the questions. In a total of twelve students, ten of them used mobile phones and two of them used Tablet PC with UP. The student is prepared with how to submit the answers via SMS and asked spontaneously to resubmit their solution in the last exercise. Those equipped with Tablet were asked to make submission via UP functions. Afterwards, an open-ended group interview was conducted. The whole lecture is video taped and further reviewed together with the student notes, submission and their observation notes.

They carefully described their observation classified by type of the problems being discussed. The observation lead to the following insights.

- The text submissions are overall brief due to inefficient text entry using mobile phones. However, the fastest six students are twice as fast as the slowest six. Such observation indicating the learning curve exists.

- Photo submission is suitable for complex problems but the readability of the image can be problematic.
- The mobility of the phone grants more physical table space for the students.
- The comparison with Tablet PCs reveals the absolute advantage of input efficiency on the Tablet PC side while the convenience and ubiquity of mobile devices are also noted.

At the end, the authors confirmed the suitability of a mobile phone as an alternative for active learning in a large class. They also admitted the limitations of the mobile phones as an input device.

2.12 INK-12: A Pen-based Wireless Classroom Interaction System for K-12

Koile et al. (Koile, Reider, & Rubin, 2010) investigated how pen-based technology works in upper elementary and middle school Science and Math. As we saw in the previous reviews, pen-based interaction has been proven to be useful when expressing abstract and condensed content. Using wireless network to share their ink inscription enables conversation between teacher and students. Such conversation is considered positive for learning and teaching.

The system consists of a set of tablet computers with customised software, Classroom Learning Partner (CLP) (Koile et al., 2007, 2007; Koile & Singer, 2008). CLP is a further development of Classroom Presenter (R. Anderson et al., 2007) we just discussed. CLP also allows the sharing of the presentation slides across the classroom. The slides are created as a PowerPoint file and loaded into all devices when the class begins. The current slide is presented on the large public screen. The teacher is able to annotate the slide, the annotation is immediately shared to all the devices including the public screen. On the other side, students could write on their device and submit the ink to the teacher. The teacher could then select and display a number of the submissions on the public display.

The CLP is customised specifically for student and teacher based on Presenter. On the teacher side, it supports receiving student notes, adding comments and sending it back privately. On the student side, the interface is simplified by removing unneeded functionality. They conducted their study of the system by observing the class and interviewing both the students and

the teachers. The result seems positive as the students were engaged throughout the entire study and the teachers were enthusiastic to adjust their course for the tablet PC. At this stage the improvements over Presenter is limited, the authors realised the issue that the slide metaphor can be restrictive and further innovation is needed on this aspect.

2.13 Measuring Increased Engagement Using Tablet PCs in a Code Review Class

In this article (Fagen & Kamin, 2013), Fagen et al. intended to stimulate discussion among students in a code review course. In this course, the students are asked to build a project in their private time and bring it to the class. Each student has 20-25 minutes to show their program on a large screen and explain it in detail. Most students found such a seminar beneficial while complaining that the discussions are getting superficial as the students get less engaged.

The system is a simple customised "shared whiteboard" application developed on the SLICE framework (Kamin, Hines, Peiper, & Capitanu, 2008). The student code gets loaded and presented as a tab. Within the tab, the content of each file is presented as the background while the user could annotate it. Every participant owns a tablet PC and all tablet PCs share the same data in real time. The tablet offers the functionality to write, draw and point while independently exploring the program. In this way, the student gets more time to read, understand and annotate the code rather than following the pace of the presenter.

Their experiment was to maintain the traditional setting for the most of the semester while introducing the tablet system for several weeks in the middle. The meetings are all audio recorded. The audio data is collected by putting microphones on every student. The moment of "cross talk" helped synchronising the timelines of different audio records. Their hypothesis was that tablets have a positive effect on engagement of oral communication. To support proving their hypothesis, they introduced a mathematical measure of "active engagement" during a meeting. The *"active engagement"* is *"how many participants had a 'turn' in the discussion within a window of time before the current time"*. By analysing all audio data, they found that after introducing tablet PC, the "active engagement" significantly increased, therefore proving their hypothesis. The author believed their measure of engagement could be helpful for studying engagement more objectively.

2.14 Office Remote

We already discussed a lot of previous academic work to make better presentations. In fact, there are also inspiring commercial products built for this purpose. Microsoft, for example, released their application called Office Remote (Microsoft, 2015b). It is an application combined with a plugin for the PC versions of the Office software and a mobile application which offers several functions as a remote. Note that to the best of our knowledge, no commercial application provides significant better functionality and usability than Office Remote.

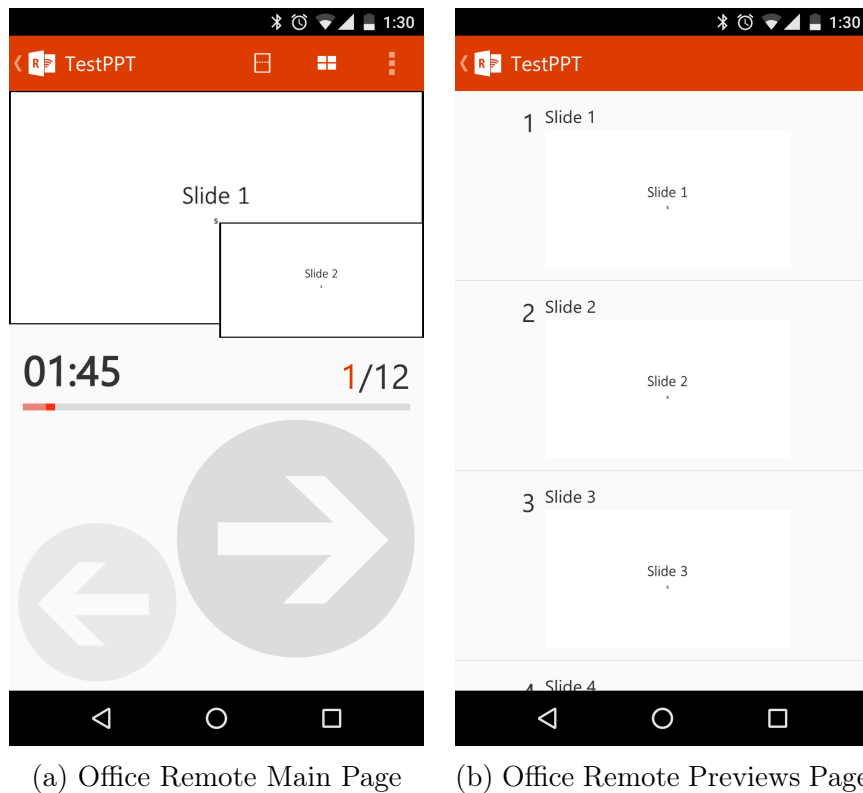


Figure 2.3: Office Remote running on an Android device

Office remote utilises a smartphone to provide convenient touch-based remote control for almost any Microsoft Office (Microsoft, 2015a) documents available on your PC. It is now at version 1.1 with the following functions also shown in Figure 2.3:

- Showing a large thumbnail of the current presented slide together with a scrollable list of previews of the other slides.

- An laser pointer controlled by touching the thumbnail view.
- Play and pause the embedded videos.

To use these functions, the user needs a installation of Microsoft Office 2013 or higher version, bluetooth on their PC and a specific desktop Office Add-In.

After proper installation, the user only has to turn the add-in on then select their PC on their mobile phone. The system also supports controlling Excel and Word however it is not related to our major concern here.

2.15 MindXpres: An Extensible Content-driven Cross-Media Presentation Platform

Although this thesis focuses on the improvement of PowerPoint, we must admit the fact that PowerPoint has a lot of limitations which need to be addressed. Roels and Signer (2013) introduced MindXpres (Roels & Signer, 2013), a new presentation platform utilising web technologies to tackle a large list of PowerPoint's issues. The goals for MindXpres are the following

1. Non-linear navigation: Some of the popular presentation tools, PowerPoint especially, only support sequential structure of their content. However, this structure has been criticised as outdated (Tufte, 2003).
2. Separation of Content and Presentation: They expect the users to focus only on the content of their presentation while MindXpres will handle the visualisation of the content.
3. Extensibility: MindXpres is expected to be an experimental platform for new concepts. Therefore they want it to be easily extensible with new content types, new presentation formats, new visualisation styles and innovative navigation techniques.
4. Cross-Media Content Reuse: MindXpres is expected to well support the reuse of external cross-media content.
5. Connectivity: As the new mobile technology arises, the authors believe that support for multi-directional connectivity helps both on content reuse and audience interaction.

6. Interactivity: Since MindXpres is web based, it is naturally suitable for dynamic content. The interactive content in the system will be a huge innovation comparing to the previous tools.
7. Post-Presentation Phase: Because of the non-linear navigation feature, the system is difficult to be played back in a same sequence when presented. However the need for a readable reviewing content still exists. Moreover, the social potential of the Web 2.0 technology is an interesting direction to explored.

MindXpres achieves its requirements by the architecture shown in Figure 2.4. As shown in the figure, MindXpres features a file format similar to the \LaTeX document authorizing language. A compiler will translate the file into a presentation bundle readable for MindXpres. Most importantly, all the behaviours are defined by plugins for flexibility concern. This allows MindXpres to address a lot of issues, which would be hard to approach in PowerPoint. It is a promising academic work to take a glimpse of what future presentation tool should be like.

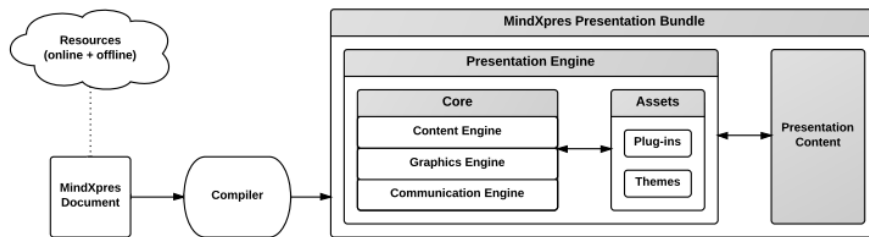


Figure 2.4: MindXpres architecture

Source: "MindXpres: An Extensible Content-driven Cross-Media Presentation Platform" (Roels & Signer, 2014)

2.16 Conclusion

We have already studied a good amount of related work. To justify our work, we need to summarise the existing solutions and finding the remaining issues. What we are particularly interested in is how to make better tools for instructors. For this purpose, we consider focusing our discussion on features mainly empowering the instructors. Our summary considers the following criteria:

1. The support of pen-based annotation.
2. The support of PowerPoint slides.
3. The learnability of the system.
4. Major deficits (if any).

Since our application is Tablet PC based, we ruled out the systems which do not feature good mobility. Ubiquitous Presenter and Classroom Learning Partner did not modify the above features of Classroom Presenter. The web-based architecture of Ubiquitous Presenter is indeed interesting and can be considered as a future possibility but not relevant with our current solution. The ZenPad system has not been updated for a relatively long time. The Livenotes are weak on PowerPoint compatibility. Moreover, DyKnow is already way too complex in terms of learnability. Thus, the remaining competitors are PaperPoint, Classroom Presenter and Office Remote. We will discuss each of them in detail with the previously mentioned criteria. We will emphasise some of our new features in the end as they have not been implemented in all other applications.

2.16.1 Classroom Presenter

Pen-based Annotation: Classroom Presenter fully supports pen-based annotations via a Tablet PC. It supports pen mode with 6 colours, eraser and highlighter. However, it does not allow creating empty pages which would be considered useful for activities such as brainstorming. What is more, it runs on the windows Platform where most of the Tablet PC featuring a built in stylus is not lightweight enough to be carried around the whole lecture.

PowerPoint compatibility: Classroom Presenter could load existing PowerPoint files and displayed them as a series of images. However, it could only save the slides later in its own format which is not compatible with PowerPoint anymore.

Learnability: It is difficult to set up the network connection for classroom Presenter. There are too many buttons on the interface while some of them are not useful. The overall learnability of Classroom Presenter is not very good.

Limitation: One of the most obvious deficits of Classroom Presenter is its incompatibility with PowerPoint. As PowerPoint has a great amount of users, such incompatibility could be a huge disadvantage. The lack of cross-platform support could also be an issue since users may not be willing to pay for a Windows Tablet PC only to use the system.

2.16.2 PaperPoint

Pen-based Annotation: PaperPoint features full support for pen-based annotations. It is even able to create new blank slides. Though the eraser and colour changing functions seem not possible to work on the paper interface, it is still overall of great usability.

PowerPoint compatibility: Since PaperPoint is an add-on for PowerPoint, it is fully compatible with PowerPoint.

Learnability: PaperPoint has a very self-explainable interface. However, to prepare the content before using could raise some troubles for normal users.

Limitation: One of the deficits of PaperPoint is because of the static nature of paper interfaces. Making any modification to the slides would result in reprinting the printouts. The paper interface also falls short when it comes to showing dynamic content.

2.16.3 Office Remote

Pen-based Annotation: Office Remote only supports the use of virtual laser pointer for pointing out content.

PowerPoint compatibility: It is compatible with PowerPoint though it fails to open certain versions of PowerPoint files.

Learnability: Since Office Remote only provides an essential amount of functions, its learnability is indeed very good.

Limitation: It is apparent that Office Remote, although the only mobile add-on for PowerPoint, is not equipped with ability to annotate at all. This issue aside, it is still a very good application serving a set of simple functions.

2.16.4 TabletPoint

We introduced TabletPoint in the first chapter. All the limitations we discussed above are compensated or even totally fixed in our system. Firstly, the system is totally compatible with PowerPoint. Despite of that it is not really a cross-platform application, we only have to migrate the client side to the new platform such as Windows 10 Mobile or iOS. Furthermore, the server side could also be adapted to other existing presentation tools as long as we could interop with them. Secondly, no extra effort is needed when users want a quick update of the slide content. The client can easily update the preview content of all slides. Thirdly, we provided several small improvements specifically for drawing such as drawing perfect squares, drawing straight lines and creating new slides while updating the preview list. Furthermore, it provides solution for storing and reproducing the digital ink. This feature, to my limited knowledge, has not been implemented in any pen-based system.

In the study we reviewed "A Study of Diagrammatic Ink in Lecture" (R. Anderson et al., 2005), where the authors discussed the case where certain diagrams become uninterpretable when reviewing the lecture slide. They concluded that it is because the phases of the diagram are no longer distinguishable. This issue remains when the lecture attempts to reuse their diagrams. According to several lecturers we know, they often find themselves reproducing the drawing they did for several times. In this case, how do we reproduce the drawing in their different phases becomes interesting. Our system provided a solution which let the presenter carry their drawing around with PowerPoint. This drawing information could only be accessed through TabletPoint. When needed, they could reproduce the saved diagrams to any slide.

3

Enhanced Presentation Interface with Tablet and Stylus

In this chapter we introduce our design of the system. In the previous chapter, we investigated and compared several existing solutions. Experience gained from these systems helped us to design our own. We will first take a look at the hardware requirements. Afterwards, we describe our design to help the readers to gain a better vision of what our system offers.

3.1 System Requirements

Our client is designed to be platform independent meaning that it could be developed for either Android or iOS devices. However, we choose the Android platform for our first prototype. Even it does not require too much computation power, a relatively large display is still necessary for better usability. As for stylus, the system in fact works well with bare hands if accuracy is not a major concern. Yet, we recommend the users to obtain styluses for fully exploiting the functionality of our system. As for the server side, a laptop PC with Java and PowerPoint is necessary to run the server software.

We carefully considered our choices of the hardware platform. For the server side our requirements are quite basic. In general cases, a user always needs to bring their laptop PC for PowerPoint presentation. However on the client side, we had many options. What we expect is a tablet with a popular operating system. Moreover, we want a built in stylus slot since asking a user to carry a stylus separately would be considered bad usability. We also noted the accuracy of the stylus. There are three different types of styluses we could choose. Among them, capacity stylus is the most affordable one but it is less accurate, the bluetooth stylus seems promising however it is a bit expensive, the digitizer stylus is accurate but only compatible with certain devices.

3.2 System Functionality

We introduce functions offered by our system by discussing every component of the interface. All system functions rely on a running server application on a laptop PC, therefore that is what we are going to describe first. After the server is up and running, we will bring our attention back to the client side where most of the interaction happens. We will bring in several use cases in later chapters to support our design.

3.2.1 Server Side

The server side is where we exploit the functions offered by PowerPoint. The technical details are very interesting but we will leave it for later discussion. The very first thing the user should do is to start the server side application. Upon starting the server side application, it shows a *file picker dialog* as shown in Figure 3.1.

The user could pick the presentation file they want to present with this file picker. After clicking the "Open" button, a new PowerPoint instance is created. The created instance automatically opens the file and goes in to presenting mode. Once the user is in presenting mode, there is no further operation needed on the server side.

3.2.2 Client Side

Once the server is started, all the user interactions would be on the client side. When the user opens the client side application, the program checks whether it could establish a connection with the server side. If not, a dialogue will

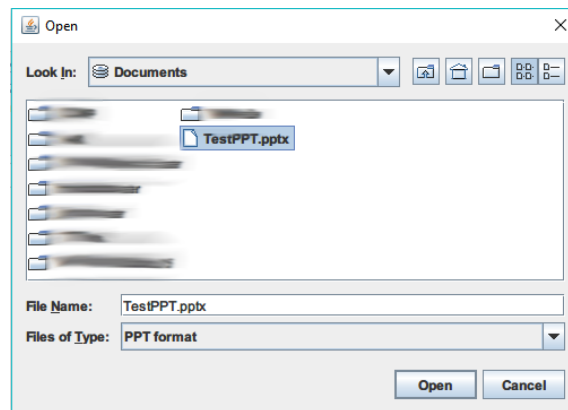


Figure 3.1: File picker dialogue

pops up asking whether the user wants to enable the relevant communication channel. When the user clicks the "OK" button, the application should enable the connection channel automatically. After the establishment of the connection, the user would see a list of possible devices as shown in Figure 3.2.

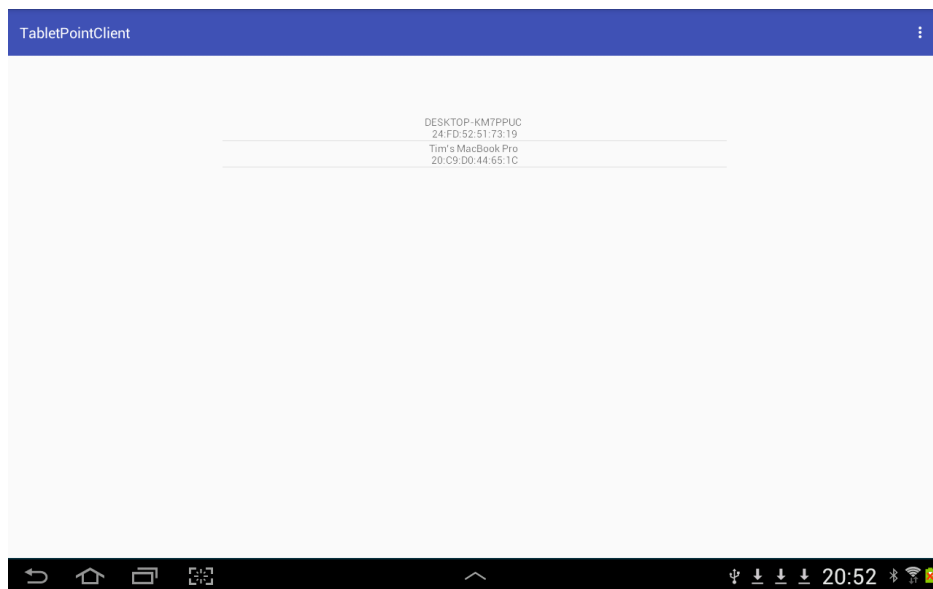


Figure 3.2: Device selecting view

The user is expected to pick the device where the server application is running on. After the PC information is clicked, the client application will attempt to establish a connection with the selected device. Of course we expect the

server is already running on the selected device. If the server is not running properly, a line of text will be shown asking the user to check the server setup. Once the connection is established, the main view of the client side is shown and the data including the previews of the slides, the digital ink user saved and the image of the presenting slide will be transferred from the server to the client side. After the data is well received, the main view's elements get populated, as shown in Figure 3.3. The elements on the main view are the following:

1. A list of slide previews at the top. We will refer to this list as the preview list.
2. A list of saved annotations on the right side of the screen. Because the saved annotations are comparable to using stamps in real life, we will refer to the list of annotations as the stamps list and the saved annotation as the stamp.
3. Four most used buttons at the top right corner of the screen. These group of buttons will be referred in the later content as the control panel.
4. A live streaming display of slideshow at the lower left, we will refer this view as the streaming view.
5. A tool box at the bottom right corner which contains a set of stroke types.

At this stage, the user is able to conduct a great amount of operations. We will discuss them separately based on the elements they interact with.

Preview List

There are not many operations supported by the preview list. It serves as an enforcement for navigating through slides. The user could scroll horizontally to see all their slides and navigate to their target simply by clicking on it. The currently presented slide will be marked by a darker background colour.

Toolbox

The toolbox contains a set of tools to be applied on the streaming view which we will elaborate later. The types of tools we offered are carefully selected because we believe that the worst thing for a pen-based interface is the redundancy of functionality. The tool box is collapsed by default as



Figure 3.3: Main view

shown in Figure 3.3 in order to save space. When the user clicks the main button of the toolbox, the 7 buttons offered by the tool box spread in an arc layout as shown in Figure 3.4. Upon selecting one of clicking one the buttons, the tool box automatically collapses again. The buttons consist of 6 different strokes and a colour selector. We will introduce them in the following list:

- Pencil stroke: The pencil stroke is the default stroke when the user enters the main view. It is a solid ink which would last through the entire presentation. The strokes shown on PowerPoint's slideshow is drawn by PowerPoint's default pen stroke.
- Pen stroke: The pen stroke utilise PowerPoint's pen stroke as the Pencil stroke does. However, the difference between the two is that the Pen stroke can be saved as a stamp and shown in the list while the pencil stroke cannot.
- Magic Wand stroke: The Magic Wand stroke is an innovation of the system where we show the stroke for 5 seconds and then erase it. According to the article "Digital Ink in Lecture Presentation" which we reviewed previously, the attentional marks can only be meaningful in a short time but take up to 50% of the lecture annotations. Our magic wand stroke is therefore specifically designed in consonance with the study and well suited for producing attentional marks.



Figure 3.4: Tool Box

- Eraser: The eraser is a complete mimic of PowerPoint's eraser tool. We made it to function exactly the same as PowerPoint's eraser to ensure consistency of the two display. The eraser erases every line it crosses with.
- Line stroke: The line stroke is a specific mode of pen stroke. It enables the user to easily draw a straight line. It locates the start of line when user put the pen tip down. Before the user lifts their pen, a dynamic line is showing on the client side locating its endpoint where the pen tip is. Finally, when the user finishes adjusting the position of the endpoint, they could lift the pen tip. The straight line will then be drawn on the slideshow.
- Square stroke: The rectangle stroke is another mode of pen stroke which help drawing perfect rectangle. It works very much like the line stroke with the only difference that it locates the top left corner of the rectangle where the tip is down and the bottom right corner where the tip currently is.
- Colour selector: The colour selector is self-explanatory. It allows the user to pick the colour of their stroke. Note here not all strokes share the same colour. The magic wand stroke uses its unique colour which by default is blue while the rest of the strokes use another colour which by default is black. the colour selector is shown in Figure 3.5

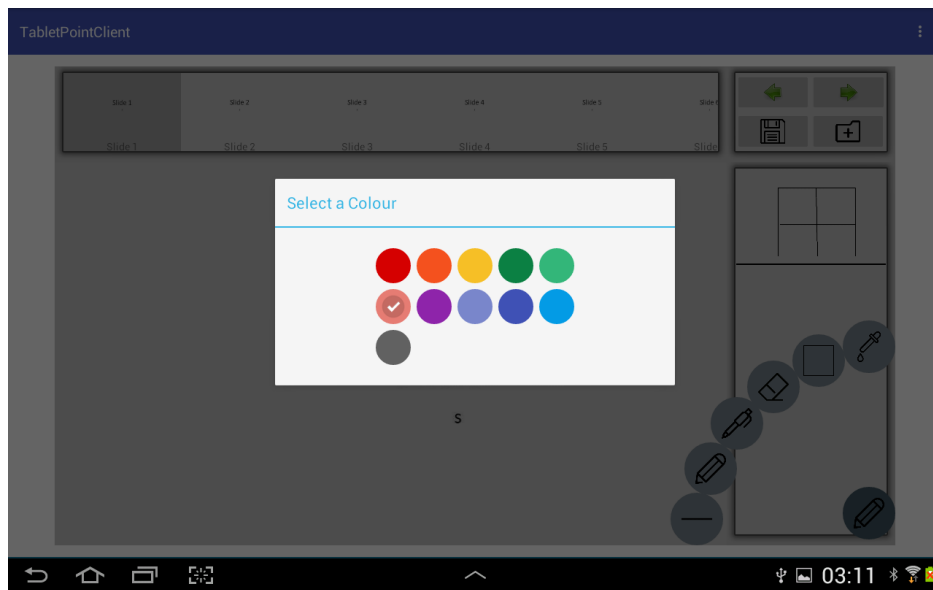


Figure 3.5: Colour picker

As we introduced, the user can switch to any one of the six strokes and change to any one of the colour options. Therefore it is important to clearly indicate the currently selected stroke. We put this indication at the entering button of the tool box. An icon of the current stroke is shown on the button while the background colour of the button is the current colour of the stroke.

Control Panel

The control panel, as we briefly introduced, contains the four most used buttons. As the icon indicates, the two buttons above enable navigating to the previous or the next slide. The button at the bottom left corner is the button to save the stamps, we will call it the "save" button. The other one, when clicked, appends an empty slide to the end of the slide queue and automatically navigates to that slide. This empty slide can then be used as a virtual whiteboard and be saved for future reviewing.

Stamps List

The stamps list is where all the stamps will be saved to when we clicked the "save" button. The saved stamps will also be saved in the PowerPoint file on the server side. Thus next time the application opens, the saved file will be loaded automatically to the stamps list. To utilise the stamps, all the user needs to do is to click on the target stamp. The program will reproduce

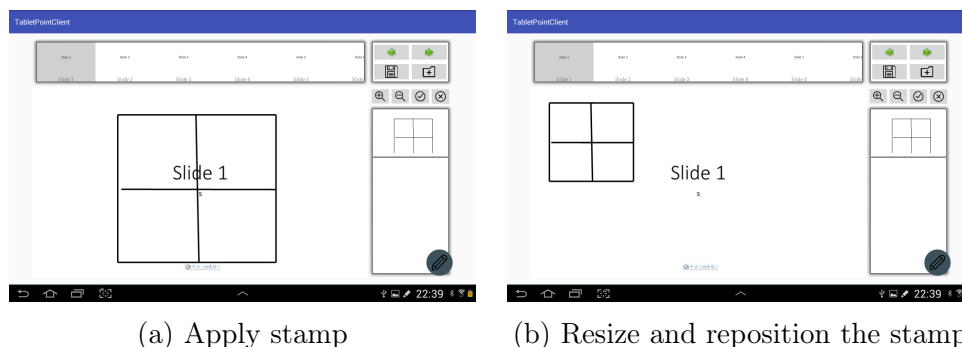


Figure 3.6: Using stamp

the stamp on the client side screen exactly where it is saved. However, the stamp will not be posted on the presentation just yet. It can be observed in Figure 3.6a that when the new stamp appears on the streaming view, four buttons appear above the stamps list. The first two of these four buttons are for zooming the stamp. When the stamp is zoomed to the proper size, the user is able to adjust its position by moving the stylus. After the stamp is in the proper position with the proper size as shown in Figure 3.6b, The user could click on the third button to confirm the stamp. Upon confirmation, the stamp will be drawn on the server side at the expected position. If the user change their mind about using the stamp, they could click the fourth button to cancel the operation. Note here though, the reproduced diagram is actually drawn with the pencil strokes therefore it will not be saved multiple times. Moreover, as the diagram is drawn with the pencil strokes, it also inherits the colour of the current pencil stroke. That means, the user has the freedom to change the colour of their diagrams.

Since the stamps list is scrollable, it could in theory contain as many stamps as possible. However, we would not want our list to be filled with stamps that are never used. In that sense, we provided the function for deleting the unwanted stamps. The options is optimized for a touch based device. By simply swiping the unwanted stamp to the left, a delete button slides in, shown in Figure 3.7. The user could then click the button to delete the stamp.

Streaming View

Finally, we discuss the streaming view where the slideshow is streamed to. By streaming the slideshow, we support animations and videos on our client side. Obviously, the streaming view is where the user draws. The stroke

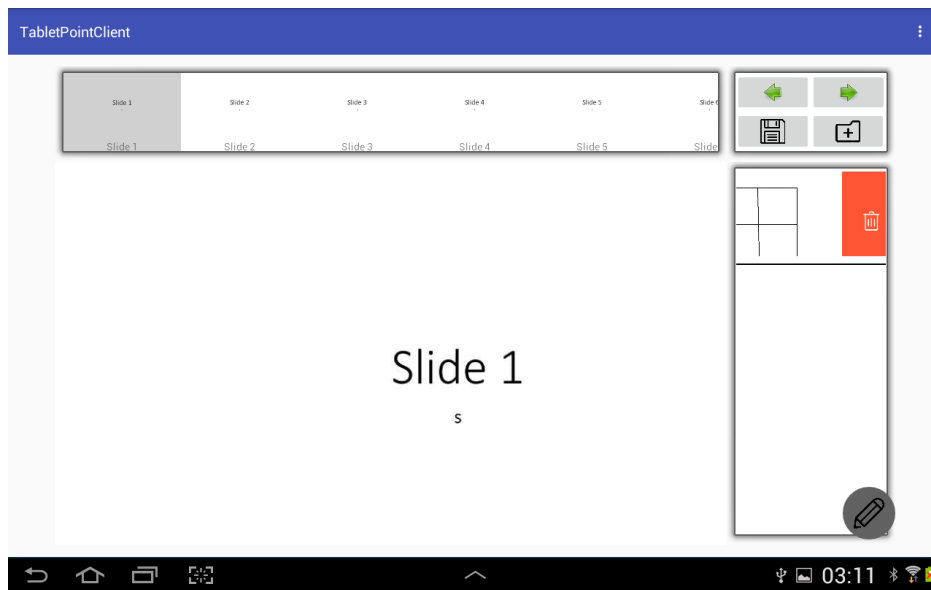


Figure 3.7: Stamp Deletion

will firstly be shown on the client side and then immediately synchronised with the server side. Because the server-side strokes are also captured in streaming, the client-side strokes are actually overlapped on the them. To minimize the confusion between the two strokes, we need to make sure the consistency of both sides by synchronising all operations in real time.

4

Implementation of TabletPoint

In this chapter we describe our implementation of the system, which we call TabletPoint. Before we introduce the implementation of TabletPoint, We first discuss about the research training we did and how the training helped the development of TabletPoint. Afterwards, we will first introduce the hardware platform and the general architecture shown in Figure 4.1. Then, for several interesting functions, we will show how we implement them in accordance to our previously mentioned architecture. Just like what we did in the previous chapter, we will discuss the server side and the client side separately. However, different from the previous chapter, we will only introduce the general solution we adopted without covering all details. As we made many iterations to finally achieve our purpose, we will focus on what we are finally settled with. In the meantime, we will mention the solutions which we did not adopt but that could be used as alternatives. We used Java as our primary programming language because of its platform independent feature, its use on the Android platform and its popularity. Our system made use of several existing libraries for the Bluetooth functionality and special UI effects.

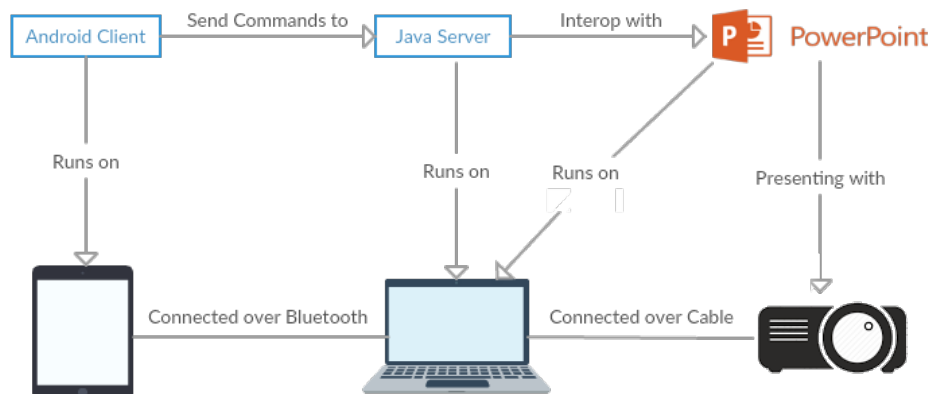


Figure 4.1: Architecture

4.1 Research Training

The research training we did is about developing a Java interface for the digital pen from Anoto¹ and finding a solution to interop with the newer version of PowerPoint. Though Paperpoint has achieved this before, it has not been updated for a while. Therefore, we attempted to make PaperPoint's interface compatible with the new versions and more efficient. The solution for the PowerPoint's interoperability is reused in the development of TablePoint thus we will discuss it later.

4.1.1 Digital Pen

To work with the digital pen technology, we must know the basic knowledge of how the digital pen works. The Anoto digital pen comes with the specially prepared papers which are covered with non-repeating dot pattern. When the pen tip contacts with the papers, The embedded camera at the tip of the pen captures the images of the dots and the built-in micro computer which recognises the unique patterns. Because the patterns are non-repeated, the pen could infer the current coordinates of the pen tip from the pattern and pass this information to another device which further utilises the coordinates to conduct some more complex tasks. There are currently two major uses of this technology.

¹<http://www.anoto.com/creative/technology/the-digital-pen/>

1. Applying handwriting text on the digital documents: Because the digital pen captures all the coordinates which the pen tip passes, it can be easy to reproduce the handwriting strokes on the digital documents.
2. Converting handwriting to digital text: This use takes one more step further and converts handwriting to digital text. This feature helps the users to produce digital documents without using keyboards.

4.1.2 Interface Implementation

The work starts from the interface for the digital pen. The major challenge here is about finding the right library to communicate with the pen. We investigated a lot of open-source libraries for interfacing with Bluetooth HID-Class devices and finally aimed for one (Signal11, 2012). This HIDAPI library is extremely light-weight while providing all the functions we need. The remaining issue here is that it seems very cumbersome when we tried to put it into a dynamic linked library and then wrap it up with Java. Fortunately, thanks to the blooming open-source community, we found a well organized Java library, HID4Java (Rowe, 2015). The HID4Java library is a Java wrapper for the HIDAPI library. It uses JNA for accessing the unmanaged code. The library comes with the pre-compiled HIDAPI library for almost all platforms. It wraps up all interfaces provided by the HIDAPI library. All we had to do is to import the package into our project and the low level issues are handled.

4.1.3 Decoding Bytes

The rest of the work is on the Java side. Since the wrapper only provides a limited amount of interfaces, we have to do a bit more work on our own. The architecture of the Java side needs to do all the work to interpret event messages. We used the "Blocking Queue" class in Java as a buffer of the bytes. One thread keeps reading messages from the pen and saving them into the queue while the other consuming all the messages. It is not challenging to correctly recognise the messages and trigger the right functions. What is left to do is to define a event-listener interface. In our case, the interface only provides the service to pass the organised message further to the subscribers. The overall architecture of the digital pen interface is shown in Figure 4.2.

There is a bit of extra effort to spend to interpret the messages from the digital pen. The Anoto pen is a proprietary hardware with its unique communication protocol. We could only decode the message in reference to the

documents from Anoto. The process of decoding messages is not difficult but a bit cumbersome since they defined many types of reports. The decoding solution is the most important part of the interface implementation.

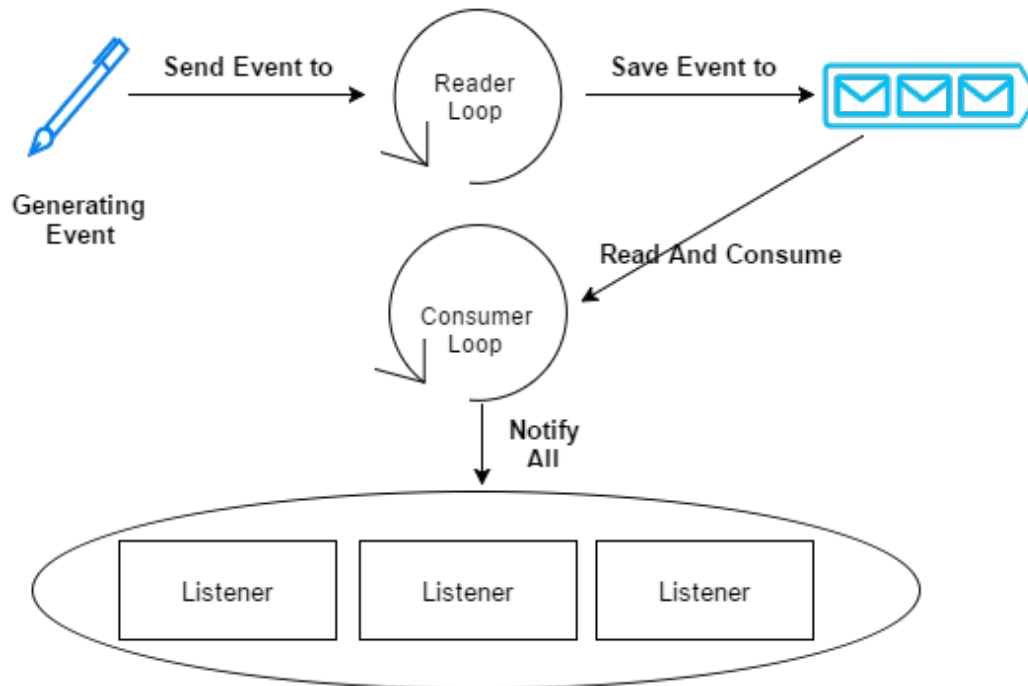


Figure 4.2: Digital Pen Interface Architecture

4.2 Server Implementation

For the server side, there are several things to consider. We need to decide how to utilise the Bluetooth functionality offered by the driver of the wireless card, how to stream our presentation to the client side, how to operate PowerPoint from Java code, and most importantly, how to cooperate with the client side. The following discussion will follow this sequence to show how the server side is implemented in detail.

4.2.1 Bluetooth Functionality

The system connects its client and server by Bluetooth connection. This means we need to find an easy-to-use solution. We decided that the Bluecove (BlueCove Team, 2008) library is a good option because it supports all the functionality we need and there is a great amount of documentation for

an easy integration. While Bluecove supports 4 types of connection: Service Discovery Application Profile (SDAP), Serial Cable Emulation Protocol (RFCOMM), Logical Link Control and Adaptation Protocol (L2CAP) and Generic Object Exchange Profile (GOEP) on top of RFCOMM and TCP (OBEX), we could only use RFCOMM protocol since we are limited by the Bluetooth support on the Android side.

To use the Bluecove library is relatively easy. What we expect is a running server waiting for connection request. We defined our universally unique identifier (UUID) to identify both sides correctly. Once the library is imported, simply run the following code and we have a running server waiting for connection.

Code 4.1: Bluetooth Connection

```
UUID uuid = new UUID(80087355);
String connectionString =
    "btspp://localhost:" + uuid + ";name=server";
StreamConnectionNotifier streamConnNotifier =
    (StreamConnectionNotifier) Connector.open(connectionString);
StreamConnection streamConnection =
    streamConnNotifier.acceptAndOpen();
```

4.2.2 Presentation Streaming

There are several options we have when trying to synchronise the presentation on both displays. For example we could use a third party library to render the slides on the android side while synchronising the position of the slides on both sides. Another option is that we could make a series of images out of the slides and simply render the images on the client side. However, none of these solutions are perfect in terms of functionality. The third party library could frequently fail to render as we tested, this issue also happened with Office Remote. The use of static images will lose all the dynamic content of the presentation such as animations, GIF images and videos. Therefore, we came to our current solution, streaming from the desktop PowerPoint application. By streaming the exact screen the audience would see, the content is by default highly synchronised. File compatibility is no longer an issue either. Most importantly, all the dynamic elements on the slide could be seen on the client side.

To achieve streaming, we still have two minor issues to solve. First, how do we generate the data for streaming? Second, how do we stream the data? The most straightforward solution for getting the data is to make a screenshot,

therefore that is what we did. However for the second issue, we need to ensure the data is loaded properly. What we did, is to send a customized header of our data. This header contains the size of the data and some signature bits to ensure the completeness of the data.

Code 4.2: Capturing Screen with Java API

```
Rectangle rec = getScreenCaptureBound();
BufferedImage screenCapture = ServerFactory.getRobot().
    createScreenCapture(rec);
```

4.2.3 PowerPoint Interoperability

Interoperability, by definition, is the "ability of a system to work with or use the parts or equipment of another system". We tried to interop with PowerPoint through PowerPoint Primary Interop Assembly (PIA) (Microsoft, 2012). PIA enables us to interop with PowerPoint through the Component Object Model (COM). Knowing there is such possibility, the issue is suddenly narrowed to how to access the COM model provided by PIA. There we did quite some research. As we introduced, PaperPoint achieved this by using a framework called Jawin (Jawin Team, 2005). However, the framework is last updated around 2005 and no longer supported. There is another popular solution called Java Native Access (JNI) (Oracle, 2016). Although JNI is an efficient solution when it comes to accessing dynamic linked library, it is powerless for COM accessibility. After a thorough investigation, we found our final choice, Java Native Access (JNA) (Java Native Access Group, 2016), a Java library which supports COM accessibility.

To use JNA, we first define a set of Java interface for the COM object we want to use. We match the PowerPoint object with its program ID as shown in the following code block.

Code 4.3: PowerPoint Interopable Interface

```
@ComObject(progId="PowerPoint.Application")
public interface ComPowerPoint_Application extends IUnknown {
    ...
}
@ComInterface(iid="{91493442-5A91-11CF-8700-00AA0060263B}")
public interface ComIApplication
    extends IUnknown, IConnectionPoint {
    ...
}
```

With this interface, we could create new PowerPoint instance as shown in the following code snippet. What is left is just creating interfaces for utilising the objects we get from this PowerPoint instance. Therefore, in theory, every single function provided by PowerPoint Interop Assemblies can be utilized in our Java code.

```
Factory factory = new Factory();
ComPowerPoint_Application powerpoint = factory.createObject(
    ComPowerPoint_Application.class);
ComIApplication app = powerpoint.queryInterface(ComIApplication.
    class);
```

Though it may sounds simple, to draw ink on the presented slide actually became another challenge because PowerPoint does not provide the interface we need. Here, we exploit a Java built-in class, Java robot. The Java robot is a program enabling us to mimic mouse and keyboard operations. Because all the fundamental events such as pen down, pen move and pen up could be easily mapped to events such as mouse press, mouse move and mouse release, we could in theory reproduce any diagrams drawn on the client side.

4.2.4 Communication

In order to control the server side from the client side, we have to define a set of commands. Every action on the server should be triggered by a command sent from the client side. Then, we associated this commands with blocks of codes. As long as we defined the commands and associated code blocks carefully, basically all functions of the system could be achieved by sending a set of commands. Then we made the server thread keep reading and handling requests so that the system can run smoothly. By building this, we would only need to focus on the client side and trying to exploit the different combinations of commands. The sets of commands for communication can be found below.

Code 4.4: Communication Commands

```
public class Commands
{
    public static final String IMAGE_REQUEST = "image_request";
    public static final String SLIDE_PREVIEW_REQUEST = "
        slide_preview_request";
    public static final String FILE_RECEIVED = "file_received";
    public static final String INKXML_REQUEST = "ink_xml_request"
        ;
    public static final String NEW_SLIDE = "new_slide";
    public static final int PenDown = 0x20;
    public static final int PenMove = 0x21;
```

```
public static final int PenUp = 0x22;
public static final int GOTO_PREV_SLIDE = 0x23;
public static final int GOTO_NEXT_SLIDE = 0x24;
public static final int GOTO_SLIDE = 0x25;
public static final int SAVE_INK = 0x26;
public static final int SET_COLOR = 0x28;
}
```

4.3 Client Implementation

The client side application runs on an Android tablet, Galaxy Note 10.1". We followed the official released best practices (Android Team, 2016) to develop our applications to ensure maintainability. There are also several issues deciding architecture on the android side. First question would be how to implement the connection thread while ensuring the easy inter-thread communication. This one is very important because it is directly related to the maintainability of the code. Secondly, we need to carefully think about how do we achieve annotation on the streaming view while synchronically drawing the annotation on the server side. Thirdly, we need to come up with an efficient data structure of stamps which could be used on the client and saved on the server side.

4.3.1 Threads

When developing for Android, one golden rule of making a usable application, is to put the heavy work load on a background thread. Doing this is to ensure not blocking the UI thread. There are many options when it comes to running background thread on Android. Creating a new thread in Java style, creating an AsyncTask or using a background service are all valid options. However, for our specific needs, we want the background thread automatically managing its "Looper" object for handling events from the main thread. The thread class fitting our requirement in Android API is called "HandlerThread".

Using HandlerThread is relatively simple, we first create our customised "BluetoothHandler" which handles "Message" objects from the main thread and further translate that message to send it to the server. When we finished defining our handler, we register an instance of this handler to the "Looper" instance of the thread. Therefore, we have the handler looping to handle every event in the message queue.

Another interesting rule about Android development is that the developers should always register a handler on the UI thread when using background thread. Because changing the UI component from the background thread is forbidden on the Android system, we need to send message to the main thread handler for delegating the job. Thus we created two handlers for two threads on the client side. The code snippet below shows how we create the background thread and assign it with both handlers.

Code 4.5: Creating Handlers

```
public BTConnectionThread(String name, int priority, Handler
    uiHanlder) {
    super(name, priority);
    mUIHandler = uiHanlder;
    mBTHandler = new BTConnectionHandler(this.getLooper(),
        mInStream, mOutStream, mUIHandler);
}
```

As for using the handlers, there are three different methods resulting in different effects. The first method is "sendToTarget" method which directly sends the message to the end of the message queue. We utilized this for streaming request since the streaming is of the lowest priority. The second method is "sendMessageAtFrontOfQueue". This method make sure the message will get executed in a very short time but not immediately. The last one is "handleMessage". This message executes the message on current thread therefore is strictly immediate. The use of the three methods is shown in the following.

Code 4.6: Using Handlers

```
\\ Will be executed
btHandler.obtainMessage(Constants.SEND_SLIDE_PREVIEW_REQUEST).
    sendToTarget();
\\ Almost immediately executed
mBTHandler.sendMessageAtFrontOfQueue(
    mBTHandler.obtainMessage(
        Constants.SEND_GOTO_SLIDE, slideNumber));
\\ Immediately executed on current slide.
btHandler.handleMessage(
    btHandler.obtainMessage(
        Constants.SEND_PEN_DOWN, new Point(x, y)));
```

4.3.2 Streaming View Implementation

The streaming view is very specialised because of our requirements. There is no existing reusable solution thus we could only fully customise our streaming

view class. In our code, the class name is "SlideshowView". To implement the streaming function is simple, when we received the image on the background thread, we pass it to the UI thread and draw it to our view. The actual complex part is handling the annotation. Since Android provides a "Path" class to record custom strokes, we could make use of this class to draw on the view. However by checking the API document we found that the "Path" class does not support erasing specific line. Therefore, we generate one "Path" object each time the "PenDown" event is triggered. When we need to delete the path, we could release the object. We saved all Path objects on a slide into an "Arraylist" object. In order to separate the strokes by slide numbers. We made use of the "HashMap" class. The slide number will be the key while its corresponding arraylist of strokes will be the object. This choice of data structure made the implementation easier.

Though we saved the "Path" objects properly, to draw them to the view, we have to employ a "Paint" object. As we stated, we need to draw different colors of strokes. That means we need to save the color information of each stroke. We achieved this by customizing the "Path" object to save the color. Before we draw a "Path" object to the view, we first set the color of the "Paint" object. Therefore the stroke on the view will be colored properly.

The one last issue would be keeping track of the different types of the strokes. What we did is just keeping them separated. We store the pencil, pen and magic strokes separately in three "HashMap" objects. We paint them in separate for-loops. Therefore when we need to save the pen strokes, we save the whole data structure with out worrying about extracting one from the others. As for eraser stroke, programmatically it draws an invisible line and deletes every other lines it crosses. The user could find a code snippet bellow of how path objects are managed.

Code 4.7: Data Structure

```
Public class SlideshowView extends View {
    private MyPath currentPath;
    private MyPath eraserPath;
    private HashMap<int , ArrayList<MyPath>> pencilPaths;
    private HashMap<int , ArrayList<MyPath>> penPaths;
    ...
}
```

4.3.3 Stamps List Implementation

The stamp is another one of our main contributions. To achieve this idea, we saved every pairs of coordinates the system captured into the Path object.

Therefore we could save the coordinates to the server if needed or we could reproduce the "Path" object basing only on the coordinates. As for stamps list, it is a "ListView" with a customized "Adapter". The customized adapter maintains lists of "Path" objects. Each list represents a stamp showing on the stamps list. Once clicked, the stamp is immediately drawn to the streaming view.

4.4 Functions

In this section we discuss how we implemented several interesting functions. The list includes how we synchronise the client annotations to the server, how we implemented the saving and loading of the stamps, how we achieved loading of the previews, and how we achieved some cool interaction like tool box and deleting of the stamps.

4.4.1 Stroke Synchronisation

Based on our previous discussion, we know that we have a robot object running on the server side mimicking the mouse movement. We are also aware that we can trigger any server function by sending commands to the server via Bluetooth connection. Therefore, we send a message to trigger the correspondent mouse event each time a pen event is triggered. Moreover, the same idea applies to features like changing stroke color, changing stroke type and so on. As we mapped all annotation related functions, the strokes on the client side is easily synchronized to the server side.

4.4.2 Loading and Saving

We mentioned previously that all initial data is loaded from the server side. The loading of the previews is achieved by exploring PowerPoint's "Save as Images" function. As soon as the PowerPoint instance finishes saving, the server sends the list of images to the client thus client could populate its previews list. When it comes to the saving of the stamps, we actually coded the stamp in to an XML file. After we send this XML to the server, the server could easily save this XML into the PowerPoint file by employing the PowerPoint's support for custom XML content. This function is meant for native PowerPoint plugins to have their own data stored inside the PowerPoint file. This saved custom XML content will not be wiped out by modifying the slides content. As soon as some content is saved on the server side. Next

time the system starts, it will be loaded and translated to stamps again on the client side. The XML file is structured as following.

Code 4.8: Path XML

```
<WISEInkXML>
  <Paths>
    <Path>
      <Points >0.1,0.1,0.2,0.2,0.4,0.3,0.5,0.6 </Points>
    </Path>
  </Paths>
</WISEInkXML>
```

We defined this XML structure only to represent the simple diagrams. However, this structure could be extended in many possible ways if needed.

4.4.3 Special Interaction

We only utilised two kinds of special interaction. The first is the swipe-to-show effect we utilised to show the delete button of a stamp. This is a customized layout called "SwipeLayout" (Damaijia, 2016) we found on Github. The other special visual effect is used in toolbox. The layout we used for the toolbox is the "ArcLayout" (ogaclejapan, 2016). We took a little customisation of these library to bend it for our needs and they look good with our system.

4.5 Conclusion

To sum up, we have the server side which launches a PowerPoint instance presenting a chosen presentation file. With the file loaded, the server could generate screenshots of the current presentation, the previews of the slides and the XML files of stamps. As soon as the client side makes a connection and requests all the prepared information, the server sends them to the client side and starts looping. Once the preparation is finished, the interaction will only be on the client side. Almost all client side operations are linked to the corresponding server side functions. When a operation happens, the client send the request with specific operation commands which get recognised on the server and soon executed.

5

Use Cases and Usability Evaluation

We introduced the system both conceptually and technically in the previous chapters. In this chapter, we would like to show how the system should be used and how the users think about it. We first define two use cases which exploit the key functions of the system. Then we will describe our usability evaluation using the System Usability Scale (SUS) (Brooke et al., 1996) evaluation method. The SUS tool has been an industry standard (Bangor, Kortum, & Miller, 2008) for assessing usability of a system. The result of the test offers us a clear vision of how the system is and what improvements we need to add to the system.

5.1 Use Cases

In this section we describe two use cases of the system. These two scenarios will not be able to cover all possible uses of the system but only the uses of the key functions. The first scenario is about the use of text annotation and attentional marks. The second will be about the reusing of a simple stamp. During the usability test, we asked the participants to reproduce the two scenarios following the guidance of the developer. Then they were asked to

give a short presentation using the system therefore they could provide valid feedback when answer the questionnaire.

5.1.1 First Scenario

The first scenario is about the use of the very basic functions of the system. We loaded a presentation created and used by one of the lecturer in our lab. He tends to write text annotation during presentation while using a lot of attentional marks. The basic functions of our system is completely suitable for his needs. When the lecture starts, he could navigate the slides, quickly check the content while moving around making eye contact with the audience.

Text Annotation

During presentations, he tends to write several keywords on the slides to better illustrate the idea. Since these keywords are not very reusable, he chooses the pencil stroke with a distinct red colour. Thanks to the accuracy offered by S Pen, the writing is well recognisable to the audiences, shown in Figure 5.1.

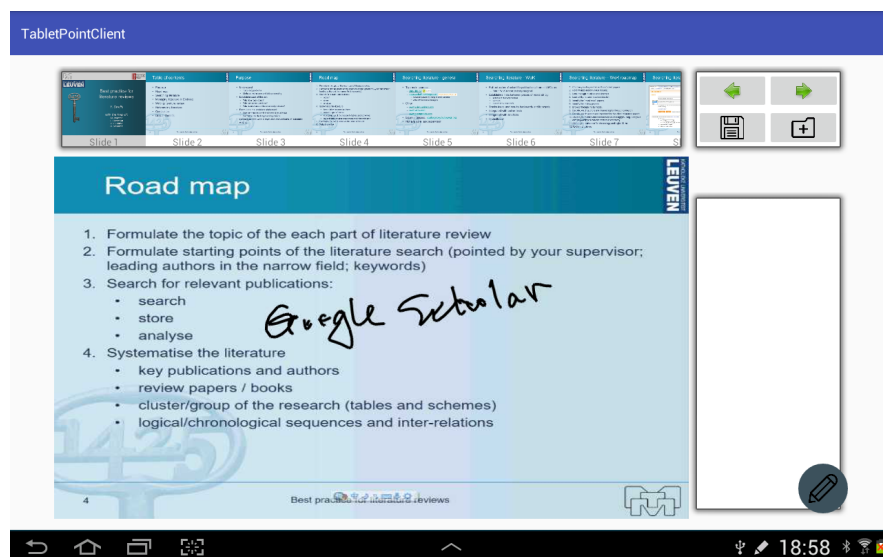


Figure 5.1: Text Annotation

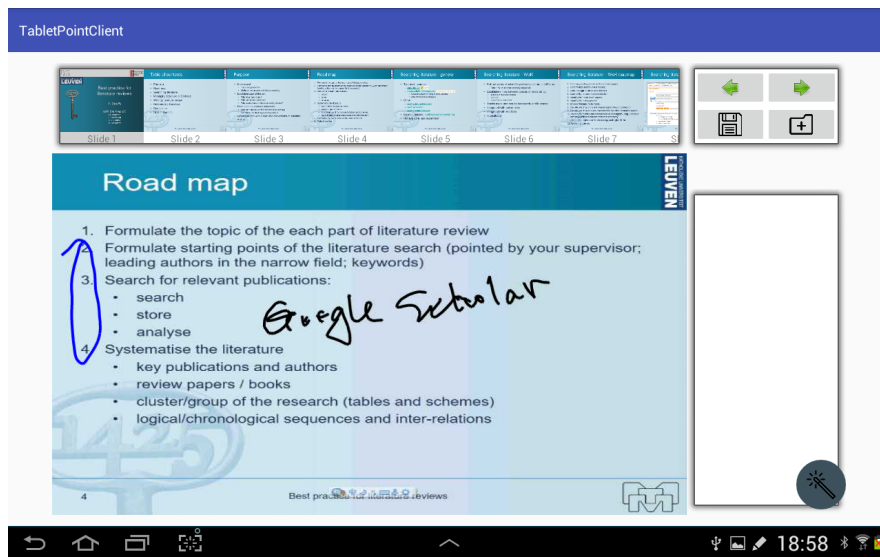


Figure 5.2: Attentional Marks

Attentional Marks

To highlight a part of content, the lecturer wants to use attentional marks here. The magic wand stroke of our system is specifically designed for this purpose. By using the magic wand, the lecturer stressed the importance of the keywords he annotated. They attentional marks disappear after 5 second of appearance leaving no confusion for reviewing the slide as shown in Figure 5.2.

5.1.2 Second Scenario

The second scenario is about using the stamps together with some of our drawing specific strokes. Imagine we are in a course teaching Chinese characters. When writing Chinese characters, the sequence and structure of the strokes are very important. For the structure, normally the beginners need baselines as a reference. As for the sequence, the lecturer has to reproduce the character stroke by stroke for the students to learn. We will see how our system helps in the lecture described above.

Baselines

To show the structure of a character, the lecturer has to draw the baseline as a reference for the students. With our square and line strokes, this task could be quickly accomplished, shown in Figure 5.3.

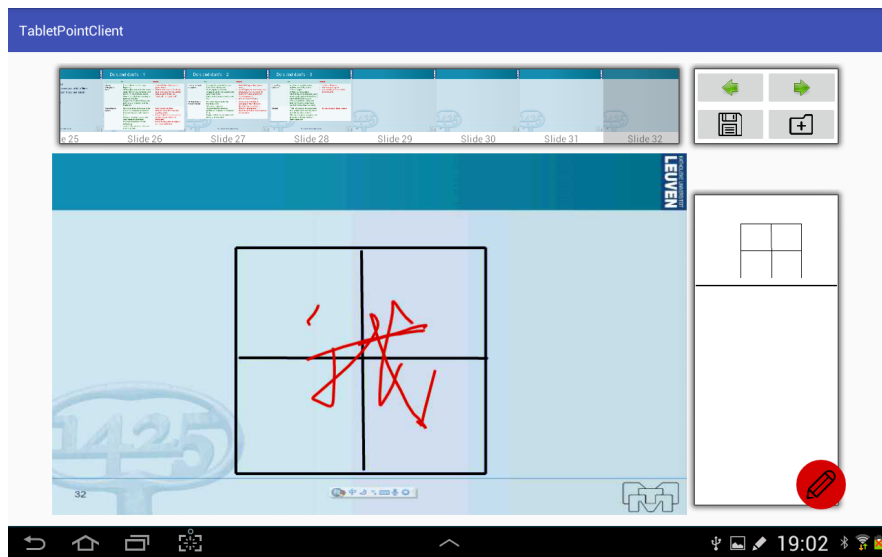


Figure 5.3: Drawing Baselines

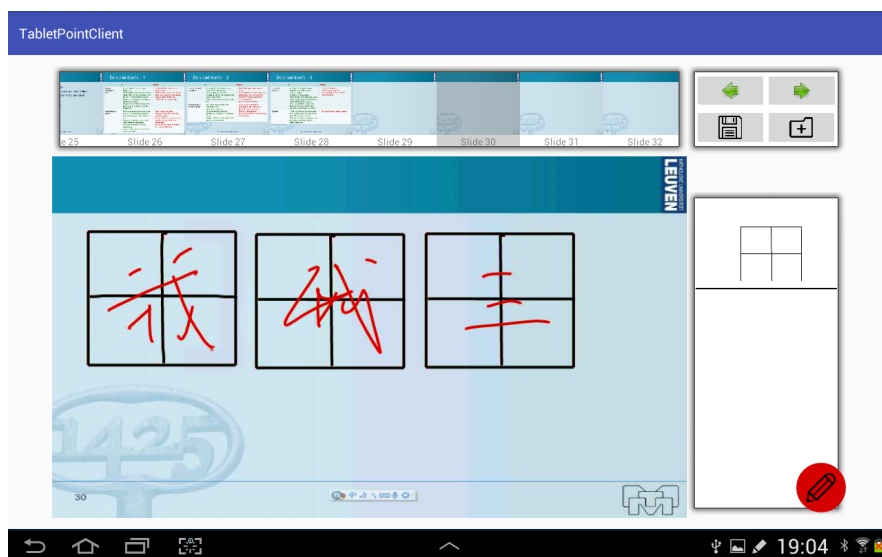


Figure 5.4: Using Stamp

Baselines Stamp

Though it is relatively easy to draw baselines in our system. We provide a better solution for such needs: stamps. When the teacher draws the baselines for the first time, they could save it as a stamp. When needed, the teacher could quickly reproduce the baselines while adjusting its position and size.

Therefore, with the help of our system, the teacher could really focus on teaching the sequence of the strokes as shown in Figure 5.4.

5.2 Usability Evaluation

In order to gain a better understanding of our system, we asked five of our fellow students to participate in the usability test. All participants are Master students in the field of engineering. As mentioned before, we will use the SUS evaluation method which consists of ten grading items. We will utilise the Typeform¹ web application for creating the questionnaire. This web application allows their users to put questionnaire online and collects the result automatically. Then we could analyse the data easily for our purpose.

5.2.1 Procedure

The procedure of the test is carefully planned. To give the participants a quick start, we gave them a demo of the two scenarios we described before in this chapter. Then the participants were asked to reproduce these scenarios as well as operate the tool freely. Then we asked the participants to give points to a series of statements prepared in the Typeform application. At the end of the usability evaluation, there is one open question where the participants are asked to give their opinion freely about the system. We expected that their answers to this question could give us a unique view of how we can improve the system. After the test, we collect all the data and analyse it.

5.2.2 Items and Scoring

The SUS contains ten items as well as a scoring system. The participants mark each item based on a Likert scale (Wuensch, 2005) ranging from "strongly disagree" to "strongly agree". For the items 1, 3, 5, 7, and 9, the score contribution is the scale position minus 1. For the other half of the items, the score contribution is 5 minus the scale position. The final result will be the sum of the scores multiplied by 2.5. The participants are asked to mark their points as fast as possible without extra thinking. The following is the list of the items.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.

¹<https://www.typeform.com/>

3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

5.2.3 Test Results

The results of the usability test are overall positive. The answers from the five participants are organised automatically by the Typeform application as shown in Figure 5.5. From the data dashboard of the Typeform application we could see a large variety of user information.

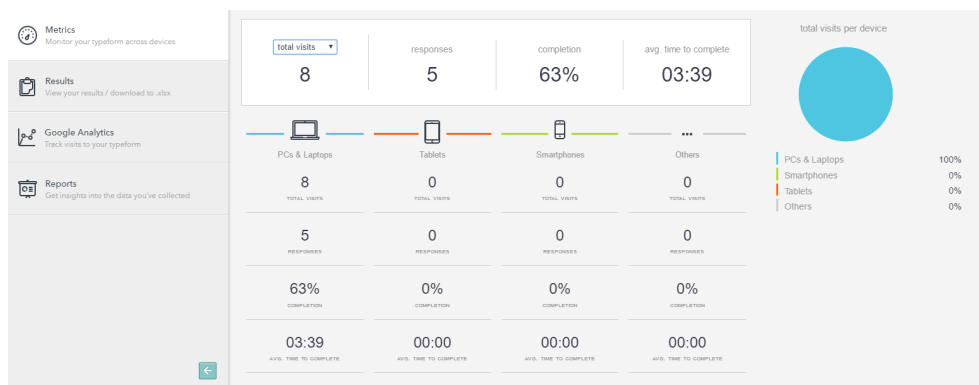


Figure 5.5: Data Dashboard

To analyse the data, we organise the data into an Excel file. With the help of Excel we could create a simple bar chart as shown in Figure 5.6. From the figure we can see that for the majority of the items, the user opinions are consistently positive. However, for the items 2, 6, and 8, there is some

disagreement among the participants. All these items are concerned with the complexity of the system. This makes sense since we introduced several relatively new concepts into the system. As we interviewed the participants after the test, they confirmed that it is the concept itself which confused them and the user interface is overall clear. Therefore we would argue the complexity is unavoidable in our case.

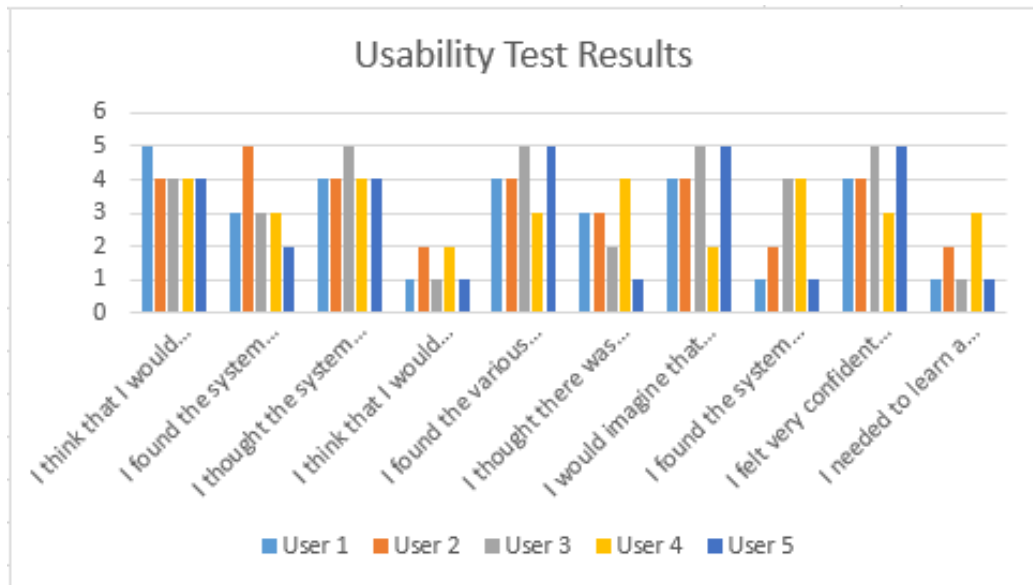


Figure 5.6: Bar Chart

We calculated the scores and put all number into a table as shown in Table 5.1. From the scores we could see that 4 of the participants are relatively satisfied with the system usability. The fourth user gives a low score due to the technical issue when she interacted with the system while the system is trying to create a new empty slide. We immediately fixed the bug therefore such unpleasant experience will not happen again. In the end, we asked each participants's opinions to the system. Most of the feedback concerns the lag between the interaction on the tablet and the reaction on the PowerPoint slide. Although this is an issue we care about, there is currently no feasible solution. Nevertheless we are still actively looking for the solutions to improve the reaction speed of the system. Interestingly, one participant brought up the idea of a voice interface for controlling PowerPoint. Together with the participant we imagined such possibility and it seems that such a feature could have quite a lot potential. For instance, the system could react to the voice commands such as "Next slide", "Previous slide" and "Go to slide

Table 5.1: Usability test data

Questions	User 1	User 2	User 3	User 4	User 5
I think that I would like to use this system frequently.	5	4	4	4	4
I found the system unnecessarily complex.	3	3	3	3	2
I thought the system was easy to use.	4	4	5	4	4
I think that I would need the support of a technical person to be able to use this system.	1	2	1	2	1
I found the various functions in this system were well integrated.	4	4	5	3	5
I thought there was too much inconsistency in this system.	3	3	2	4	1
I would imagine that most people would learn to use this system very quickly.	4	4	5	2	5
I found the system very cumbersome to use.	1	2	4	4	1
I felt very confident using the system.	4	4	5	3	5
I needed to learn a lot of things before I could get going with this system.	1	2	1	3	1
SUS score	80	72.5	82.5	55	92.5

5". With emerging hardware such as Google Glass², it might convey a very pleasant presentation experience.

5.2.4 Conclusion

The usability test provides us a lot of insights to our system. Although the overall results are positive, it unveils the aspects we need to improve as well. Moreover, the voice control interface brought up by the participant is indeed very interesting to consider.

²<http://www.google.be/glass/start/>

6

Conclusion

In this chapter we conclude this thesis. We will briefly go through a list of major contributions of our work. After that, we will discuss some remaining issues we would want to solve. Last but not least, we will offer some thoughts about the future possibilities.

The purpose of this thesis is to further explore the pen-based system utilising the new generation of tablets and PowerPoint. We designed our system to minimise the incompatibility with PowerPoint while providing a number of new interesting features. While enormous effort for a pen-based tablet presentation system has been made, they all tried to develop a new system for their purpose. Slight differently, we consider the pen and the tablet an interface for an existing system. That system, most importantly, is the most well-known presentation software, PowerPoint. We utilised PowerPoint as the core component for slides creating and presenting. Therefore our interface becomes a plugable remote interface which could be switched off anytime without influencing the presentation.

We also noticed some limitations which are because of how PowerPoint organise the presentation content. Although solutions such as MindXpres address this issue, a pen-based mobile interface can still be a good add-on to these systems.

6.1 Contributions

Our first contribution is the architecture of our system. We have already proven the feasibility of our architecture. The interoperability with PowerPoint enables us to control the behaviour of PowerPoint at a very subtle level. The idea of saving our data into PowerPoint without any risk of losing it is also very interesting. Moreover, the architecture can be easily reused to utilise other interfaces. For example, our client side could be easily replaced with a voice controlled or a gesture based interface. In that case, the only component needs to change on the server side is the command sets. As long as the interface interactions could be properly mapped to the PowerPoint functions, any client side could be compatible.

The second contribution of this thesis is our design of the interface and functions. Our system fully utilised the screen area offered by the tablet and defined some useful strokes for the stylus. For example, drawing square and straight line are, according to some of the colleagues, "both simple and useful".

The third contribution is the concept of a stamp in a pen-based system. In our implementation, the stamp could only be drawn and saved with the stylus. Because the tablet stylus is still not perfect for drawing, the stamps are normally just simple diagram. We believe that when user could easily create more complex diagram, the use of stamp could be more frequent and pleasant.

6.2 Limitations

Although our system has all the functions needed as a PowerPoint controller, we made some compromises along the way as we could not really change anything within PowerPoint. The very first workaround we introduced is that we tried to stream the presentation to the tablet instead of rendering the file. Doing this is because we don't have a proper library to render slides from a PowerPoint file. Even though this workaround is good to have because a live streaming ensures the complete synchronisation of both sides, we still think it would be nice to utilise the computing power of the tablet.

The other issue we noticed is that sometimes the reproduced image on the client side is not as smooth as the server side. We assume that this is because the robot moved the mouse too fast thus PowerPoint failed to capture the path. We set a minimal delay to the robot and the issue is solved. However,

this is not a perfect solution, we need the PowerPoint API which is not yet there to directly create the diagram.

6.3 Future Work

Since we spent most of our time developing the basic architecture of the system, we still have in mind a lot of future possibilities. The system could be improved in detail as well as extended in the general architecture. We will first discuss the subtle parts then some architectural changes.

6.3.1 Desktop Operation

As we all know, sometimes lecture content is not only the slides. Right now, the presenter could not show anything else from their computer only by using our system. However, it is definitely possible. Since we streamed the presented screen to the tablet, we could, in theory, control the Windows system as well. All there left to do is to properly define a set of interaction and mimic them as groups of mouse actions. Though we indeed used a robot to mimic mouse in order to draw on the presentation, to obtain fully control of the Windows system is a totally different story. Simple mimicking the mouse with the tablet interface could very likely results in an unusable system. Therefore, we put this into the future work for anyone who is particularly interested.

6.3.2 Universal Compatibility

As we introduced, the system could be quickly ported to the iOS and Windows 10 Mobile systems. What is also portable is the server side. As long as we could find a solution to interop with another presentation software, we could support that presentation software on the server side with a small amount of coding. As for client side, we could develop a set of interfaces for all supported software. When first connected, the server could notify the client which software it is using. Thus the client only need to load the corresponding interface for the system to work. Doing this, our system could become real universal compatible on both server side and client side.

6.3.3 Multi-user Support

One last feature we really expect this for the system to have is the multi-user support. Benefiting from the client-server architecture, we could allow

the audiences' devices to connect to the server. However, we need to think carefully here about how do we manage different user rights. We would like to give the audience the ability to submit digital ink only when permitted by the presenter. To achieve this, some careful consideration and implementation are required.



Appendix

References

- Abowd, G. D. (1999). Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment. In (Vol. 38, pp. 508–530). IBM.
- Abowd, G. D., Atkeson, C. G., Feinstein, A., Hmelo, C., Kooper, R., Long, S., ... Tani, M. (1997). Teaching and Learning as Multimedia Authoring: the Classroom 2000 Project. In *Proceedings of the Fourth ACM International Conference on Multimedia* (pp. 187–198).
- Anderson, R., Anderson, R., Davis, K., Linnell, N., Prince, C., & Razmov, V. (2007). Supporting Active Learning and Example Based Instruction with Classroom Technology. *ACM SIGCSE Bulletin*, 39(1), 69–73.
- Anderson, R., Anderson, R., Davis, P., Linnell, N., Prince, C., Razmov, V., & Videon, F. (2007). Classroom Presenter: Enhancing Interactive Education with Digital Ink. *Computer*(9), 56–61.
- Anderson, R., Anderson, R., Hoyer, C., Prince, C., Su, J., Videon, F., & Wolfman, S. (2005). A Study of Diagrammatic Ink in Lecture. *Computers & Graphics*, 29(4), 480–489.
- Anderson, R., Hoyer, C., Prince, C., Su, J., Videon, F., & Wolfman, S. (2004). Speech, Ink, and Slides: the Interaction of Content Channels.

- In *Proceedings of the 12th annual ACM international conference on Multimedia* (pp. 796–803).
- Anderson, R. J., Hoyer, C., Wolfman, S. A., & Anderson, R. (2004). A study of Digital Ink in Lecture Presentation. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems* (pp. 567–574).
- Android Team. (2016). *Android Development Best Practices*. Retrieved from <http://developer.android.com/guide/practices/index.html> ([Online; accessed 15-May-2016])
- Apte, A., & Kimura, T. D. (1993). A Comparison Study of the Pen and the Mouse in Editing Graphic Diagrams. In *Visual Languages, 1993., Proceedings 1993 IEEE Symposium on* (pp. 352–357).
- Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An Empirical Evaluation of the System Usability Scale. *Intl. Journal of Human-Computer Interaction*, 24(6), 574–594.
- Berque, D., Bonebright, T., & Whitesell, M. (2004). Using Pen-based Computers across the Computer Science Curriculum. In *Acm sigcse bulletin* (Vol. 36, pp. 61–65).
- Bligh, D. A. (1998). *What's the Use of Lectures?* Intellect books.
- BlueCove Team. (2008). *BlueCove - Bluecove jsr-82 Project*. Retrieved from <http://bluecove.org/> ([Online; accessed 15-May-2016])
- Brooke, J., et al. (1996). SUS-A Quick and Dirty Usability Scale. *Usability Evaluation in Industry*, 189(194), 4–7.
- Center for Science and Mathematics and Engineering Education, & Committee on Undergraduate Science Education. (1999). *Transforming Undergraduate Education in Science, Mathematics, Engineering, and Technology*. Joseph Henry Press.
- Creed, T. (1998). *PowerPoint No! Cyberspace Yes! The National Teaching and Learning Forum 6 (4)*. Greenwood Publishing Group.
- Damajia. (2016). *damajia/AndroidSwipeLayout*. Github. Retrieved from <https://github.com/damajia/androidswipelayout> ([Online; accessed 15-May-2016])
- Dimond, T. (1957). Devices for Reading Handwritten Characters. In *Papers and Discussions Presented at the December 9-13, 1957, Eastern Joint Computer Conference: Computers with Deadlines to Meet* (pp. 232–237).
- Fagen, W., & Kamin, S. (2013). Measuring Increased Engagement Using Tablet PCs in a Code Review Class. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 465–470).
- Frederick, R., Jacobson, V., & Design, P. (2003). Rtp: A Transport Protocol for Real-time Applications. *IETF RFC3550*.

- Gaskins, R. (2007). PowerPoint at 20: Back to Basics. *Communications of the ACM*, 50(12), 15–17.
- Gray, E. (1888, July 31). *Telautograph*. Google Patents. (US Patent 386,815)
- Iles, A., Glaser, D., Kam, M., & Canny, J. (2002). Learning via Distributed Dialogue: Livenotes and Handheld Wireless Technology. In *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community* (pp. 408–417).
- Java Native Access Group. (2016). *Java Native Access*. Retrieved from <https://github.com/java-native-access/jna> ([Online; accessed 15-May-2016])
- Jawin Team. (2005). *Jawin*. Retrieved from <http://jawinproject.sourceforge.net/> ([Online; Accessed on 05/11/2016])
- Kam, M., Wang, J., Iles, A., Tse, E., Chiu, J., Glaser, D., ... Canny, J. (2005). Livenotes: a System for Cooperative and Augmented Note-taking in Lectures. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 531–540).
- Kamin, S., Hines, M., Peiper, C., & Capitanu, B. (2008). A System for Developing Tablet pc Applications for Education. In *ACM SIGCSE Bulletin* (Vol. 40, pp. 422–426).
- Kim, H.-s., Kim, H., Yoon, I., Jung, C.-H., & Hwan, Y. (2015). User Performance Differences between Graphics Tablet and Mouse in Graphic Applications: Focus on Controllability and Accuracy. *International Journal of Multimedia and Ubiquitous Engineering*, 10(9), 167–174.
- Koile, K., Chevalier, K., Rbeiz, M., Rogal, A., Singer, D., Sorensen, J., ... Wu, K. (2007). Supporting Feedback and Assessment of Digital Ink Answers to In-class Exercises. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE* (Vol. 22, p. 1787).
- Koile, K., Reider, D., & Rubin, A. (2010). *INK-12: a Pen-based Wireless Classroom Interaction System for K-12*. Purdue University Press.
- Koile, K., & Singer, D. (2008). Assessing the Impact of a Tablet-pc-based Classroom Interaction System. *The Impact of Tablet PCs and Pen-based Technology on Education. Evidence and Outcomes*, 73–80.
- Lindquist, D., Denning, T., Kelly, M., Malani, R., Griswold, W. G., & Simon, B. (2007). Exploring the Potential of Mobile Phones for Active Learning in the Classroom. In *ACM SIGCSE Bulletin* (Vol. 39, pp. 384–388).
- Markett, C., Sánchez, I. A., Weber, S., & Tangney, B. (2006). Using Short Message Service to Encourage Interactivity in the Classroom. *Computers & Education*, 46(3), 280–293.

- McNeill, D. (1992). *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press.
- Microsoft. (2012). *Office Primary Interop Assemblies*. Retrieved from <https://msdn.microsoft.com/en-us/library/15s06t57.aspx> ([Online; accessed 15-May-2016])
- Microsoft. (2015a). *Microsoft Office*. Retrieved from <https://products.office.com/en-US/> ([Online; Accessed on 05/11/2016])
- Microsoft. (2015b). *Office Remote*. Retrieved from <http://research.microsoft.com/en-us/projects/officeremote/>
- Norrie, M. C., Signer, B., & Weibel, N. (2006). General Framework for the Rapid Development of Interactive Paper Applications. In *Proc. CoPADD* (Vol. 6, pp. 9–12).
- ogaclejapan. (2016). *ogaclejapan/ArcLayout*. Github. Retrieved from <https://github.com/ogaclejapan/arclayout> ([Online; accessed 15-May-2016])
- Oracle. (2016). *Java Native Interface Specification*. <http://docs.oracle.com/javase/7/docs/technotes/guides/jni/spec/jniTOC.html>. ((Accessed on 06/08/2016))
- Parker, I. (2001). Absolute Powerpoint: Can a Software Package Edit Our Thoughts? *The New Yorker. Annals of Business Section*, 76–87.
- Parker, I. (2004). Presentations and the PowerPoint Problem. *Business Communication Quarterly*, 67(1), 78–80.
- Ploetzner, R., Dillenbourg, P., Preier, M., & Traum, D. (1999). Learning by Explaining to Oneself and to Others. *Collaborative learning: Cognitive and computational approaches*, 103–121.
- Reuss, E. I., Signer, B., & Norrie, M. C. (2008). *Powerpoint Multimedia Presentations in Computer Science Education: What do Users Need?* Springer.
- Richard Anderson, B. S. S. A. W. T. V. K. Y., Ruth Anderson. (2004). Experiences with a Tablet PC-based Lecture Presentation System in Computer Science Courses. In *ACM SIGCSE Bulletin* (Vol. 36, pp. 56–60).
- Roels, R., & Signer, B. (2013). An Extensible Presentation Tool for Flexible Human-information Interaction. In *Proceedings of the 27th International BCS Human Computer Interaction Conference* (p. 59).
- Roels, R., & Signer, B. (2014). MindXpres: An Extensible Content-driven Cross-media Presentation Platform. In *Web Information Systems Engineering–WISE 2014* (pp. 215–230). Springer.

- Rowe, G. (2015). *GitHub - gary-rowe/hid4java: A cross-platform Java Native Access (JNA) Wrapper for the signal11/hidapi Library*. <https://github.com/gary-rowe/hid4java>. ((Accessed on 06/02/2016))
- Shallcross, D. E., & Harrison, T. G. (2007). Lectures: Electronic Presentations versus Chalk and Talk—a Chemist’s View. *Chemistry Education Research and Practice*, 8(1), 73–79.
- Shilman, M., Wei, Z., Raghupathy, S., Simard, P., & Jones, D. (2003). Discerning Structure from Freeform Handwritten Notes. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on* (pp. 60–65).
- Signal11. (2012). *GitHub - signal11/hidapi: A Simple Library for Communicating with USB and Bluetooth HID Devices, howpublished = https://github.com/signal11/hidapi, note = (Accessed on 06/02/2016)*.
- Signer, B., & Norrie, M. C. (2007). PaperPoint: a Paper-based Presentation and Interactive Paper Prototyping Tool. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction* (pp. 57–64).
- Subrahmanyam, K., Kraut, R. E., Greenfield, P. M., & Gross, E. F. (2000). The Impact of Home Computer Use on Children’s Activities and Development. *The Future of Children*, 123–144.
- Taylor, D. (2007). Death by PowerPoint. *Developmental Medicine & Child Neurology*, 49(5), 395–395.
- Tufte, E. R. (2003). *The Cognitive Style of PowerPoint* (Vol. 2006). Graphics Press Cheshire, CT.
- Wilkerson, M., Griswold, W. G., & Simon, B. (2005). Ubiquitous Presenter: Increasing Student Access and Control in a Digital Lecturing Environment. *ACM SIGCSE Bulletin*, 37(1), 116–120.
- Williams, S., & Steinweg, S. (2009). Use of Tablet PCs in Teacher Education and K-12 Settings. In *Society for Information Technology & Teacher Education International Conference* (Vol. 2009, pp. 2189–2192).
- Wuensch, K. L. (2005). What is a Likert Scale? and How Do You Pronounce ‘Likert’? *East Carolina University*.