synchronous MediaSharing: Social and Communal Media Consumption for Geographically Dispersed Users

Maarten Wijnants Jeroen Dierckx Peter Quax Wim Lamotte
Hasselt University – tUL – IBBT – Expertise Centre for Digital Media
Wetenschapspark 2, 3590 Diepenbeek, Belgium
{firstname.lastname}@uhasselt.be

ABSTRACT

Real-time social interaction possibilities are increasingly disappearing from the media consumption process. A decisive contributing factor to this tendency is growing user location disparity. This paper proposes *synchronous MediaSharing* (sMS), a distributed, purely web-based framework consisting of APIs plus a back-end that enables geographically separated persons to socially consume multimedia content in a synchronized fashion. The sMS service currently resides in a proof-of-concept stage, awaiting qualitative evaluation by means of user experience research methods, yet its underlying network infrastructure as well as the majority of its principal functional components have already been designed, implemented and evaluated. A chief innovation of the system is that its exclusive reliance on open web standards warrants cross-platform support and unlocks seamless content synchronization across the physical and virtual worlds. We outline our vision for the sMS service, motivate the need for such a system, discuss its current implementation, present tentative practical results that confirm the feasibility and validity of our design, and overview the sMS feature roadmap. The sMS functionality has apparent utility in education, training and professional settings, but also in the field of entertainment and the recreational market.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; H.m [**Information Systems**]: Miscellaneous

General Terms

Design, Experimentation

Keywords

Synchronized media consumption, content sharing, social interaction, user location disparity, IDMS, web technology

1. INTRODUCTION

Certain traditional forms of media consumption entail an important social and communal aspect. For instance, friends or relatives tend to convene at a single location to watch a particular TV show in the presence of each other. As another example, people returning from a holiday and wanting to share their experiences often do so by showing photographs to invited friends. Both cases illustrate situations in which multiple persons are physically co-located and in addition are concurrently consuming identical content. Such scenarios not only scaffold social interaction between people, they even are likely to act as a catalyst for it [4].

Technological advancements as well as evolving user habits are however causing fundamental changes, both in the media provision landscape and in the way people access and consume content. On the one hand, new dissemination methods such as video-on-demand (VOD) and online media repositories are rapidly becoming legitimate alternatives for traditional broadcast solutions. These methods allow users to consume exactly the content in which they are interested. On the other hand, innovations such as Personal Video Recorders (PVRs) and mobility features allow consumers to access content at their own convenience, both in terms of time and place.

Although these evolutions definitely have their merits for the media consumer, they also introduce issues in the field of user interactivity. In particular, many of them intrinsically individualize the media consumption process and transform it into a much more isolated activity. This increased isolation manifests itself at two separate levels. First of all, temporal disparity emerges as it is no longer guaranteed that users consume content simultaneously. For example, technologies such as PVRs and VOD have removed the need for viewers to adhere to the broadcast schedule imposed by the TV station. The second level of individualism relates to location: multimedia data is nowadays ubiquitously accessible via a plethora of devices and platforms, some of which are explicitly intended for single-person use (e.g., smartphones and tablet PCs). The negative impact of these trends on social interaction and community-centric media consumption is apparent.

This article introduces the web-based *synchronous MediaSharing* (sMS) framework, a combination of a collection of APIs and a back-end, that aims to emulate the feeling of concurrently consuming media content with co-located users for geographically distributed persons. Stated differently, the sMS project attempts to bridge the social interaction divide that is caused by media consumers' spatial disparity. Prominent already implemented features include multi-device and cross-platform support, the ability to incorporate the service in (3D) virtual environments, the option of seam-

MMSys'12 February 22-24, 2012, Chapel Hill, North Carolina, USA

lessly transcending the boundaries of cyberspace and the physical world with regard to communal multimedia content consumption, support for heterogeneous media content types, synchronous (i.e., real-time) inter-participant communication facilities and integration with social networking websites. Envisioned future properties entail elaborate utilities for presence and user status information, and "second screen" exploitation in case the sMS service is visualized on a TV display. Temporal asynchrony, the second origin of communal interaction decline in media consumption patterns, is at the moment deliberately not tackled by the service because it represents a radically different research topic.

The outline for the remainder of this work-in-progress report is the following. Section 2 reviews related work. Section 3 describes the design and current implementational status of the sMS service. A proof-of-concept demonstrator and tentative practical results are presented in section 4, while planned extensions and future improvements are enumerated in section 5. Finally, section 6 draws our conclusions.

2. RELATED WORK

2.1 Inter-Destination Media Synchronization

To achieve a sense of shared and synchronous media consumption, the sMS service requires some form of Inter-Destination Media Synchronization (IDMS, sometimes also referred to as "group synchronization"). A comprehensive body of IDMS research is available in the literature. Boronat et al. present a comparative study and taxonomy of the subject in [1]; it appears that the majority of proposed IDMS systems is based on the "synchronization maestro" scheme, in which a (central) management entity accumulates output timings from the involved destinations and subsequently distributes control information to synchronize media playback at each client. Tasaka et al. examine in [11] the impact of handover on inter-destination synchronization accuracy in integrated wired/wireless environments. ETSI TISPAN has standardized an IDMS solution as part of its IPTV architectures and protocols specifications [10].

Although IDMS functionality is crucial for the correct operation of the sMS framework, it is not a research topic we will be actively pursuing. This implies that we will rather adopt an existing IDMS scheme than personally advance the state of the art in this research field by engineering a novel approach ourselves.

2.2 Social TV Research

Social TV viewing research is concerned with fostering communication and social interactivity between viewers of (broadcast) TV content. A number of efforts in this domain focus on resolving the time asynchrony issue (i.e., users watching the same TV show at divergent times). Example systems in this category include neXtream [5] and CollaboraTV [6]. Other systems are primarily concerned with addressing the issues introduced by geographical dispersion (i.e., concurrent TV viewers that are separated by location). Notable examples here are AmigoTV [2] and ConnecTV [10, 12], a social TV system that complies with the previously mentioned ETSI TISPAN IDMS standard. Social TV research is additionally interspersed with argumentations for the inclusion of social interaction assets in media consumption environments, which also represents a chief aspiration of the sMS service.

2.3 Commercial Web-Based Media Services

A number of web-based frameworks that encompass synchronous media sharing possibilities are commercially available. Examples include Rounds (`http://www.rounds.com/`), an interactive video chat tool with media sharing options, and Cisco's WebEx platform (`http://www.webex.com/`). Both exhibit significant conceptual similarities with the sMS service. Rounds as well as WebEx are however centered around the desktop sharing metaphor and as such implicitly inherit real-time media sharing opportunities. In contrast, the sMS framework has from the very outset been designed for synchronous content consumption purposes and consequently treats media objects (instead of the entire desktop) as its atomic synchronization units. The explicit emphasis on individual content items yields a more fine-grained approach and therefore is expected to unlock extensive and powerful functionality that exceeds the technical scope of desktop sharing software. In addition, Rounds capitalizes exclusively on the recreational market, while WebEx is principally intended to facilitate the hosting of meetings for business customers. The sMS system on the other hand does not limit itself to a specific target audience and hence seems more versatile and universally applicable.

There also exist extensions for some popular applications that enable synchronous media consumption. For instance, the synchtube service (`http://www.synchtube.com/`) enables synchronized playback of YouTube clips, whereas the Photo Swap feature of Microsoft Live Messenger allows users to share photographs. Such solutions are however directed exclusively at one particular application and are furthermore fairly modest in terms of provided functionality (e.g., only a single type of media content is typically supported). In contrast, the sMS service is completely application-agnostic and exhibits a richer feature set.

2.4 Non-Web-Based Media Synchronization

A powerful property of the sMS service is its ability to be embedded in arbitrary non-web-based applications such as collaborative digital spaces or so-called Networked Virtual Environments (NVEs). Related work in this context includes the previously mentioned CollaboraTV project, which encompasses an avatar-based 3D "virtual audience" user interface [6], but limits this to a fixed synthetic setting, namely a movie theater-like environment. The sMS framework is not bound to similar restrictions. As another example, the so-called Shared Media update fairly recently introduced the option to appropriately display and interact with webpages in Second Life (`http://secondlife.com/`); inter-participant synchronization is however constrained to the URL of the website and hence disregards dynamic content as well as the effects of user interactions (e.g., scrolling). In contrast, the sMS service is able to achieve complete and accurate media synchronization in digital environments.

2.5 Innovations & Contributions

It is apparent that the sMS idea shares interfaces and overlaps with previous as well as ongoing research efforts, and also with commercial products. It however differentiates itself from these related systems by means of its unique mix of (envisioned) properties. For instance, the sMS service has been designed with heterogeneous media types in mind. This is unlike typical social TV installations, which exclusively consider TV content, or, for example, the Sto-

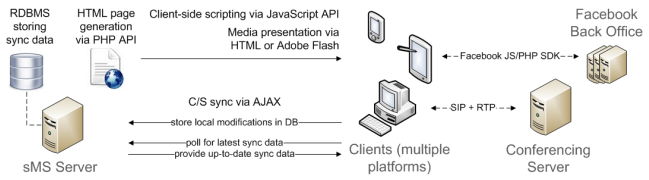


Figure 1: System design and mode of operation

ryVisit system [7], which is restricted to the synchronization of a limited number of predefined electronic books. Another distinctive characteristic of the sMS framework is its level of applicability: not only web-based services and applications, but also distributed desktop software like NVEs can leverage the sMS functionality. As such, it becomes feasible to seamlessly and transparently bridge the real and digital world in terms of synchronous multimedia sharing. Comparable provisions are lacking in the vast majority of related systems, as the focus is either on providing a strictly web-based solution (e.g., Rounds, WebEx, neXtream) or on offering dedicated media sharing implementations for individual applications (e.g., Second Life).

3. SYSTEM DESIGN

A schematic overview of the sMS system's current architectural setup and the technologies utilized by its composing functional entities, is provided in figure 1. As can be seen, the sMS service adopts a client/server network communication model. The sMS server forms the heart of the platform, as it hosts a relational database which maintains the media synchronization data. The clients are simple end-user devices and are primarily responsible for content presentation. In the following subsections, the various aspects of the system's design and implementation will be elaborated on.

3.1 Design Objectives & Resulting Implications

A first important sMS requisite is multi-device and multi-platform support. In other words, the service is intended to be maximally accessible for end-users. This requirement is motivated by the desire to be able to cater to the plethora of devices via which end-users can nowadays access multimedia content. Particularly mobile and handheld devices are rapidly gaining popularity for this purpose, as they enable content consumption while users are commuting. The multi-device design objective advocates a standards-based solution. The sMS service has therefore been exclusively designed around open, standardized web technologies and languages that are available on most current major platforms (i.e., Microsoft Windows, Unix-like operating systems, Apple's Mac OS, Google's Android platform, and so on). As such, it is guaranteed that users on a multitude of platforms can engage in sMS sessions via a basic web browser.

Heterogeneity is also strived for in terms of multimedia content. The sMS service intentionally does not restrict itself to one particular type of media data; instead, it aims to enable the sharing of diverse content types. A straightforward candidate in terms of content is a collection of pictures. Conceptually, such a sMS session represents the digital analogy of browsing through and discussing a (physical) photo album with a number of (co-located) friends. It is however equally possible to synchronize sets of continuous media via the sMS service. At the moment, only video clip sharing has been implemented and validated in this content category, but an extension to audio fragments seems trivial. The single requirement for the media input is that it must be syndicated and specified by means of a Media RSS feed [8]. For each individual content item (i.e., each picture or video clip), the feed must incorporate an entry specifying at least the item's Globally Unique IDentifier (GUID) and the source URL from which it can be retrieved.

A final pivotal design aspiration for the sMS framework is the option of cross-platform media sharing. In particular, it should be feasible for users on different devices and platforms to jointly engage in sMS sessions. This requirement is again automatically fulfilled by opting for a standards-compliant solution.

3.2 Underlying Technologies & Content Sync

A PHP scripting API is provided for the assembly of the HTML page via which the sMS functionality is made available to the end-user. Rendering and presentation of the media content is taken care of via either standard HTML or Adobe Flash (see section 3.4). Finally, a JavaScript API facilitates any necessary sMS-related scripting at client side.

Client/server synchronization occurs by means of two-way AJAX communication. First, in case a user actively influences the state of the sMS session, his client initiates an AJAX request to update the corresponding synchronization data in the database. Secondly, clients periodically solicit the most recent synchronization data that pertains to their current session from the server. Once provided, the synchronization is performed locally. In a picture sharing session, this will lead to the presently selected image being displayed at all participating sites as users advance through the input set. In a video sharing setup on the other hand, the additional temporal aspect of this content type will also be synchronized. Stated more elaborately, the synchronization will not be confined to the current video clip, but will also encompass the playback time of the currently showing clip.

It is evident that the adopted AJAX approach yields a straightforward, relatively loose synchronization. As has already been mentioned, achieving strict IDMS is not the focus of this project. The present synchronization mechanism is expected to be satisfactory for various types of applications and deployments (e.g., digital entertainment for residential users). For application domains and communities that pose tighter consistency requirements for the shared media, the current implementation may be insufficient. This implies that the sMS platform in its current form is probably not universally applicable. Adoption and integration of a more robust and efficient IDMS scheme would likely eliminate this deficiency. To still permit some control over the synchronization process and hence the attained level of content consistency in the current design, the time interval at which the client polls for synchronization data is configurable.

3.3 Database Design

The synchronization records that are stored in the central database encompass the following information. First of all, they comprise the ID of the sMS session (in the form of a unique session name) as well as the URL of the Media RSS feed that syndicates the content that is being shared in the context of this session. Combined, these two entries uniquely identify a single media sharing instantiation and hence act as primary key in the database. This implies that it is possible to include multiple Media RSS feeds within a single

sMS session and, conversely, to independently synchronize the same Media RSS feed in various sMS sessions. Next, the actual synchronization attributes consist of the GUID of the currently selected content item and, in the event of continuous media, the current playback time of the item as well as a flag indicating the status of the playback (i.e., playing or paused). Finally, each record also includes a timestamp value and a so-called master client ID. The latter identifies the user who lastly influenced or modified the state of the media sharing session. This user is the author of the synchronization record and is therefore assumed to currently "control" the session.

On an incoming synchronization request from a client, the sMS server fetches the correct record from the synchronization database and disseminates it. The received data is subsequently enforced locally by the client, this way achieving synchronization of the shared media content. Recall from section 3.2 that the local application of synchronization instructions might for instance entail progressing to another photograph in the shared collection or adjusting video clip playback time. The timestamp value that is included in the database record enables clients to discard duplicate or stale synchronization instructions received from the sMS server. The master client information on the other hand is exploited to periodically refresh the playback time of continuous media types in the synchronization database (see section 3.4).

3.4 Content Presentation & User Interaction

For image content, two distinct presentation methods have been included in the sMS JavaScript API; both are founded on the slideshow metaphor. The first approach is exclusively engineered by means of HTML functionality and shows images of the shared collection one at a time (see the smartphone client in figure 2(c)). By clicking on the currently presented image, the next image from the collection will be selected and visualized. As a result, content traversal is in this implementation limited to a single direction and is completely sequential. The second method leverages Cooliris (`http://www.cooliris.com/`), an Adobe Flash-based media browser. This instantiation considerably improves content traversal possibilities compared to the former approach, as Cooliris supports the presentation of a thumbnail overview of the shared picture set. Users can freely browse through this overview and can randomly click on an image to select it; the selected image is then shown nearly full-screen, with cues of both the previous and the next image in the collection alongside it (see the virtual media screen in figure 2(c)). The Cooliris-based implementation clearly outperfoms its HTML counterpart in terms of functionality and usability. The rationale for still including the latter in the sMS platform, is the lack of official Adobe Flash support on Apple's iOS devices such as the iPad and iPhone. As these devices are highly popular and hence occupy an important segment of the consumer market, it would be unfortunate to exclude them from the potential sMS userbase.

Video clip content is visualized via flowplayer (`http://flowplayer.org/`), a video player for the web again built on Adobe Flash technology (see figure 2(b)). Interaction with the content in this case occurs by means of a controlbar that enables playback pausing/resumption, bidirectional video clip navigation (i.e., users are able to skip to either the previous or the next item in the collection), and random seeking in the playback of the current item. A special flowplayer plug-in allows it to serve video content on the iOS platform by providing a fallback to Apple's native QuickTime player. Besides video clips, flowplayer also supports playback of audio files in the MP3 format, a notable advantage in case it would be decided to extend the sMS framework with support for this content type.

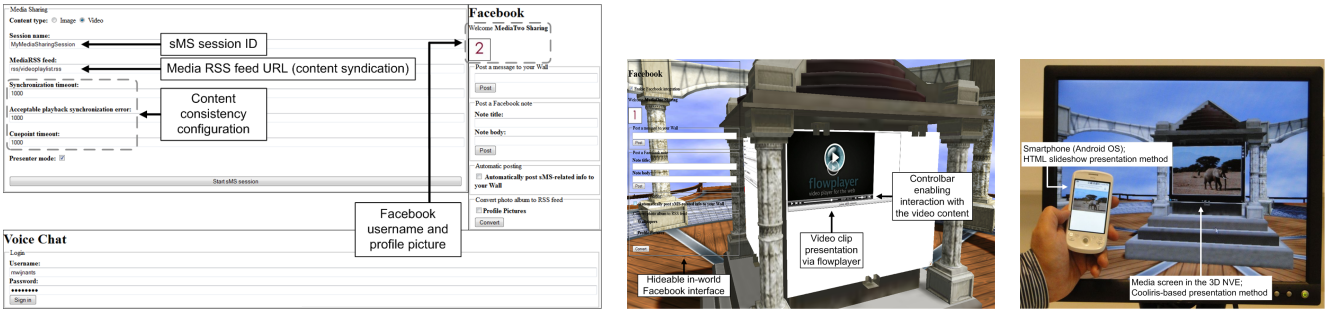
As soon as the user interacts with the shared content (i.e., by switching to a different image or video clip, or by adjusting the playback time of the current video clip), the sMS JavaScript API pushes synchronization instructions to the sMS server via AJAX as described in section 3.2. For continuous media, this basic behavior is extended with the construct that the master client of the session (see section 3.3) periodically updates the playback time of the current video clip in the database. By doing so, newcomers in ongoing media sharing sessions are able to "catch up" on the playback time and hence start playback of the current video clip at the correct location. Analogous to the synchronization data soliciting time interval, the pace of these periodic playback time updates is configurable.

3.5 Real-time Interpersonal Communication

To create a true sense of connectedness and of concurrently and socially sharing media, adequate synchronous inter-participant communication facilities might prove to be of paramount importance. The sMS architecture therefore incorporates voice conferencing features. Due to the multiplatform objective of the sMS service, an important requirement for the communication mechanism was that it needed to support a web browser as execution environment (i.e., it had to function as part of a HTML page). The communication functionality has been implemented via Androme's Intellivic SDK (`http://www.intellivic.com/`), a software development kit that offers JavaScript, C++ and .NET APIs for the realization of audiovisual conferencing products. Like the sMS service itself, the Intellivic SDK is strictly based on open standards, with a prominent role being reserved for the Session Initiation Protocol (SIP).

3.6 Social Network Interfacing

Social networking websites such as Facebook and Twitter foster social interaction by enabling people to communicate and keep in touch with their friends. As social aspects and stimuli might be indispensable factors for the adoption of a media sharing system by end-users [3], the sMS service has been interfaced with Facebook. From within the confines of the sMS environment, users are able to log in to Facebook, post messages to their Facebook news feed (often referred to as "the Facebook wall") and transform entire photo albums from their Facebook profile into Media RSS feeds (which can subsequently serve as subject in media sharing sessions). An optional feature enables sMS-related actions or events to trigger automatic posting of Facebook messages. When activated, a message will for instance automatically be published on the local user's Facebook news feed when he starts or joins a sMS session. The post mentions the name of the session, a concise description and an URL which allows Facebook friends to participate in the session via a single click. These automatically published messages are intended to incite end-user interest in the sMS service and to encourage participation in ongoing sMS sessions. In summary, the social network interfacing at the moment serves the dual purpose of content acquisition and aggregation on the one hand,



(a) Web interface (b) NVE front-end (video sharing) (c) Cross-platform sharing

Figure 2: sMS prototype

and the manual or automatic dissemination of information and events (sMS-related or otherwise) to friends in the outside world on the other. The Facebook integration has been realized via the official Facebook JavaScript and PHP SDKs (`http://developers.facebook.com/docs/sdks/`).

3.7 Synchronous Sharing in Digital Spaces

A significant advantage of implementing the sMS functionality by means of standardized web technologies is that it unlocks the option to incorporate the service in any environment that supports HTML rendering. In particular, nothing prevents the sMS service from being encapsulated in other software, and distributed synthetic environments or NVEs in particular. Such environments typically embody users via an avatar and allow them to explore a 3D virtual space. In case the NVE affords constructs for in-world web browsing, the sMS functionality automatically becomes accessible for its users. The benefit of staging sMS sessions inside a (3D) NVE is that it enables the creation of an appropriate framing and that it automatically induces a sense of shared presence and togetherness, which will in turn likely evoke a more immersive experience for users compared to a simple HTML page.

4. PROOF-OF-CONCEPT DEMONSTRATOR

A prototype has been developed to assess and validate the general methodology of the sMS service, the feasibility of its architectural design and the properties of its current features. The proof-of-concept demonstrator includes both a web interface (i.e., a HTML page that enables access via a web browser) and a 3D NVE implementation as front-ends.

The HTML website for the web interface instantiation is divided into three logical panes that are respectively devoted to sMS session control and content presentation, Facebook interfacing, and voice communication configuration (see figure 2(a)). Via the session control and content presentation pane, users can configure, start/join and leave media sharing sessions. After a user starts or joins a session, the configuration options disappear and this portion of the webpage is used for content presentation and interaction purposes (i.e., to visualize the shared media content that forms the topic of the session, and to enable the local user to interact with it). The Facebook pane clusters the functionality that is related to the social network interfacing (see section 3.6). The third and final pane allows users to login with their SIP account for voice communication, and also to select their desired audio I/O devices. Both the Facebook and the voice communication sections of the website are optional and can be omitted from the HTML page (e.g., to free up screen space on handheld devices with a limited form factor).

The NVE front-end offers users a 3D virtual world that is populated by a number of so-called media screens which afford in-world participation in sMS sessions (see figures 2(b) and 2(c)). Users are represented by animated avatars and can freely navigate through the digital environment. The media screens host a HTML page that logically and functionally corresponds to the session control and content presentation pane in the web interface. Facebook interfacing is possible via a (hideable) HUD-like virtual world overlay. In contrast, voice communication options do not appear in the GUI but are enforceable via a configuration file. The 3D world is rendered by means of the open-source OGRE graphics engine (`http://www.ogre3d.org/`). Via its NaviLibrary plug-in (`http://navi.agelessanime.com/`), OGRE allows users to in-world interact with websites.

Figure 2(c) displays a sMS picture sharing scenario involving both a desktop PC and smartphone. The desktop client presents the sMS functionality inside the developed 3D virtual environment setting, whereas the smartphone user participates in the session via a standard web browser. This use case demonstrates that our design achieves seamless content synchronization between physical devices and the digital world. It also makes the differences in contextual framing that is offered by both front-ends evident. In particular, the 3D NVE instantiation is expected to yield a substantially more convincing and immersive usage experience.

5. ENVISIONED FEATURES

Since the development of the sMS service is still in progress, a subset of its envisioned features and properties remains to be implemented. These aspects were nonetheless taken into consideration during the design phase and should hence nicely fit in with the current system architecture.

5.1 Social Presence Information

Since the sMS service targets real-time and synchronous interactions, it is advisable for end-users to be aware of the current status and activities of their friends [3]. Such knowledge is typically denoted by the term “social presence”. Although this topic is already partly addressed by the ability to interface with social networking websites, it might prove worthwhile to include facilities for social presence within the sMS service itself. In particular, with an internal mechanism in place, it would become possible to cover valuable, more elaborate social presence information such as the participant lists of active sessions, the media content that is being

shared in ongoing sessions, a direct and clear overview of which sessions friends are currently attending, etcetera. It is apparent that it is neither advisable nor attractive to (publicly) publish such detailed presence information on external social networks. Stated differently, the provision of internal presence facilities is expected to significantly augment our present services for presence information dissemination. An implementation based on the XMPP standard [9] is currently ongoing.

5.2 Personal Second Screen

A standard sMS session involves a number of geographically isolated participants. A special use case emerges when multiple co-located users share a single (large) display for sMS purposes. Such scenarios hold opportunities to leverage users' personal devices (e.g., smartphones, PDAs, tablet PCs, ...) as so-called second screens to supplement the primary display. The second screen could, for instance, be employed to present information regarding the active sMS session, to visualize (personalized) presence information, to initiate private communication sessions with remote session participants, and so on. An appreciable benefit of a second screen solution is that it eliminates the need to allocate precious shared screen real estate to marginal information that is unlikely to be of interest to all co-located users. As such, it becomes possible to exclusively preserve the common display for the visualization of the actual content, which in turn prevents the sMS experience of co-located users from being disrupted by the presentation of irrelevant information.

6. CONCLUSIONS & FUTURE WORK

Drastic evolutions in the media ecosystem, as well as shifting usage models and consumers' expectations are evermore transforming content consumption into an individual experience. As a result, a significant negative trend is evident in communal and co-located media consumption. This paper has reported on the ongoing development of sMS, an integrated framework which aims to afford geographically dispersed users with a digital equivalent of concurrently and synchronously consuming media content, which in reality is only possible for physically co-located consumers. The sMS platform does not necessarily innovate in terms of the objectives it sets forth nor the individual techniques it employs to achieve them. What does make the service unique, is the aggregation of these methods in a single system and the way they are combined to provide remotely located users with a sense of social and community-centric media consumption. A defining characteristic of the sMS platform is that it is completely web standards-compliant, which maximizes its utility and deployment potential. At the same time, it enables the framework to achieve cross-platform interactivity and ubiquitous content synchronization by seamlessly uniting physical devices and virtual environments, a scenario that has not yet been (thoroughly) explored by related research efforts. The outcome of an interim experimental evaluation, in which a prototype sMS implementation was employed to synchronize a collection of pictures between a desktop and a smartphone user, has been presented and demonstrates the viability of the sMS design.

There remains ample room for future work in the sMS research. Some of our short-term actions points have already been described in section 5. In parallel to these developments, the sMS system is scheduled to be (iteratively) subjected to user trials. The objective of these studies will be to accumulate both quantitative and qualitative feedback from end-users, as well as to gain insight in pertinent open questions such as "Does the incorporation of voice communication facilities actually aid in inducing an effective communal experience for remotely located participants of a sMS session?", "Does the currently achievable level of content synchronization and consistency suffice for home entertainment purposes?", and "Does the provision of elaborate social presence information increase the users' feeling of involvement and does it promote participation in the sMS system?". The trials will be conducted in the users' natural environment (i.e., in their homes as opposed to an artificial lab setting). Finally, in the longer term, we would like to explore alternative hosting and deployment strategies for the sMS system.

Acknowledgments

This research is part of the IBBT ICON $3DTV2.0$ project, funded by the Flemish government. Part of this research is also funded by the EFRD.

7. REFERENCES

[1] F. Boronat, J. Lloret, and M. García. Multimedia Group and Inter-stream Synchronization Techniques: A Comparative Study. *Information Systems*, 34(1):108–131, March 2009.

[2] T. Coppens, L. Trappeniers, M. Godon, and F. Wellesplein. AmigoTV: Towards a Social TV Experience. In *Proceedings of EuroITV 2004*, pages 1–4, Brighton, UK, April 2004.

[3] G. Harboe, C. J. Metcalf, F. Bentley, J. Tullio, N. Massey, and G. Romano. Ambient Social TV: Drawing People into a Shared Experience. In *Proceedings of CHI 2008*, pages 1–10, Florence, Italy, April 2008.

[4] J. Lull. *Inside Family Viewing: Ethnographic Research on Television's Audiences*. Routledge, 1990.

[5] R. Martin, A. L. Santos, M. Shafran, H. Holtzman, and M. J. Montpetit. neXtream: A Multi-device, Social Approach to Video Content Consumption. In *Proceedings of CCNC 2010*, pages 779–783, Las Vegas, NV, USA, January 2010.

[6] M. Nathan, C. Harrison, S. Yarosh, L. Terveen, L. Stead, and B. Amento. CollaboraTV: Making Television Viewing Social Again. In *Proceedings of uxTV 2008*, pages 85–94, Silicon Valley, CA, USA, October 2008.

[7] H. Raffle, G. Revelle, K. Mori, R. Ballagas, K. Buza, H. Horii, J. Kaye, K. Cook, N. Freed, J. Go, and M. Spasojevic. Hello, is Grandma there? Let's Read! StoryVisit: Family Video Chat and Connected E-Books. In *Proceedings of CHI 2011*, pages 1195–1204, Vancouver, BC, Canada, May 2011.

[8] RSS Advisory Board. Media RSS Specification. Online, `http://www.rssboard.org/media-rss`, December 2009.

[9] P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core. RFC 6120, IETF, March 2011.

[10] H. Stokking, M. O. van Deventer, O. Niamut, F. Walraven, and R. Mekuria. IPTV Inter-Destination Synchronization: A Network-Based Approach. In *Proceedings of ICIN 2010*, pages 1–6, Berlin, Germany, October 2010.

[11] S. Tasaka, Y. Ishibashi, and M. Hayashi. Inter-Destination Synchronization Quality in an Integrated Wired and Wireless Network with Handover. In *Proceedings of Globecom 2002*, pages 1560–1565, Taipei, Taiwan, November 2002.

[12] M. O. van Deventer, H. M. Stokking, O. A. Niamut, F. A. Walraven, and V. B. Klos. Advanced Interactive Television Services Require Content Synchronization. In *Proceedings of IWSSIP 2008*, pages 109–112, Bratislava, Slovak Republic, June 2008.