A QUALITY OF EXPERIENCE TESTBED FOR VIDEO-MEDIATED GROUP COMMUNICATION

Marwin Schmitt, Simon Gunkel, Pablo Cesar
Distributed and Interactive Systems
CWI: Centrum Wiskunde & Informatika
Amsterdam, The Netherlands
{schmitt,gunkel,P.S.Cesar}@cwi.nl

Abstract— **Video-Mediated group communication is quickly moving from the office to the home, where network conditions might fluctuate. If we are to provide a software component that can, in real-time, monitor the Quality of Experience (QoE), we would have to carry out extensive experiments under different varying (but controllable) conditions. Unfortunately, there are no tools available that provide us the required fined-grained level of control. This paper reports on our efforts implementing such a testbed. The testbed provides the experiment conductor full control over the complete media pipeline, and the possibility of modifying in real-time network and media conditions. Additionally, it has facilities to easily develop an experiment with custom layouts, task integration, and assessment of subjective ratings through questionnaires. We have already used the testbed in a number of evaluations, reported in this paper for discussing the benefits and drawbacks of our solution. The testbed have been proven to be a flexible and effective canvas for better understanding QoE on video-mediated group communication.**

Keywords- QoE, testbed, video-mediated communication, multiparty video conferencing

I. INTRODUCTION

Video-mediated communication services have moved from dedicated solutions, to rich media communication over an open infrastructure. The impact of these asymmetric and fluctuating conditions on the QoE on the participants is currently investigated by industry and academia [1]. In order to adequately address these issues, we need a fine-grained understanding of what effects different system parameters have. on the ongoing conversation. Such knowledge can be gained through extensive user trials under diverse, but controlled, conditions. Unfortunately, currently there is no tool publicly available that gives us the flexibility and level of control required for such studies. This paper reports on the Video-Mediated Communication Testbed (VMC-TB) which is designed for controlled QoE studies. The VMC-TB features a multi-party video-communication client, shown in Figure 1, which gives the experiment designer control over the QoE influencing factors. Additionally, it has integrated features that assist with subjective and objective QoE measures, experiment conduction and post-study analysis. In this paper we detail the requirements for such a testbed and show how they are addressed by the VMC-TB architecture
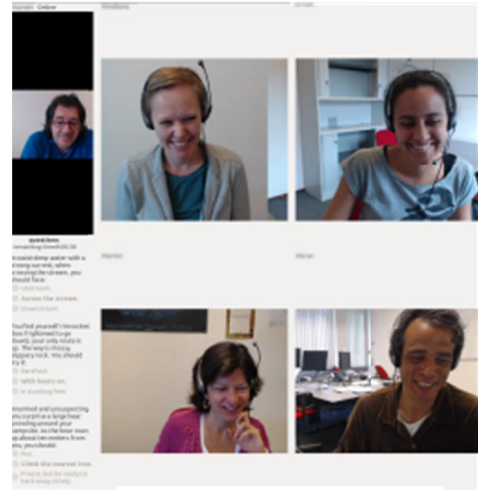

Figure 1 VCMC Client

II. REQUIREMENTS

We derived the following requirements, by modeling the QoE influencing factors [2] and through extensive pre-trials.

R1. Direct real-time manipulation of application layer QoS parameters [2]

Instead of indirect manipulation through network simulators, this approach allows abstraction from a specific system

R2. Monitoring of the application layer QoS parameters.

This is necessary to ensure that parameters not under test are stable.

R3. QoS parameter manipulation for each participant individually.

This allows the investigation of asymmetric scenarios and context.

R4. Session recording from each participant's perspective.

This is necessary for a fine-grained post-study analysis.

R5. Live monitoring and hideable interaction capabilities for the experiment conductor.

In order to ensure that the study progresses correctly monitoring and interacting are necessary. To not influence participants tmust not be visible during the whole experiment process.

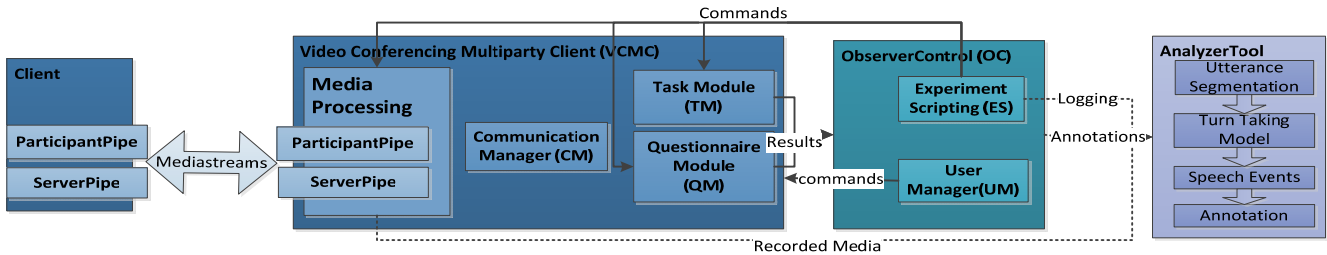**R6. Optional: Experiment Progress integration.**

Figure 2 VMC-TB Architecture

Integration of the experiment status (set parameters, conditions, conducted questionnaires) assures a smooth experiment progress.

R7. Optional: Integration capabilities for activities.
This allows for easy synchronization with the data processing during the analysis. Necessary in some activities (e.g. synchronized media sharing).

R8. Optional: Questionnaires for subjective assessment.
Allows making the questionnaires an integral part of the experiment and reduces idle waiting times during the experiment.

III. VIDEO-MEDIATED COMMUNICATION-TESTBED

An overview of the VMC-TB components can be seen in Figure 2. The media processing is implemented via GStreamer[1]. We detailed the pipeline in [2] showing how the manipulation and control over each system parameter can be achieved (R1&R2). VMC-TB uses a peer-to-peer infrastructure which allows for a fine level of control of asymmetric conditions (R3). We employed a command pattern for communication between components. The commands are tailored for logging, experiment scripting and easy GUI integration. The logs and recorded media streams enable us to analyze the session from each participants perspective (R4). This can reveal for example why a simultaneous start [3] is not noticed by every involved participant. The ObserverControl (OC) is an extended VCMC which is normally only a hidden spectator in the session. If necessary, an audio or audiovisual connection to all or specific participants can be opened (R5). The TaskModule has a set of standard elements that can give instructions about the task and collect answers from the participants. The logged results can be used in the analysis to derive task scores as objective metrics [2]. We also integrated VMC-TB with ambulant [4] for synchronized multimedia sharing (R7). The QuestionnaireModule displays questionnaires defined in a simple document format, saves results and provides live feedback to the OC (R8). All non-mediastream communication (e.g. session initiation) is centralized at the CommunicationManager. Results of the tasks and questionnaires, as well as the status of each client, are collected at the UserManager. The fine-grained steps of an experiment can be scripted. Each step can specify commands to be executed and conditions when to proceed to the next step. Also complete manual items can be included as a checklist. After the experiment the AnalyzerTool (AT) assists in analyzing media and logs. The conversation is utterance segmented with help of the Adintool[2]. Details of the turn-taking [3] of each participant can be exported for further statistical processing. AT allows synchronized playback of the media, viewing the turns and manual annotation.

IV. DISCUSSION

The VMC-TB has enabled us to successfully investigate the effect of delay on semi-structured group discussion with 5 participants. We used a decision making task, similar to the survival scenario [1] and assigned one of the participants to be the moderator. First we conducted a study with 39 participants, setting symmetric conditions for all participants. The integration of the task allowed us to reveal how the preexisting knowledge about the task was highly correlated to the objective performance while the delay had little influence. The perceived performance assessed with questionnaire, was highly correlated with the delay. The qualitative analysis of the feedback and session showed how the communication broke down with high delays and revealed that most groups developed mechanisms to cope with the delay.

The testbed is a first step towards our final goal: detailing the effect from different factors (QoS, context, roles) on the individual and overall QoE in video-mediated group communication.

REFERENCES

[1] P. 130. ITU-T RECOMMENDATION, "ITU-P.1301 - Subjective quality evaluation of audio and audiovisual multiparty telemeetings." 27-Feb-2013.

[2] Marwin Schmitt, Simon Gunkel, Pablo Cesar, and Peter Hughes, "A QoE Testbed for Socially-Aware Video-Mediated Group Communication," in *ACM MultiMedia, Socially-Aware Multimedia Workshop*, 2013.

[3] A. J. Sellen, "Remote conversations: the effects of mediating talk with technology," *Hum-Comput Interact*, vol. 10, no. 4, pp. 401–444, Dec. 1995.

[4] Jack Jansen, Pablo Cesar, and Dick Bulterman, "Multimedia Document Synchronization in a Distributed Social Context," in *Proc. of ACM DocEng*, 2013.

---

[1] www.gstreamer.org

[2] http://www.voxforge.org/home/docs/julius-manual/adintool