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

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. We discuss lessons learned from using the Framework to prototype a multi-display application within a Brazilian healthcare facility, and outline future work.

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 [5], their role in public places and communities[1]. 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 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 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 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.

2. THE RED FRAMEWORK

The Really Easy Displays Framework 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, and 3) a DOM manipulation layer based on Javascript, which provides control of displays and easy access to data.

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. In the *Thing Broker* "things" are used to hold and relay state, events and raw data related to physical entities or virtual objects, providing a core data model that allows for data manipulation common in a variety of scenarios including multi-display situations. To accommodate multiple instances of the same application, the RED Framework provides a threading mechanism to couple "things" to an interaction *context*. The third layer of the RED Framework provides an API to develop multi-display applications following conventions used to create web documents. This layer is a jQuery plugin that leverages popular web document manipulations, and AJAX-based access to RESTful APIs.

3. CASE STUDY AND LESSONS LEARNED

We created a public display application to evaluate the RED Framework. This case study took place within a community of healthcare professionals in a Centre for Mental Care in Brazil. In this community people have little or no previous exposure to digital information and communication technologies. Our ongoing research focuses on understanding how Natural User Interfaces can affect the functioning of similar communities. In this particular case study, HCI researchers required the iterative development of a public kiosk allowing healthcare professionals to submit new tasks or mark previously submitted tasks as completed. Access to tasks was required via a public display, tablets and mobile phones.

The application was designed in terms of *things*, rather than *data*. It was determined that data across all displays needed to be synchronized, i.e. that it belonged to the same *context.* Tasks were modelled as a *task thing* that generating task events; each *task event* was modelled with a boolean "resolved" field that could be updated as required. Each user was considered as a separate *user thing* generating either *create* or *resolve* events. The underlying *thing* abstractions provided by the Thing Broker simplified the understanding of the application for developers. The kiosk application consisted of a web-based application creating a *userID thing* for each person in the community that automatically generated or updated the *tasks thing* events. All of the user *things* were then requested by a Javascript-based visualization to generate a network graph. The mobile application, shared across the tablet and the mobile phone, simply requested the *tasks thing* and rendered its past events in list form. The underlying framework ensured that data was synchronized across applications, timestamped and stored in a remote database for future analysis.

With respect to *supporting spontaneous interaction* our case-study reinforced the generally accepted belief that walk-up interaction, without the need to install native applications, was greatly preferred by people when interacting with multi-display scenarios. In our deployments within the healthcare setting this was not only welcomed by developers, but mandated by the need for rapid and iterative development that would have been difficult through development of native applications. Equally, when considering *ease of development*, by liberating developers from the task of creating infrastructure for cross-display data management and data persistence, we were able to reduce both the development time and expertise needed by researchers.

In addition to our original motivation, the use of the RED platform also highlighted several issues in our underlying model.

Low level abstractions for pervasive displays. In our case study, the "thing" model was useful to conceptualize a multi-display application. Although we found this model useful, we also found the need to provide a contextual information container (e.g. interaction between "things" according to a place or a set of displays) on top of the Thing Broker abstractions.

Applications as inter-connected things. While developing the application for our case study we noticed that our understanding of multi-display and Ubicomp applications shifted from "visual representations of data" to "things within an ecosystem of inter-connected things". Being able to model web document elements (divisions, paragraphs, images, etc.) as *things* in an Internet of Things freed the application from limits imposed by devices and web-browsers. For example, being able to modify a division "thing" rendered at a tablet through a text-box "thing" residing on a public kiosk was a trivial task. This approach of 'pushing our model to the limit', i.e. modelling web elements as things, goes beyond current approaches in the Internet of Things (IoT) community and offers a tantalising glimpse of a merger of web and IoT technologies that we hope to explore further.

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. Using this framework we have developed and deployed a multi-display application in a Brazilian Healthcare facility and discovered the need to support better interoperability between application elements, accommodate contextual information, and provide data security. Our future work consists of implementing these missing functionalities, and increasing the ease of application development. 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] 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.