A Flexible Multi-Screen Solution Based on UPnP

J. Clarke Stevens
Principal Architect, CableLabs
Technical Chair, UPnP

Abstract

The popularity of smart phones and tablet devices has created a viable and attractive new user model for interaction with television. There are numerous use cases for interaction between these devices. Universal Plug-n-Play Forum has created a new and open multi-screen specification that leverages the billions of UPnP devices already deployed, leverages W3C specifications in development and offers flexible numbers of primary and secondary screen devices to interact with each other in a synchronized fashion over IP networks. Some gaps in W3C specifications are identified and the new UPnP specification is described as a solution. A further explanation is given on how the new multi-screen protocol ties in to the Internet of Things to enable even more multi-screen use cases.

Multi-screen Use Cases

Development of a multi-screen solution is driven by a few key use cases. These use cases, in turn, are driven by a need for a shared context. Either the user is simultaneously interacting with multiple devices (e.g. a tablet and a TV), or several users are sharing a common experience (e.g. playing a game, watching TV together, etc.). Such applications require real-time sharing, information synchronization, and multiple paths of control.

Gaps in W3C Specifications

W3C has taken major steps in the last couple of years to address new use cases that allow HTML interfaces loaded from the Internet to access and interact with multiple devices in the home network. The key example is the Network Services Discovery (NSD) specification currently in Editor's Draft.

NSD uses CORS to allow HTML interfaces hosted in one domain to securely discover devices hosted in a different domain. Current NSD sample implementations use UPnP (SSDP) and Zeroconf (mDNS/DNS-SD). Other discovery protocols can also be used. Interfaces like XML Http Request (XHR) and WebSockets can be used to send communications between the HTML user interface and discovered devices.

NSD is an important specification for device discovery, but W3C specifications still lack information on how HTML can be used to provide coordinated user experiences

between multiple devices. There is not a commonly specified way for coordinated interaction between a dynamic set of screen devices. There is not a specification for synchronizing multiple user interfaces, nor is there a way to move user interfaces between devices.

The following key use cases were important in designing the UPnP multi-screen solution.

Synchronization

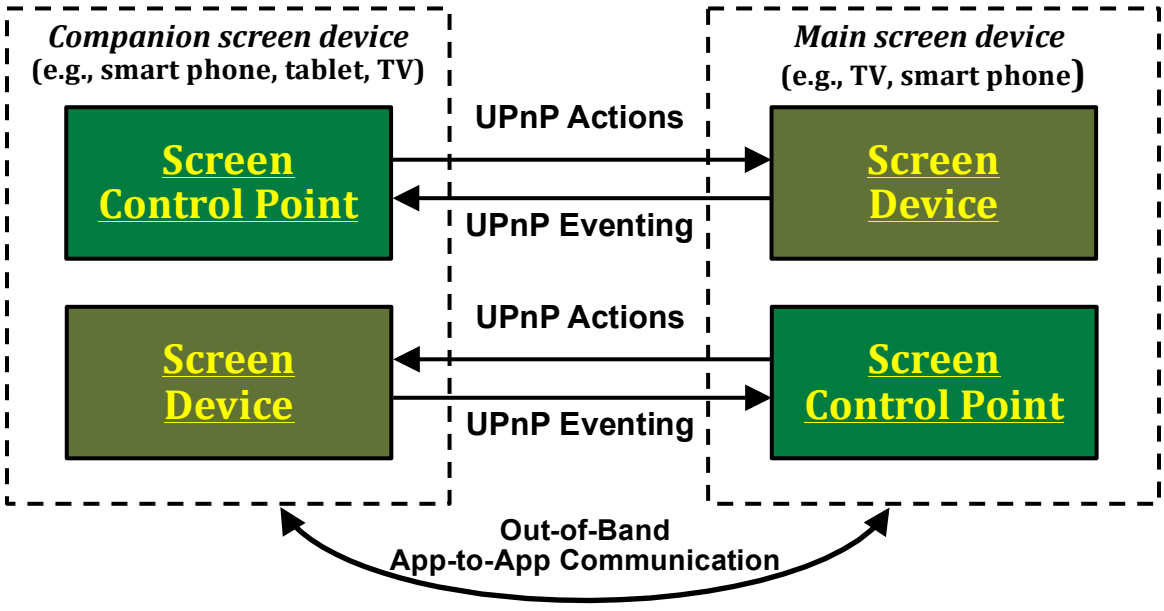
Synchronization is one of the key features that move a simultaneous experience into an integrated experience. Shopping for clothes online while the TV is on is very different from being provided one-click purchase of the clothes the actors are currently wearing on screen. This second example requires a coordination method, and information about the content of the television programming.

Screen Passing

Screen passing is another primary multi-screen use case. People generally want to watch their content on the best nearby device. This suggests the need for transferring content between devices. Displaying the same content on multiple screens is a use case with similar requirements. Standards are required for a common way to modify context.

Main & Companion Screens

Several use cases can be understood in the context of main and companion screens. A "main" screen is a rather flexible term, but can be considered the screen of primary focus. The companion or secondary screen provides information or interaction in support of the main screen. The main screen might be a television with the companion screen serving as a remote control. The main screen might be a common game board while the companion screen provides the game view for an individual player. A standard way to model and implement these complex and dynamically changing interface requirements is necessary



UPnP Multi-screen Architecture

The UPnP multi-screen architecture supports all of these use cases through a model defining a "screen device" and a "screen control point." Any screen can have either or both of these features. A screen device defines actions and state variables that can be controlled. A screen control point launches those actions and can read and set the state variables. Since any physical device, like a tablet or a television, can operate as either or both, the types of multi-screen applications that can be developed are limitless. A television operating as a screen device could change channels based on instructions from a screen control point on a tablet. Additionally, the television could also act as a screen control point and send product advertisement links to the tablet acting as a screen device. This architecture enables a flexible n-by-m interaction model so applications can be implemented with any number of screen devices and screen control points.

Since the UPnP device architecture provides device discovery and notification when devices disappear (and allows access to this architecture through W3C NSD), setting up a dynamic system for interaction between multiple screens is straightforward for most any type of application.

The UPnP multi-screen solution does not prescribe any particular application design. Although you can design an application that does all communication via UPnP, you could also design a multi-screen application that uses any other communication model and just uses UPnP communication to set up the screens and load the application. For example, a multi-screen application could discover a television screen and a tablet and use UPnP to manage the control of the application, but then stream the video over a separate path using HTML. This design allows for tremendous flexibility in application and communication design while providing a common way to manage the devices participating in the application. UPnP architecture provides a common framework for discovery, description, control, eventing and notification, yet still allows developers great flexibility in application design.

Following the example of NSD, a flexible W3C specification could be created to describe a common framework for creation and management of multi-screen applications. NSD could be used to discover potential interface devices and a W3C API to enable the features of the UPnP specification would allow any number of screens to be discovered and added to multi-screen applications implementing HTML interfaces.

Multi-screen in the Internet of Things

Two new UPnP specifications can be leveraged to extend the multi-screen experience to new devices and to cloud-hosted shared experiences. The UPnP SensorManagement specification allows almost any new type of device to be quickly implemented by specifying an XML structure. It extends the multi-screen experience by quickly adding control for rolling down projection screens, managing ambient lighting or setting up special features like alternate audio tracks.

The new UPnP Cloud annex uses XMPP to coordinate communications across the cloud using a simple “chat room” based paradigm. It enables multi-screen experiences to extend beyond the LAN. A multi-screen experience can be shared with family across the Internet or be accessed remotely from a smart phone. UPnP devices that implement the cloud annex can be immediately added to a multi-screen experience from any location and the billions of UPnP devices already deployed can be added to a cloud multi-screen experience through a UPnP Cloud bridge.

Conclusion

HTML5 is clearly the interface of choice for emerging services. By creating an HTML multi-screen API in W3C, these services can be implemented in a common way and accessed through browsers on multiple platforms. The UPnP multi-screen specification provides a flexible open interface that can be made available through NSD and can access Web and TV devices in the home. Any number of primary and secondary screens can be combined to support shared applications. The benefits of multi-screen applications can be further extended to control additional devices that are part of the Web and TV experience.

By using the UPnP Cloud annex, multi-screen applications can be further enhanced to share common experiences, like watching video programs with friends and family, across the Internet.

W3C applications can be enhanced through the adoption of a common framework for multi-screen services that leverage a flexible combination of screens and other real-world devices for new classes of shared experiences.