Enhancing Social Communication Between Groups

Tim Stevens, Doug Williams, Peter Hughes, Roland Craigie, and Ian Kegel, BT I&D, Ipswich, UK
Pablo Cesar and Jack Jansen, CWI, Amsterdam, The Netherlands
Marian Ursu and Michael Frantzis, Goldsmiths University of London, UK
Nikolaus Färber, Manfred Lutzky and Sebastian Vogel, Fraunhofer IIS, Erlangen, Germany

Abstract—This paper describes a prototype software platform that supports advanced communications services, specifically services enabling effective group-to-group communications with a social purpose, between remote homes. The architecture, the individual components, their interfaces, and the rationale for building the software platform, such as the usage of standard consumer equipment, are described. A specific novel network based capability, called orchestration, is described as are the results of initial user and technical trials and some plans for future work.

Keywords-component; Orchestration, Social TV, communication

I. INTRODUCTION

Television, as both a service and a device, is changing rapidly. And with these changes comes opportunity. Twenty years ago the television experience comprised a few channels of analogue broadcast displayed on a relatively dumb box in the corner of the living room. Today we have a myriad of distribution possibilities; digital has usurped analogue and satellite, while cable, terrestrial, fibre and copper all compete as distribution mechanisms. And the machine is no longer a dumb box: it can be connected to other in-home devices playing DVDs, Blu-Ray discs, or accessing files from a shared hard drive. It can also stream video relayed through consumer games consoles such as the Wii, the PS3 and the X-Box and of course play pay-per-view content from a range of service providers. Internet connectivity allows new uses, such as SkypeTV (1) (2) and network-based gaming experiences from providers such as OnLive (3) as well as generic web browsing.

In this maelstrom of change, new opportunities are emerging which come under the umbrella term of “Social TV” (4) (5) (6). Television has always had a social aspect (7) (8), but the availability of an Internet connection, the rise of social networking, and the commercial success of consoles such as the Nintendo Wii (9) supporting social, fun games (designed to be played by groups in one location) together enable new ways to incorporate television into a social experience.

The work reported in this article is focused on television as a device capable of supporting social communications between several groups of people with each group in a different location. Current technology (such as the mobile phone or computers) serves the needs of individuals, who may coalesce on line to form groups or teams, but it is not well geared towards enabling a group of people in to interact with teams or groups in other locations. Interaction between groups is common (and vital) in real life and, if supported by technology, we believe that it should be possible to build attractive services that people will use.

This paper describes the network and software capabilities required to support a range of Social TV applications that are characterised by involving more than two locations, with two or more people at each location. Use cases include, but are not limited to the playing of a social game, of a communal interactional nature, much like a parlour game or board game. A screen-based application shared between locations allows participants to see and hear two or more people at each location. In order to achieve this, we use multiple cameras at each location, with intelligent algorithms for compositing the different visual elements (such as the outputs from cameras or elements of the game). The algorithms are based on film editing skills, which allow good representation of the interaction between people in the different locations. For example, one such technique might automatically choose a specific angle to best frame the person speaking in a given moment; we call this orchestration.

This paper also describes the overall technical architecture that enables this use case. It explains the functionality and performance of the main components and discusses the design choices considered, as well as elements of the overall system design. As far as is possible, the architecture tries to build upon existing device types, and existing standards and protocols.

II. MOTIVATION AND USE CASE

The motivation of our work is to support social interaction between geographically separated friends and families. The intention is to support the nurturing of strong ties through remote social activities such as playing a board game using video-mediated interaction. While the majority of the research on video-conferencing has focused on the office environment, some very recent work is starting to consider video-conference within the context of home social activity. For example, a recent study on video-mediated free play between children found that different kinds of views lead to different types of play (10).

The ideal game for this experiment should bring the experience of a family (board) game to separated households, to recreate an experience which normally relies very much on people’s interaction within one room.

978-1-4244-7445-5/10/$26.00 ©2010 IEEE
The game chosen for our experiments is based on the board game Space Alert\(^1\) which has been adapted to be played using a touch screen.

In Space Alert, all players are members of one spaceship who operate in constant risk of attack by hostile enemies. The ship’s crew has to cooperate to be successful in defending the spaceship and in attacking the invaders. The players plan their actions together and then let their actions run their course in the ensuing twelve rounds. Communication is crucial to the success of the mission since all players take on different roles and have different resources. The whole crew must work together to succeed. The game presents an animated layout of the spaceship, its status and instructions for the game using an implementation in Adobe Flash. The application generates instructions for the other components in the system, such as the lobby (which is also implemented in Adobe Flash).

The lobby is the user-interface component that forms the initial point of interaction for users of the system. Displayed on the touch screen on the table, the lobby allows users to logon, and to see who is active on the system, and where they are located. It also allows a user to contact another user to initiate a (video and audio), or to send a game request. When a user starts a game, the game logic is able to dictate the behaviour of the system: in the case of Space Alert, this includes changing the background image which surrounds the video image so that the users can see that the game has started, and switching the view on the touch screen from lobby to game (which restricts interaction to those involved in the game). The lobby is also implemented in Adobe Flash.

III. SYSTEM DESCRIPTION

The technical architecture is shown in Figure 1. There are two logical sets of components: Communications and Gameplay, the latter being tailored to specific scenarios. The focus of our development activity has been on communications, but the gameplay has been instrumental in defining some of the capabilities that the communications system should support. The components in this architecture are described in the following sections.

The architecture aims to work, as far as possible, with standard items of consumer equipment. Where this is not possible, research prototypes have been developed. For example, in the case of the video components, existing cameras and commercial capture cards have been used, and mature software sub-components such as the Ambulant SMIL player (11) and the x264 video encoder (12) have been incorporated.

In general, components have been designed to be independent from each other. Messages are passed between components using standardised, cross-platform protocols. Abstraction elements have been developed in some parts of the system, in order to allow messages to be transformed, or to simulate the behaviour of components which are not required in this release. An ultimate goal is to see elements of this system deployed in a domestic environment, where large numbers of powerful

---

\(^1\) Space Alert was awarded a special prize in the 2009 Spiel Des Jahres (Game of the Year) competition.
computers are unlikely to be available. Abstracting functionality will allow some elements to be refined and perhaps ported to domestic equipment such as a home hub or a television or a games machine, without adversely affecting other parts of the system.

A. Home Setting

In order to validate the concepts underpinning this work we have equipped a number of laboratory spaces as living rooms. Two pairs of rooms exist, one in Antwerp (Belgium) and the other in Martlesham Heath (UK), with functionally identical rooms at each site. These rooms are set up as shown in Figure 2. Each contains a widescreen LCD television and 3.1 sound system (two stereo speakers plus a centre speaker), with a powerful PC for video processing whose output is displayed on the television screen.

Additionally, each room has a table upon which is an LCD touch-screen display, driven by a notebook computer. An array of four small microphones is located in the table. The use of four microphones, together with signal processing, enables the captured sound to be spatialised at the remote end using the 3.1 sound system. There are also three HD video cameras; two of which are self-contained and provide side views of the seating area. The third (centre) camera has a feed that is captured, cropped and streamed by the visual composition engine. Both laboratory setups include a control room space which holds the additional computers and network components required to operate the system. These two living rooms and the control room are interconnected using a dedicated gigabit LAN.

B. Components

The components in the architecture (see Figure 1) include:
- the game/lobby as explained in the previous section,
- the Visual Composition Engine (VCE), the videoconferencing and composition engine,
- the Audio Communication Engine (ACE), which digitizes, transmits and renders 3D audio,
- Analysis, which receives a video and 3D audio feed, identifying human beings by examining the video data, locating them in the room by spatial analysis of the audio data in conjunction with the video data,
- Orchestrate, which takes cues from Analysis and instructions from the Space Alert application and determines the video shots to be presented to the participants, and
- the Communication Manager (CM), which handles the routing of messages (local and remote) and overall control.

C. Visual Communication Engine

There is one Visual Communication Engine (VCE) per room. The VCE is a custom component which provides two-way multi-camera video communication with the lowest possible latency. The VCE switches seamlessly between one of five possible camera views, following commands provided by the orchestrator, as described later.

The hardware is a quad-core PC running Microsoft Windows XP, with a BlackMagic Decklink HD Extreme video capture card. The card allows video from a High Definition video camera to be captured at full resolution at 25 frames/second, which would not be possible using a conventional USB 2.0 interface owing to the high data rate required.

The VCE software is based around a modified version of the Ambulant player (11). The custom additions allow the player to simultaneously receive three remote video feeds from the other room, whilst simultaneously capturing images from a local HD camera, transforming and streaming them. The VCE can also composite the video feeds with graphic and text regions on the TV screen, plus audio files, allowing the video to be mixed with visual and audio effects.

The VCE receives its commands as SMIL fragments which specify the visual elements and dictate how and when they should be displayed. The centre HD video camera captures images at 1920 x 1280 pixels, and the VCE uses the Microsoft DirectShow API to capture them from the BlackMagic card. SMIL allows the input video to be used
either as a scaled 1280 x 720 source (full frame, but smaller), or for one of two 1280 x 720 SD crops to be selected from the original. These options permit three virtual cameras to be formed from one physical camera, with any one available at a time. Once acquired, the images are encoded into an H.264 stream using a modified version of the open source x264 encoder (12), carried over User Datagram Protocol (UDP) and using Real time Transport Protocol (RTP) for streaming and Real Time Streaming Protocol (RTSP) for port negotiation. The VCE also provides a scaled-down capture of the main HD camera to the Analysis components using Transmission Control Protocol (TCP). The VCE discards alternate frames, plus 75% of the remaining pixels for this feed; the remaining smaller images provide sufficient detail for face recognition, without overloading the Analysis machine with data.

Simultaneously, the VCE receives three H.264 streams from the other room. One of these streams is the 1280 x 720 output from the other VCE, and the remaining two are the feeds from the two physical side cameras, transmitting at 1280 x 720. These streams are simultaneously decoded using the H.264 decoder, with one of the three being rendered onto the screen at any time. The VCE is also capable of loopback video mode, to allow viewers in the room to see themselves prior to making a video call.

D. Audio Communication Engine.

There is one Audio Communication Engine (ACE) per room. This is a set of software components developed by Fraunhofer IIS, running under Apple’s OS X on a Mac Mini (implementations are also available for Microsoft Windows and Linux) (13). The ACE components can be classified in three categories: an acoustic interface, audio codec, and network transport.

The first element of the ACE is the acoustic interface. This covers the physical setup of microphones and loudspeakers, plus the additional signal processing between these and the audio codec. The acoustic interface processes the microphone signals using Directional Audio Coding (DiaAC) (14). The current release of the ACE uses a 4-channel microphone array, which is processed into 3 audio channels (left, centre, right) for encoding, transmission, and reproduction. Additionally, the acoustic interface performs echo control (EC) which allows people to talk naturally to others in the partner room, with the system retaining the spatial relationship between the various voices in the rooms, and at the same time minimising background room noise and removing echo from both ends.

The audio codec forms the core of the audio transmission chain, and implements the AAC Enhanced Low Delay (AAC-ELD) audio coding algorithm, standardized in MPEG (15). The AAC-ELD codec supports the full audio bandwidth (44.1 kHz sampling rate) and is typically operated at a bit rate of 64 kbit/s per channel. For the three-channel configuration described above, this results in an audio bit rate of 192 kbit/s. The algorithmic delay of AAC-ELD is 15-35 ms depending on the exact configuration of coding tools and sampling rates (13).

The audio stream is transported to the peer ACE using RTP over UDP/IP, where the stream is decoded and fed to the room’s loudspeakers. In order to minimize delay despite delay variations on the IP-network, a jitter buffer management and adaptive play out algorithm is employed at the receiver. This algorithm minimizes buffering-time while reducing late-loss to an acceptable level. As a result of this low-delay streaming component, the end-to-end delay (“from mouth to ear”) is kept as low as 50-80ms. However, since the VCE introduces approximately 200 ms delay (16), the ACE artificially delays the audio in order to synchronize with the video signal.

An additional audio feed is also sent to Analysis, where the speech activity is analysed in conjunction with the images from the VCE. This audio feed carries uncompressed PCM samples for all four microphone signals and is echo controlled by the ACE. Hence as long as the echo control operates correctly, the Analysis will only “hear” the local audio signals.

E. Analysis

There is one Analysis computer per room. The Linux-based Analysis software components receive video images from the centre camera via the VCE, and an audio feed from the ACE. Analysis identifies human faces and tracks their movement and orientation in the video streams, and also uses the spatial relationship information in the audio data from the ACEs to produce output messages which indicate people’s position and show if they are speaking. Analysis is implemented as three discrete processes on each machine (17). The first of these is a Video Cue Detection Engine which identifies faces, tracks them (allowing for people entering and leaving the scene), and a gesture detector. An Audio Cue Detection Engine receives signals from the ACE and performs various signal processing tasks including voice-activity detection, noise reduction and word detection. The final section is the Unified Cue Detection Engine, which synchronizes the outputs from the other two detectors, and provides higher-level reasoning, such as identifying specific keywords in the voice activity, and matching specific people to the “people change” events from the Video Cue Detection Engine.

F. Orchestrator

The Orchestrator (18) is a Java process running under Apache Tomcat on Microsoft Windows. The Orchestrator receives inputs from other components, in particular from Analysis, and attempts to make “editorial” predictions to control the camera views. For example, if a person was identified as talking to another person in the same room, the Orchestrator might instruct the VCE to frame a close-up shot on the first person for a few seconds, then cut to a wider shot for a reaction, then back to the talker. To change a camera view, the Orchestrator may have to switch both VCEs: the sending VCE may be required to change its
centre view, whilst the receiving VCE may need to switch input from a side camera to the centre camera. In the simpler case where the receiving VCE is switching to a side camera, in the other room, then the sending VCE does not need to be switched.

The Orchestrator receives as inputs the voice activity and person-identification cues from Analysis, state information from the game and system mode information from the CM. The Orchestrator’s reasoning engine applies logic from an interaction ontology, to identify whether, for example, there is an ongoing dialogue or a single-person monologue. The output from the Orchestrator is a set of cut/crop camera instructions, which are passed as SMIL fragments via the CM to the appropriate VCE using XML-RPC.

G. Presence Server

The Presence Server uses the Extensible Messaging and Presence Protocol (XMPP) to track who is logged on to the system, and to identify whether they are currently chatting, playing a game or otherwise available. The Presence Server will become more important in subsequent versions of the system which will have more than two end points.

H. Communication Manager

This consists of two components: the Local Controller (CM-LC), which is responsible for coordinating the operation of the other platform components and also acts as a communications hub, and the Session Controller (CM-SC), which acts as an interface for the XMPP communications with the Presence Server. The CM-LC is the largest component in the CM. At platform start-up it ensures that all platform components are present and properly synchronised, then initialises the VCE and ACE, registers its presence with the Orchestrator and then, following user log-on, puts the platform into a self-view configuration so that users can identify themselves and log onto the system. It then starts up applications under user control via the Lobby.

The protocol employed for all communications between TA2 components is XML-RPC. This is non-proprietary, relatively lightweight, and cross platform. All control communication passes through the CM-LC, allowing the CM-LC to provide an abstraction interface, appending information which may be required by certain components, or masking changes in components. In its role as a central controller, the CM-LC may be able to instruct components to alter their behaviour, for example informing a VCE that a specific camera is unavailable.

The Gameplay components have been customised for Space Alert, allowing features specific to that game to be accommodated. For example, Space Alert includes a “communication failure” scenario, in which players must work together to restore voice communications. To achieve this, the game engine interacts with the VCE and ACE to cut voice communications at certain points in the game. Further game-specific events can also be incorporated.

IV. EXPERIMENTAL FINDINGS

Our long-term goal is to determine whether orchestrated communication is better than conventional video-conferencing consisting of static cameras connected directly to TV screens. We are investigating this through (a) studio experiments in which orchestration is performed semi-automatically, and (b) experiments carried out on the TA2 platform where orchestration is fully automatic. Early results are encouraging and are summarised below while also being reported more extensively elsewhere (18).

The setup described above has been fully assembled and tested. The overall video delay (event to remote screen) has been measured at 365ms (16) across a LAN between two rooms, only slightly higher than equivalent delays measured between commercial video conferencing systems. Complex screen composition and seamless switching between video sources in the VCE has been achieved. At the mechanistic end, the whole chain (capture, analysis, orchestration and composition) works adequately. At times it has been necessary for human intervention to correct the automatically-extracted cues. Appropriate shots have been chosen for mediating communication patterns such as dialogues and monologues.

Initial observations have been undertaken of four children, two in each of two locations, playing Space Alert, with orchestration switched off. The children were invited to play the game over three sessions, two before lunch, and one after. During the first session the game logic was explained to the children. In the second two sessions they were left to play on their own. During the third session their game was recorded from one end.

At this stage we sought to understand whether game play requiring communication “through the screen” was possible; as much as anything we wanted to observe the dynamic between the players to ensure that they were indeed operating as a single team and were not operating as two isolated teams.

Reviewing the recordings of the games it is clear that the players were all engaged in the game and eager to communicate through the screen in order to complete the task. As described previously, communications are occasionally interrupted as part of the game-play. When this happened the body language was very revealing. With communication enabled, players were on the edge of their seat, peering intently into the screen and communicating with the remote party. When communications were interrupted, players’ shoulders sagged, they sat back in their seats and started local two-way communication with the player in their room. During this time the players remained “on task”, discussing strategies to win the game, but were clearly disengaged from people in the remote room. As soon as communications were re-established, they again sought to discuss strategies through the screen. These initial tests
show that the technical set-up can support the complex video composition and transmission, and support the basic premise that shared activities requiring communication through the screen may be engaging. However further work is required.

V. FUTURE WORK

Three lines of work are planned: Orchestration tests (18), lab-based user trials, and a “lite” implementation for long term user evaluation.

To test whether intelligent sequences of camera views enhance immersion in and engagement with the narrative portrayed on the screen (as continuity editing does in cinema), comparative tests are planned in which users engaged in some shared activity are observed both with and without orchestration. Three other orchestration experiments are planned. The first (already completed) were designed to prove that the overall system communication between Analysis, Orchestrator and VCE can control the video composition. To test this, the system has been built and Analysis has been allowed to run, providing cues (no matter how spurious) to the Orchestrator which then switches the camera views based on the received cues and its internal reasoning. A second set of experiments will observe the interaction between groups of users, with and without orchestration, in which the orchestration is controlled manually based on human interpretation of cues. These will test whether the subjective experience of the interaction improves when excellent cues are being provided to the Orchestrator.. A third set of experiments test the analysis performance against the ground truth derived from the second set of experiments. This will confirm how closely a computer-based system can match a human interpretation of the cues.

User trials are also planned, including lab trials designed to explore more innovative capabilities of the system. These will be performed on specific functionality and will provide sufficient data for understanding how users perceive the system, what they like and do not like, and what other functionality they are expecting. The trials will include experiments carried out over the Internet, designed to confirm that system properties are robust enough to cope with the additional and variable delays that operation over the Internet will introduce.

Long-term field trials are also planned, deploying a simplified or “lite” version of the system in selected households. The final goal is to provide results to validate our assumption that videoconferencing combined with social activities supports the nurturing of strong ties between groups of friends and families living apart.

VI. CONCLUSION

This paper reports on a prototype software framework that integrates the capabilities of a range of devices found in today’s homes, in order to allow groups of people in different locations to share a social experience in which participants can see and hear each other, and within which communications is central, important and natural. Early results are encouraging; they have shown that the system allows users to become fully engaged with complex game play relying on verbal communication through a screen, and that the end-to-end architecture is viable. Further experiments are planned to validate some of the more ambitious capabilities of the system such as the Orchestrator, which instructs the system, (based on inputs from analysis and the game play), to deliver selected camera views and sequences of camera views to better represent the communication. User-testing will provide results on the desirability of the functionality provided by the system.

REFERENCES


