Multimedia Services : Final Report

Group BeNine
TI2316 Context Project

Bryan van Wijk (bryanvanwijk, 4363329)
Dorian de Koning (tcmdekoning, 4348737)
Ege de Bruin (kedebruin, 4400240)
Jochem Lugtenburg (jlugtenburg, 4370805)
Naomi de Ridder (nderidder, 4383109)

June 16, 2016

Supervisor: Dr. Cynthia Liem
Software Aspect TA: Valentine Mairet
Context Aspect TA: Alessio Bazzica

Delft University of Technology
Faculty of EEMCS

Contents

1 Introduction

This document contains information about the developed, implemented and validated software product. The document is divided into seven main chapters. First, a brief explanation about the problem description of the product and the end-user's requirements will be given. Second, an overview of the developed and implemented software product will be given. Third, the product and process will be reflected on from a software engineering perspective. Fourth, a description of the developed functionalities will be given. Fifth, an explanation of the interaction design part of the project will be given. Sixth, the functional modules and the product in its entirety will be given. Last, an outlook of the product will be given in which we will look at the possible improvements for in the future.

For this project team BeNine has worked with (and for) an already existing company named PolyCast. PolyCast is a recording company, with its primary focus on classical music productions. Currently PolyCast works with a control panel that is very basic but to execute a simple task, a lot of mouse clicks are needed. Because this control panel is inefficient and requires more man-power than it should need, a new web interface with the same possibilities as the control panel has been made this project. With this web interface it is not only possible to create presets and recall them but also search for them with tags or auto-create presets from a certain position. All of the different features will be explained further in chapter four.

The main focus of this project was optimizing an already existing idea and prototype. The final product can replace the old control panel of PolyCast and tasks can be executed in way less clicks than before.

2 Developed and Implemented Software Product Overview

During the project, the team has been actively developing and implementing a solution improving the preset work-flow of the camera operator. The software product is divided in two main parts, namely a front-end and a back-end server. This section is composed in three subsections. First, a high-level overview of the software architecture division is given. Second, important technical details about the back-end server will be given. Finally, the same will be given for the front-end server.

2.1 High-Level Software Architecture Overview

The system is divided in a front-end and a back-end server. The front-end server being the view, the back-end server being the model. The system has been decomposed in these parts to make sure the back-end server can be used without a view. The back-end server provides a clear API which can be used to control camera's and manipulate presets. On the other end is the front-end server. The front-end server provides the user with a clear user interface, using the back-end server to communicate with the camera's and store presets. In Figure 1, a global overview of the system components is given.

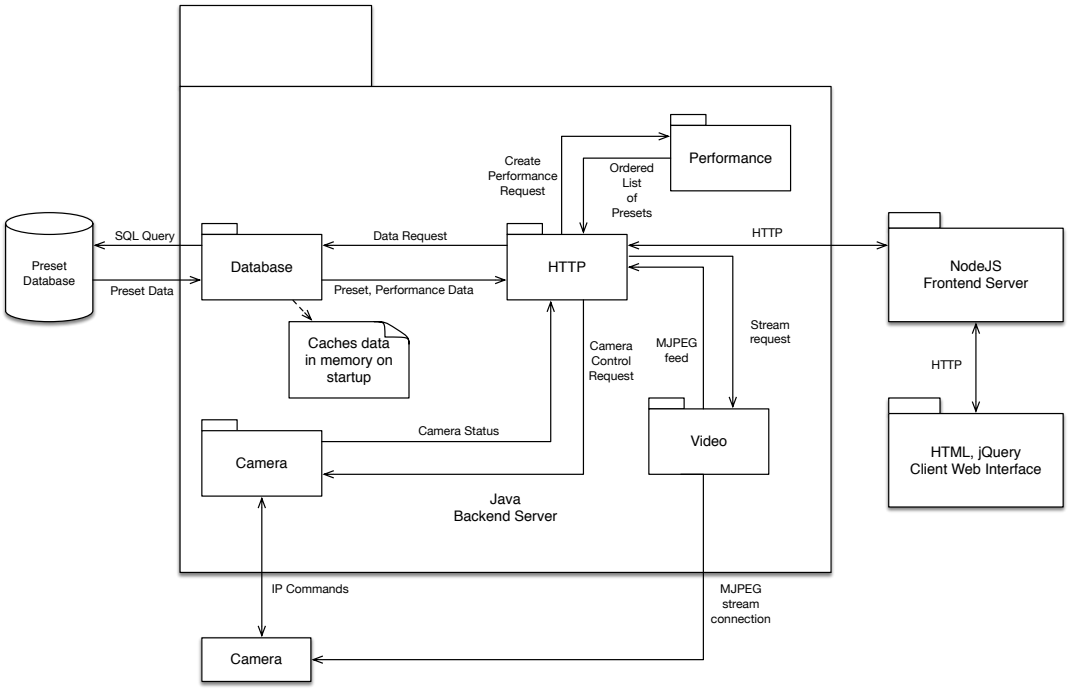

Figure 1: Global overview of the system

2.2 Back-end Server Overview

The back-end server consists of a model written in Java. The back-end server is responsible for all computations, camera communication and preset management. The main computations involve camera positioning and an automatic preset generation algorithm which will be discussed later. Cameras and presets are stored in a MySQL database, so they may be retrieved later. Camera's are stored with a MAC address, so they can also be identified on a different local area network. Presets contain location and image data and can be ordered in a sequence to make recalling more efficient. All features above are presented to the client (in this case the front-end server) using a clearly documented API.

2.3 Front-end Server Overview

The NodeJS based front-end server consists of a Twitter Bootstrap based web interface supported by custom HTML, CSS and jQuery. The NodeJS server acts as a bridge between the user interface and the back-end server API preventing cross-origin resource sharing security restrictions by various browsers. This setup makes clear that the front-end server can be removed from the product. For instance if the client wishes to use the camera control functionality as part of a different product, they may choose to remove the front-end server and only use the back-end API.

3 Software Engineering based Reflection

This section contains a reflection on the development process and the product from a software engineering perspective. For both the development process and the product, a short description and lessons learned are given.

The development process was distributed over eight SCRUM sprints, with each sprint iteration lasting a week. Each sprint concluded with a TA and supervisor meeting. After this meeting the team reflected on the previous sprint in a retrospective meeting, taking into account feedback received from the supervisor meeting. After creating the retrospective the team used experience from previous sprints to create a plan for the next iteration. Creation of a plan involved writing user stories, discussing about the tasks involved and finally assigning these tasks to individual group members. A problem which arose frequently, was the difficulty of estimating the amount of work involved with a task. Even if big tasks were split into smaller parts problems persisted. Because the system is relatively small, changes affecting each other were common. This required other team members to adapt, which took most of the wrongly estimated time. An important lesson learned, was taking into account other team members tasks and making sure a task is executed on time so conflicts can be prevented.

A supporting pillar of the development process was the use of GitHub and Travis CI. By preventing merges into the master branch if Travis failed, chances of the system becoming unstable were minimized. The team frequently discussed problems and reviewed changes in GitHub Pull Request, further preventing issues. Being strict on these reviews helped maintaining quality and is definitely something to keep in future projects.

The product itself has been created using SOLID design principles. Stimulating code reuse with inheritance and design patterns. With the help of the tool supplied by the Software Improvement Group, further design issues such as a high fan-in and high amounts of parameters were identified and solved. A tool identifying these issues definitely helped the team to improve quality. Next to a tool identifying architecture issues, tools such as PMD and CheckStyle were used to maintain comment, naming and convention issues. Because the SIG tool was introduced later in the process, some problems had already spread into the program. Using such a tool earlier in the process might further improve quality.

4 Developed Functionalities

This section gives an overview of the main developed functionalities of the product. Since the team has created a web interface, all of the functionalities can be executed on this.

4.1 Camera Control

Since a large part of the features implemented depend on the camera, camera control is a large part of the system. Not only camera control with a touch interface and visible Motion JPEG streams is implemented, but also the actual HTTP communication with the Panasonic HE-130 camera. The communication can be used to recall presets, a core feature of the program which will be discussed in the next paragraph.

4.2 Live performance view and edit view

The web interface has two different views: a live performance view and an edit view. In the edit view a visual representation of the streams as well as the controls for the camera can be seen. The camera can be moved around and presets can be created, edited and recalled. It is also possible to adjust the settings of the controls so that little things can be easily adjusted. In the live performance view of the web interface all of this has been left out. The live performance view will be explained later in the performance scheduling paragraph.

4.3 Preset Creation and Management

To create a preset, the user is presented with a small modal, enabling naming and tagging a preset. A preset consists of three main parts: a name, a list of tags and an image representing the state of the camera on preset creation. The list of tags can be easily searched, returning presets with similar tags. When a preset has been created, it is possible to recall this preset. This is done by simply clicking on the desired preset. It is also possible to edit already existing presets using an easily accessible interface.

4.4 Performance Scheduling

In the edit view it is possible to prepare a sequence of presets that will be loaded in the live performance view. An easy to use menu has been built for this purpose. A new performance is created and named and the presets to be included are selected. The order of the presets can also be adapted. When a performance is created, a list with the order of the presets is given in the live performance view. The next preset in this list can be selected and then the camera will automatically move to the position specified with the preset.

4.5 Automatic Preset Generation

To further improve the preset creation work-flow, the camera can now automatically create a preset hierarchy. To do this, an area of the camera can be selected, after which the camera divides this area in selectable subviews. The amount of rows and columns can also be selected. Each subview will then be converted to a preset.

5 Interaction Design

For the interaction design part of the project, several people have been interviewed. It would have been useful to interview PolyCast so that we could have adapted the product to their needs but because they are busy all the time, this wasn't possible. For this reason the team has interviewed 3 other people that did not have any prior knowledge of the product. The different persona's, the method of interviewing and the conclusion of the interaction design test will be explained in the following paragraphs.

5.1 Personas

Persona 1 and 2:
Persona 1 and 2 are very much alike. They are both male students from the TU Delft and they are currently working on the HI project. They did not know anything about the product beforehand but they are familiar with different kinds of web designs and they know what they have to look at critically.

Persona 3:
The last person interviewed is a girl that just finished high school. She does not have any previous experiences with web interfaces or computer science related topics at all. Because she has no experience yet with any of our related topics, she can give good feedback on how user-friendly the web interface is. By looking at her feedback we can make our product easily understandable for people with zero computer science experience as well.

5.2 Method

As said before, several people have been interviewed because this would give us the most information and useful feedback on our product. With all this feedback the product can be improved to make it more user friendly. The interview started with a short introduction about our project and PolyCast. After that the persona's were asked for the first impression and this already gave the team some useful feedback. The team then walked through all of the different features of our web interface by asking the user to do something like 'recall the preset with the tag piano'. A question was asked, the timer was started and the persona's figured out how to do it in the web interface, without any help. When they finished the task the team stopped the timer and wrote down how many seconds the task cost. It is important for us to know that the system is fast because this is an important feature of our product. When they finished the task of e.g. recalling the preset, more questions on that part of the system were asked to get to know what the user was thinking about a certain screen, button or functionality. All of the questions asked can be found in the appendix.

5.3 Results

The whole interview and the answers from the test persons can be found in the appendix. All of the users gave really useful feedback, mostly about the user interface. Something that came back several times is that the button to edit presets is unclear. Some other feedback was about the way the web interface is styled, e.g. that the sliders in the setting screen are too small, or that there are too many colours used. Another feature that came back was that the tagging wasn't optimal yet, e.g. it is possible to add an already existing tag in the

UI.

Persona one and two were equally fast but persona three was a bit slower than the first two. This is probably because persona 3 had zero experience with using a web interface like this. However, the slowest that anyone was, was 35 seconds, which is not that slow if you are using something for the first time. When PolyCast is going to use the interface, they will get an explanation of all the different features and they get the opportunity to practice with the system. The result of this would be that the same tasks can be executed in way less time.

5.4 Conclusion

All in all, the users were really positive about our web interface and mostly gave some small improvement ideas. However, some of the things that the users pointed out, like the empty space in the bottom right, were due to the fact that the laptop we used is bigger than the iPad that will be used by PolyCast. So these problems will not occur when using the product. A lot of the feedback has been processed by making issues of them. The three different parts that the team has changed are the tags screen, the edit preset button and the settings screen. Some features that the team hasn't implemented are e.g. searching for multiple tags because the library doesn't support this or pressing the space bar to save a tag because then you can't add tags with spaces in it.

6 Functional Modules

This section gives an overview of the four main functionalities of the program and evaluates if they fulfill the needs of the user.

6.1 Camera Control

The Camera Control module mostly acts as a supportive feature for the Preset and Auto preset features. The team communicated with PolyCast about the controls in the user interface during the middle of the project. The visual representation of the streams in the user interface was unnecessary, since they have faster real time screens and controls in front of them. This lead to the decision of splitting the program into an edit and live performance view, removing the controls and streams. The client also stated that they might want to move to different 4K enabled cameras in the future. For this purpose it is made possible to easily extend the program with new cameras, without requiring a complete rewrite of the product.

6.2 Live performance view and edit view

As a result of communication with PolyCast, the product has been adapted to match the workflow more. In the early stages of the project, the team focused on implementing preset editing and management. After a chat with PolyCast, it became clear that they would like a way to schedule presets so they can recall and prepare. This resulted in splitting the user interface into a live and edit view.

6.3 Preset Creation and Management

Currently, a camera operator manually needs to press a lot of buttons to create and recall presets. The program simplifies this by presenting the user with a small modal, enabling naming and tagging a preset. While preset creation is not much faster compared to the current method of preset creation, a real speed gain is achieved in recalling presets. During a live broadcast performance, the camera operator can quickly search all presets using the tags he or she has assigned to a preset to make quick recalling possible. During a rehearsed performance, the camera operator may prepare a sequence of presets. Recalling a preset is as easy as tapping the preset on a tablet device, instead of pressing multiple buttons to recall a scene. This time can now be invested in the more creative aspects of the job, such as re-framing the shot.

6.4 Performance Scheduling

When the live view was created, preset scheduling became an issue. For this purpose the performance scheduler was created. This scheduler allows the camera operator to create a list of presets based on the director script. This makes selecting the next preset much faster because it is already present in the User Interface, further improving speed.

6.5 Automatic Preset Generation

The Automatic Preset Generation feature supports the creative aspect of preset creation. It allows a set of presets to be created, and selected afterwards. This makes it more easy to see

different preset variants. If the process is done, the camera operator may delete unnecessary presets and tweak interesting ones.

7 Outlook

While the team has striven to implement all features in the best possible way, further improvements can still be made. A few features were considered, but not implemented due to time constraints. These features are described in the list below.

- Preset similarity detection

  A comparison between presets can be made using computer vision algorithms. This allows the end-user to look for similar shots.

- Integration with other products

  The current camera preset management and recalling features could be used to extend other projects. Currently, team 'goto-fail' uses the back-end server API to facilitate this. This notion of integration can be used to create one overarching software product eliminating the need to use multiple products by PolyCast.

Appendix A User Interaction Design Script

Introduction

For our project we have made a web interface for PolyCast. PolyCast is a company that records mostly classical concerts. Currently they work with a control panel but this does not work optimal because a lot of mouse clicks are needed to for example select the right camera and preset. We have made a web interface with which you can not only control the camera's but also create presets, recall them and search for them with tags. All of this can be done in way less clicks than before.

Interview

Before you you see our web interface for PolyCast. I will not give an explanation of it yet because you will find out all the different features yourself by answering some questions.

First impression:

Before we are going to work with the web interface, I would like to ask what you think of the style of the web interface.
Some supporting questions:

- What do you think of the different blocks placed on the page/ how everything is arranged?
- What do you think of the colours used on the page?
- Are there buttons that are already unclear or that you would have placed somewhere else?

Features:

Create presets: If you would want to create a preset, how would you do this?
What do you think of the different options in the create preset screen? Are there buttons that are unnecessary?

Editing preset: If you would want to edit a preset, how would you do this?
We created the editing of presets this way so that you can turn it on and off. Because the app will be used on an Ipad, we thought it wouldn't be a good idea to make editing an option below the presets because it is easy to miss-click.

Recalling a preset: If you would want to select a preset with the a "piano", how would you do this? Are there things that you miss when searching for preset with the tags?

Updating tags: If you would want to update a tag, how would you do this? What do you think of the different options/ buttons in the tags update screen?

Changing the settings: If you would want to change the settings of the joystick, focus, iris and zoom, how would you do this? What do you think of this screen? Are there too many buttons/ options or is it clear this way?

Reviewing:

Now that you have tried several features yourself, I would like to ask what your impression of our web interface is.

- Are there things that are still unclear?
- Which things would you change now that you have tried our system?
- Do you have any further questions and/or comments?

Appendix B The interview

Test person 1:

First impression:

- The gray box in the bottom right corner is empty. Maybe the joystick can be moved to the right a bit.
- The images of the preset are stretched out.

Creating a preset: 9 seconds (clicking on the button, creating a preset and saving it)

- It isn't possible to press enter to save the preset.
- When adding tags it would be useful to press the space bar to add the tag.

Editing a preset: 12 seconds (clicking on the button, editing a preset and saving it)

- The name and description aren't saved.
- It is unclear that you can turn editing a preset on and off.
- When a preset is visible on your screen, it would be useful to immediately be able to edit this preset. Now you have to enable editing and click a preset again.

Recalling a preset: 12 seconds (searching for the tag and clicking the preset)

- It isn't possible to search on multiple tags.

Updating a tag: 14 seconds (clicking on the tag button and changing a tag)

- The position of this updating a tag button isn't very logical. It can maybe be placed somewhere else.

Change the setting: 3 seconds (clicking on it and moving a setting)

- If you are changing the settings you want to be able to play with it a bit before saving it.
- Zoom, iris etc. should start with 1, not 0.

Test person 2:

First impression:

- Nothing to point out yet.

Creating a preset: 8 seconds (clicking on the button, creating a preset and saving it)

- The name of the preset isn't visible yet.
- To create a preset you have to move the camera to that position. It would be useful to be able to fill in the position yourself.

Editing a preset: 9 seconds (clicking on the button, editing a preset and saving it)

- It is unclear that you can turn editing a preset on and off.
- The values that you already filled in should be visible when editing a preset because it is possible that you want to change only the name and not the tags.

Recalling a preset: 25 seconds (searching for the tag and clicking the preset)

- This was difficult to find because of the gray background with text that looks a lot like the background and because it is placed in a corner.
- A button like e.g. the create preset button would be useful and makes it consistent.
- the tag search bar disappears when scrolling down.

Updating a tag: 13 seconds (clicking on the tag button and changing a tag)

- When adding a new tag it is entered at the bottom. A pop-up or adding it at the top would be better. Now I added 5 new tags without noticing it.
- Empty tags are also added as 'new'. This shouldn't be the case.
- You can have the same tag multiple times. This shouldn't be the case.

Change the setting: 2 seconds (clicking on it and moving a setting)

- The sliders could be bigger because now they are really small.

Test person 3:

First impression:

- Looks really basic.
- Confusing where the different blocks on the screen are used for.

Creating a preset: 15 seconds (clicking on the button, creating a preset and saving it)

- The name of the preset isn't visible yet.
- Not clear what the colors below the live camera's mean.
- What is the difference between name and description?

Editing a preset: 30 seconds (clicking on the button, editing a preset and saving it)

- It is unclear that you have to click on the presets to edit them.
- After clicking on save, editing could be disabled again without clicking on the edit preset button again.

Recalling a preset: 30 seconds (searching for the tag and clicking the preset)

- Was looking for the preset image with the piano without using the search bar.
- Preset images are a bit small.

Updating a tag: 23 seconds (clicking on the tag button and changing a tag)

- When changing a tag, the tags are not changed in every preset. The tag is only added as a new tag.
- The tags in the update tag window could be the same size instead of depending on the length of the tag text.

Change the setting: 35 seconds (clicking on it and moving a setting)

- The settings button in the top is a bit small and difficult to find.

Reviewing:

- The editing buttons like edit presets are on the right while they control the presets on the left hand side of the screen.
- The colors used in the UI could be changed to the colors of the logo.
- The text in the update tags button could be changed to edit tags to be consistent with the edit presets button.