Video conferencing

Jonathan Kasongo

OCR A-level Computer Science NEA

Full name: Jonathan Kasongo
Candidate number: N/A
Centre number: N/A
Centre name: Harris Academy Purley
Qualification code: H446

Contents

Chapter 1

Analysis

1.1 Problem identification

1.1.1 Context

My client Axel Alabi has asked me to create an interactive video conferencing application to allow others to view talks in realtime. The current solution is to use the *Zoom* video conferencing application. While it is true that the application is technically sound and can work fine, there is a large number of elderly users that also try to connect to the talks. These users often don't fully understand how to correctly use the application and then end up accidentally disturbing the conference/talk, by leaving their microphone's on, accidentally raising their hands and so on. This makes my client's job difficult since he is in charge of managing the *Zoom* call. To combat this situation he would like a simple and user friendly video conferencing application that provides the features needed for people to view and interact with the conferences/talks in real time. This includes features like (but not limited to) audience participation, the ability to speak to others via one's microphone and the ability to vote on polls. The application should be created specifically to help elderly people have a better experience whilst watching any conferences/talks, so may also include extra accessibility features to ensure comfortable viewing for all, irrespective of one's age and/or disabilities.

1.1.2 Features the problem solvable through computational methods

Before examining the features that make this problem solvable through computational methods, it is necessary to first establish a guideline for some of the basic requirements for features that should have to be included within the application, based on our contextual knowledge of the problem discussed in 1.1.1.

Requirement	Justification
Real-time video footage from users webcam's.	This is necessary for a *video* conferencing application by definition.
Real-time audio from users microphone's	This is also necessary by definition.
End to end encryption.	Users should be able to interact freely without having to worry about anyone intercepting their data.
A simplistic/intuitive design.	We should ensure that the client's main problem is solved.
Reliability and robustness	The application should be reliable enough so that it is performant (minimal lag) and also has measures in place to handle any errors e.g. network failure.
Some accessibility features	The application should have some features that enable elderly people, or disabled people to comfortably view the talks/conferences like everyone else.

Now that we have defined some base requirements we can now describe the computational techniques that can be applyed to solve the problem, and the features of the problem that make the technique suitable in the context of our problem.

Feature	**Justification**
Real-time audio/video feeds.	To complete this part of the application we could apply decomposition. This problem can be decomposed into multiple sub problems, for example: 1) Establish a connection to server. 2) Ensure user has connected a webcam. 3) Access the webcam via the relevant API. 4) Send the video feed data to the server so that everyone in the call may view the footage. This idea of breaking the problem down into smaller steps allows for a clear and logical approach to implementation. We can also apply a generalisation to the design of the application systems by making use of *structured analysis* where we break down an entire system into smaller components with flow diagrams. [1]
The application should be simple and user friendly.	We may apply the technique of abstraction in implementing this feature. By removing irrelevant information from the user interface we can ensure that the user only sees information that is relevant to them in a simple and clear manner, directly achieving one of our client's requests.
Reliability and robustness	

Bibliography

[1] Tom DeMarco. *Structured Analysis and System Specification.* Yourdon, 1978.