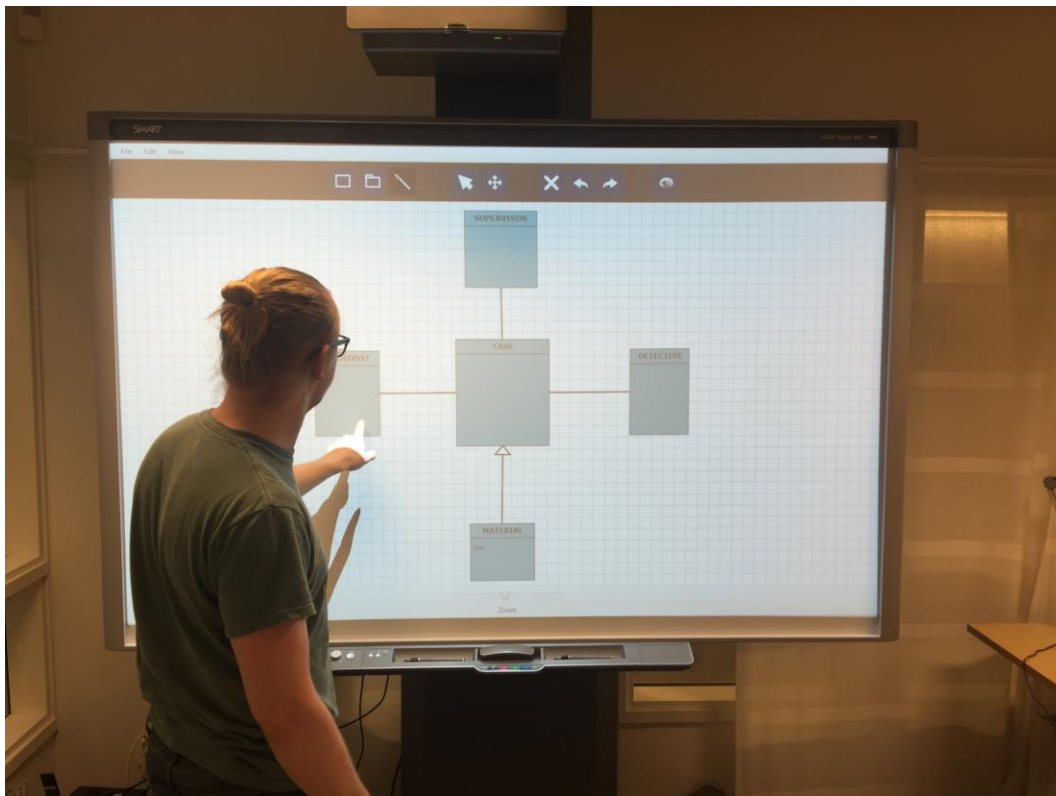




UNIVERSITY OF GOTHENBURG



Integrating Voice Commands for Software Modelling using UML on a Smart Board

A Qualitative Study

Bachelor of Science Thesis Software Engineering and Management

JOHAN HERMANSSON

EMIL SUNDKLEV

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
Göteborg, Sweden, June 2016

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Integrating Voice Commands for Software Modelling using UML on a Smart Board A Qualitative Study

Johan Hermansson
Emil Sundklev

© Johan Hermansson, June 2016.

© Emil Sundklev, June 2016.

Examiner: Regina Hebig

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden June 2016

Integrating Voice Commands for Software Modelling using UML on a Smart Board

A Qualitative Study

Johan Hermansson
Software Engineering and Management
University of Gothenburg
Gothenburg, Sweden
johan.hermansson@hotmail.com

Emil Sundklev
Software Engineering and Management
University of Gothenburg
Gothenburg, Sweden
emil.sundklev@gmail.com

Abstract— Software modelling on a Smart board is a fairly new approach and using the same interaction modalities as on a computer might not be the most efficient way. The software modelling process could potentially be improved by integrating voice recognition for some of its functions. We implemented voice commands to a software modelling tool and let 14 variously experienced students test it and reflect on the possibilities of using this tool in the future. Our results will show that the Smart board presents a good environment for collaborative work and that the voice commands can eliminate the need for using a keyboard and thus making the process more efficient and interactive.

Keywords—Software Engineering, Smart Board, Voice Recognition, UML-Modeling

I. INTRODUCTION

The Unified Modelling Language (UML) is a well-known and a commonly used language for modelling the system architecture of a software [1][2]. Today there are many different applications for different platforms (computers, tablets and smartphones) that can help any software architect or programmer to draw a UML-model that will represent their system. As new technologies are introduced to the software industry, the software that already exists for the currently used platforms, are not always functional for the newly introduced platforms.

Smart boards are one of the platforms that has been proven to be more and more usable for sharing information. They increase the learning outcome for students in the classroom [3][4] and enhance the effectiveness and interaction at company meetings [5]. The evolution of the Smart board and its functions, makes it a potential tool for improving the effectiveness and usability for the development of software, and more specifically, UML-modelling. The usage of the Smart board is most likely to be done while standing up in front of it, therefore, mouse and keyboard are neither the most effective nor user friendly way of using it, as bending down to a table or using the built in keyboard of the Smart board would be both counterproductive and not very ergonomic. So to bypass this

issue, another way of possibly working would be to introduce voice recognition, as this could both terminate the need of the keyboards and that the usability and effectiveness could be improved even further [6].

A. Purpose of the Study

The purpose of this study is to study the usability and effectiveness of a modelling tool on a Smart board, by implementing voice recognition for some of its functions. Our research strategy will be of a design research methodology [7], and a qualitative data collection, as we want to go more into depth how a user would preferably interact with the Smart board.

B. Research Questions

1. In what way can voice interaction help improving the modelling process on a Smart board?
2. Which tasks in modelling are candidates for voice-interaction on a Smart board?

II. BACKGROUND

A. Smart Board

A Smart board is an interactive whiteboard that is mainly used for education in schools and courses, but also for common areas in offices. It has been shown that it increases the learning outcome for students, as it helps both teachers and students to be more interactive with each other, since the students attention to the teacher and the board increases, and the teachers will to engage the students are also increased [3][4].



Figure 1. SMART Board 800 Series [8]

The Smart board to be used in this study is a Smart Board 800 (see Figure 1), provided by the company SMART, with a UX80 projector above the board, with a multi-touch interactive whiteboard, which makes it possible for two people to work on it simultaneously.

B. PenguinUML

PenguinUML is an environment software for modelling, made by two master-students at Chalmers, University of Technology. It is a UML-modelling tool written in Java, that are made to be used on a Smart board to design UML-diagrams. They developed PenguinUML for their Master thesis and they gave us permission to modify it, so we could implement voice recognition into the application.

C. Sphinx4

Sphinx4 is an Open Source voice recognition library in Java, made by Carnegie Mellon University(CMU). By implementing Sphinx4 in PenguinUML we can take voice-inputs for different tasks and features, while creating diagrams on the Smart board.

D. Our version of PenguinUML

We took the latest version of PenguinUML, and added the Sphinx4 library to it, so we could implement a few features with voice control.

The two features that are supported with voice recognition in our version of PenguinUML are:

- Changing tools
- Naming classes

Changing tools means that you can change between the different modes, like creating classes, packages and edges. Naming classes simply means that you can name the classes, which would be one of the more important function in the prototype for our tests, as this is where we expect the users to appreciate voice recognition the most. For a complete list of voice commands see Appendix 1. Because of time constraints for the development on our version of PenguinUML, we could not use a dictionary to give the classes and packages any possible name. Instead we had to create a grammar file with expected words to be used as a title. As long as we take the precautions to make sure the titles needed for the tests are in the grammar file, this would not affect the results of our tests,

as it would still represent a functional application for any sort of system, and not just the one used in our test.

Our application and instructions how to install it can be found on our GitHub page:

<https://github.com/SodaSaft/PenguinUML>

III. RELATED WORK

Chaudron and Jolak [9], write about their vision for a new generation of software design environments, a multiplatform where everything related to software development should be available. It includes several interaction modes, like voice, touch and gesture in different environments, like Smart boards, PC and tablets.

In the study by Lackey et al. [6], many different software and tools are tested and evaluated with productivity in mind. Speech recognition software was evaluated to save a lot time, by saving time on speaking instead of typing or writing, for any sort of activity. So according to this study, our prototype should prove to increase the effectiveness and productivity while designing UML-models.

In the study by Lahtinen and Peltonen [10], they are talking about computer-aided software engineering (CASE)-tool and their hidden information and difficulty of how to use them properly, therefore they looked at new ways to improve the tools, by making it more effective and user-friendly, and they put their focus on voice recognition. They build a prototype and test it on different users. Their findings stated that “UML is a favourable domain for speech recognition, speech recognition can be used to enhance the usage of UML CASE-tools, and our approach is viable”. This study has full focus on using the tool on a computer and not a Smart board, so our work must have focus on other aspects of the prototype application, as there are quite a few differences between the standard desktop computer setup and a Smart board.

Even though modelling is an important stage of software development and that there are many different CASE-tools supporting modelling in different levels, it’s still not as widely used as it perhaps should be [2][11]. Smart boards have been tested and proven to be a useful tool for modelling UML, not only effective, but also user-friendly and supports co-operative working environments [11].

Repetitive strain injury (RSI), is a condition that makes it difficult for some people to do different tasks, like using a keyboard or simply moving specific body parts. Several studies have gone out to help programmers with RSI, by using voice recognitions [12-14], but in the study of Mills et al. [15], they point out that these aids have a high rate of faults and errors, and that it is not commonly used. It would be interesting to see if modelling might be more suitable than programming to be integrated with voice recognition.

IV. RESEARCH METHOD

A. Research Strategy

As the goal of this report is to identify ways of improving the modelling process, specifically on a Smart board using

voice recognition, and the best possibly way of doing this, we decided that Design Research were the most suitable approach for us. Design Research is focused on the development of solutions to fulfil the needs of the end-users, just as our research is focused on, by increasing the effectiveness and usability of modelling. This is done using our prototype application, by comparing the results from our tests, where our subjects are using the tool both with and without voice recognition, and this helps us to answer our research questions.

B. Data Collection

Our data collection consists of four different parts: *Introducing the application*, *The actual test*, *System validation* and *Interview*.

1) Introducing the application

This is where we introduce the different functions of the application that we developed and requires the subjects to test. They will not be thrown into the deep side of the pool when starting *The actual test*. We provide them with a sheet which contains voice commands that can be used. We also talk about how the entire test will be done by shortly describing each of the three next upcoming phases.

2) The actual test

Subjects are given a task, where the goal was to develop a software for management of investigations of a Police Department, for the full task see Appendix 1. Based on this text, the subject models the system using the Smart board and the PenguinUML application with voice recognition. Since this is a qualitative study and not an experiment, we only have one and the same task for each subject, and they only test the application with voice recognition, and not the one without it. The task was chosen since it has been used in an experiment before, and this will help us to predict as many possible titles as we could for the classes and packages, for defining a good grammar list, by looking at the data from the experiment.

3) System validation

In this phase our subjects fill in a form for the system that they just tested. We use the System Usability Scale (SUS) [16-18], SUS is a questionnaire with ten fixed questions with five alternative answers scaled between strongly disagree to strongly agree, see Appendix 2 for the complete questionnaire. Based on the different alternative answered for each question we calculate a score based on the SUS algorithm. The calculated score can land between 0-100, where higher is better. Later our applications SUS-score is compared with the version tested by the two master students from Chalmers, which is the same system, but without voice recognition. This comparison helps us to some degree, to see if we have been taking the development of PenguinUML and also the modelling process in general, in the right direction, which is to increase its usability and effectiveness.

4) Interview

The final part is done by discussing the system with the subject. Unlike the SUS questionnaire, this interview is a more open discussion about the test and application with only a few fixed questions and answers, thus to get feedback and a more in-depth knowledge of their opinions of the applications and

their thoughts on the general idea of using voice recognition while modelling UML on a Smart board. The interview is structured into two parts, the first one is about asking six questions regarding the subjects modelling experience etc. The second part is about the subjects' 16 questions on their thoughts on the application, its functions and future possibilities, for the complete list of interview questions see Appendix 3.

C. Selected Subjects

The subjects who are selected for the test, must fulfil these two requirements:

- English speaking capabilities
- Basic knowledge of UML

These two requirements are based on the fact that if you cannot speak English and do not have any knowledge of UML, you are out of the scope of the target group that this application is intended for. Since this system has been tested before, by the two master students from Chalmers, we have the opportunity to test a few of the same people that already has some experience with the Smart board and PenguinUML (without voice recognition), so they can give us some additional feedback and data on the difference between modelling on a Smart board with and without voice recognition. But we want the majority of our subjects to have no experience with PenguinUML, since we can already look at some of the data gathered by the tests held by the two master students.

D. Data Analysis

During the entire test, we observed, recorded video and audio of the subjects while they were trying and completing their given tasks, and when we had our discussion with them afterwards.

To analyse our data, we decided to go with the grounded theory, which is a rather general methodology for research, and is suitable for "every kind" of research questions [19]. We coded the transcripts of the interviews, but also collected data from observations during *The actual test* phase. As more and more code is generated, it will be shown to be similar to each other, and you will be able to map them in a different number of concepts. As all the coding is done, and no more concepts can be discovered, we came up with a few number of categories, that are made from a group of similar concepts. These are the categories we present in the result section:

- General Findings
- Prototype Testing
- System Usability Scale (SUS)
- Comparing different Tools and Interaction Modes
- Importance of different Voice Functionalities
- Smart Board Modelling using Voice Recognition in Collaboration with other People

From these categories, we can come up with a conclusion for our research questions.

V. RESULTS

In this section we present our findings from the tests, interviews, SUS-questionnaire and some of the data taken from the two master students from Chalmers. They are presented according to the categories listed in the *Data Analysis* section.

A. General Findings

We gathered our data evaluating our software and interviewed 14 subjects individually. We asked them all to put their modelling skills on a scale between one and five, where one represented a novice modeller and 5 a professional modeller. The majority of the subjects put themselves in a skill level of three, but there was a deviation between two and five on the skill level scale (see Figure 2). All the subjects had used pen and paper to sketch or draw a UML-diagram before, 12 out of them had modelled using a whiteboard and 13 of them had used one or more different software-tools to draw models, like Papyrus, Lucidcharts, Rhapsody etc.



Figure 2. Subjects software modelling skill level

Four of the subjects had previous experiences with a Smart board before the test. Two of them had participated in the previous PenguinUML tests, without the voice recognition utility. Only one of those four had used a Smart board more frequently during school time.

We asked the subjects on their opinion of how important the modelling process is when developing a good system/application, and on a scale from one “not important” to five “very important” (see Figure 3). The majority thought that it was important or very important, whereas only two of them thought that it was not that important.

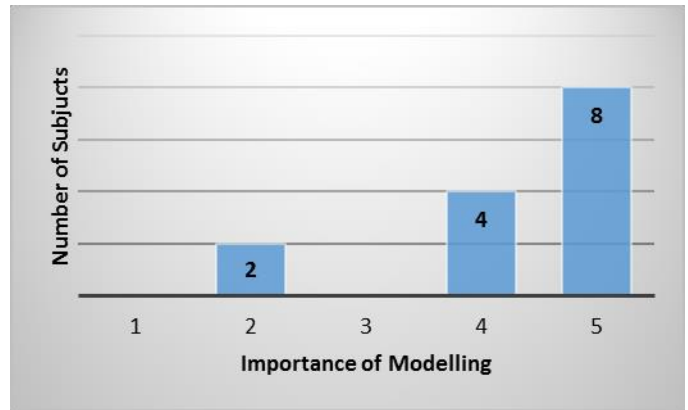


Figure 3. Importance of modelling

We asked the subjects on their general thoughts advantages on modelling on a Smart board, both with and without voice recognition, one bachelor student from Software Engineering and Management (SEM) said:

“My first thought was that this was fun, and I think it would be easy for everyone in the room to see what’s going on”

And another bachelor student from SEM said that:

“It’s easy to use, and you don’t need to be able to draw because if you draw it on the whiteboard it could get really messy, but if you do it on the Smart board you can keep it neat and you can expand as big as you need it”

We also asked them about their thoughts on disadvantages for modelling on a Smart board, both with and without voice recognition, a SEM bachelor student said:

“I don’t really see any big disadvantages right now, except that maybe if you are in a big room with a lot of people and uses the voice recognition it might pick up wrong commands”

And a software engineering (SE) student at Chalmers pointed out:

“The board is expensive. And just another extra cost to a projects initial modelling process, which is quite short”

B. Prototype Testing

While the subjects executed their task, we gathered some data on how well the application performed, by looking on how many voice commands were used, and how many times it failed in three different ways, faulty tool change, faulty name inputs and unrecognized commands (see Figure 4). This data can be used to see where the system fails the most and where it needs improvement.

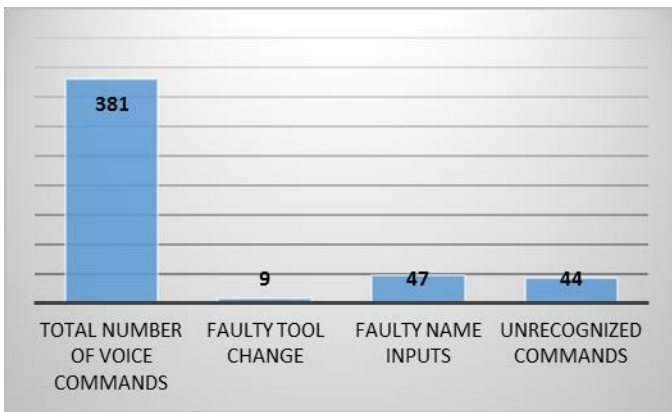


Figure 4. Usage and faults of voice commands

The average number of voice commands used by each subject were 27, the lowest amount of voice commands used by one subject were nine and the highest number of voice commands used by one subject were 42.

The average number of faults per voice command was 0,26. The subject with the highest number of faults per voice command were 0,42 and the subject with the lowest number of faults per voice command were 0,12.

We asked them after the test what they liked about the application, and a PhD student said:

“I liked the voice commands and the idea of using them when doing the task”

The subject who used the most voice commands in total said:

“You can switch tools vocally, and that’s quite good because at first you tap on the screen every time, but then you get familiar with the system, so you use more and more voice commands”

A SEM bachelor student thought:

“This application is really good. Simplistic, you have these commands and they are the only things you need. There are a lot of other tools that I have seen that have a lot of things that you never use or isn’t necessarily. So this is perfect when you want to do a UML-diagram”

The same student also mentioned this when asked if there was anything he disliked about the application:

“No, not really, besides the voice is a bit weird sometimes. You have to speak up. But you get used to it after a while, then it became easier”

We also asked the subjects to rate the level of difficulty on three different aspects, learning the system, using the system

overall and the difficulty in using the voice commands. The scale was between one “very easy” and five “very hard” (see Figure 5).

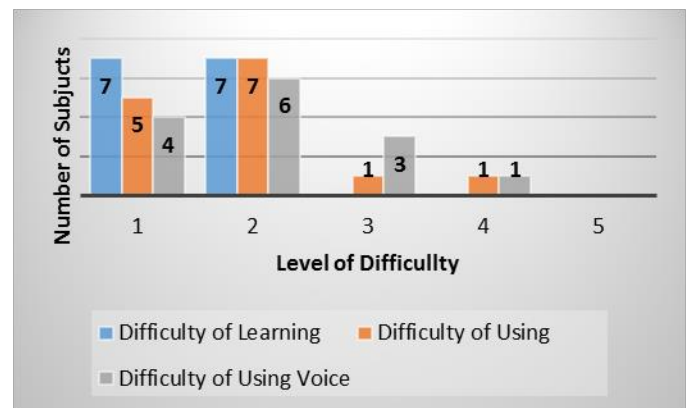


Figure 5. Difficulty level on learning and using the application

Overall, everyone thought it was very easy or easy to learn the system, and the majority of the subjects thought that it was easy to use, both the system as a whole and the voice commands.

A Research Engineer at Chalmers and Gothenburg University, that thought it was hard to use the voice commands said:

“Sometimes I needed to think twice and forgot the voice commands. But I think it will be easier after using it more.”

We also asked them if they thought the application was effective while using the voice commands, on a scale from one “not effective” to five “very effective”.

The majority of the subjects thought that it was quite effective, but some of them thought that it was not that effective and that the application needed to be improved (see Figure 6).

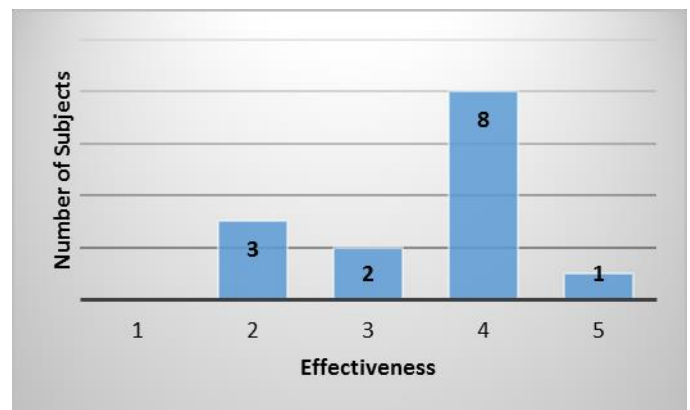


Figure 6. Effectiveness of using voice commands

A master student at Chalmers that rated the effectiveness with a four, said:

"It made it easier in a way that you could multitask, tell it to do one thing and start thinking about the next thing to do while doing the first thing and so on. Like say "Create class" and then be ready to tap the screen"

A PhD student who rated it with a two because the application did not behave as expected at all times, said:

"I would say that it's not very effective in the way it is working right now"

C. System Usability Scale (SUS)

When the test phase was over we gave the subject a SUS-form to fill in. The final average SUS-score was 74,6, which is a pretty good score. The lowest individual score given was 35 and the highest individual score given was 95. The meaning of this score is brought up in the discussion section.

D. Comparing different Tools and Interaction Modes

We asked the subjects about comparing their previous modelling tools that they have used before the test (pen and paper, whiteboard, software-tools etc.), with modelling on the Smart board using the voice commands as an interaction mode.

Comparing the Smart board and a whiteboard, a SEM student said:

"It scales much easier, so if I would like to add more functionalities and more attributes then a tool like this helps more than a whiteboard, because then I don't have to redraw everything"

And when comparing the Smart board with software-tools on a regular computer, another SEM student mentioned:

"If you are drawing the UML yourself, I think it's easier to do it on a regular computer if you're used to that, but if you are having a discussion with a few people about the design, then I think this approach is much better"

A PhD student that prefers pen and paper said this:

"I haven't used voice recognition for any modelling-tool before, but I think telling the computer what to do by using voice commands could be good. Although I have a fear that the system could be very complex to develop and use. I would like to use voice but I'm a little bit of resistant of it"

We also asked if the voice commands simplified the modelling process on the Smart board, where one SEM student said:

"When naming the classes, it's simpler because then you don't have to use the keyboard"

And a PhD student also said:

"I think it will reduce the number of clicks on the Smart board and you don't have to use the keyboard"

A SEM student thought about the current state of the application and said:

"At this level it didn't simplify it since it was a learning curve involved. But I can imagine it will in the future but not in this moment"

When we asked the subjects if they thought that the usability increases for a modelling tool that uses many interaction modalities (keyboard, mouse, touch, voice etc.), one SEM student pointed out:

"Yes, but the Smart board loses some of its "magic" if you use the keyboard next to it. You should not have to walk away from the smart board because then it would be faster to do it on a regular computer"

And a research engineer who does not like the concept of voice recognition said:

"Yes of course it can. I wouldn't use it, but I think other people would find it very useful, especially those with disabilities"

A master student also mentioned:

"It would if you don't force the user to use all of them, so you have it as an option. But if you have to combine all of them it would just confuse the person"

Another master student who thought touch and voice was enough said:

"I think touch and voice is enough. Just make sure the application covers more functionalities"

A PhD student also said:

"I think it's helpful, because I have read some papers that say modelling tools are complex to learn and complex to use, and I think new interaction techniques can solve this problem, and between these interaction techniques, I believe that voice recognition can be a solution"

During the test, all the subjects had to name the classes as they saw fit, and all of them did that by using both voice commands and by simply using the keyboard. Those who wanted to add attributes and operations had to use the keyboard at the side or the built in keyboard on the Smart board, since it was not supported to add those using the voice commands. When asking them on which interaction mode they thought were the best, one master student said:

“I think you should only use voice commands, and I also want to be able to add attributes and operations with my voice instead of typing, so yes I think it’s better. It’s more interactive and as soon as I start using my voice I want to continue to use it, instead of going back and forth between the keyboard and the smart board”

And a PhD student who thought using the keyboard with that Smart board might not be that ergonomic, said:

“Yes I think it’s better, since it was not very adequate or suitable to change from the smart board to the keyboard to write, and also you don’t see what you write on the screen”

E. Importance of Different Voice Functionalities

When the subjects were testing our prototype, it only supported two different voice commands, naming the classes and the selection/change of tools. We asked them to rate the importance of those two functions, but also a few additional functionalities that could be implemented in the future that supports voice recognition. The scale was one “not important”, to five “very important”. These were the average rate of importance for each functionality:

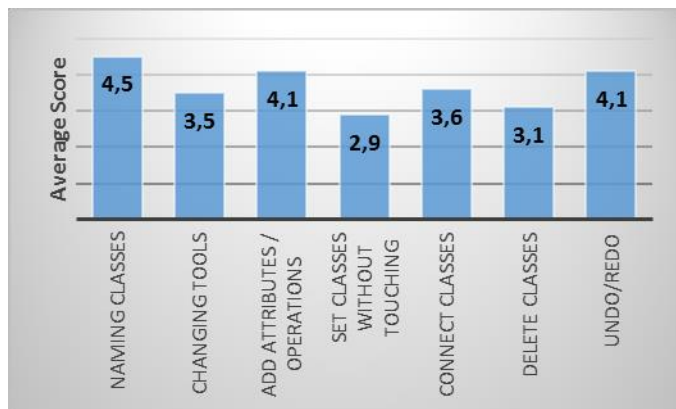


Figure 7. Average score of functionalities importance

Besides these functionalities that we brought up, we also asked everyone if there were any other features that they could think of, that were supported by voice recognition. These features listed below were mentioned by at least one subject:

- Name, change association types on edges

- Save, open, import and export files
- Create a new diagram
- Exit the application
- Rearrange the classes and packages
- Select and deselect classes, packages or edges
- Zoom in and out
- Define your own voice commands

F. Smart Board Modelling using Voice Recognition in Collaboration with other People

The final topic of discussion was about using this modelling tool with voice recognition, in collaboration with other people. In general, the thoughts on this were very positive and most responses were quite similar. One master student said:

“It’s maybe the most important part and utility of this tool”

Another master student similarly said:

“That’s the only way of using it. If I would be home alone I would use a computer. The good part with this is to show others what you have done and what you are doing”

A third master student said:

“I think it depends on people if they want to use this alone and/or in collaboration with other people, because when you draw those models standing up, you’re thinking a bit more, and are bit more active and so you can focus on your task”

A SEM student also said:

“If it works correctly then I think it would be good, because then you get the ideas on the board fast, and you can remove things fast”

A PhD student pointed out the difficulty of implementing this into the application, and said:

“I think it could be very complicated in terms of voice detection, especially when it comes to when 2 or 3 people using the board at the same time and all try to do different commands. I mean it will happen all the time when you collaborate with other people. Theoretically I think it’s quite complex

but when you have implement it is quite cool”

VI. DISCUSSION

In this section we discuss the results presented in the previous section and the within the same categories. The discussion of the results is going to help us to come to a conclusion and answer our research questions.

A. General Findings

When we chose our subjects we made sure that they all had some prior knowledge of modelling using UML. As our system is a new approach for modelling, it was important that they could compare it with their previous experiences in modelling. The majority of the subjects were bachelors or master students, and since that is where they are introduced to modelling, it makes sense that most of them scaled their skills in modelling somewhere in the middle between novice and professional. They have a good knowledge of modelling, but have not used it professionally yet i.e. in the industry. If they had all rated themselves as beginners or slightly beginners (1-2), the feedback might have been a bit out of scope, more focused on that this is a cool way of modelling or that modelling is boring, and not why the modelling process might be good or not, in comparison to different tools that are used both in schools and industry.

Considering the importance of modelling to develop a good system, 12 out of 14 subjects thought that it was an important phase. This further indicates that modelling is an important part in the development phase of a system. Since the vast majority of the subjects thought that modelling is important, we might have missed out on some feedback from people that think otherwise. That is because people with bad experiences with anything (software, products etc.) will point out the reasons behind it, while the people with good experiences usually won't say anything at all. The two other subjects that thought modelling was not that important told us they rated it low because they simply do not like to model, and from their previous developments, it has not been a necessary phase to make their systems work as expected. Saying that, those two subjects still had a positive experience with our prototype, and saw potential in using it in the future.

When asking about the advantages about modelling on a Smart board, not restricted to only using voice recognition as an interaction tool, every subject had at least one advantage, but not necessarily a disadvantage. The most common advantage brought up were in the topic of using it in a group, while being more interactive with both the system and your colleagues and the most common disadvantage brought up was about modelling on a Smart board would not be very useful while modelling alone. Our results are then indicating that if you are to use the Smart board for modelling, do so while being in a group, since that invites everyone to contribute or at least not being left out of the progress because there are too many people blocking the screen, compared to a regular laptop or computer screen. But if you are modelling alone, it would still be useful but perhaps a bit unnecessary, as you do not need

to include other people in your model or interact with anyone else but the screen.

B. Prototype Testing

Throughout all the tests, every subject used the two different functions supported with voice recognition, changing tools and naming classes. They also chose not to use them at some point, which helped them see the difference in having and not having the voice commands to some extent, and therefore got some ideas on where voice commands could improve the modelling process on the Smart board.

On average every subject thought that it was very easy to learn the different functions in the application, and easy to use the system as whole and with voice commands. Even though a few subjects thought some usages of the system was a bit hard. Considering that the voice commands failed more than ones in every forth try, (26%) on average, you could expect that the subjects would have given the learning and usage scale would be much lower than the usage of the voice commands. But the reason why it is not because at the beginning of working on their task, they did a few mistakes while trying out the voice commands, but as they learned what they did wrong, they corrected their usage accordingly. So after getting more used to the system and its functions and commands, most subjects thought that it was easy. This does not mean that no mistakes were made at the end of the task, but if a mistake was made, the subject knew exactly what was done incorrectly and could easily correct it. This is also connected to how the subjects rated the effectiveness of using the voice commands, as the average score was 3,5, which means it is just slightly more effective than not using voice commands. But that is a score based on how it worked while testing it, and many of the subjects pointed out that if improvements were made, they thought it would eventually be more effective to use the voice commands.

When asking about what the subject liked and disliked about the applications, the responses were very positive, and they saw a lot of potential in using this tool on the Smart board in the future. Most of the subjects thought that the system needed improvements on making the voice commands work better and support more functionalities, and some thought that the system needed to support more different UML-diagrams, for example use cases. One of the subject thought that this system did not need more functions as it was already simple and included the most important aspects of sketching a class diagram. Two subjects also pointed out that many modelling tools have many functions and subsystems that are rarely used, which makes them way too complex. This shows us that people are considering modelling and how to model in different ways. Also if a system does not work as expected, it could become quite frustrating, so that the voice recognition needs to be further improved, and as this was a prototype test, it was already obvious before the tests were held.

C. System Usability Scale (SUS)

We obtained an average SUS-score of 74,6 while PenguinUML (without voice recognition) had an average SUS-score of 78,75, when tested by two master students from Chalmers. This does not look like that big of a difference, but if

we look at their lowest score, 65, and our lowest score, 35, there is quite a huge difference. The reasons for the lower average score and the bigger spread between our lowest and highest score could be traced to a few different reasons.

First thing to mention is that we removed one functionality from PenguinUML for our test, which is the ability to sketch informal classes and edges. We removed them as we did not want to introduce the subjects to too many functionalities, so they did not lose too much focus on testing the voice commands implemented by us.

Even if we removed one functionality from the system, we also added two new functions, each with several combinations of voice commands. Considering that together with what one of the subjects told us about tools having many functions that it becomes too complex to use. This could also be one of the reasons we got a lower average score.

Finally, as almost every subject pointed out, the voice recognition needed to be improved and better integrated into the system. When considering that we removed one function, and replaced it with several functions, that still needs improvements, you can expect the usability score being lower than the score in previous tests.

Previous studies that had a goal of validating SUS and to get an idea of what a good SUS-score would be, say that an average score lands on 70,1, which was estimated after collecting 2324 SUS questionnaires [12]. This means that we obtained a SUS score above average, so our system is still quite usable.

The biggest reason that we calculate a SUS-score, is to be able to track development progress of PenguinUML, as more improvements will be done and to make sure that the development is moving in the right direction.

D. Comparing different Tools and Interaction Modes

Comparing our system on Smart board and its functions with and without voice commands with other tools available for modelling (whiteboard, pen and paper, software tools etc.), is of big importance. This is because we cannot come to a viable conclusion of our system if we cannot show to have any potential beneficial uses in the future compared to these tools.

According to our results we can see that modelling on a Smart board with and without voice recognition compared to modelling on a whiteboard and pen and paper, has some benefits. One thing that was brought up by several subjects was the fact that the model becomes much cleaner and neater on the Smart board, since drawing models on a whiteboard or paper can become quite messy if your handwriting and drawing skills are low. It also scales much easier, in case you want to add, remove or rearrange something in the model, which happens very often during the modelling process, you can simply drag the objects around and add information on the model without the need to redraw anything. Finally, if you have completed a model, you can save it as a file for later use and improvements instead of the need to redraw it again in a modelling software tool when using a standard whiteboard.

Many of our subjects think that when modelling individually, it would be easier and more comfortable to do it on a regular computer, as this is what more or less all of us are the most familiar with. However, at the same time they think that if modelling within a group, the Smart board is preferred over the computer. Because modelling on a computer or laptop would be more cumbersome than to do it on the Smart board, because you have a bigger screen to work on and do not have to fight over who is going to use the keyboard or mouse since you can all stand next to the Smart board, using the touch and voice commands.

There were some differences in opinions between the subjects when asking if the voice commands they could use during the test simplified their modelling process. The function to select the different tools with the voice commands had a few various opinions. Some subjects think that since they are already standing by the Smart board, it is not that big of a reach or hassle to just click on the icons. But some of the other subjects think that it was a good addition to be able to use the voice commands for selecting the tools, so they did not have to make that reach each time they wanted to change the tool.

The majority of the subjects think that using the voice commands for naming the classes is much simpler and better than to use the keyboard. The simplest reason was that it is easier to speak than to write on a keyboard that is not ergonomically placed accordingly to a few subjects. Also that if they use the keyboard next to the Smart board, they think that they lose some of their focus, since they are walking back and forth from the Smart board quite a lot.

Even though there were some differences in opinions about the voice commands simplifying the modelling process, everyone saw the potential of it. When we asked them if they think having more interaction modalities (keyboard, mouse, touch, voice, etc.) available would increase the systems usability, some people said that the keyboard and mouse should be removed since voice and touch would be more than enough to be able to create a good model if it worked correctly and supported every available function. The other subjects think that they should all be available, as they realise that some people might prefer different interaction modes or even might not be able to use some of them because of some sort of handicap. But they all thought that voice should be available when modelling on a Smart board.

E. Importance of Different Voice Functionalities

It is quite clear that the functions that require the use of a keyboard, like naming classes and adding attributes and operations, were the ones that our subjects think are most important to be supported with voice commands. And as discussed before, that is where they think it helped the most. The other functions that can be used by a simple click on the screen, were less important for our subjects, but were still something they think should be available, with the exception of an undo and redo command, which was rated just as high as adding attributes and operations. One reason why the undo and redo functions were rated so high could be that this system is just a prototype, and every subject had to use them at least once, because it was a new way of modelling and that the voice

commands and touch functions did not always work as expected. If the entire system were to be improved and become less faulty, so that the user would not have to use the undo and redo functions as much, the importance of the redo and undo commands could possibly be given a lower rate of importance in the future. But nevertheless, it should definitely be supported in some way with voice recognition. From the tests conducted by the two master students from Chalmers, to use the Smart board for modelling, they asked their participants about some specific future features. The lowest rated feature they brought up, was to name classes using voice commands. They used the same scale as us, and the average score presented in their results were 2,7. The difference between their score and ours (4,5), could possibly be that our subjects actually tested the voice commands for naming the classes, and realised that it was much easier than to name the classes by using the keyboard.

Some of the subject were convinced that it would be really good if they could model on the Smart board, even without the need to touch it. This means that the voice recognition must be able to cover every possible action to create a complete model on the Smart board. It would be an amazing achievement and could prove to be very useful for some people.

When asked about other features or functionalities that the subjects might think of and be useful with the support of voice recognition, we got a lot of good responses and ideas. We did not ask the subjects to rate the importance of the features, but considering how the functions that were rated, the most important features that the subjects brought up by themselves would be the ability to name and set the association types of the edges, since that would also involve using a keyboard. Like mentioned before, even though the functions that can be done by simply clicking the screen i.e selecting, moving, delete, etc. were lower rated, a lot of the new features and functionalities brought up by the subjects i.e saving, exporting, rearranging, etc. could work and be implemented in the same way (for complete list see Result section E). This tells us that even though its lower rated, it is still something that the subjects wants to be able to do using voice commands.

F. Smart Board Modelling using Voice Recognition in Collaboration with other People

Modelling on the Smart board with voice recognition while working with other people, was something that all the subjects think is the best scenario for this specific tool and interaction mode. Almost every subject brought up the topic using this tool alone, would probably be unnecessary. Some of the subjects did point out that it could be difficult to develop a complete system considering issues like background noises, people talking at the same time etc. however, they still liked the concept of it. For us this indicates that future work needs to look even more into collaboration, as this is where Smart board modelling using voice commands should have its main focus on.

VII. THREATS TO VALIDITY

Considering construct validity, for our test we let the subjects try and complete the same task, which could mean that

the test results and their opinions are biased towards this small scenario. The reasons for this is because of the time constraint of the development phase, as we had to adapt our grammar file to include every possible class name for the task and did not have the time to expand the grammar file for additional tasks. Also if each subject would have done two different tasks at two different times, their knowledge of the test and system could have led to some positively biased results.

For internal validity, most of the subjects did not have any knowledge of our application and its functions. To make sure that everyone was given the same possibility to complete their task, we had a small introduction for using the application and gave everyone the same sheet that consisted of the voice commands and the task.

Finally considering external validity, we only had a limited number of subjects for our test because our research strategy does not require that we have a specific minimum amount of subjects to answer our research questions. Most of our subjects were students, and only a few opinions are taken in consideration from people that have been working in the software industry, but as this study is about generalizability of the voice recognitions impact on the software modelling process, this should not be a concern. But future work testing and interviewing more people in the software industry would identify if their opinions might differ from the students.

VIII. CONCLUSION AND FUTURE WORK

Software modelling tools used today can be quite complex to use and can also be time consuming. They might not be suitable to use on some of the new technologies like the Smart board, because it is a completely different way of working i.e. the position of the keyboard is not ergonomic and can cause unnecessary time consumption. That is why the main focus of this study was to find out if voice recognition could improve the usability and effectiveness of a software modelling tool, specifically on a Smart board. We did this by implementing voice commands on an already developed prototype, which was made to make it possible to create UML class diagrams on a Smart board. With the tool supporting a few voice commands, we conducted a test, with a population sample of 14 subjects, which mostly consisted of bachelor, masters and PhD students and they gave their opinions on our application and the concept of using voice commands while modelling on a Smart board.

For our first research question we wanted to answer how voice commands could improve the modelling process on a Smart board. According to our results, the biggest improvements could take place during collaborative modelling sessions. Our subjects pointed out that if working in a group while modelling on a Smart board, using the keyboard, mouse and touch, it can be difficult for everyone in the group to contribute and to understand what the other people are doing. Voice commands in this scenario would eliminate the need for the keyboard and in that turn also the mouse. This makes sure that there is not just one single person using the keyboard and mouse, and thus excludes other people to contribute in the work. Also when people are using voice commands, they are saying exactly what they are doing or what they are creating,

and so the knowledge is shared amongst all of the other group members faster. Another improvement brought up by many of our subjects was that using the Smart board in combination with voice commands, makes them become more interactive with the application and their group members at the same time. That is because they do not lose as much of their focus by walking back and forth between the screen and the keyboard, and can also select the tool needed using voice commands without the need of reaching for buttons.

Our second research question was about identifying what kind of tasks or functions that candidates to be used with voice commands while modelling on a Smart board. We can identify two different categories of the features used while modelling on a Smart board. The first one is where the user is required to use the keyboard, for text input, and the second one is the features that can be done by just a few clicks on the Smart boards screen, like changing the tools or selecting objects. The first category is the one our subjects think is the most important task to be done by voice commands, as described earlier it is more comfortable and easier to just speak instead of typing, and that it is faster as well. The second category is not as important as the first one, but still quite a strong candidate to be supported with voice commands. That is because clicking one time on the screen is not that time consuming or that much of a hassle, as typing on a keyboard next to or on the screen. One thing to take note of is that some of our subjects wanted to be able to create UML diagrams, using only voice commands. For such a scenario, every possible feature and functions is a candidate for voice recognition.

Since we have been focusing our voice recognition integration on the Smart board, tests on using it on other platforms or mediums, for instance tablets, would be interesting, as it might prove to be just as useful on something else than just a Smart board. Other future work for this topic could include many different topics and strategies. The first thing to focus on would be to test this modelling tool with several people at the same time, to see if voice recognition works in such an environment and if there could come any other possible complications. This is needed since a part of our conclusion points towards this tool being mostly suitable for collaborative work. Further implementations and improvements of the different features and functions mentioned in previous sections needs to be done, to identify if some of them might be suitable for voice recognition or not in practice.

ACKNOWLEDGMENT

We want to acknowledge and thank the two master students, Christophe van Baalen and Marcus Isaksson for letting us implement voice recognition to their application, PenguUML, and that we could use some of their collected data to compare with our own. We also want to thank all the participants of our tests, for providing us with a lot of valuable data, feedback and opinions. We want to thank the Department of Computer Science for letting us use their equipment and working space, that we required for this study. We want send a special thanks to Rodi Jolak and our supervisor, Michel

Chaudron, for providing us with their vision and valuable feedback throughout this entire study.

REFERENCES

- [1] Petre, Marian. "UML in practice." Proceedings of the 2013 International Conference on Software Engineering. IEEE Press, 2013.
- [2] Nugroho, Ariadi, and Michel RV Chaudron. "A survey into the rigor of UML use and its perceived impact on quality and productivity." Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement. ACM, 2008.
- [3] Drigas, Athanasios, and George Papanastasiou. "Interactive White Boards in Preschool and Primary Education." *iJOE* 10.4 (2014): 46-51.
- [4] Gursul, Fatih, and Gulsah Bilgic Tozmaz. "Which one is smarter? Teacher or Board." *Procedia-Social and Behavioral Sciences* 2.2 (2010): 5731-5737.
- [5] Fast-Berglund, Åsa, Ulrika Harlin, and Magnus Åkerman. "Digitalisation of Meetings—from White-boards to SMART-boards." CIRP CMS 2015-48th CIRP Conference on MANUFACTURING SYSTEMS Research and Innovation in Manufacturing: Key Enabling Technologies for the Factories of the Future. 2015.
- [6] Lackey, Amanda E., et al. "Productivity, part 2: cloud storage, remote meeting tools, screencasting, speech recognition software, password managers, and online data backup." *Journal of the American College of Radiology* 11.6 (2014): 580-588.
- [7] Koskinen, Ilpo, et al. *Design research through practice: From the lab, field, and showroom*. Elsevier, 2011.
- [8] [Online] Page visited 2016-04-28, <http://touch-smartboards.com/p/smart-board-885-with-ux80-projector-and-smart-meeting-pro/>
- [9] Chaudron, Michel RV, and Rodi Jolak. "A Vision on a New Generation of Software Design Environments." First International Workshop on Human Factors in Modeling (HuFaMo 2015). CEUR-WS. 2015.
- [10] Lahtinen, Samuel, and Jari Peltonen. "Adding speech recognition support to UML tools." *Journal of Visual Languages & Computing* 16.1 (2005): 85-118.
- [11] Damm, Christian Heide, Klaus Marius Hansen, and Michael Thomsen. "Tool support for cooperative object-oriented design: gesture based modelling on an electronic whiteboard." Proceedings of the SIGCHI conference on Human Factors in Computing Systems. ACM, 2000.
- [12] Arnold, Stephen C., Leo Mark, and John Goldthwaite. "Programming by voice, VocalProgramming." Proceedings of the fourth international ACM conference on Assistive technologies. ACM, 2000.
- [13] Begel, Andrew. "Spoken language support for software development." *Visual Languages and Human Centric Computing*, 2004 IEEE Symposium on. IEEE, 2004.
- [14] Hubbell, Thomas J., David D. Langan, and Thomas F. Hain. "A voice-activated syntax-directed editor for manually disabled programmers." Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility. ACM, 2006.
- [15] Mills, Stella, Sohrab Saadat, and Dominic Whiting. "Is voice recognition the solution to keyboard-based RSI?." *Automation Congress*, 2006. WAC'06. World. IEEE, 2006.
- [16] Brooke, John. "SUS-A quick and dirty usability scale." *Usability evaluation in industry* 189.194 (1996): 4-7.
- [17] Bangor, Aaron, Philip T. Kortum, and James T. Miller. "An empirical evaluation of the system usability scale." *Intl. Journal of Human-Computer Interaction* 24.6 (2008): 574-594.
- [18] Lewis, James R., and Jeff Sauro. "The factor structure of the system usability scale." *Human centered design*. Springer Berlin Heidelberg, 2009. 94-103.
- [19] Charmaz, Kathy. "Grounded theory." *Qualitative psychology: A practical guide to research methods* (2003): 81-110.

Appendix 1

Task:

Criminal Investigations

We want to develop a software for management of investigations of a Police Department.

The system must support detectives, archivists, and supervisors. The detectives assigned to a case have read access to the data of a case; archivists can create new cases or modify /add details to existing cases, by uploading various material (facts, photographs and other media related to the case). Each material is characterized by a title, a short description, author, and its archiving code. Furthermore, the material can be tagged with the names of the persons associated to them (with role: suspects, witnesses and victims). The materials can be connected between each other. Cases can be closed. Only supervisors can reopen closed cases.

Voice Command Sheet

Tool Selection:

Create Class (Selects the **class** drawing tool)

Create Package (Selects the **package** drawing tool)

Create Edge (Selects the **edge** drawing tool)

Selection Mode (Selects the **select** tool)

Moving Mode (Selects the **moving** tool)

Naming:

Double tap on the class, then say "Name" followed by the name you want to give the class

Appendix 2

System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
2. I found the system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
3. I thought the system was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
5. I found the various functions in this system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
8. I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
9. I felt very confident using the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5

Appendix 3

Interview Template

Questions regarding the participant

1. What is your current occupation? (Student, PhD, Professor etc.)
2. Where are you originally from? (Do we need this information?)
3. How experienced are you in software modelling? (Scale from 1 to 5 where 1 is Novice – 5 Professional)
4. What tools for software modelling have you been using before this? (whiteboard, pen & paper, online tools, or/and eclipse)
5. Do you think software modelling process is important for developing a good system/application? (Scale: 1 Not important – 5 Very important)
6. Have you any sort of experience with a Smart board before this test? (In what way, for what purpose etc.)

Questions regarding the application

1. What are the advantages of using a smart board for software modelling?
2. What are the disadvantages of using a smart board for software modelling?
3. What did you like about the application?
4. What didn't you like about the application?
5. Was it easy to learn the application? (Scale: 1 Very easy, 2 easy, 3 neutral, 4 hard, 5 Very hard)
6. Was it easy to use the application? (Scale: 1 Very easy – 5 Very hard)
7. What was the level of difficulty of using the voice recognition in the application? (Scale: 1 Very easy – 5 Very hard)
8. Did you feel the application was effective while using the voice recognition? (Scale: 1 Not effective – 5 Very effective)
9. Comparing your previous usages of different tools when modelling, what are your thoughts on using the voice as an interaction mode in this modelling tool?
10. Did the voice recognition simplify the modelling process on the Smart board? In what way?
11. Do you think having many interaction modalities (like keyboard, mouse, touch, voice) in one modelling tool can increase its usability? Yes/no... why?
12. Do you think using voice for text input and interaction is better than using other interaction modes like mouse & keyboard when modelling on a Smart board? Yes/No... Why?
13. How important do you think the two voice recognition features that are currently in the application are? (Scale: 1 Not important – 5 Very Important)
 - a. Naming classes
 - b. Selection of interaction tools e.g. create class, create edge, etc.
14. How important do you think these following possible feature is to add to the application that uses voice recognition? (Scale: 1 Not important – 5 Very Important)
 - a. Add attributes
 - b. Add operations
 - c. Set classes directly on the model without touching the board
 - d. Connect edges between classes
 - e. Delete classes
 - f. Undo/redo
15. Do you have any other feature in mind that could be added to the application that supports voice recognition?
16. What are your thoughts on using this application with voice recognition in collaboration with other people?