Social & Interactive TV: An Outside-In Approach

Venu Vasudevan & Jehan Wickramasuriya
Applied Research
Motorola Mobility
600 N. U.S. Highway 45, Libertyville, IL
{venu.vasudevan, jehan}@motorola.com

ABSTRACT

Social and interactive TV has been held back by both technical and ecosystem limitations over the past two decades. Technical approaches to enabling Social TV have relied on proprietary TV middleware, whose success in turn depends on widespread and expensive technology upgrades by cable and satellite TV operators. Additionally, users have increasingly adopted to socialize on the web, even where it pertains to TV [2]. This makes any attempts at creating TV specific social communities unattractive to users, who already have a social presence (and identity) on the web. This paper proposes an alternate *outside-in* overlay architecture for Social TV. Such an architecture leverages the emergence of a second screen with strong and built-in interactivity, and an approach to reuse rather than replace web social software to enable Social TV. In this paper, we briefly describe the motivations, architectural elements and implementation experience around such a system.

Categories and Subject Descriptors

H.3.4 [**Systems & Software**]: Current Awareness Systems

General Terms

Management, Design, Experimentation, Languages

Keywords

Social networking, interactive TV, ambient interfaces

1. INTRODUCTION

The growth of interactive and social TV over the last two decades has been a story of fits and starts. On the interactivity front, there have been a number of intriguing research and platform experiments [1] without true commercial success. Commercially successful TV interactivity has been inhibited by the inability of the combination of 9-button remotes (the lowest common denominator for input-output) and TV middleware to support truly compelling interactivity on a shared device. Chicken-and-egg issues around community creation have inhibited commercially successful social TV. Solutions where Social TV communities have been enabled as separate from web communities have not resonated with end users because issues around the effort involved and the value received. On the effort side solutions that require users to subscribe to yet another community, with yet another identity discourage participation and come in the way of a frictionless and largely lean-back experience that TV is and should be. Assuming discounting the effort, the value derived from socialization is a direct function of the size of the community and the extent to which a user's social graph is already subscribed to that community. Given the existence of large web-based social networks (e.g. Facebook & Twitter), the absolute and relative lack of scale of a brand new social TV community has not proven attractive enough to end users to garner large scale participation. More recently, a wave of TV 'check-in' applications (e.g. Philo, Miso, IntoNow [8]) have attempted to bring social interactivity to TV in the same way that companies like FourSquare have popularized location sharing. The general problem is that each is a vertical (application) that require a sizeable community to really be useful. Others, like GetGlue have attempted to build a platform and go beyond check-in to include social recommendations with some success.

In short, TV architectures of the past decade have been built on an *inside-out* architecture, one that creates specialized industry specific middleware for interactivity and socialization as it pertains to TV. This means that application developers who wish to cater to the Social TV market have to adopt industry specific standards & build application software specifically tailored to TV (both the device and the content). This inside-out architecture has failed to attract the armies of application developers and the collective creativity that has enabled other devices (notably mobile) and other media (notably web video) to benefit from vigorous in-market experimentation. Other work has done a more comprehensive survey of efforts in TV interactivity [10], in this short paper we focus on our proposed solution.

This paper proposes an alternate *outside-in* architecture that minimizes the impedance mismatch between interactive TV and the web, and maximizes the reuse of 'non-TV' content and applications to enable interactive and social TV experiences. Key components of an outside-in interactive and social TV architecture are as follows:

- Portable devices (e.g. tablets and smart phones) are now so numerous and pervasive, and are almost omnipresent on the couch and in the vicinity of users as they watch TV [3-5]. When on the 'couch' these devices are used less in their traditional roles as communication or enterprise devices, and more for media multiplexing while watching TV [6]. We refer to this class of devices (or devices adopting this role on the 'couch top') as *companion devices*, in that they are companions to the main TV screen in terms of engaging the user with simultaneous but complementary content. Given the rich capabilities and rapid evolution of companion devices, the interactive TV platform is migrating entirely to the second screen as a layer on top of software platforms (such as Google's Android and Apple's iOS) that are already familiar and attractive to a large population of software developers.

- Given emerging trends around the 'appification' of interactivity [7], the component model for TV interactivity is inherently application focused, Thus *TV apps* can be authored and published to standard application marketplaces (e.g the Android marketplace, Apple iOS App Store, Amazon AppStore etc.) like any other application. To an application developer, there is no extra technological learning curve in creating a TV app, and the added bonus of utilizing a software platform that is robust, well supported and possibly familiar to them.
- The TV specific framework is almost transparent to the application developer, and entirely located on the companion device (therefore not requiring expensive and time consuming upgrades to set-tops or cable back office). The main goal of such a *companion device framework* is to reduce or eliminate user effort in discovering, selecting and invoking interactive and social capabilities in a manner that is matched to the content and user. The value added by the companion device framework is in three areas: *adaptive synchronization*, *flexible personalization* and *seamless integration*. Adaptive synchronization enables the automated and flexible means for synchronizing content on the TV screen and the companion device at various levels of fidelity (e.g. program, program segment, scene) appropriate to the genre. Flexible personalization enables customizable experiences that mix branded and user generated content in flexible ways. And seamless integration enables easy context sharing between (and across) component 3rd party applications and the companion device framework.

In the rest of the paper, we briefly describe the technical components of an outside-in architecture (referred to as the Companion Device Framework or CDF). We focus our efforts around the Android platform and application ecosystem, and describe an application container for TV-related activity that helps you discover and interact and coordinate second-screen experiences (in the form of apps) with what is being consumed on the primary screen. We then discuss the application orchestration framework and model for driving these experiences on the client, and conclude with potential futures

2. ARCHITECTURE & REALIZATION

The Companion Device Framework (CDF) encompasses a variety of network and client services to enable dual-screen TV/video experiences across a shared video screen (primary) and a personalized supplementary interactive screen (secondary). In the base case, the primary screen is a TV, and the supplementary feed is directed to a user's personal mobile device – which may be a mobile phone or tablet. Advanced use cases include video feeds to public displays, time and place shifted supplementary feeds, and sophisticated ways of synchronizing user engagement with feeds (supplementary and base video) into a unified analytics around user engagement. The client-side framework is currently Android-specific. However, the concept and protocols may be applicable to other platforms subject to the inherent limitations they may impose (e.g. iOS).

The base user experience requires viewing context information, notably the channel identifier and EPG program metadata – related to the primary video feed to be pushed to the companion device in order to drive appropriate second-screen

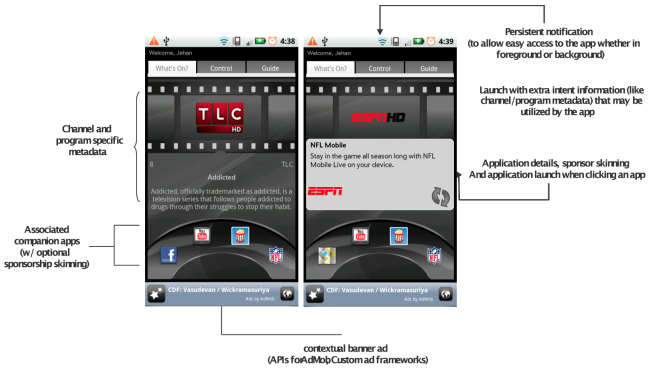

Figure 1: Companion Dashboard Application (Android)

companion experiences. The framework is agnostic to where this information is obtained; we have integrated it both with traditional back office solutions (where the device gets notifications whenever the user navigates through content) as well as custom solutions that support Internet video sources such as Hulu & YouTube. However, as we discuss in Section 3 this metadata can be obtained through a variety of audio/visual methods that analyze the primary screen content.

CDF encompasses a number of components that are utilized by the client-side components and integral to the experiences that are built using CDF. The main platform components are summarized as follows:

- **Companion dashboard**: Integrates application container and related client-side services to support use cases discussed previously (namely contextualized suggestions and related meta-data inherent to the TV-viewing experience). The main application screens can be seen in Figure 1 (the app also provides Remote Control & Guide functionality which is not shown), and enable a credible ambient experience as well as easy 1-click access to suggested/recommended applications. Eventually we plan to incorporate a diversity of *adaptive synchronization* mechanisms herein.
- **Model-based orchestration**: This component supports *flexible personalization* in ways that combine branded recommendations (from the content studios and distributors) with user favorites (personal or from their social network). Specifically, this component provides authoring and run-time support for media rights owners (studios/MSOs) to author controlled companion experiences associated with their content.
- **Identity management and device rendezvous**: User authentication supports both web identity (via OAuth) such as Facebook Connect as well as custom TV-specific identities that are integrated with the back-office.
- **Application Support**: Applications built to take advantage of CDF as well as third-party applications appearing in the marketplace (also includes support for web applications).
- Advertising and analytics: Leverages the personal ad space in the form of traditional banner or second screen

ads. These ads may be stand-alone or coordinated with content/ads appearing on the primary screen and personalized with viewing context.

Model-based orchestration is achieved via the Companion Model that is provisioned as a cloud service (currently hosted via Google App Engine) that providers the administrative interface to publishers and MSOs that allow specification of the rules dictating app suggestion on the client. This deterministic model facilitates contextual app recommendation via a mix of business-specified and personalized suggestions. For businesses, the model provides an interface for content publishers and MSOs to specify the mappings between applications and TV-related content (i.e. what the user is currently watching). For example, HBO may choose to promote it's own application on the Dashboard when one of its programs are being viewed on the primary screen. This can be enabled via the campaign management style interface (front-end) to the Companion Model. Additionally, users can personalize application recommendations by 'pinning' favorite applications, as well as receive recommendations from friends (social). The Dashboard interface allows access to the user recommendations by moving the arc clockwise via a gesture on the client.

Applications that appear on the Companion Dashboard can further be categorized by be a) applications inherently developed to support CDF (more on this below), b) any third-party Android application available in the marketplace and c) a web application or URL. Other than serving as a contextual application launcher, there is a benefit to launching those apps directly from the Dashboard as opposed to the traditional Android Home Screen. If supported (more on this in a moment), Companion Dashboard is able to pass metadata (such as what you're watching right now) directly to the application as part of the extra-generated intent. This behavior is transparent in that if the application doesn't support it, it will launch just like any other application would. Adding support to capture the extra intent information is trivial and an extremely minimal amount of code. Some of the metadata (intent info) is being standardized in an effort to get re-use among support apps. Currently, we are modeling metadata generated by the Companion Dashboard as Activity Streams. This is reflected in actions that can be performed via the Dashboard on both applications and channel/program specific metadata (See Figure 1.). Long pressing on either of these areas presents a toolbar to users that allow them to rate/share and comment on them. This data is captured in the Activity Stream format and can be exported to a backend for analytics purposes.

We developed several halo experiences to showcase the capabilities of CDF. These applications are developed such that they inherently support all the benefits CDF provides (as outlined above). However, as it's primary capability of being an application launcher - the Dashboard supports any third-party application available through the marketplace or otherwise (both native and web) as well as URLs. One particular application that illustrates the capabilities of supporting CDF was an application called TweetTV (see Figure 2). This was similar to other third-party Twitter applications with one twist, it utilized the extra intent passed to it about what the user was consuming on the primary screen to automatically segment, filter and curate tweet streams that related directly to the primary screen content. For example, when the user was watching a sports event, such as a MLB game, upon launching TweetTV they would be presented with tweets corresponding to Team A, Team B, and a general stream of MLB-related tweets without having to do nothing more than simply launching the application. This demonstrates the ability to not only provide suggestions and recommendations of applications based on what the user is watching, but also contextualize the delivery of those experiences to the user.

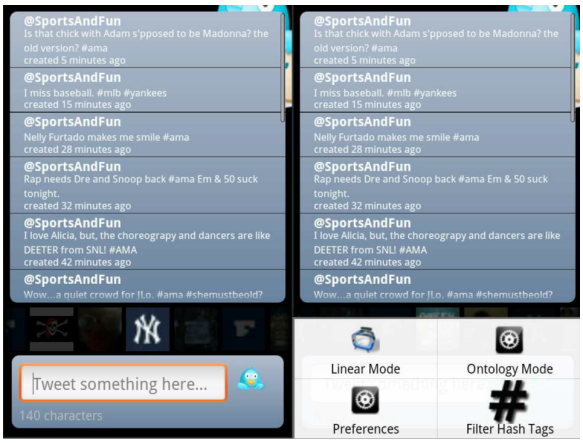

Figure 2: TweetTV (Android)

3. FUTURE DIRECTIONS

Key future directions involve broadening the diversity of synchronization choices (e.g. to include techniques such as ambient audio [8,9], increasing the style and variety of applications that can be integrated into the ecosystem (both Android and HTML5 applications) and to further simplify the integration effort in incorporating marketplace applications into a single, cohesive TV companion experience.

4. REFERENCES

[1] http://en.wikipedia.org/wiki/Interactive_television_standards

[2] http://www.deseretnews.com/article/700111884/The-online-water-cooler-TV-networks-trying-to-integrate-shows-with-social-media.html

[3] http://www.wirelessweek.com/News/2011/02/Technology-Second-Screen-Key-TV-Engagement-Mobile-Video/

[4] http://www.intomobile.com/2011/04/07/motorola-socialtv-lets-phones-tablets-interact-tv-shows/

[5] http://www.yctvblog.com/blog/2011/01/25/connect-your-phone-and-tablet-to-your-tv/

[6] http://blog.nielsen.com/nielsenwire/wp-content/uploads/2009/09/ThreeScreenReport_US_2Q09REV.pdf

[7] TV Apps: Evolution from Novelty to Mainstream, http://pro.gigaom.com/2010/05/tv-apps-evolution-from-novelty-to-mainstream/

[8] SoundPrint Platform. http://www.intonow.com/ci/soundprint

[9] "Mass Personalization: Social and Interactive Applications using Sound-Track Identification", M. Fink, M. Covell, S. Baluja, Journal of Multimedia Tools and Applications, vol. 36 (2008), pp. 115-132.

[10] "Usages of the Secondary Screen in an Interactive Television Environment: Control, Enrich, Share, and Transfer Television Content", P. Cesar, Dick C.A. Bulterman, and A.J. Jansen, EuroITV 2008