Prototype a solution for multi-track premium music score web player.

Author:
Haotian Wu

Supervisor:
Dr. Per Rohdin

Examiner:
Dr. Anne Håkansson

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in the

School of Information and Communication Technology

September 2014
Abstract

The current commercial version of ScorX multi-track premium music score (MPMS) player has poor support on mobile device, extra cost could be required for current product to support mobile device. This project aims to focus on developing a MPMS player\textsuperscript{1} using web based cross-platform technology. The company’s commercial version is build on Adobe Flash Player. An investigation is conducted and result shows that a possible solution exist that by using which building such a player could be done by using Web Audio API and Adobe Edge Animate; a prototype is built upon this solution. Study of this project shows that alternative solution for support mobile device could be native application for single mobile hardware platform and Haxe for multiple hardware platform including desktop and mobile device.

After building a prototype, a series of tests are run to evaluate both Flash player solution and HTML5 solution. It turns out that Flash player will still be the first choice for desktop device solution in current phase since it is more stable with a rather satisfactory performance, while HTML5 solution supports better in mobile device. The most of the disadvantage of HTML5 solution lies on the performance of the audio engine (i.e. Web Audio API) rather than the animation engine. As more and more major web browser has fully support Web Audio API, it will for sure make the audio engine perform even better. The future development of ScorX multi-track premium music score should therefore focus on HTML5 and native application if necessary.

**Keywords:** ScorX, multi-track premium music score player, HTML5, Web Audio API, Adobe Edge Animate, Flash player, Cross-platform application, HAXE

\textsuperscript{1}The term of “multi-track premium music score (MPMS) player” is defined as a “volume control enabled multi-track music player along with the corresponding music score demonstrated at the same time”
Abstract

(In Swedish)


När prototypen var byggd kördes en serie tester för att utvärdera både Flash player lösningen och HTML5-lösningen. Det visar sig att Flash player fortfarande kommer att vara ett första val för stationära enheter i nuvarande fas eftersom den är mer stabil med en ganska tillfredsställande prestanda, medan HTML5-lösningen fungerar bättre i mobila enheter. Den huvudsakliga nackdelen med HTML5-lösning ligger på utförandet av ljudmotorn (dvs Web Audio API) i stället för animeringen av motorn. Eftersom fler och fler stora webbläsare har full support Web Audio API, kommer ljudmotorn presteras ännu bättre framöver. Den framtida utvecklingen av ScorX flerspårig premium musik poäng bör därför inriktas på HTML5 och ursprungliga programmet om det behövs.

Keywords: ScorX, multi-track premium music score player, HTML5, Web Audio API, Adobe Edge Animate, Flash player, Cross-platform application, HAXE
Acknowledgements

I would like to thank Dr. Anne Håkansson for her patient assistance of the project planing and project report. I am grateful of the support and suggestion from Dr. Per Rohdin and Dr. Thomas Arctaedius as well as other colleague I worked with during the project period in ayond AB. Thanks to my Sabrina who is my lovely wife, teacher, friend, wake-up alarm and official food feeder.

Last but NOT least, I would like to give my very very very special thanks to Jonas Nyström, a music teacher, an orchestra conductor, a technical geek, the tech leader and the developer of the Flash player solution. During the period of this project, Jonas gave me countless numbers of help, assist and suggest which is without doubt the best thing happened to me.
## Contents

Abstract i

Abstract in Swedish ii

Acknowledgements iii

Abbreviations vii

1 Introduction 1

1.1 Background ........................................ 1
1.2 Problem statement .................................. 2
1.3 Purpose ........................................... 2
1.4 Goal(s) ............................................ 2
1.5 Methods ........................................... 3
   1.5.1 Quantitative Method and Qualitative Method .... 3
   1.5.2 Inductive method and Deductive Method .......... 4
1.6 Delimitations ...................................... 5
1.7 Outline ........................................... 5

2 Method 7

2.1 Philosophical Assumptions ........................ 7
2.2 Research Methods .................................. 7
2.3 Research Design ................................... 8
2.4 Data collection method ............................. 8
   2.4.1 Introduction of related data collection method .... 8
   2.4.2 The data collection method used in this project ... 9
2.5 Data analysis method ................................ 10
   2.5.1 Introduction of related data analysis method .... 10
   2.5.2 The data analysis method used in this project ... 10
2.6 Quality assurance ................................... 10
2.7 Sustainability ...................................... 11
2.8 Social impact ..................................... 11
2.9 Software development method ....................... 12
2.10 Software development method used in this project .... 12
   2.10.1 Introduction of Agile development and Scrum .... 12
   2.10.2 Argument about the software development method .. 13
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HTML5</td>
<td>Hyper Text Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>MPMS</td>
<td>Multi-track Premium Music Score</td>
</tr>
<tr>
<td>OOP</td>
<td>Object Oriented Programming</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Socket Layer</td>
</tr>
<tr>
<td>SWF</td>
<td>Shock Wave Flash</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>UX</td>
<td>User Experience</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This chapter presents the general introduction of this thesis background, problem, purpose, goals and selected research methodology as well as the outline of this report.

1.1 Background

ScorX [1] is a web-based online music score reading and playing service for choir singing created and operated by Mellansels folkhögskola[2], Sensus[3] and ayond AB[4]. ScorX player is a current commercial version of an Multi-track Premium Music Score (MPMS) player. The service allows users to listen to each individual part\(^2\) or combined parts of the music score (multi-track play in this case) while reading it. Such a product which contains music score and music sound track is defined as a “Premium music score”, thus the corresponding player is defined as “ScorX multi-track premium music score player”. The next version of this service will build a community for different organization to organize their activity. The user will be able to purchase regular music score or premium music score from the web page (provided by Gehrmans Musikförlag and other providers in the future). It could help choir singers to learn and practice easier and faster.

\(^2\)The word “part” refers to “a strand or melody of music played by an individual instrument or voice (or group of identical instruments or voices) within a larger work”[5]
1.2 Problem statement

The current ScorX player is implemented with Flash Player which cannot be used by mobile device (including mobile phone and tablet product) users where Flash Player is not available since Adobe has announced that there will be no future development on mobile device support of Flash Player. It is also required to install both web browser and Flash Player running environment in the desktop device to run this service. It is necessary to find another web based technology to re-create ScorX player that could be more independent to the running environment and support more kind of device.

The problem in this thesis project is how to build an working MPMS player using a current available web technology called HTML5 while keeping and possibly improving the performance and functionality of current Flash Player solution as much as possible.

1.3 Purpose

The purpose of this thesis project is to present the work with developing a prototype of MPMS player with web technology. An evaluation and comparison of this prototype with other possible solutions and conclusions will be presented as well.

1.4 Goal(s)

The goal of this project work is to provide a web solution for MPMS player. One or several prototype will be implemented based on the research. The prototype will be tested and suggestion will be given in actual product and future development. Meanwhile, the author will contribute to current development process of front-end implementation on the web service other than music player.

After the goal is achieved, future ScorX player could be able to support mobile device which lead to a large amount of potential customers (i.e. potential revenue source). The study of HTML5 multi-track music player will be a useful reference for other organization which seeking on-line audio playback and process solution.
Investigation part: An investigation will be conducted on possible choices of HTML5 implementation with Java Script application programming interface (API) or other suitable tools. The solutions will be evaluated based on implementation feasibility, browser support, real performance compared with current Flash Player solution. Meanwhile, less feasible or suitable solution will be presented, evaluated and compared in the theoretical level.

There will also be a conclusion based on current HTML5 development to describe the improvement space for future product.

Implementation part: One or more prototype will be implemented using HTML5 based on the need from research part and will be tested for compatibility and performance. Another part of the work is to contribute to current ScorX development process. Skills and knowledge learned and improved in this process will benefit back to the research work.

1.5 Methods

In this section, quantitative study, qualitative study, inductive method and deductive method will be introduced as well as the argument of certain selection of research theory and method.

1.5.1 Quantitative Method and Qualitative Method

Qualitative research “is designed to reveal a target audience’s range of behavior and the perceptions that drive it with reference to specific topics or issues. It uses in-depth studies of small groups of people to guide and support the construction of hypotheses. The results of qualitative research are descriptive rather than predictive.”[8].

Quantitative method is described as “to develop and employ mathematical models, theories and/or hypotheses pertaining to phenomena. The process of measurement is central to quantitative research because it provides the fundamental connection between empirical observation and mathematical expression of quantitative relationships. Quantitative data is any data that is in numerical form such as statistics, percentages, etc”[9].
The study on HTML5 is described by properties and features. All these combined together turns out to be a non-numerical functionality requirement for the prototype where qualitative method is more useful.

Meanwhile, the evaluation of the prototype could lead to numerical data analysis such as response time. Other conclusion from this study would be highly relied on statistical data such as the usage percentage of different desktop and mobile browser where quantitative method seems necessary. However, the purpose of this method is to mathematical result to describe a qualitative feature instead of proof or disproof a pre-set hypothesis.

In this case, the method in this project should be described as qualitative method with certain mathematical analysis to conclude certain qualitative feature.

1.5.2 Inductive method and Deductive Method

**Inductive method** (i.e. inductive reasoning) is “reasoning in which the premises seek to supply strong evidence for (not absolute proof of) the truth of the conclusion. While the conclusion of a deductive argument is supposed to be certain, the truth of the conclusion of an inductive argument is supposed to be probable, based upon the evidence given.” [10]

**Deductive method** (i.e. deductive reasoning) is “is the process of reasoning from one or more general statements (premises) to reach a logically certain conclusion” [11]

The MPMS player, which is an instance presented in digital device such as PC, mobile phone and tablets, can be implemented in various ways given certain requirements such as satisfactory user experience and device compatibility. In the study of implement the MPMS player in HTML5, the problem is described as can or cannot using HTML5 to develop such a player. In this case, separated tools and concept should support the purpose of developing such a player using HTML5. As long as there exists all necessary tools and concepts that could support this solution, the conclusion could be drawn. Abstractly, this could be described as “seek to supply strong evidence for the truth of conclusion” [10] as mentioned in the definition of inductive method.

In order to compare whether Flash player solution is better or not compared to HTML5 solution. An evaluation will be conducted that would generate several individual factors of interest. The combination of these factor could lead to a conclusion regarding to which solution is better. This approach also meet the definition of inductive reasoning.
As mentioned above, such a reasoning method fulfills the definition of inductive method. In this project, inductive method will be used.

1.6 Delimitations

Since the current version of ScorX implemented with Flash is up and running with a quite satisfactory user, the HTML5 version might not be a firm solution for the future due to possible performance flaws, compatibility and potential risk of mass production solution. In this case this project will not provide the development solution of a full featured HTML5 version of ScorX player including public ready user experience design and possible performance improvement based on programming skills, which could be a waste of time for current business strategy of ayond. There will also be some requirements with lower priority that will not be implemented during the limited time period, however, a solution of such requirements will be delivered.

1.7 Outline

- Chapter 2 presents the selected philosophical concept, research design research method including data collection method and software development method and other logical and social discussion.
- Chapter 3 presents the industrial and technical background of the current ScorX product.
- Chapter 4 presents the theory involved in the study and development work of this project.
- Chapter 5 presents an in-depth study of Flash Player version of ScorX MPMS player. The study suggests a guideline for the requirements of HTML5 solution.
- Chapter 6 presents various choices of technology that could be used in the HTML5 solution, after a series of test and reasoning, one set of audio and animation technology will be used for prototyping.
- Chapter 7 presents the actual procedure of prototyping the HTML5 solutions part by part including design, process and tools.
• Chapter 8 presents the evaluation conduct between HTML5 and Flash Player solution. The evaluation suggests advantage and disadvantage of each solution.

• Chapter 9 presents the conclusion and future work of the prototype, the development process and this project.
Chapter 2

Method

The description of chapter will go through the model of the portal of research methods and methodologies.[12]

2.1 Philosophical Assumptions

A philosophical assumption “is the starting point for the research”[12].

Positivism is used as the philosophical assumption in this project. Positivism “assumes that the reality is objectively given and independent of the observer and instruments”[12][13][14].

The previous product of ScorX MPMS player, the technology in Flash Player, HTML5 and others mentioned in this project is objectively exist fact. They are independent to the observer and thus Positivism is the suitable philosophical assumption for this project.

2.2 Research Methods

Research methods “provide procedures for accomplishing research tasks”. [12]

Applied research is used as the research method in this project. “Applied research method involves answering specific questions or solving known and practical problems.”[12]
The problem in this project is a practical problem which is using current technology to find a solution that solve the current issue of ScorX MPMS player, thus it is the suitable research method of this project.

2.3 Research Design

The research designs are “the guidelines, or the methodologies, for carrying out the research”[12].

Action research is used as the research design in this project. It is “performed by actions to contribute to practical concerns in a problematic situation”[12].

This project present a HTML5 prototype of ScorX MPMS player which is considered to be a improvement for ScorX service, thus it is contribute to the practical concerns. In this case, it is suitable for this project.

2.4 Data collection method

According to the description in the previous chapter, qualitative method is the main selected methodology. Hence the most common used data collection method are Questionnaire, Case Study, Observations, Interviews, and Language and Text[12]. What is more, the evaluation requires mathematical analysis to conclude certain qualitative feature, thus the Experiments method which is a common quantitative data collection method will also be used in this project.[12]

2.4.1 Introduction of related data collection method

Experiments is a method to design and conduct certain experiments to collect sets of variables for future mathematical analysis.[12]

Questionnaire collects data through a designed list of questions. The data could be qualitative data based on closed questions such as multiple choice questions. It could also be quantitative data based on open questions with no standard answer.[12]
**Case Study** is “analyses of persons, events, decisions, periods, projects, policies, institutions, or other systems that are studied holistically by one or more method. The case that is the subject of the inquiry will be an instance of a class of phenomena that provides an analytical frame an object within which the study is conducted and which the case illuminates and explicates”[15].

**Observations** is a common social science method that involves participation in certain circumstance and observe the behavior of participants with emphasis on culture and situations.[12]

**Interviews** is to conducting “structured, semi-structured or unstructured [12]” interview to acquire in-depth understanding of interviewee on certain system and questions. [12]

**Language and Text** is a method trying to interpret the meaning of an interview, a document or any text based raw data. [12]

### 2.4.2 The data collection method used in this project

There are several data collection method selected that will apply to the different phase of this project.

Before the HTML5 prototype is actually implemented, the data that will be collect would be the feature and the concept of the previous Flash player version of MPMS player. An in-depth case study will be conducted on the Flash player version of MPMS player. The case study involves studying related design and development documentation and interacting with the product based on the requirement from the company. One or more interviews will be conducted with the tech-leader of the Flash player MPMS player for concept and method of design and implementation. The data that describe the feature of the MPMS player, the key concept of design and implementation which is considered as qualitative data and will be collected from these two studies.

After the prototype is implemented, several experiment will be conducted on the prototype and the Flash player version of MPMS player to evaluate performance, the data is described by certain numerical data such as time. This kind of data is considered as quantitative data but to describe qualitative feature.


2.5 Data analysis method

The data analysis method used in this project are based on qualitative methodology.

2.5.1 Introduction of related data analysis method

Coding is “an interpretive technique that both organizes the data and provides a means to introduce the interpretations of it into certain quantitative methods. Most coding requires the analyst to read the data and demarcate segments within it, which may be done at different times throughout the process”[16].

Narrative Analysis is a method based on literature and applying discussion and analysis on it to extract useful data and draw conclusion.[12]

2.5.2 The data analysis method used in this project

The collected qualitative data will be analyzed by Narrative Analysis during literature study, interview and case study. The key representative data will be extracted from the documentation, record and interview transcript using this method. Coding theory are used when evaluate both Flash player and HTML5 solution. Some quantitative data is extracted from the general usage behavior and user experience and then will be used to generate a qualitative conclusion.

2.6 Quality assurance

This project use qualitative method with inductive approach, thus the quality assurance will apply and discuss validity, dependability, confirmability, transferability and ethics[13].

Validity “makes sure that the test instruments actually are measuring what is expected to be measured”[12][17][18].

Dependability “corresponds to reliability, and is the process of judging the correctness in conclusions, using auditing”[12][13].
**Confirmability** “confirms that the research has been performed in good faith without personal assessments that have affected the results”[12][13].

**Transferability** “is to create rich descriptions that can become a database for other researchers”[12][13].

**Ethics** “independent of quantitative research or qualitative research, is the moral principles in planning, conducting and reporting results of research studies”[12][13].

For the actual discuss of Quality assurance, see Chapter 8.

### 2.7 Sustainability

There is no potential issue of environmental sustainability considering that the project is working on an digital media service with the same physical environment as the old product. However, this project do have a positive impact on the economical sustainability impact. Since the technology that will be used in this project could support more than one kind of hardware platform, by applying this technology to the new product, it will reduce the cost of developing and maintaining separate solution for each platform. Therefore the whole cost this product will be reduced. Furtherly, it is possible to reduce the price for using this product from where the customer can benefit.

### 2.8 Social impact

If the new technology used in this project has been proved to be possible to use in the future, the ScorX will provided the MPMS player with better user experience with the same and even lower cost. The targeted customer could be benefit from the convenience that brought by ScorX.

The new technology is also believed to be able to inspire more Web service developer and provider to create other forms of multi-media Web service while do not need to suffer from the limitation of old standard and technology. The potential popularity of the usage of the new technology will therefore drive the original developer of this technology and third-party organization to further improve this technology.
2.9 Software development method

“The software development models are the various processes or methodologies that are being selected for the development of the project depending on the project's aims and goals. There are many development life cycle models that have been developed in order to achieve different required objectives. The models specify the various stages of the process and the order in which they are carried out.”[19]

A similar definition is called Software development method. “A software development method presents a set of guidelines or approaches for software development processes, and presents optimal choices for development tools and techniques, thus facilitate high quality and cost-effective software products”[20].

In the following section, the term software development method will be solely used.

2.10 Software development method used in this project

The software development method used in this project are Agile development method. Scrum is the specific method that will be used in this project, the detail will be explained in this section.

2.10.1 Introduction of Agile development and Scrum

Agile development is a method of developing software in an incremental and rapid cycles [21]. Compared with traditional software development, the agile model has break the development into smaller tasks, each task has its own development cycle including planning, implementing, testing and demo. The delivery of one cycle is based on the delivery of previous cycle, thus the function, structure and even planning are incremental. Because of it, a single delivery could be far from fully functional, in this case the deployment or release of each delivery is optional.[21] For a more direct explanation, see Figure 2.1.

Scrum is a method of conducting agile software development. The way to conduct a scrum software development method can be generally described using the description of agile development models. However, there are some unique concept behind scrum
method. “empirical process control” is an important concept of scrum, which means the planning could vary based on the delivery, observation and experience of current development progress instead of a single plan from beginning to the end based on guess and prediction.[22] It is common to NOT define the final product from beginning. Mistake, innovation and attempting could constantly be made and therefore change the whole plan. The developer in this procedure are highly self-organized and self-motivated to interact with customer and each other.[23]

2.10.2 Argument about the software development method

The main reason to select agile and scrum method is because the nature of them allows uncertainty from beginning and the development cycle is small and multiple. The HTML5 version of ScroX has no settled tool and API choice in the beginning, all the possible choice obtained from pre-study will be tried and narrowed done to a best one. Meanwhile, the requirement from the customer, (i.e. requirement from Ayond AB), could change, thus the development plan will be changed accordingly.

To sum up, the fast pace and acceptance of modified planning is the main reason to choose Agile and scrum.
2.10.3 In this project

As mentioned in previous section, the scrum method can be generally described using the description of agile development models. However, there are some unique concept behind scrum method. Thus the way to conduct agile development model and scrum method will be explained together.

- In the beginning, the planning is to produce an HTML5 solution of MPMS player. The planning did not specify any technique and tool to be used but only define the requirement of the product including graphical and functional design.

- In the first phase, in each development cycle, the developer will try different tools and choice to develop audio engine, animation engine. The best choice of solution will be find after a few cycles.

- Then the developer will try to combine the audio and animation engine together in the new cycle to test its interaction ability. After this cycle is done, the basic product is complete.

- Another few new cycle will be conducted to alter the basic product based on customers feedback and other update such as tools version.

- The new cycle could be kept added to the process even after this project is done to constantly improve and update the current product.
Chapter 3

Background

This chapter provides detailed background description of the industrial area (i.e. the product) that this thesis project worked with.

3.1 ScorX multi-track premium music score player

As introduced in chapter 1, The ScorX multi-track premium music score player is a combination of audio player and music score reader. The General purpose of creating such a music player is to make the user be able to read the music score even if the user has insufficient musical theory and music note knowledge while be able to hear the music playing at the same time. Before HTML5, such kind of web based multi-media application is done generally by using Adobe Flash Player, example such as Facebook based game application and youtube player (though it is possible to use HTML5 player in youtube) are still in use today. As mentioned in the first chapter, Flash player has no support on mobile device, this could cause the loss of potential customers for ScorX service.

The choir singers (major target user group) who are full time worker may have quite limited time to practice. A pre-study shown that a fairly large number of church choir singers is elderly people who could forget to bring certain music score to the rehearsal. Under this situation, certain conclusion could be drawn that mobile device will bring convenience to users.
For example, elder people can simply bring a tablet every time for the rehearsal, all necessary music note can be accessed directly from the device with sufficient reading quality. The extra playing function could help the singer to learn faster and easier, especially for those who has no sufficient skill of reading a music note.

The bottom line of ScorX music player is that such a player will be run in the ScorX website in the beginning. In this case, supporting the mobile device is an extension of the web version of ScorX music player. To support this feature, an early plan of Ayond AB is to create independent mobile application based on iOS and Android platform. However, the cost of this solution is beyond acceptance. Ayond AB would like to re-use current technical asset as much as possible while minimum the extra cost for developing new solution.

The best solution is to use multi-platform technology to create such a service. In this project, HTML5 and HAXE has been discussed as possible candidates while HTML5 is the main target technology to study and work with. The technical features and requirements of ScorX MPMS player will be described in the Theory chapter.

In the next subsections, the detailed requirement will be explained.

### 3.1.1 Audio

The designed Audio feature of ScorX multi-track premium music score player is to be able to play multiple music track at the same time (i.e. the term “part” mentioned in 1, footnote 2). The Audio requirement is listed below:

- The performance should be smooth and the each single track playing should not interfere with other tracks.

- The volume of each track should be able to be controlled (including mute and maximize) individually.

- A time synchronization function should be created to interact with animation. User should be able to control the time point of all tracks at the same time (the time control function is not applicable for a single track).
3.1.2 Graphical interface

The Graphical interface of the ScorX multi-track premium music score player is to display a music score as well as the interface for volume control of the music player. The user could read the music score while listening the music.

In order to help user follow the music score correctly and easily, a time marking method should be used to implement such a function (e.g., a cursor moves along with the music playing). In this case, the time marking method can be defined as an animated graphical interface in this player. Besides, the time marking method should also be able to interact with the music player to control the playing timing, which means by certain interface created in the music score reader (i.e., the animation), user should be able to use such an interface to control the multi-track music playing. If there is multiple pages of music score, the animation should be able to auto focus to different pages of music score.

3.1.3 Printing function

The printing function grants user limited rights \(^3\) to print a certain music score after user has purchased the score or the score is free of charge. The security (i.e., the copyright) issue should also be considered, which means the printer should not allow user to print a certain music score via a virtual printer such as a PDF printer \(^4\) but only physical printer. The printing function is not a focus in this project, thus no further detailed will be mentioned and discussed.

3.1.4 Target user and requirement

The original and majority of the target user is church choir singers. As the current business has a potential pop musical market, the future target group will also be pop music band and studio. Certain requirement can be found in previous sub-sections. There is some extra requirement listed below:

\(^3\) Limited rights means that the user could only print music score in a limited time period or printing limited numbers of copies

\(^4\) A PDF printer is a software which treated as a physical printer in the operating system but do not printing actual papers. It however convert a document into a PDF file and save it. In this case, users could use such a method to print countless of copies.
• The ScorX multi-track premium music score player should be embedded in web which is able to use in desktop computer and mobile device (i.e. tablet and mobile phone.) in iOS and Android platform. The windows phone is not listed in current phase as the windows phone user in Sweden is a minority group. See figure 3.1.

![StatCounter Global Stats](image)

**Figure 3.1:** Data from StatCounter, period 2013-01 to 2013-12, including mobile and tablet device in Sweden [24]

• The music score audio engine should have potential ability to simulate pop music processing device and procedure such as a synthesiser.

**3.2 ScorX music product**

The premium music score contains music score and combination of one or several recorded sound track (depends on the music score). The current phase of ScorX music product only church choir songs. The sound tracks are ordered by the owner of this

---

5 A sound synthesizer “(often abbreviated as "synthesizer" or "synth") is an electronic musical instrument capable of producing a wide range of sounds. Synthesizers may either imitate other instruments (“imitative synthesis”) or generate new timbres. They can be played (controlled) via a variety of different input devices (including keyboards, music sequencers and instrument controllers). Synthesizers generate electric signals (waveforms), and can finally be converted to sound through loudspeakers or headphones. [25]

---
project and recorded by professional studio. The company that owns ScorX service has the full copyright of both music scores and sound tracks.

3.3 Support of mobile device

The future version of ScorX MPMS player should support mobile device with touch screen. The implementation is not limited to one technology. There are generally two way of doing such an implementation.

The first concept is to utilize the browser as the application host for both desktop and mobile device. In this case, the web service should support both desktop web and mobile web. As HTML5 is introduced as a future replacement of current HTML technology, the nature of HTML5 has better support on multi-media function such as audio and video/animation playback and interaction. Thus HTML5 could be a potential solution.

The second concept has separated desktop and mobile support, however, there will be only one solution for all platform. The difference between this these two concept is that the service host is not same on different platform. This kind of solution should be further converted to different target device. This means that the desktop solution could be any possible solution such as Flash Player, HTML5 and other similar solution while the mobile solution could embrace a much mature solution such as native application. All of the solution in the different platform could trace back to one set of code. HAXE is a possible choice for this concept. The detail of is concept is not the main topic of this thesis, thus it will not be discussed in detail.

For more detailed introduction, see the next chapters.
Chapter 4

Theory

This chapter presents the theory, concept and technology that involved in this thesis project. Certain technology are used by previous solution while others are considered to be used in this thesis project.

4.1 Cross-platform

Cross-platform often refers to “Refers to developing software for, or running software on, more than one type of hardware platform.”[26]. One of the typical multi-platform application or running environment is a Web browser which means the rendered web page are almost the same on different platforms.[26]

There are generally two primary way in order to achieve this goal. The first way is to use different programming language or different source code to be compiled to the software which has the same user interface and function.[26] The drawback for this method is of course developer has to maintain several set of source code. The Second way is to use a interpreter such as Java virtual Machine, the source code will be compiled into a language that the Java Virtual Machine will recognize and run, however, the Java Virtual Machine could be developed under different code and method based on different hardware platform.[26]
4.2 Web Application

Web Application or Web-based application is “any program that is accessed over a network connection using HTTP, rather than existing within a device’s memory”[27]. Web application will commonly run but not limited to Web browser, it could also run as a part in the application installed on a hardware platform.[27] According to the definition of cross-platform application, Web application is a typical type of cross-platform application. The HTML5 ScorX MPMS player is a web application.

4.3 Mobile Application

Mobile application, or simply Mobile app, is a kind of software that is designed to run on the mobile device such as smartphone and tablet computer[28]. There is another more specific definition called Native Mobile App which refers to the application build on specific programming language such as Java Android SDK for Android application and Objective-C for iOS application[29].

The difference between these two definitions is that the Native Mobile App has excluding the Web application which is cross-platform and thus could run on mobile device as well. In this case, a Web application that can run on mobile device can also be defined as mobile application.

The prototype created in this project is a Web app and has include mobile device as target device, thus the prototype is also a Mobile app. The term Native application will not be discussed in detail in this project, however, some discussion in the Haxe section has involved native mobile app.

4.4 Adobe® Flash® Player and Air

The tools and applications introduced in this section have been used in the previous solution of ScorX multi-track premium music score player, the next chapter will have further explanation of how these tools and applications are used.
4.4.1 Adobe® Flash®

Adobe® Flash® Player “is a lightweight browser plug-in and rich Internet application runtime that delivers consistent and engaging user experiences, stunning audio/video playback, and pervasive reach.”[30]. The next generation of Flash® Player called Adobe Flash Platform is build upon Adobe Flash Player[30]. Besides audio/video playback, this application is widely used for advertisement and providing interactive multi-media content by embedded into web as a plug-in or an ActiveX control 6 [33][34].

Adobe® Flash® Player stores all multimedia and interactive content in a Shockwave-Flash (SWF) file [34]. “SWF content can range from simple animations to online advertisements to complete applications that communicate over the Internet.”[34]. Adobe® Flash® Player also supports various different Internet protocol such as HTTP,SSL,etc, which could be used for front-end/back-end data communication and building on-line application[34].

The current version of ScorX multi-track premium music score player is built upon Adobe® Flash® Player, the detailed description will be mentioned in the next chapter.

4.4.2 Adobe® AIR

Adobe AIR is “a cross-platform runtime that enables you to use your existing Flash/ActionScript or HTML/JavaScript development skills and tools to build and deploy applications, games, and videos outside the browser and on mobile devices”[35]. It can be concluded that as long as the AIR runtime is installed, it can perform multi-media interactive function on various platforms and devices. Besides, AIR can also be used for building extensions for mobile platform such as iOS and Android[35].

In the ScorX project, AIR is used to build current solution of printing and music score time marking planning/producing. The music score time marking planning/producing solution will be mentioned in the next chapter.

---

6 An ActiveX control is a proprietary technology developed by Microsoft that allows linking and embedding content and object together while also support to connect other interfaces[31][32].
4.5 HTML5

HTML or HyperText Markup Language “is the standard markup language used to create web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>).” [36]. HTML5 is the latest version of HTML standard comes after HTML and HTML4.01, it is designed to be the replacement to the previous version of HTML[37].

The significant update of this HTML5 is that “it was specially designed to deliver rich content without the need for additional plugins. The current version delivers everything from animation to graphics, music to movies, and can also be used to build complicated web applications” [37]. HTML5 is also designed to be device-independent and to support cross-platform, which means that it can be used in different desktop, mobile and other smart device (e.g. smart TV) with different platform[37].

One of the most important new feature, which is also the big advantage of this project, is that HTML5 has introduced lots of API that developer can use it along with new element to develop sophisticated Web application[38].

The main reason to seek a HTML5 solution for ScorX multi-track premium music score player is that HTML5 is a cross-platform solution, which means a one-application-for-all solution. Such a implementation could potentially fulfill all requirements while no extra work will be done for each individual platform.

As mentioned before, Flash Player has been widely used for web application, yet the statistics shows that the usage of Flash player in Web has become less popular compared to HTML5, for more information, see figure 9.1 in Appendix.

4.6 Haxe: A cross-platform solution

Haxe is “an open source toolkit based on a modern, high level, strictly typed programming language, a cross-developer, a complete cross-platform standard library and ways to access each platform’s native capabilities.”[39]. To use Haxe as a cross-platform, developer should use Haxe language, which is a modern high level programming language [40]. It
is strictly typed and fairly simple if the developers are already familiar with Java, C++, PHP and other object oriented programming (OOP) language[40].

Though both Haxe and HTML are described as cross-platform, the core concept is different. The nature of HTML is to be used to build websites which is contained in a web browser. Haxe language, however, will be compiled into source code of different kind such as Actionscript 3, PHP, C++, etc by using Haxe compiler. The compiled source code will then be compiled (if the target language is not a script language) or run on other platform, thus Haxe has also introduced a third concept of developing cross-platform application compared to the two primary method introduced in the cross-platform section.

An advantage of Haxe is its rich APIs, especially that Haxe has introduced Flash APIs that could be used to mimic Flash player multi-media features[41]. According to Jonas Nyström, the syntax of Haxe is similar to Actionscript 3 which makes it fairly easy to migrate the Flash player application into Haxe language[42]. The comparison of Haxe language and AS3 code also suggest the same conclusion [43].

4.7 Related work

The related work of this project will be discussed in two part.

The first part covers the previous ScorX product which is created using Flash Player. As mentioned before, the Flash Player has no natural support on mobile device, which causes higher cost if mobile device is planned to be supported while using Flash player (i.e. other technology will be used to solely develop mobile version of applications). The requirement and design concept of the prototype of this project will be based on the flash player version of ScorX MPMS player, thus this work will be thoroughly studied, the detail will be shown in the next Chapter.

The second part is related to the potential choice that could be used to create HTML5 solution of MPMS player, the detail will be described in chapter 6. The related work involved in this part is the current development example of the selected audio and animation engine. Some detailed could be found in 6. However, the most important related work in this case the example application made by Web Audio API.
most typical example that has the similar function as the prototype of this project is called Mix.js (Link: http://kevvv.in/mix/#{}).

Compared with the related works mentioned above, this project could potentially improve the Flash player version of ScorX player. This project will also enrich the user interaction of Web Audio API by adding a co-operational animation will be played along with sound playback. More detail could be found in the conclusion chapter.
Chapter 5

Study of the Flash Player Solution

This chapter provides an in-depth study of the Flash Player version of ScorX player, certain requirement and concept will be used in the prototyping of HTML5 solution.

5.1 Overview

The original purpose of Flash Player solution is to integrating a functional ScorX multi-track premium music score player into the ScorX website. The previous developer have rich experience in developing Flash based application, the multi-media features of Flash are very mature, which makes it easier to be a choice of first version of solution.

The current commercial version of ScorX multi-track premium music score player is implemented based on this solution which contains the following key design:

- Front-end user interface design
- Time marking algorithm and implementation methodology.
- Audio and animation synchronization
- Front-end to back-end data communication

The detailed feature will be described in the following sections.
5.2 Graphical User Interface (GUI)

![Figure 5.1: Graphical User Interface of Flash solution](image)

The GUI of Flash solution is shown in figure 5.1. It can be seen that the main note display area are placed at the left area of the window. The right sidebar contains the following control interface:

- Full screen view
- Zoom out view (list view of music score pages if there are multiple pages)
- Play/Stop control
• Volume control of each sound track (i.e. parts in the music score, the definition of parts can be found in footnote 1 in first Chapter). The number of volume control bar is determined by the music score.

The orange colored vertical line positioned at the beginning is the cursor that will move along with the music playing. The behavior of the cursor will be described in the next section. If there are more than one page of the music score, the window will automatically focus on the specific page according to the current playing time point.

5.3 Time marking concept and tools

Time marking is one of the most important function in the ScorX multi-track premium music score player, this function allows user to listen and read the music and corresponding music score more easier. The details of the concept, method and tools of the Flash version of time marking function is introduced in this section.

5.3.1 Concept design

As different part of the music score will be played (i.e. sung by choir singers) at the same time, thus the cursor is designed to be a long vertical line that covers G-clef and F-clef notes.

The cursor will move along with the music score. However, the speed of the movement will be varied according to the music note. The principle of the cursor movement behavior is to find a bar or a bigger piece of the music note that allows the cursor to move in a fixed speed, then the whole music score will be divided into such kind of sections. Each section have a start time point which is considered as a key time point. From one key time point to the next one, the cursor will move in a fixed speed (i.e. linearly) between them.

The example shown in 5.1 is one of the simplest one regarding to movement speed. The nature of the church choir song gives a smooth and static tempo. This nature make it possible to have a rather accurate cursor movement plan based on each bar. It means that for this kind of music score, simply measure the time of first note of each bar in the corresponding sound track will be good enough.
Since the ScorX service in the future will provide different type of music scores, special cases should be considered. Some type of note, for example the classical piano composition in each bar is more complex (the example note is shown in appendix Figure 9.2), which means the cursor movement could be non-linear/non-steady in one bar. What was "worse", this composition has been played by countless of talented piano master that each piano player has his own interpretation. This cause a significant difference tempo-wise. If a cursor is place in this music score, one cannot predict the movement behavior by reading the music score. This means the recorded sound track should be studied carefully in order to make a accurate cursor movement.

5.3.2 Tools

In order to make the marking work easier, an Adobe AIR application is developed to measure the time point. The algorithm will be similar to the one mentioned above. See figure 5.2.

![Figure 5.2: ScorX grided editor in Flash version solution](image)

The visual grid-editor is used for graphically "draw" the "systems" (numbered 0-9 with red figures) on the page, and placing out the sync points (typically one per bar) that sub-divides a system. While playing the sound tracks, the user double clicks in real-time directly to the right of the sync-point which is currently played. As a result we
get a sync playback position for each graphical sync point. This position is represented by a decimal value from 0 to 1, where 0 is the playback beginning of the file and 1 is the playback end. (This can be seen in the screenshot, where the first system’s first sync-point is 0.049, meaning 0.049 of the full playback length of ca 2 minutes, which means about 2 seconds after the playback started etc.) [42]

5.3.3 Results

The result of the edited music score is a .xml file which contains necessary timing information. The AIR based timing editor has a build-in player that similar to the one in Flash based ScorX player. The editor could read a xml file and display it along with music score as shown in figure5.2. The xml file will be stored in the data base along with the sound file and the music score image file, which will be loaded when required from front-end. A snippet of xml code is shown below:

```xml
<item page="1" type="bar" origpos="0,5049311903061"
  h="0.166105499438833" w="0" x="0.133333333333333"
  y="0.312008978675645" pos="0,506950843506732"/>
```

5.4 Music Score conversion

The copyrighted music score are in PDF format, which need to be converted into a proper picture file such as PNG or JPEG to be properly displayed in the ScorX player. Several tools are used to do this work.

5.4.1 ImageMagick

ImageMagick® is “a software suite to create, edit, compose, or convert bitmap images”[44]. It can be used to convert a PDF file into bitmap picture[44].
5.4.2 GhostScript

Ghostscript “is an interpreter for the PostScript language and for PDF”[45]. In this project, it will be used for ImageMagick to convert PDF files[46].

5.4.3 PdfTK Server

PDFtk Server is “a command-line tool for working with PDFs. It is commonly used for client-side scripting or server-side processing of PDFs”[47]. In this project, this tool will be used to divide a multi-page PDF file into a single page PDF file.

5.4.4 Cooperation of each tools

In this solution, a executable(i.e. .exe) file is developed to integrate and run Ghostscript to drive PDFtk server and ImageMagick. This .exe file will also upload the converted file into the website server[46].

5.5 Audio, Graph and Synchronization

The build-in audio engine in Flash player is used as the audio engine in this solution. The case is the same in graphic engine, however, Haxe is used and compiled into AS3 codes instead of directly using AS3. The reason of using this method will be discussed in the later chapter.

5.5.1 Audio streaming

The separate sound-channel-files are stored on the server side. They will be loaded by the player one by one by simple HTTP(s) requests, decoded from mp3 data to raw wav data format and stored in memory. In the playback procedure, the separate sound channels are programatically mixed by simple byte addition (depending on channel volume information). The summary data is then fed back into the Sound object, with performs the playback just as if it was one single stereo file.[42] The procedure could be seen on 9.3 and 9.4.
5.5.2 Synchronization

An event listener is attached to the flash “stage” (player root object) to listen to Flash playback event, which is running at about 60 frames per second (fps). If playing is active, the current playback position. This position value is then fed into “sync engine” with holds the synchronizing information consisting of an array of “sync points”. Each “sync point” consists of playback position (also a value between 0 and 1) and the x-y-height-coordinates for the current position on the page. The sync engine then finds the two framing sync points where the current playback position fits in between, and simply finds the current pointer position by using simple linear interpolation.[42]

5.6 Communication and loading sequence

The data communication and loading sequence is shown in figure 9.3 and 9.4. After the user has requirement a certain premium music score, a series of URL will be generated contains necessary identification information of the music score and send to the server (i.e. back-end). These URLs are treated as a requirement of corresponding image files, sound track files and time marking information on the server side. These files will be then sent to the front end and load into the Flash ScorX player. The more detailed action could be read from the figures.

There are sections that goes very detail into the solution specifically for Flash Player solution, even though it seems to be less relevant to this thesis regarding to HTML5 solution, however, it is believed that these section will be helpful for ScorX project. These information could be considered as a good recording for the development history and could for sure contribute to the future development design of ScorX project.
Chapter 6

Study of Possible HTML5 Solution

This chapter presents the study and discussion of candidate technology that could be used in prototyping.

6.1 Overview

To implemented a HTML5 of ScorX multi-track premium music score player, to main part should be considered as the most important task, which is the audio and animation solution. To find a proper solution and to implement a prototype is the major task in this project. Besides, more should be considered when integrated to the commercial page with massive number of product, which will be discussed in the later chapter.

6.2 target web browser

As the study is conducted in 2013, a result of major browsers usage is shown in figure 6.1. It can be seen that the minority falls on Safari and Opera. Therefore, these two browsers should not be considered as the target browser. However, since iOS device user mentioned in Chapter 1 is massive, thus combined with the ScorX AB policy, a second consideration should be made that both mobile version and desktop version of Safari are considered as target browsers.
After the prototype is developed, the prototype will be tested on all target browsers.

6.3 Audio

As introduced in Chapter 1, HTML5 gives a cross-platform device independent multimedia solution including introducing new tags and new APIs. In this case the Audio solution will be find in one of those two choices.

6.3.1 Audio tag

An `<audio>` element “represents a sound or an audio stream”[49]. By adding argument controls, the audio elements will shown progress bar, start/stop button and volume control[49]. The source of the sound file could be simply included in the tag, which could be a local file or a online source[49].

The supportability of audio element is shown in figure 6.2. It could be seen that the supportability of audio element is sufficient in major browsers.

A simple test has been conducted on audio element to test its multi-track playing function. An HTML web page is created contains several audio element, each element contains a different sound track. A JavaScript script is used to trigger all element to
play at the same time. In turns out that each sound track will be played in a slightly different speed, result in an unsynchronized playing after a couple of seconds.

To sum everything up, a conclusion could be made that audio element is not suitable for multi-track playing.

### 6.3.2 Web Audio API

Web Audio API is a “*high-level JavaScript API for processing and synthesizing audio in web applications*” [38] which is introduced as an API in HTML5 standard [51].

The basic function of this API is to play sound. However, it brings more function such as special sound effect such as reverb, biquad filter, panner, compressor which is common audio studio technique [52]. The concept to achieve such a feature is to introduce audio routing graph (also called modular routing) which is shown in figure 6.3 [52].
The dashed box represents an audio context which includes everything involved in audio playing. Creating an Audio Context means creating a sound player framework. A source node is a sound file, which could be any type supported by host browsers. The source could go through several processing nodes and eventually connect to destination node which means it is ready to play. A much complex example is shown below, see Figure 6.4.

![Audio Context Diagram]

**Figure 6.4:** A complex example of audio context model. [51]

It could be seen that the Web Audio API is capable of performing complex sound processing. The detailed advanced function will not be discussed in this project.

The supportability of Web Audio API is shown in Figure 6.5. It could be seen that the supportability of audio element is sufficient in major browsers.

It can be seen that the supportability is poorer compared to audio element, however, according to Figure 6.1, the supportability of Web Audio API covers a majority of web users, which makes it to be a candidate of the HTML5 audio solution.
Figure 6.5: Supportability of Web Audio API in major browsers. Green block represents a supported version and a pink block represents a non-supported version. [53]

6.4 Animation

The concept would be similar to seeking audio solution. HTML5 has its own built-in animation solution, there will be also fairly large numbers of external library in JavaScript that can be used as a potential animation solution. In this section, only some popular library will be discussed. Some small test will be done to test the possibility of using the specific method as the animation solution. The test is only done in browsers which support Web Audio API which means Internet Explorer is excluded.

6.4.1 Canvas

“Added in HTML5, the HTML <canvas> element can be used to draw graphics via scripting in JavaScript. For example, it can be used to draw graphs, make photo compositions, create animations or even do real-time video processing or rendering”[54].

The basic example provided[54] is to draw a solid square. Therefore, a cursor could be drawn as a long thin square and use JavaScript to control its movement path along the music score. An experiment is conducted for such a function (create the cursor and let the cursor move along a straight line), which is a success. However, canvas is controlled by JavaScript which need to be told how to move the cursor and where to move the cursor. In this case, one has to measure on the music note picture for each key time point based on pixels.

The measurement could be done with Adobe Photoshop or windows drawing tool. Each key time point will be measured for horizontal and vertical axis based on pixels and
recorded into the control JavaScript. The recording work is done manually which is not convenient.

6.4.2 Jquery animate

“jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.”[55].

Jquery animate is a method introduced in Jquery library, the concept is to control the HTML elements’ CSS properties such as position[56]. There is a factor called speed which could change the position in minimally 1 millisecond[56]. A fast speed of change an element’s position (such as a cursor) could be considered as a set of picture frame which could theoretically up to 1000 fps, thus an animation is created.

Similar test is done on Jquery animate to test the possibility of manipulate an HTML cursor. It is also a success which makes it possible to be a solution for animation. The disadvantage is the same, the lack of convenience caused by pure code method.

6.4.3 Adobe Edge Animate CC

Adobe Edge Animate CC(creative cloud) is a tool developed by Adobe to create HTML5 animation and interactive multi-media content[57]. It enable user to use its GUI editor to create HTML5 in a rather easy drag/drop way compared to pure code development[57].

The project is done during the period that this product is till in a beta test phase which means this tool is free at that time. The original name is called Adobe Edge animate. Now the product is officially released and join Adobe creative cloud suite with new features added. The current version is not free any more, however, the original user who joined the the beta test program has the right to download and use the beta version for free for eternity. In this case the current commercial version and its new feature will not be discussed in this chapter.
The editing window is shown in figure 6.6. A simple test has been done to learn how this tool works by creating a very simple animation. The output result in a set of JavaScript files and an HTML file. The JavaScript files set contains the animation library including a so-called “edge library” and Jquery library. Besides, there are also several control JavaScript for HTML elements. The animation could be successfully run in target browsers.

![The main window of Adobe Edge Animate.](image)

The advantage of this solution is that the measurement and axis recording task could be done at the same time by drag and position the cursor element in the correct position of the HTML page which makes it similar to the AIR solution for cursor movement planning mentioned in chapter 5.

6.5 Final choice for prototyping

The bottom line of developing a HTML5 prototype of ScorX multi-track premium music score player is to meet the requirement as much as possible while taking the lowest time cost. In this case, the final choice of HTML5 solution will be the combination of Web Audio API and Adobe Edge Animate.
Chapter 7

Implementation of the prototype

Based on the study conducted mentioned in the previous chapter, this chapter describe the design and actual process of prototyping.

7.1 Graphic design and implementation

The graphical design aims to meet the functional requirement of ScorX player. The basic structure is similar to Flash solution with a music score display area and a control area including play/stop function and single track volume control. The interface is implemented in HTML with CSS. The GUI is shown in figure 7.1

The left area is an area contained in an HTML element (i.e. a ”div”) to load the music score in image file format. The right area contains several progress bar, button and check box build from HTML elements. There is no extra design with colored decoration as the reason described in the Delimitation. A new feature has been introduced as a mute function for each sound track.

7.2 Implementation of Audio solution

The audio solution is implemented in Web Audio API. The logic is demonstrated in figure 7.2. This prototype contains a music score with five sound tracks.
Several sound track source files (e.g. mp3 file) will be loading by calling an XMPHttpRequest(), the file will be loaded in to a temporary file for decoding. After the sound files are decoded into playable raw sound data, these sound tracks will be loaded into another buffer. Then an AudioContext will be created using Web Audio API.

In the AudioContext, five source node will be created to load the decoded sound track from the buffer. The source nodes are then connected to five individual gain nodes which are used to control the volume for each individual sound track. These gain nodes will then connect to the destination node to be ready to play.

### 7.3 Implementation of Animation solution

The picture of the Adobe Edge Animate is shown in figure 7.3 working sample.

The designed GUI will be loaded to Edge Animate editing window as the base layer of the player. A cursor is drawn over the base layer, the key time point for cursor to move in between is measured by sound recording software called Audacity\footnote{Audacity\textsuperscript{\textregistered} is “free, open source, cross-platform software for recording and editing sounds”[58] The key time resolution is up to one millisecond.}. The key time
points are arranged in the playback time control interface. For each key time point in the control interface, the cursor is positioned in the corresponding area on the music score. Then between two key time point, the cursor is configured to move in a fixed speed.

### 7.4 Interaction with Audio and Animation

The implementation of the Audio and Animation is running relatively independent to each other. In order to use the GUI to control the audio playback, volume and animation playback, the user interface should have some function to connect these element together. In this solution, such a connection is done using Javascript via Edge animate editing window.

#### 7.4.1 General playback and volume control

In figure 7.1, a play button and a stop button are placed in the right area. The function behind it is edited from Edge Animate editor, see figure 7.4.
The play button, which is an HTML element, can be inserted certain JavaScript code in this editor, in the figure it can be seen that the function has some checking in advance. If the sound is not fully loaded to the buffer, or the player is already playing currently, the play button will be disabled. Otherwise, once click the play button, both audio and animation will be triggered to play at the same time. Same case goes to the stop button. There is a pop up notification to tell user when the loading procedure is completed.

The volume control is done by directly creating extra JavaScript file. For each time a user moves the control nonius, the value of that volume control bar is changed. This changing will trigger the a JavaScript function to manipulate the gain node for corresponding sound track. For the muting function, the concept is the same. When check the mute box, the sound track will be muted. When uncheck the box, the volume will be 100
7.4.2 GUI control for playback time

The HTML5 solution also has a playback time control. When user double click a specific area of the music note, the sound play and the cursor will automatically jump to that point and playing from there and goes on. This is also edited in the Edge Animate editor. Similar as the play button function inserting, a function is added in the overall environment of the HTML page and will be triggered by a double click action.

When the function is triggered, the function will use a method provided in Jquery library (pageX and pageY [59]) to determine the horizontal and vertical position of double click in pixel. If the click is located in the area that the cursor path covered, the double click action will be considered as a valid action, otherwise the click will be ignored. For a valid action, a calculation will be done to determine what the corresponding time point would be, and then control Web Audio API to start play at that time point.

7.5 Extra pointing support on mobile device

One of the advantage of HTML5 is that it has a build-in touch control support (e.g. touchstart)[60]. This has also be built into Edge Animate. Similar to time control, the HTML5 solution has added a touch triggered function. When using a tablet or mobile phone, the player will listen to at touch action made by users. If a touch is considered as a valid touch (same way as a valid click), the playback time control will be triggered for both audio and animation.
Chapter 8

Evaluation

This chapter presents the design, conduction and result of evaluation on both Flash player solution and HTML5 solution followed with the discussion of quality assurance.

8.1 Testing

The evaluation will be done with a comparison with Flash player solution and HTML5 solution based on following factors:

- Supportability on different device and browsers
- Performance of different solutions
- Functions and features of different solutions

The Flash player solution is hosted on the live website of ScorX. [1] while the HTML5 solution is hosted on a development server of ayond AB. The version of browsers mention in this section, if not version specified, will be the latest available non-beta version during the November of 2013. The test conducted with Web Audio API used the latest version of Web Audio API.
8.2 Supportability

The evaluation of the supportability is conducted on both desktop computer and mobile device, several browsers of each kind of device are used to test whether each solution could successfully run in them.

8.2.1 Desktop browsers

A test is conducted trying to run both solution on major desktop browsers. The type of browsers are chosen based on the study of browser usage in early chapter. The result is shown below:

<table>
<thead>
<tr>
<th></th>
<th>Internet Explorer 9+</th>
<th>Google Chrome (latest version)</th>
<th>Firefox (Latest version)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash player solution</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>HTML5 solution</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 8.1: Supportability of both solution in different browsers

Since there are two major versions of Web Audio API with different syntax, a problem has been noticed that Firefox browser has only support the later version of Web Audio API where to start and stop play, function `start()` and `stop()` should be called[61], while the older version has the corresponding different functions called `noteOn()` and `noteOff()` [51]. The earlier version cannot be running in Firefox browser.

There used to be a problem that the iOS Safari version tested in mid 2013 (the latest version in that time) do not support new version of Web Audio API. However, the iOS7 contains a new version of iOS Safari which support later version of Web Audio API. Thus the the older version of Web Audio API should not be used anymore.

8.2.2 Devices

A test is conducted trying to run both solution on desktop device and mobile device. The type of mobile device is chosen based on the study of mobile device usage in early
chapter. The target host software (i.e. browsers) will be chosen based on the study of supportability in earlier chapters. The result is shown below:

<table>
<thead>
<tr>
<th></th>
<th>Desktop (Google Chrome)</th>
<th>Android phone (nexus 4 with Chrome and Firefox)</th>
<th>iPhone4 (Safari)</th>
<th>Android tablet (Samsung galaxy Tab II with Google Chrome)</th>
<th>iPad3 (Safari)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash player solution</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>HTML5 solution</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 8.2: Supportability of both solution in different devices

It could be seen that HTML5 has support all devices while Flash player only supports desktop devices. The supportability of Flash in difference device is very poor compared to HTML5 solution. Combined with browsers test, HTML5 also covers around 90% of the users (see figure 6.1).

8.3 Performance

There are many different aspect to evaluate the performance of a web music/animation player. In this project, three factors are used to be the testing target which are believed to be the most important factor for such a player. The factors are:

- Loading time
- Stability
- Animation user experience

8.3.1 Loading time

From the web page is loaded to the player is ready to play, there is a time in between, a longer loading time would be considered as a fairly bad user experience. Two tests
has been conducted on Flash solution and HTML5 solution based on their different supported browsers and devices. For certain browsers, the loading time varies a lot, in this case the loading time will be shown as a range from shortest time to longest time. The result is shown below:

<table>
<thead>
<tr>
<th></th>
<th>Internet Explorer 9+</th>
<th>Google Chrome</th>
<th>Firefox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash Player solution</td>
<td>2s-3s</td>
<td>2s-4s</td>
<td>2s-3s</td>
</tr>
</tbody>
</table>

Table 8.3: Loading time for different browsers and devices of HTML5 solution

<table>
<thead>
<tr>
<th></th>
<th>HTML5 solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Chrome</td>
<td>5s-6s</td>
</tr>
<tr>
<td>Firefox</td>
<td>1s-3s</td>
</tr>
<tr>
<td>Google Chrome with Samsung Galaxy Tab II in Android 4.1</td>
<td>5s-8s</td>
</tr>
<tr>
<td>Google Chrome with Nexus4 in Android 4.4.2</td>
<td>10s-22s</td>
</tr>
<tr>
<td>Firefox with Nexus4 in Android 4.4.2</td>
<td>5s-15s</td>
</tr>
<tr>
<td>Google chrome with iPhone4 in iOS 7</td>
<td>5s-10s</td>
</tr>
<tr>
<td>Google chrome with iPad3 in iOS 7</td>
<td>1s-2s</td>
</tr>
<tr>
<td>Safari with iPhone4 in iOS 7</td>
<td>5s-10s</td>
</tr>
<tr>
<td>Safari with iPad3 in iOS 7</td>
<td>2s-4s</td>
</tr>
</tbody>
</table>

Table 8.4: Loading time for different browsers of Flash player solution, all mobile devices are tested under same network environment

Compared with the desktop test on both solution, Flash player loading faster than HTML5 except in Firefox. The loading time is stable in Flash player solution while HTML5 solution could varies a little bit more. Regarding to the mobile device test on HTML5 solution, the result has varies even more.

Several factor has been noticed that could effect the loading time of HTML5 solution in mobile devices:

- Type pf device
- Type of browser in same device
• Network connection speed and stability of different device

• Battery level and number of background application running in the device

There is hardly a trend or a logical pattern could be found for how much it effect the loading time. The only conclusion that can be draw in this test is that the mobile tablet loading time is slightly short, this conclusion still need to be further verified by introducing more tablet devices.

8.3.2 Stability

The principle of testing the stability of both solution is simple. As long as the player does not crash during data loading period or playing period, the performance could be considered as stable. 15 times of repeated test has been done on each device and browsers.

The test result shows that the flash player solution is almost stable at all time, except the case that the testing desktop device has an incorrect installation or an unstable version of Flash Player. This problem could always be fixed when the newest version of Flash player is installed in the device. The average crash possibility is 2%.

The case for HTML5 solution will be different. On the desktop Chrome, if one load to many sound tracks or too large sound tracks (during the test there are up to 10 sound track and up to 60MB of sound files), the Web Audio API will crash, the average crash possibility is 90%. This problem has been tested on both old and new version of Web Audio API. The new version has much less times of crashes, the average crash possibility is 5%. In the Firefox version, there is no crash ever happened under same test scenario.

On the mobile version, all of the browsers in all kinds of devices has experienced crash. However, the crash only happens on iOS devices when the old version of Web Audio API is used, the average crash possibility is 5%. On Android device, the Samsung tablet crashed about 10% of times while the Nexus4 crashed in about 2% in both old and new version of Web Audio API.

The stability of Flash player solution is better than HTML5 solution.
8.3.3 Animation user experience

The animation user experience test is a visual test conducted during the animation is playing. In this case is mainly the movement behavior of the cursor.

Compared with HTML5 solution, the Flash player solution, in all target browsers, the cursor moves with less frame per second, the visual experience could be described as mild jumping and not smooth. The HTML5 solution has a smoother cursor movement in all device and all browsers. However, an other test conducted in some non-major browser like Sougou browser developed in China, the cursor has a dragged trace that covers along the path. It could only be eliminated when scrolling the page up and down a little bit.

The overall conclusion would be that the Animation user experience of HTML5 solution is better than Flash Player solution.

8.4 Other functions

The function mentioned in this page is not implement in HTML5 solution but has been implemented in Flash Player solution. The reason is explained in Delimitation. However, this section will provide a possible solution.

8.4.1 Auto focus

The auto focus function means the target page will be automatically focused on the center of the ScorX player then there are multi-pages. To implement such a feature, Haxe solution for the Animation could be used. As a matter of fact, there is a HTML5 version of compiled Haxe example that has been developed by the developer of Flash player but it has not yet been tested with Web Audio API.

8.4.2 Synchronization

As mentioned in the study of Flash player solution, the animation will be periodically synced as a slave with sound playing to make sure that the audio and animate are
synchronized. The HTML5 solution prototype did not have such a mechanism. The reason is because during the playing test, there is no out-of-sync happened even the animation and audio are running independently, which is sufficient for current phase. The only exception happened when try to scroll the page using iOS device. If the page is scrolled when the playing is taking place, the sound will keep playing while the animation will be stopped. The Animation then will be resumed after scrolling is complete, then the cursor is out-of-sync with the sound playing.

The Web Audio API did not provide a method to get the current playing time but only provide a method that can provide current AudioContext time[51]. The AudioContext time is 0 in the beginning and start counting when the playing function is triggered[51]. Then it will not stop even the playing is stopped. A test suggested that it would only stop when the AudioContext is destroyed.

The solution would be to create a start-time variable to record the current AudioContext time every time the playing is started, including the case that the playing is re-started. If the playing is not start from beginning, an offset will be calculated and recored in the start-time variable. Then along with the playing, a function could be periodically called to get the current AudioContext again. This time point minus the start-time point will be the actual time point in the whole sound playing. This makes it possible to sync the animation play with the sound play in HTML5 solution.

8.5 Quality Assurance

As mentioned in Chapter 2, dependability, confirmability, transferability and ethics will be discussed.

8.5.1 Validity

The data measured in this project are based on the evaluation factor described by numbers. The measurement design, implementation and the actual result can be and has been acknowledged and proof-read by the supervisor.
8.5.2 Dependability

The delivery of this project has been verified by the supervisor of Ayond AB, the correctness has been proved.

8.5.3 Confirmability

In this project, the confirmability could be verified by testing the final delivery, all the design and implementation of experiment and evaluation can be and already has been audited and verified by the supervisor from Ayond AB.

8.5.4 Transferability

The delivery in this project has clearly indicated that it is possible to build a fully functional HTML5 substitution of Flash player ScorX player. The solution could also be a feasible guideline for other similar multi-media web based service. The delivery has been proof-read by supervisor and CEO from Ayond AB, the feedback is that it is clear and helpful for the ScorX project and future development.

8.5.5 Ethics

Generally the design, method and result of this project reveals no personal information and no commercial secrets, thus it is considered that no ethical issue is involved in this project.
Chapter 9

Conclusions And Future Work

This chapter presents the conclusion of this thesis, discussion of both technical research/implementation results and the logical design, research and performance of this thesis work. Finally the future work will be presented.

9.1 Conclusions

This thesis presents the work with developing a prototype of ScorX MPMS player with web technology as well as the evaluation of this prototype compared with a Flash Player solution.

This project, in the strategic level, has found one theory or technology to solve this problem which is using HTML5 as the substitution of Flash Player solution. The initial requirement and planing is based on the study of Flash player solution and requirement from Ayond AB. Meanwhile, among various of HTML5 API, programming solution and tools, this project finally pick Web Audio API as the audio engine, Adobe Edge Animate as the animate engine and embedding tool. The final solution is a cross-platform Web application. In this case, the goal, which is provide a web based MPMS player solution, is fulfilled.

Since the study of HTML5 shows multiple choice for building audio and animation solution. As this nature of uncertain choice of different HTML5 API and tools, Agile and scrum are the most suitable software development method. Several development lift
cycle will be conducted until the prototype has meet the requirement. After prototyping, evaluation is conducted by taking certain test and measuring certain data. It shows that the HTML5 is fully capable of implementing the ScorX multi-track music player, however, Flash player solution for desktop ScorX Player has a better performance and stability and therefore is still the first choice of the current product solution. As the purpose of this thesis is to present the work of prototyping of HTML5 MPMS player, along with the short procedure summary mentioned above in this section and in previous chapters, it could be considered that the purpose is fulfilled.

Even though evaluation shows that HTML5 is not mature enough, a fact should be noticed that during the development period, the support and performance of Web audio API has been "quietly" improved in the major web browsers, people involved in the ScorX project has realized this fact which means it is more likely that the next generation of ScorX MPMS player will take HTML5 solution into consideration.

The work of study and implementing HTML5 player as well as comparing it with different solution could be a usable reference and suggestion for the further work of ScorX service development as well as other developers who is think to have a substitution of Flash player based multi-media web service. The current delivery and the further work could also be helpful for other people who is trying to explore the multi-media feature of HTML5 and provide similar product or service for personal and commercial use.

9.2 Discussion

As mentioned before, the initial problem of the subjected industrial area of this thesis project is the current Flash Player solution cannot support multiple hardware platform, thus this limitation leads to potential customer loss.

To solve this problem, the solution presented in this thesis project has utilized HTML5 technology to support multiple hardware platform. Even though it is less stable nor compatible, as more and more major web browsers start to support or improve the current support of the Web Audio API audio engine as well as animation, the future of the development capability is brighter. Besides the cross-platform solution conducted
with HTML5 Web application, there are also other possible technology to conduct cross-
platform application such as using Haxe. Specific native mobile application for different
platform will be the complement mobile support solution besides Flash player.

During the test, some developer in ScorX project has also discovered some mobile
browser could support flash player, however, as there is no official documentation from
Adobe mentioned the mobile support, this browser is suspected to used a so called
"hacked" version of flash player support which could be less stable. This provides a
reason to hope Flash player could be supported in the mobile device in the further work
by a third-party customized support.

Regarding to this thesis project, the problem statement is clearly addresses the issue to
be solved, and the definition and categorization of the research design and methodology
are considered to be correct and straightforward. It is believed that the study of this
thesis has found a fairly balanced point between over use the previous work’s concept
and totally redefine the product (i.e. start from scratch).

However, there are certain part that could be improved:

- The interview with the previous Flash Player solution has not been well designed,
  neither dose the recording method. This caused the communication to be less
efficient and harder to extract useful information from the interview.

- The author was new to the HTML5 area. This caused the limitation of the study
  scope by ignore the nature of HTML5 area, which is developing fast and variously.
The related work and relative technology could be much more than what mentioned
in this project. A better awareness of this situation could potentially lead to a
better solution.

- The design of evaluation concept and method is rather scattered than systemic,
certain factors are picked solely based on experience. A better design could lead
to a more constructive and comprehensive conclusion which will make this thesis
more useful.
9.3 Future Work

The Flash player solution will be less improvable than HTML5 solution, the remaining problem of Flash player solution could last for a long time without a proper solution. In this case, the future work of ScorX player should not be focus on Flash Player anymore. Instead, more effort should be put on using HTML5 or another solution such as native application for mobile devices.

9.3.1 Current HTML5 solution

Besides the performance problem mentioned in earlier section, the development of HTML5 could be improved by adding the not-yet-implemented feature with a full stack of front-end to back-end solution. In the coding, the performance could also be potentially improved by have a more optimized coding structure.

One of the advantage of HTML is the corresponding style design (CSS) of the element could varies depend on the size of the browser windows. A proper CSS design could make the HTML web page to adapt the size of the window to the size of the mobile device screen automatically. This feature makes it possible to design only one web page that will be suitable for all kind of supported devices. (the term could be refered as a responsive web design).

9.3.2 Other possible web solution

In chapter 6, it could be seen that the audio tag supportability is better than Web Audio API. Thus it would be nice to somehow use this method with a proper plan. One thing that could be done is that instead of process the sound in the front-end, the sound could be pre-processed from the back-end.

This means that the user will try to request a combination of sound tracks, could be one of the whole set or several tracks from all tracks. The request will be sent to the back-end. The back-end server will fetch the sound tracks that are requested and mix them into one sound tracks and send to the front-end. In this case, there will only one audio tag instance in the front-end which eliminate the out-of-sync problem for multi-track playing.
The drawback will be that every time a user trying to listen to a different track mix, a new request will be sent to the back-end, in this case the data communication times are increased.

**9.3.3 Haxe based native application for mobile device**

As introduced in Chapter 1, the Haxe code could be compiled into different target. This makes it possible to use one Haxe solution to build native application for all kinds of mobile devices. This choice will be further discussed when the web based cross-platform solution cannot meet the company’s requirement of the new version of ScorX MPMS player.
Appendix

Figure 9.1: Flash Market position in 2014, the term Javascript in this figure represented HTML/HTML5 [33]
Appendix A. Appendix Title Here

Figure 9.2: Snippet of Nocturne No. 2 in E flat major Op 9 [62]

Figure 9.3: ScorX Player Sequence Overview 1
Figure 9.4: ScorX Player Sequence Overview 2
Bibliography


Bibliography


[42] Interview with Jonas Nyström, the technical leader of Flash version of ScorX player. Interview, 2014.


