Developing Cross-Display Applications Using the Really Easy Displays (RED) Framework.

Roberto Calderon, Michael Blackstock,
Rodger Lea, Sidney Fels
Media And Graphics Interdisciplinary Centre
The University of British Columbia
roberto@alumni.ubc.ca,
mblackst@magic.ubc.ca,
rodgerl@ece.ubc.ca, ssfels@ece.ubc.ca

Ricardo Almeida
Departamento de Computação
Universidade Federal de Sao Carlos
ricardoalmeida@dc.ufscar.br

ABSTRACT

Public displays, mobile phones, and personal computers have become pervasive in public spaces. Yet, developing, deploying and maintaining applications that promote spontaneous interaction between displays on these devices is often challenging. We present the Really Easy Displays (RED) Framework that provides easy development and deployment of multi-display applications. The Framework leverages the concept of an *Internet of Things* where physical (people, places, sensors and actuators) and digital entities (applications, paragraphs, images, etc.) are able to interact with each other. This video outlines the motivation for our framework, talks through its usage and demonstrates the ease of developing and deploying a number of sample applications.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

1. INTRODUCTION

There has been significant research activity in the Ubiquitous Computing (Ubicomp) community in recent years to investigate issues related to pervasive displays. These have included deployment considerations [2], technologies [6], their role in public places [1] and communities [5]. Although displays are pervasive components of our world, developing and deploying applications that provide spontaneous interaction in multi-display scenarios is often challenging. In our own research over the last 6 years we have been exploring the use of pervasive displays in public settings, initially focussing on the necessary infrastructure for such displays [3], on engagement and casual use and more recently on understanding the core design issues when pairing personal displays with large screens [4]. During the course of this research, we have developed and deployed a large number of infrastructure components and applications and have identified a number of key 'practical' issues associated with the development, deployment and maintenance of multi-display applications. These include supporting and facilitating spontaneous interaction between devices (e.g. phones and situated screens) without the need for specialized software or hardware; simplifying application development, deployment, and maintenance to encourage the creation of a viable ecosystem for pervasive displays; and providing an easy-to-understand ontology of elements for multi-display interaction (e.g. gestural data, data streams, display content, context) that can be assembled together to create interactive applications.

To meet these needs, we propose the Really Easy Displays (RED) Framework that leverages web technologies and an *Internet of Things* approach to allow the easy combination of physical objects (situated screens and mobile phones) and content often found associated with such objects (text, images, or videos). In turn, these collections of physical and digital "things" can together form interactive applications that span multiple types of displays (mobile phones, tablets, large screens) and development platforms (Windows, Linux, Android, iOS). In this paper we discuss the motivation for our work, and how it has shaped the design of the RED Framework for cross-display interactive applications. We provide an overview of the framework's architecture and illustrate its ease of use. We discuss the lessons learned from initial usage of the Framework, and outline future work needed to improve the proposed technology.

2. MOTIVATION

The proliferation of public displays and personal devices such as cell phones or tablets offers a unique and growing environment for multi-screen interaction.To support these and similar requirements we designed a set of abstractions and developed a core publish/subscribe platform providing RESTful web services to allow for the easy development of interactive multi-display applications. Once built, we have used this platform to develop and deploy several applications that explored use of personal mobile devices and large public screens. According to our previous findings, mobile phones play a significant role when more interaction accuracy or privacy is required. These personal displays promote a stronger feeling of possession, reducing the embarrassment of utilizing public screens. Large displays, by contrast, allow for an increased sense of social engagement and accessibility of content to a broader audience.

We have found [4] that two of the most critical challenges for building applications in such multi-display public scenarios are firstly, promoting spontaneous interaction by simplifying the flow from bystanders to actors, and secondly, leveraging complementary capabilities from different types of displays (e.g. mobile phones, large displays). Spontaneous interaction with situated displays is difficult to achieve partly because of the need for people to install custom mobile phone applications to support interaction, and partly because developing applications for multi-display applications requires the use of technology native to each kind of display, and custom protocols to communicate between displays. The plethora of operating systems, mobile platforms (e.g. Windows, Linux, Android, iOS) and display devices makes this task challenging.

3. THE RED FRAMEWORK

The Really Easy Displays Framework (Figure 1) consists of three layers. 1) A supporting infrastructure providing a suitable ontology, protocols and data management (*the Thing Broker*), 2) an application container, either web-based or native to a display, providing a) contextualization of interaction flow based on place or coupled devices, and and b) access to native capabilities of a display – like accelerometer or camera –, and 3) a DOM manipulation layer based on Javascript, which provides control of displays and easy access to data.

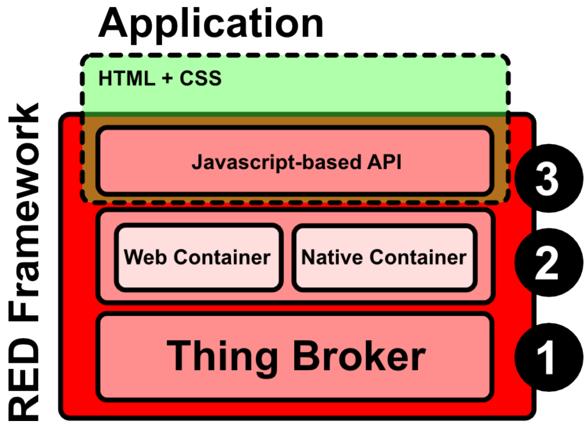

Figure 1: Framework Architecture providing a (1) data model,(2) application container, and (3) DOM manipulation layer.

The Thing Broker provides a data model supporting the notion of "things" as a way to encapsulate multiple types of data sources and real-world objects. The Application Container is formed of two components: 1) a native mobile container and b) a web-accessible container. The native mobile container provides access to device-dependent capabilities or sensors (e.g. camera, or accelerometers), while the web-accessible container provides interaction flow threading depending on contextual information. The third layer of the RED Framework provides an jQuery based API to develop multi-display applications following conventions used to create web documents.

3.1 Video Demo

The video provides an overview of the motivation for a framework highlighting the key need for ease of development. A short example is used to show the ease of developemnt and provide an indication of the power and flkexibility of the framework and its simple develop-deploy model. The video finishes with some examples of the framework in operation and draws out some of the lessons learnt from our experiences with RED.

4. CONCLUSION

We have presented the Really Easy Displays (RED) Framework, a set of technologies that can be used to develop, deploy and maintain multi-display applications. The RED Framework leverages web documents to create interactive applications that bridge across display characteristics (size, orientation, etc.) and operating systems (Windows, Linux, Android, iOS, etc.). Our experiences with the framework have shown the benefits of a rapid development environment for large screen applications and the need for implicit support of mobile access and control of such interactions. The RED Framework is currently being used by several researchers to develop applications in different scenarios ranging from healthcare, public spaces, third places and situated gaming. These new case studies will prove essential in testing our approach to multi-display application development, and eliciting new requirements for our framework.

5. REFERENCES

[1] E. F. Churchill, L. Nelson, L. Denoue, J. Helfman, and P. Murphy. Sharing multimedia content with interactive public displays: a case study. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '04, pages 7–16, New York, NY, USA, 2004. ACM.

[2] N. Davies, M. Langheinrich, R. Jose, and A. Schmidt. Open display networks: A communications medium for the 21st century. *Computer*, 45(5):58–64, May 2012.

[3] A. Erbad, M. Blackstock, A. Friday, R. Lea, and J. Al-Muhtadi. Magic broker: A middleware toolkit for interactive public displays. In *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pages 509 –514, march 2008.

[4] N. Kaviani, M. Finke, R. Lea, and S. Fels. Investigating a design space for multidevice environments. In *International Journal of Human-Computer Interaction (IJHCI) Special Issue on Distributed User Interfaces. Special Issue on Distributed User Interfaces, ed. Taylor and Francis.*, volume 28, pages 722–729, 2012.

[5] N. Memarovic, M. Langheinrich, F. Alt, I. Elhart, S. Hosio, and E. Rubegni. Using public displays to stimulate passive engagement, active engagement, and discovery in public spaces. In *Proceedings of the 4th Media Architecture Biennale Conference: Participation*, MAB '12, pages 55–64, New York, NY, USA, 2012. ACM.

[6] J. Müller, F. Alt, D. Michelis, and A. Schmidt. Requirements and design space for interactive public displays. In *Proceedings of the international conference on Multimedia*, MM '10, pages 1285–1294, New York, NY, USA, 2010. ACM.