Web-Based Applications for
Open Display Networks

Constantin Taivan

UMinho | 2014

Universidade do Minho
Escola de Engenharia

Constantin Taivan

Web-Based Applications for Open Display Networks

**Programa de Doutoramento em Informática
das Universidades do Minho, de Aveiro e do Porto**

universidade de aveiro

Universidade do Minho

U.PORTO

setembro de 2014

Universidade do Minho
Escola de Engenharia

Constantin Taivan

Web-Based Applications for Open Display Networks

Programa de Doutoramento em Informática das Universidades do Minho, de Aveiro e do Porto

universidade de aveiro

Universidade do Minho

U.PORTO

Trabalho realizado sob a orientação do
Professor Doutor Rui João Peixoto José

setembro de 2014

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração da presente tese. Confirmo que em todo o trabalho conducente à sua elaboração não recorri à prática de plágio ou a qualquer forma de falsificação de resultados.

Mais declaro que tomei conhecimento integral do Código de Conduta Ética da Universidade do Minho.

Universidade do Minho, ____de ______________de _________

Nome completo:____

Assinatura:____

Acknowledgments

Thank God that I am finally here and I am getting relaxed writing these special thanks to all the people involved in my PhD journey. This thesis is primarily the fruit of a close collaboration with my supervisor Prof. Rui José. He deserves my sincere acknowledgments because he trusted, encouraged and invested considerable time both as a teacher and as a parent. This collaboration thus consolidated my critical thinking and enabled me to be more productive in everything I am doing.

I also would like to thank my colleagues from the Ubicomp group, their close support and daily good will. Especially, I thank Prof. Helena Rodrigues, Bruno Silva, Hélder Pinto and Jorge Cardoso for their devotion to give me support and explain things for me. As well, all my professors and colleagues I have met in my doctoral program deserve kind thanks. They never forgot to adapt the language for me or even teach me Portuguese. This thesis was also part from a fruitful collaboration with the members of PD-Net European project. I may thank Dr. Adrian Friday who helped us to frame this thesis from the outset. Then, Ivan Elhart deserves special thanks as he enjoyed coming in Portugal and spending time together working to frame our research topics. I also thank Nemanja Memarovic, Florian Alt and Sarah Clinch for their kindness and useful conversations. I thank Prof. Nigel Davies, Prof. Albrecht Schmidt, Prof. Marc Langeinrich and Prof. Rodger Lea who were pleased to give me interesting insights.

My family is happy to see me finishing this doctoral program. They always taught me not to lose time and do things in life that I can benefit afterwards. Without them I could not do what I put in mind to do – considering that to be of value. Finally, in the last year, the most beautiful support came from my son, Mihail. My wife supported from the beginning. Roxana deserve my best thanks because she guided and encouraged all my steps in my PhD. Her family also deserves kind acknowledgments due to their devotion and trust. In the end, I thank all people to whom I collaborated and I also apologize for my mistakes.

This research was supported by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 244011 (PD-Net) and Fundação para a Ciência e Tecnologia (FCT) under the PhD grant SFRH/BD/75868/2011.

Abstract

Web-Based Applications for Open Display Networks

Digital display technology is facing considerable progress and large screens are increasingly pervasive in our public environment. Most display systems exist as non-interactive units, broadcasting multimedia content, such as images, text and videos and work under tight control of their owners. These installations usually do not take much consideration for the audience, as information is pushed regardless of viewers' interests. Although the pervasiveness of these displays has increased in the last decade, they do not provide considerable added value for people's lives. This means that public displays have not yet managed to integrate the emerging ecosystem of services and devices that constitute a ubiquitous computing infrastructure.

Open Display Networks are an emergent paradigm that considers the possibility of opening the currently closed networks and invite audience into new levels of engagement through the usage of sensing and interaction capabilities. It also entails the idea that display owners will benefit from opening their display networks for third-party software applications and multimedia content. This originates multi-application displays that can engage viewers in a wide range of usage scenarios similarly to what happened with smartphones.

This research addresses the conceptual and technical challenges for the concretization of the applications for interactive public displays. Our work contributes with the fundamental understanding of what a display application might be and how it should be designed, developed and used as part of multi-application display experiences. We came out with a set of application design principles and characteristics that may guide application creators in the design space of multi-application displays. This thesis also contributes to the understanding of the implications that display applications might have on the use of Web technologies as an appropriate technological framework for the creation of this type of applications. We identified a set of Web development specificities and insights that help third-party developers to understand in what ways a display application is different from its desktop or mobile counterparts. The new findings developed as part of this thesis are expected to have an impact on the emergence of potential web-based application models and related application ecosystems. This in turn would determine the deployment of multi-application display systems that would embed many use cases and therefore increase benefits for all the parties involved.

Keywords: Pervasive displays, Interactive public displays, Open display networks, Public display applications, Third-party applications, Web technologies.

Resumo

Aplicações Baseadas em Tecnologias Web para Redes Abertas de Ecrãs Públicos

As tecnologia de ecrãs digitais estão a enfrentar um progresso considerável e é cada vez mais comum a sua presença em vários tipos de espaços públicos. A maioria dos sistemas consistem em unidades não-interativas, para difusão de conteúdos multimédia, como texto, imagens e vídeos. Estes sistemas funcionam sobre um controlo direto do proprietário e sem grandes preocupações com o público e os seus interesses. Apesar de serem cada vez mais comuns, dificilmente se poderá dizer que essa maior presença de ecrãs públicos corresponda a um maior valor acrescentado para a vida das pessoas.

As Redes Abertas de Ecrãs Públicos são um paradigma emergente, que considera a possibilidade de abrir as redes actualmente fechados e convidar o público para novos níveis de apropriação dos ecrãs como meio de comunicação. Este paradigma baseia-se também na ideia de que os proprietários de ecrãs poderão partilhá-los com aplicações de software e conteúdo multimédia de terceiros, saindo a ganhar com o valor gerado por essa partilha. Isto origina ecrãs com múltiplas aplicações que podem envolver os espectadores em uma ampla gama de cenários de uso semelhante ao que aconteceu com os smartphones.

Esta investigação aborda os desafios conceptuais e técnicas para a concretização das aplicações de ecrãs públicos interativos. Esta investigação contribui com a compreensão fundamental do que uma aplicação de ecrã pode ser e como deve ser concebido, desenvolvido e usado como parte de ecrãs com experiências de múltiplas aplicações. A contribuição desta tese inclui um conjunto de princípios de design de aplicações e características que podem orientar os criadores de aplicações no espaço de design de ecrãs com múltiplas aplicações. Esta tese também contribui para a compreensão das implicações que as aplicações de ecrãs pode ter sobre o uso de tecnologias da Web como uma estrutura tecnológica adequada para a criação deste tipo de aplicações. Identificamos um conjunto de especificidades de desenvolvimento Web e recomendações que visam ajudar os desenvolvedores de aplicações a entender de que forma a aplicação de ecrã é diferente de uma aplicação web clássica. Este resultados deverão ter um impacto sobre o surgimento de modelos de aplicações baseados na web e ecossistemas de aplicações relacionadas. Este por sua vez, será uma contribuição para a implantação de sistemas de ecrãs com múltiplas aplicações que sustentem muitos casos de uso e possam assim servir melhor os objetivos de todas as partes envolvidas.

Palavras-chave: Ecrãs públicos, Ecrãs públicos interativos, Redes Abertas de Ecrãs Públicos, aplicações de ecrãs públicos, aplicações de terceiros, Tecnologias Web.

Table of Contents

List of Acronyms

AJAX Asynchronous JavaScript and XML

API Application Programming Interface

CMS Content Management System

CSS Cascading Styles Sheet

DOM Document Object Model

GPS Global Positioning System

GUI Graphical User Interface

GWT Google Web Toolkit

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

IDE Integrated Development Environment

IM Instant Messaging

LCD Liquid Crystal Display

MOO/MUD Multi-User Dimension/Domain

NFC Near Field Communication

OBEX Object Exchange

ODN Open Display Networks

OS Operating System

PDA Personal Digital Assistant

QR Code Quick Response Code

REST Representational State Transfer

RFID Radio-Frequency Identification

RSS Really Simple Syndication

SMS Short Message Service

SNS Social Networking Service

SVG Scalable Vector Graphics

UI User Interface

URL Uniform Resource Locator

USB Universal Serial Bus

WIMP Windows, icons, mouse, pointer

WLAN Wireless Local-Area Network

WSN Wireless Sensor Network

XML eXtensible Markup Language

List of Figures

List of Tables

What Hath God Wrought?[1]

[1] In the memory of May 24, 1844 when Samuel F.B. Morse inaugurated the world's first commercial telegraph line with this message: "What Hath God Wrought?" taken from Bible (Numbers 23:23).
Source: http://www.history.com/this-day-in-history/what-hath-god-wrought

Chapter 1

Introduction

Digital technologies are increasingly embedded in urban spaces aiming to provide new types of support for our everyday activities (Greenfield 2006). Large digital displays, in particular, have the properties to become an important infrastructural element for intelligent urban environments. They have always been part of the ubiquitous computing[2] vision. In the original Weiser's work (Weiser 1991), large digital displays were called "boards", where they were essentially seen as interactive, yard-size displays that would complement mobile devices such as pads and tabs to enhance spaces by enabling a broad range of casual information exchanges.

The increasing availability of affordable projectors and especially flat-panel displays has contributed to significantly expand the number of displays deployed into public spaces, and nowadays, as we look around in our cities, we are constantly facing a very broad range of public displays with multiple form factors and serving very diverse purposes (Kostakos & Ojala 2013). Such displays are primarily driven by advertisement and they are commonly used for showing special offers in shopping malls, list interesting facts and events at universities, display schedules and news in metro stations, or advertise a product on a building facade. In other cases, public displays employ entertainment content or useful, interesting information and services. Being a ubiquitous element of our visual culture, digital public displays may be recognized as a technology that may radically change communication in public and semi-public spaces.

However, despite the considerable technological advances in networked services and screen hardware and the increasing interest on public displays as a research topic and as a commercial field, the fact remains that the value proposition of the displays we can today find in public spaces has not changed much for the past decade. It seems clear that more displays, as we know them today, is not necessarily something that people would perceive as an added-value to their lives. Even though digital displays are pervasively deployed, they have not yet managed to integrate the evolving ecosystem of services and devices that constitute a pervasive computing infrastructure. There are two major limitations that are inherent to the current assumptions upon which these systems are designed: *centralized content distribution* and *single domain control*.

Centralized content distribution. The first major limitation of the existing display infrastructure is that the single and centralized content management model can be seen as mimicking traditional TV-scheduling. While it is possible in public displays networks to have fine-grained control over how to distribute different content to different locations, a model known as narrowcast, the paradigm is essentially the same as in a broadcast model. The infrastructure is mainly designed to enable advanced scheduling of pre-arranged

[2] Ubiquitous computing and pervasive computing refer to the same computing paradigm

content on a network of public displays, featuring tight control of injected content. Content is mainly driven by advertising and pushed to the public irrespective of their wishes or preferences. Most displays are not interactive at all, and active engagement by people seeing the display is not common. The displays are thus essentially seen as end-points for the distribution of centrally created content. This content distribution mind-set does not offer much consideration for public displays as strongly situated artifacts that should be deeply embedded in their physical, social and cultural settings.

Single domain control. The second major limitation is that current display systems are essentially a world of multiple closed networks that operate as isolated islands under single management domains. Because the network is closed, each provider must be able to guarantee by itself the entire value proposition. The integration of more entities in the process needs to be individually negotiated in order to create formal partnerships for specific display networks. This is a complex process that does not scale well to a large number of partners and is excessively demanding to be viable for small deployments. As a consequence, value chains are typically very short, often involving a single entity that needs to assume all the functions of network operation, from creating content to monitoring the network or selling advertising space. This is only possible for relatively large sized networks and even then the end result tends to be very limited.

1.1 Open Display Networks

In our research, we considered a new vision for public displays. Our motivation is that public displays can be more valuable if they become part of open large-scale networks, in which there is no single point of control for display nodes. Such displays would rather be situated in their physical, social and cultural settings, employing rich interactive features and allowing passersby to influence the content shown. We believe that 21st-century public displays should form the backbone of a new communications medium with the same potential impact on society as radio, television, and the Internet.

A parallel can be drawn between the current state of the art in display systems and the state of computing prior to the invention of the Internet. Machines were networked together in small clusters to facilitate resource sharing, control and communication, but there were no mechanisms for interconnecting these networks to enable end-to-end applications. The development of mechanisms to enable communication between computers on these different networks created a robust, scalable, distributed and open platform that transformed life in the subsequent decades with the emergence of applications such as email, the Web, e-commerce and social networking.

A similar breakthrough is needed to create a robust, secure, scalable, distributed and open platform that enables the formation of large-scale networks of interactive displays and associated sensors that may effectively become a communication medium that is open to multiple communication purposes and social practices. Therefore, unleashing the full potential of public displays as a new communication medium will require a radically different paradigm that fully explores openness as a source of value for all the stakeholders. We will refer this new paradigm as Open Display Networks (ODNs) (Davies et al. 2012).

Currently, there are no such open displays systems and their materialization may significantly change the way we perceive public displays and bring a set of new and innovative usage scenarios regarding the information propagation in public spaces. According to (Davies et al. 2012), in order to build a communications medium via an open network of public displays, it would require significant advances along five key building blocks:

1. Open architectures that extend the use cases of display networks by serving the interests of many stakeholders including display or venue owners, viewers and content providers.
2. Deploying public displays as situated artifacts is a key factor for building a positive perception of currently ignored public displays. ODNs are considered pervasive, not just because we can anticipate having displays almost anywhere, but mainly because the displays will be seen as an integral part of their physical, social and cultural setting.
3. Privacy-compliant personalization and control mechanisms would enable people to personalize the experience with public displays. This opportunity might have the potential to transform digital signage from a simple broadcast medium to an interactive system that responds to the needs and preferences of viewers.
4. Engaging and efficient user interaction models constitute a significant part of the value proposition of ODNs and can lead to stronger user engagement and user-generated content.
5. Viable business models would attract more interest from the industry in a way that ODNs would bring benefits for all the parties involved such as place/display owners, content creators and display viewers.

1.2 Novel Use Cases of Public Displays

Current digital signage systems, without considering the research deployments are mainly used for scenarios including advertising (improving customers' experience), information presentation, signage and entertainment (art, culture and aesthetics). The recent developments in sensing and interaction technologies are enabling entirely new use cases for public displays including tailoring content to the situation and audience of the display. As a consequence, interactivity will determine a change from viewing experience to a user experience that enables public displays to be re-framed as rich platforms for communication and interactions.

While the research community very often uses the notion of *public display application* at a conceptual level meaning a practical use of a public display, e.g., advertising, information presentation, signage or entertainment, in this thesis we consider a *public display application* at a concrete level referring a *piece of software or computer program,* which can be developed by third-party developers. We thus define a software application for public display as a third-party display application or for short, *display application*.

In our work, a *display application* is a software mechanism that is primarily designed to render content, i.e., multimedia items (text, images, animations, videos, soundtracks or any documents authored with standard Web technologies), across a network of large digital public displays and additionally, depending on the interaction capabilities, provide various content-related functions. We see these applications as publication mechanisms that pack data (content) and behavior (logic). The notion of application is very similar with the notion from related computing artifacts such as desktop or smartphones. We consider that content, i.e., information intended for the audience, is part of the entire application data space. The applications are responsible for the logic to present its content and in the same time, gather and process both users' requests and environmental data. In public displays, we shift the value proposition from content to applications. Rather than the simply distribution and presentation of content, like most digital signage systems do today, displays would be running multiple applications, each handling and presenting in a semantically and contextually meaningful way, content from many sources, e.g., display audience, display owners. Therefore, a public display, with its physical environment, interaction mechanism and possibly sensors, becomes the execution environment for these applications.

The first main motivation of enabling applications in public displays is to decentralize the system itself from complex behaviors (logic) in handling with potential content items. While a traditional display system has a dedicated behavior of dealing with its content, new benefits are emerging from enabling autonomous software applications to manage particular content types and thus extend the overall system functionality, even in non-interactive scenarios. Perceiving public displays as information systems employing many software applications rather than simply content is an inherent decision that technically should be taken when considering the extension of display use cases and the fact that such display will satisfy the interest of many stakeholders. As content may come from multiple sources and will serve many purposes would it be hard to manage it outside its meaning and usage case. Therefore, public displays require these software programs to handle how we consume and create digital content. From a computational perspective, applications, and not content, become the primary driver for the experience offered by public displays.

"The Appification of Everything" metaphor (Wing Kosner 2012), seems to provide an answer and a motivation to the public displays case. In his article, Anthony Wing Kosner, propose a paradigm shift in how we metabolize information and entertainment. Referring to the Web as the universal storage medium, he favors perceiving the Web content as an underlying service layer for application-based interfaces rather than a hierarchical tree of documents (Wikipedia of Wikipedias). While, the content in public displays may come from the Web (which is the case of many digital signage systems) moving towards multi-application displays would imply abstracting away from particular content items and develop applications that can provide a unified and integrated experience in using a public display. This means that content will be an integral part of applications but not necessarily bounded to them. In conclusion, displays are thus much more likely to exhibit a functionality that corresponds to a composition of applications rather than just a single one.

Opening current closed display networks raises the requirement of satisfying the interest of a large set of stakeholders including display or space owners, viewers and even applications and content creators. As previously stated, this imposes public displays to embed many applications thus becoming multipurpose artifacts by definition. We distinguish between various usage models in which display applications could be employed: 1) a display system cycles through applications using a combination with static and dynamic scheduling algorithms (Elhart, Langheinrich, Memarovic, et al. 2014); 2) general public can browse and select an application (from many choices) to use for an undetermined amount of time (Ojala et al. 2012); and 3) viewers or audience in general will be able to open their own applications on a public display and use them at will. In all these usage modes, the applications' content can be supplied either by display owner or audience or it is part of the application experience itself.

In our work, we consider that display applications may come from many sources or third parties. These third parties can contribute with applications and content to be used at any display across multiple administrative domains. Numerous recent examples, such as the Apple' App Store and Google's Play Store, have demonstrated the immense potential of opening creativity to a wide range of contributors. By using content and applications developed anywhere in a global network, there will be a shift from today's environments where information is pushed to people in the form of advertisements, to spaces that can utilize displays network and ubiquitous computing to create new opportunities for educational and economic activities or to reflect hopes and aspirations of its occupants. This open model in which the control of displays is shared by the audience has the potential to foster a rapid innovation and co-creation of value by a global community.

A key characteristic of display applications is that they would entail a clear separation of concerns between content creators and particular displays, reflecting the need to create applications that may potentially be used anywhere. Therefore applications must be developed without any assumptions about their execution contexts. This is a tremendously powerful idea for both applications developers and display owners. For developers, it would mean that instead of developing applications to be deployed in a specific display system, they could now make their applications available anywhere, significantly increasing their potential reward for investing in application development. For displays owners, this would mean access to a wealth of interactive applications from a variety of sources and a wide range of general purpose content without the need to implement applications by themselves.

The currently closed model of public displays is analogous with that of mobile devices early before the iPhone advent, when the handsets were effectively locked down in terms of functionality and use cases, with network operators acting as gatekeepers to new services. Things radically changed with the emergence of Apple' iOS and Google's Android open ecosystems, which encouraged innovation and allowed a wide range of developers to create new applications for mobile devices. The newly created mobile app stores have spurred the emergence of a huge diversity of application offers, transforming those devices from single-purpose communication artifacts into dynamic tools that are nowadays analogous to a "Swiss Army Knife" (Satyanarayanan 2005) with an excess of readily-accessible functionalities for everyday life, e.g., navigation, sports, playing games, listening to music and sightseeing.

Similarly, we envision an analogous evolution for future public displays in a way that they will extend their usage scenarios and employ many software applications or interactive services which can support people in a wide range of activities. This means that displays will shift from single-application or dedicated displays to multi-application displays that are considered being multipurpose rather than single-purpose.

Over the years, the research community tried to describe what a multi-purpose or multi-application might be. According to (Ojala et al. 2012), one approach is to consider that the multi-purpose character of a display is given by the number of functions rather than information type. In their work, Ojala et al., describe a display with multiple information types about a city (maps, picture and text) as a single-purpose display because it has one function – to supply information (this behavior is usually encountered in information kiosks). Alternatively, a multipurpose display would provide additional functions, such as browsing, games, galleries, and polls. While this explanation may raise some confusion between multipurpose displays and multi-application displays, in our work, we consider a multi-application display implicitly a multipurpose display but the contrary not. While it is clear that a display embedding multiple applications each one with its own goal can serve various purposes, as well, a dedicated display with just one application can offer a diverse set of functionalities beyond merely supplying information, e.g., uploading user-generated content, commenting, etc. Therefore, such display may be considered multi-purpose but not necessarily embedding many independent applications. For instance, the content posted in such an application can serve both commercial purposes and any general information needs. In consequence, it should be noticed, as is the case of single concept displays (Section 2.2) that a single-application display may have various operating modes or functionalities around the same theme, e.g., improve awareness or supplying information. In this case, we do not consider them as multi-application displays.

1.3 Challenges

Building these multi-applications displays as rich platforms for communication and interaction carries a vast potential and will enable new scenarios for information access that have not been seen in the analog signage landscape. In this thesis we specifically consider both the conceptual and technical challenges to enable the concretization of *third-party display application* concept. In the following, we thus provide a detailed description of these challenges organizing the discussion on three different dimensions: *design space*, *development* and *usage*.

1.3.1 Design Space

Currently there are no abstractions to model what display applications should be like or what execution environment they should expect. The prevailing model in digital signage does not consider applications and those installations that are deployed in public spaces are largely non-interactive and their key focus is to disseminate information serving commercial purposes. On the other side, the research efforts have not yet address this topic in a systematic way. This means that most research display prototypes are designed as single concept or single application displays without an explicit focus on the notion of application. Even for the case of multi-application displays, due to the novelty of such systems, their design, implementation and evaluation is challenging. Such systems exist as research testbeds controlled mainly by researchers and currently, there are no established infrastructure where display applications are already being created and used in everyday life. Therefore, building a design space of display applications as software components that can be developed by third parties requires an understanding of applications' design principles and properties. This would enable both researchers and developers to grasp the notion of third-party display application as a distinct computer program that is a clear concept that embeds the experience offered by public displays.

Both in the commercial digital signage and in most of the research prototypes, there is a broad diversity of technical solutions and each of them decides about its own architecture. There is no shared view of what an application might be or what execution environment it should expect. The challenge is to understand the appropriate components of the execution environment needed for public display applications. By deploying public displays in open networks with associated sensors, e.g., video camera, Bluetooth or RFID scanners, it is expected that the application execution environment would no longer be bounded to a single system. Instead, it will be extended to the entire network and the encompassing digital ecosystem including, for instance, distributed user interfaces among many devices such as smartphones and displays with different sizes and form factors. From a design or conceptual perspective, there is a need for abstractions that make display and network resources available to applications so that applications might run across many types of display systems and communicate among them. From a technological or computational perspective, the issue revolves around the system assumptions and the choices between executing code on the display nodes and executing code in the network, possibly hosted as web-based applications at some specific server.

1.3.2 Development

Developing applications for open public display networks require novel paradigms and adequate programming tools. Researchers often build distributed applications for dedicated displays based on the Web approach. The main advantage of web-based applications is that they can serve a much broader range of application scenarios, including the ability to execute on multiple types of display nodes. Therefore, portability might be a fundamental property when considering large scale application deployment for display networks. However, the use of web-based applications does not rule out the existence of native applications for particular types of display nodes. While native code approaches can be more powerful in many settings including the access of hardware resources, there is no display platform that is widely accepted to serve as the target platform for the development of those applications. The Web approach, regardless of its inherent limitations, may thus provide the most appropriate alternative for a widespread development in large scale networks.

Despite the obvious potential and widespread adoption of the Web technologies in creating display systems, including both infrastructure and applications, there is a lack of understanding about the development specificities within this particular domain. The traditional usage model of Web does not fit well the usage model of public displays. Simply rendering content on a browser does not provide a good user experience and specific solutions that package web engines with the display-specific functionality may be needed to fully explore the value of this integration. In addition, each display infrastructure has its own way to describe the application assumptions and requirements, which mean different development solutions – that are custom based, laborious and system dependent. While many applications are making use of the Web, not much is known about the implications of the characteristics of public displays on these technologies.

The emergence of third-parties display applications will reflect the need for developing applications that can be distributed and deployed across a global network of public displays. A major challenge to be addressed is how to enable these global applications to be deployed to the entire display network while being able to exhibit a situated behavior on each of the displays where they are used. While not all applications need to be situated, public displays are inherently situated artifacts that are deeply embedded in their specific physical, social and cultural setting. Therefore, display applications should support this feature natively, providing by default the necessary mechanisms to generate localized content.

Most of commercial digital signage solutions consider displays as non-interactive artifacts. Interactive display applications are mainly common in research prototypes, in which developers normally implement interactive features depending on the available interaction mechanisms in a certain display. This behavior leads to solutions that are specific for each infrastructure thus limiting their applicability within the context of an open network of public displays. While it is recognized that rich and diversified support for interaction is as a key feature for public displays, there are no widely accepted paradigms or interaction abstractions for developing or interacting with public display applications. This means that there is too much specific work that needs to be done outside the core application functionality to support even basic forms of interaction. This lack of proper interaction abstractions might be considered as a big obstacle that could have also impeded the emergence of any concept of third-party display application. Therefore, the existence of some kind of interaction abstraction is especially important if we want to be able to develop public display applications that run on heterogeneous networks of public displays. In this case application developers do not have to replicate a specific application for every different type of display and users will be prompted with consistent interaction models that enable them understand and effectively enjoy in the interaction (Cardoso & José 2012)(Cardoso & José 2013).

1.3.3 Usage

Due to the novelty of multi-application displays, which are currently deployed only as research prototypes, the usage space of such systems is still under investigation and not well understood. It entails many challenges including application diversity and presentation, interface design, multi-user interaction, and dynamic scheduling models.

Building multi-application displays involves dealing with application diversity or the range of applications that could be available at a given moment. While the huge application diversity as seen in mobile landscape is the effect of individual preferences because people own and use personal devices, in public displays things can change towards valuing common expectations and consideration for managing conflicting interests of many stakeholders. This may suggests that displays could be primarily used to serve shared interests rather than a diversity of highly personal activities. Additionally, besides designing displays as shared artifacts, the number of applications deployed has direct implications on their presentation and discoverability. Theoretically, the highest the number of applications installed or deployed in a display, the harder it becomes to locate or discover an application amongst others. However, while application discoverability is tightly connected with how a display may present applications, e.g., menu browsing, or keywords-based application searching, its management can affect the adoption and potential success of the applications (Hosio et al. 2013).

Traditionally, the classical usage model for public displays is based on cycling static content items in which the displays act as distribution nodes for information dissemination – behavior known as broadcasting. Starting with the deployment of interactive displays such as touch-based or smartphone-based interaction, users changed their role from passive observers to active performers – who may, at any moment, want to influence and manage the content presentation as well as add new personalized content. Many of these displays can be subject to multi-user interactions and support a broad range of applications. In this context, a fundamental implication emerging from the interactive nature of the applications is that users should have access to appropriate control techniques that would allow them to drive the way applications are shown and used in the respective environment. Such techniques should enable each user to reason and express intentions about the system behavior, while also dealing with concurrent requests from multiple users in a way that is fair and clear.

Another implication from opening the current closed display networks to a wide range of content sources and applications is that such installations would require appropriate control and scheduling mechanisms to mediate the potentially conflicting requirements of different stakeholders, such as display owners and display viewers. Such multi-user and multi-application display environments require new forms of application and content scheduling that go beyond a predefined sequence of content (Elhart, Langheinrich, Memarovic, et al. 2014).

1.4 Research Objectives

The overall objective of this thesis is to extend current knowledge on the concept of *third-party display application.* The specification of *display application* concept will determine the development of various application models and software implementations that can be explored by the industry to create display systems that are open to applications from third-party developers. This in turn would give raise to diverse application ecosystems that would increase the value proposition of public displays and consequently start an era for new products and services. Building on the arguments we described so far and the entire research we conducted as part of this work, we state our position as:

> *Web-based applications created by third parties are adequate to be repurposed for the execution environment of Open Display Networks, both as content publishing mechanisms and interactive tools that pack features and functions related with their content.*

In the following, we refine our overall objective in a set of *five* specific objectives. The objectives cover three main dimensions: *design*, *development* and *usage.* In Table 1 we provide a brief overview of the study dimensions, objectives and chapters.

1. **Design:** Structure a design space for display applications by identifying their key design principles and attributes.
2. **Development:** Analyze the implications of the identified design principles for the use of Web technologies as an appropriate technological framework for creating this type of applications.
3. **Usage:** Uncover the impact of multi-application displays on how people might perceive the range of applications that make sense in public circumstances as well as on the user experience itself.

Objective 1: Systematize a design space for display applications. This objective explores various concepts of display applications and identifies their main design principles and properties. The outcome is a structured design space describing the main design alternatives in form of a set of key specificities that would inform the emergence of various display application types.

Objective 2: Analyze the implications of design principles on the use of Web technologies for creating display applications. This objective studies the applicability of the identified design principles and properties on display application development based on Web technologies. The outcome is a set of technological insights that build the distinctive case of third-party web-based application development for public displays.

Objective 3: Identify third-party development perspectives. This objective assesses the extent to which web-skilled developers can leverage on their experience to create display applications. The outcome is a set of development considerations that consolidate the definition of third-party display applications.

Objective 4: Identify elements that may affect the diversity of display applications. This objective uncovers and describes people's expectations and perceptions regarding a diverse set of display applications. The outcome is a set of insights into people's perceptions about the potential factors that may affect application diversity in future display application ecosystems.

Objective 5: Formulate concepts for display application control. This objective identifies, characterizes and applies relevant GUI concepts, which can frame user interaction with multi-application displays. The outcome is a set of concepts for application control and design considerations that are meant to help developing appropriate application control techniques. These techniques would allow people to control applications in multi-application and multi-user public displays.

There are also several challenges emerging from the vison of display applications that are not addressed. In our work, we do not intend to provide any contributions towards application and content scheduling, i.e., deciding when a particular content item or an interactive application would show up (Storz, Friday, Davies, et al. 2006). Moving from content to application scheduling entails a specific set of challenges as clearly identified in (Elhart et al. 2013)(Elhart, Langheinrich & Memarovic 2014)(Elhart, Langheinrich, Memarovic, et al. 2014). Studying particular interaction techniques, as well as providing interaction abstraction toolkits or middleware are also not the focus of this work.

Table 1: Mapping between research objectives, research dimensions and chapters.

Objective	**Chapter**
Design	
1. Systematize a design space for display applications.	Chapter 3
Development	
2. Analyze the implications of design principles on the use of Web technologies for creating display applications.	Chapter 4
3. Identify third-party development perspectives.	
Usage	
4. Identify elements that may affect the diversity of display applications.	Chapter 5
5. Formulate concepts for display application control.	Chapter 6

1.5 Resources

This work was conducted within the context of the European PD-Net project[3], which explored the foundations of creating a global communication medium through open display networks. Associated to PD-Net, Instant Places display infrastructure from University of Minho had the role of a localized research testbed for the various experiments conducted in this thesis. Instant Places display infrastructure and its web-based platform was thus the main resource used in this work and it acted as a research tool shared between multiple projects and also with researchers from other institutions. Being deployed in a real environment, the infrastructure brought an excellent medium for validating the research results. This is valuable because the public display deployments are very challenging and very time-consuming with many unexpected problems. The approach behind Instant Places was to create an open environment for experimentation and co-creation in situated displays that gathers and orchestrates a community of users and stakeholders. Instant Places was set-up as a Living Lab within Living Lab of Minho, a member of European Network of Living Labs (ENoLL).

Many of our display applications that were used in this thesis as reference examples for specific studies were made in a close collaboration with colleagues as well as external researchers. While a part of applications were just used as proof of concept applications and did not imply too much burden, other were intended to be deployed in the field and thus were the subject of a long run development and evaluation cycle that spanned across all four years. A particular and fruitful case of collaboration was with Ubisign[4] that provided us with the necessary tools and software.

[3] pd-net.org
[4] www.ubisign.com – Portuguese company that provides software services for digital signage systems.

1.6 Methodology

At the moment, there is not any established system where *third-party display applications* are being created and used in everyday life. Within the research context, the role of displays as a new execution environment for applications is currently being explored by some display infrastructures that anchor a significant part of their value proposition in the ability offered to passersby to select from multiple application alternatives and personalize the content shown. This means, that the development of Open Display Networks is at an early stage. Due to the novelty of multi-application displays, the design space is under debate and construction, and there are no commonly accepted principles and guidelines.

Our work combines mainly two types of research: *exploratory* and *testing-out* on both qualitative and quantitative measures (Phillips & Pugh 2005). While the exploratory research is involved in tackling a new topic about which little is known, testing-out research tries to find the limits of previously proposed generalizations or to apply certain principles and theories in order to improve them (specifying, modifying, clarifying). According to their classification, Philips and Pugh mention also problem-solving research. This involves a particular problem in the real world for which a strong documentation (seeking various theories that can be used outside the problem domain) is needed in order to discover a solution. The problem has to be defined very well with clear formulation and the method towards the solution has to be uncovered. In our work, we do not address problem-solving type of research.

As part of the PD-Net collaboration, we employed firstly a bottom up approach in which, based on various display system deployments, the members of the entire team, located in different countries, worked to inform the formulation of a design space for display applications by describing their key design principles and properties (Chapter 3). As design principles were built up, a top-down approach was used to assess their implications for creating display applications by third-party developers using Web technologies (Chapter 4). We thus frame as exploratory research the approach in which our research went from specific studies to high level concepts or principles by generalizing the notion of display application (Chapter 3). Then, we consider testing-out research the application of the concrete principles or concepts for specific contexts, i.e., Web technologies, by analyzing their impact and limits. Since our research involved tackling a new topic about which little is known, at the beginning, our research ideas could not be formulated very well. For this reason Chapter 5 and Chapter 6 are pieces of work that build on Chapter 3 and move further to explore specific design principles of display applications and thus extending the understanding of this new concept.

We employed different study types ranging from descriptive, experimental and grounded theory, under different evaluation paradigms such as asking users, lab studies, field studies, and deployment-based research, and made use of a variety of data collection methods such as interviews, questionnaires, focus groups and observations (Alt et al. 2012). Our research contributions intend to inform the wide emergence of multi-application displays based on a set of considerations regarding the design space, development specificities and usage implications of display applications. Overall, we adopted a user-centered design approach in which most of our prototypes were evaluated both in lab and in real environment. Based on different user studies we were able to iteratively improve the initial application designs.

1.7 Contributions

The main contribution of our research is the consolidation of the concept of third-party display application over there fundamental dimensions: *design*, *development* and *usage*. The benefit of this work is threefold: 1) inform designers of multi-application display about possible application design alternatives with their underlying properties. 2) provide application developers with key development considerations that orientate them to effectively create engaging and motivating user experiences, and 3) frame users and researchers expectations towards the emergence of a display application ecosystem by identifying relevant concepts for application control and uncovering existing behaviors regarding the envisioned usage of a broad set of display applications. In the following, we provide a brief overview of the particular ways in which we pushed further the state of the art of digital public displays.

1. **Reframing the Scope of Public Displays** that provides display owners, application developers and application users with a systematic approach in defining, building and usage of multi-application public displays that are opened to applications and content from many sources.
2. **Design Principles and Properties for Display Applications** that provides abstractions guiding designers and researchers to different aspects of the application design and facilitate thinking about trade-offs between one design and another. This has impact towards the emergence of various applications models for Open Display Networks.
3. **Web Development Considerations for Display Applications** that provide third-party developers with a set of web development specificities and insights, into the case of display applications. This has a direct impact into understanding the effective ways of repurposing existing web-based services and applications for public displays.
4. **People's Perceptions towards Display Application Diversity** that frame researchers' expectations regarding the emergence of a display application ecosystem in this area by providing some indication about possible factors that might hinder the development of a wide range of display applications.
5. **Concepts for Display Application Control** that can serve as a basis for the creation of appropriate application control techniques for multi-application and multi-user displays.

1.8 Scientific Publications

In regard to publications, this work has generated *nine* papers: 3 journal papers[5], 3 conference papers, 2 workshop papers, and 1 demo paper. Table 2 matches these papers with chapters and objectives. As previously stated (Section 1.6), our work was part of a collective approach in which the entire PD-Net team was involved in the specification of

[5] Paper 2 is the third one

the design principles for display applications (Chapter 3). We are currently compiling a new publication (Paper 2) to reflect this collaborative work conducted as part of PD-Net where the thesis author was involved. While some of our publications are dedicated for specific chapters, in Chapter 3 we briefly introduce many of them and highlight their sense within the entire PD-Net efforts towards instantiating the vision of display applications for Open Display Networks.

Table 2: Mapping between papers, chapters and objectives

Publications	**Chapter / Objective**
1. Taivan, C., & José, R. (2011). An Application Framework for Open Application Development and Distribution in Pervasive Display Networks. In On the Move to Meaningful Internet Systems: OTM ’11 Workshops (LNCS 7046., pp. 21–25). Springer Berlin Heidelberg. doi:10.1007/978-3-642-25126-9_4	Chapter 1 / Objective 1
2. José, R., Taivan, C., Silva B., Davies, N., Clinch, S., Elhart, I., Memarovic, N., & Alt, F. (2014). Fundamental Design Principles for Applications in Open Display Networks (Planned for submission by the end of October)	Chapter 3 / Objective 1
3. Taivan, C., José, R., & Silva, B. (2014). Web-Based Applications for Open Display Networks: Developers’ Perspective. Special Issue, International Journal of Computer Systems, Science & Engineering (In Press).	Chapter 4 / Objective 2, 3
4. Taivan, C., José, R., & Silva, B. (2014). Understanding the Use of Web Technologies for Applications in Open Display Networks. In International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops) (pp. 500 – 505). IEEE. doi:10.1109/PerComW.2014.6815257	Chapter 4 / Objective 2
5. Taivan, C., Andrade, J. M., José, R., Silva, B., Pinto, H., & Ribeiro, A. N. (2013). Development Challenges in Web Apps for Public Displays. In 7th International Conference on Ubiquitous Computing & Ambient Intelligence, UCAmI’13 (LNCS 8276., pp. 135–142). Springer International Publishing. doi:10.1007/978-3-319-03176-7_18	Chapter 4 / Objective 2, 3
6. Taivan, C., José, R., Rodrigues, H., & Silva, B. (2013). Situatedness for Global Display Web Apps. In Demo Session, Adjunct Proceedings of the 2nd International Symposium on Pervasive Displays, PerDis’13.	Chapter 4 / Objective 2
7. Taivan, C., & José, R. (2014). Application Diversity in Open Display Networks. In The 3rd International Symposium on Pervasive Displays, PerDis’14 (pp. 68–73). ACM. doi:10.1145/2611009.2611035	Chapter 5 / Objective 4
8. Taivan, C., Rui José, Silva, B., Elhart, I., & Cardoso, J. (2013). Design Considerations for Application Selection and Control in Multi-User Public Displays. Journal of Universal Computer Science, 19(17), 2526–2542. doi:10.3217/jucs-019-17-2526	Chapter 6 / Objective 5
9. Taivan, C., José, R., & Elhart, I. (2012). Selection and Control of Applications in Pervasive Displays. In 6th International Conference on Ubiquitous Computing & Ambient Intelligence, UCAMI’12 (LNCS 7656., pp. 165–172). Springer Berlin Heidelberg. doi:10.1007/978-3-642-35377-2_23	

The thesis main author's contributions to the listed publications are depicted in Table 3, together with the chapters where the content from the publication is included. Our approach in writing the thesis was mainly based on the papers. Additionally, we provided further insights and extended the explanations.

Table 3: The thesis's author contributions to the publications:
● Primarily thesis author; ◐: collaborative work; ○: primarily co-authors;

Topics/Papers	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Concept or software design	N/A	N/A	N/A	◐	○	◐	◐	◐	N/A
Software development	N/A	N/A	N/A	◐	○	◐	●	◐	N/A
Study Design	●	◐	●	◐	●	●	◐	●	●
Study Execution and Analysis	●	◐	●	●	●	●	●	●	●
Literature Review	●	◐	●	●	●	●	●	●	●
Paper Writing	●	○	●	●	●	●	●	●	●
Chapter	1	3	4				5	6	

1.9 Thesis Outline

To clarify the association between the chapters and their role in the thesis structure we provide here a brief description of each chapter.

Chapter 2: Related Work – This chapter offers a detailed description of the many concepts employed in applications for public displays. It starts with single concept displays that are designed, in most of the cases, to serve a single purpose. Then, we present the field of multi-application displays with their opportunities and challenges. We also present a review of toolkits and middleware for developing interactive display applications. Finally, we consider the field of SmartTVs from the perspective of third-party application support.

Chapter 3: Design Space – This chapter brings a closer view of public display installations that employ different applications ranging from single-concept to multi-purpose public displays. The goal of this chapter is to reach a common understanding of the main design principles that governs different application types.

Chapter 4: Web Development – This chapter looks on how to develop display applications using Web technologies. It starts with understanding the key considerations of a web-based application for public displays when compared with desktop or mobile ones and asses the implications for Web technologies. Then, it is presented a detailed journey into the development process as experimented by third-party developers.

Chapter 5: Application Diversity – This chapter uncovers people's expectations towards a diverse range of applications that might populate future multi-application public displays.

We took a top-down approach to investigate the people's preferences of different application types and how such applications might be relevant for different places.

Chapter 6: Application Control – This chapter explores another dimension of the usage space of display applications. It studies how people will be able to control applications, looking for concepts that can frame the interaction in displays with many applications.

Chapter 7: Conclusions – This chapter concludes the thesis by aggregating the contributions of this work. It also presents further research directions that can extend the knowledge towards having world-wide display application ecosystems.

Chapter 2

Related Work

In this chapter, we present a detailed description of the related work that helps to position our contribution within public displays research and offer a global picture of the motivation in addressing the topic of this thesis. We deepen into public displays literature having in mind the way in which various display systems describe their design and usage concepts. In a first step, we look for digital signage systems that are currently deployed in public spaces. Then we move to research prototypes by describing single concept displays that are usually conceived for one particular usage case or application scenario. We continue our survey by addressing research display prototypes with multi-application support. Finally, we also provide a description and analysis of research toolkits and middleware that facilitate the creation of interactive display applications and are important building blocks for future multi-application displays. In addition to public display research, we present the distinctive case of third-party applications for commercial SmartTV platforms.

2.1 Digital Signage Systems

Digital signage systems come in various sizes, shapes and forms factors. They are usually deployed in networks having only a single function – distribute information or content in which the audience is mainly passively viewing. The content of these installations is described in terms of multimedia items including text, images, videos, live streams, web pages. We know digital public displays from a range of diverse places including train stations, airports, shopping malls, hotel lobbies, city squares and many others. Being under the tight control of their owners, such systems have a particular purpose and broadcast information (usually) in a cycle including transport schedules, events and advertisements (Davies, Clinch, et al. 2014). We provide a few examples of these systems as part of four principal application domains, i.e., advertising, information, signage and entertainment.

Adverting is a major application domain for public displays and is the principal business model behind most display networks. Large-scale networks of screens (thousands of displays) are typically owned by advertiser brokers such as JCDecaux[6], Ströer[7], ClearChannel[8], which sell advertising space to a wide range of customers.

Another major application area for digital displays is the information boards. These can be seen in airports, public offices (Figure 1). Besides showing travel information, displays may present the order status of persons to take turn as is the case with waiting rooms systems at various public services in city councils. While in these two examples, the displays are simple information systems, there are also many public displays that support interaction. These systems are usually classified as kiosks where people can search for information

[6] http://www.jcdecaux.com/ Accessed September 26, 2014
[7] http://www.stroeer.com/ Accessed September 26, 2014
[8] http://clearchanneloutdoor.com/ Accessed September 26, 2014

using touch-enabled screens. They are mostly encountered in tourist offices or hotels where the information is rich serving various purposes.

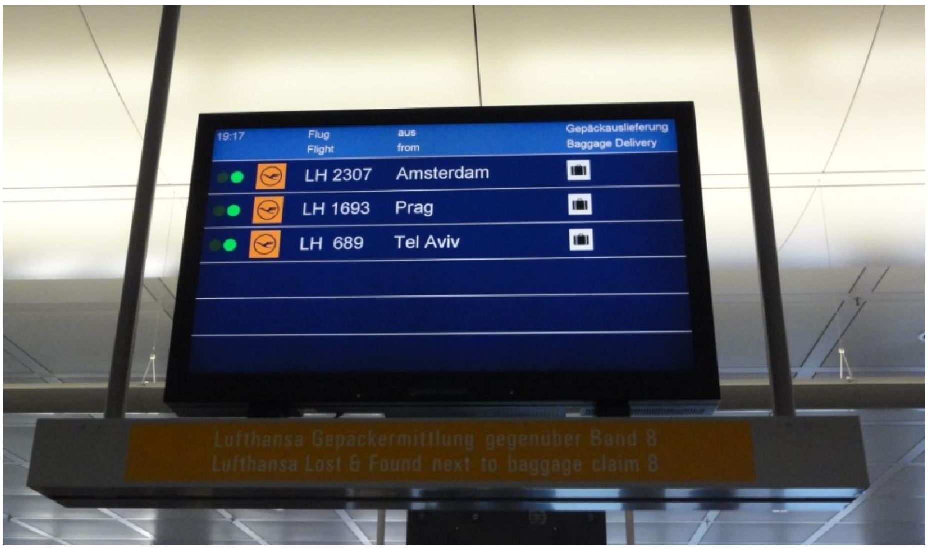


Figure 1: Airport information display (By Mattes, Own work, Public domain, via Wikimedia Commons)

Regarding the application area of digital signs (signage), the classical usage of displays includes identification, navigation and warning. For example, street signs such as speed limits are replaced by digital versions that facilitate a dynamic management of the traffic flow through variable speed limits.

Figure 2: Variable speed limits digital signs (By GilPe, Own work, CC-BY-SA-3.0, http://creativecommons.org/licenses/by-sa/3.0, via Wikimedia Commons)

Finally, large scale digital displays (display walls) have a wide range of uses in the field of art and entertainment. One notable example is BBC Big Screens (Thomson 2014). The infrastructure features twenty two of 25-square-meter outdoor displays deployed in UK (Figure 3), including Belfast, Plymouth, Norwich and Edinburgh and other cities. At the moment, displays are operated only by the city councils and show a wide range of content including preprogrammed BBC television channels, live broadcasts and information from non-commercial organizations and local communities. The screens are known for a number of competitions to enable artists to create content specifically for them. They are also popular during large sporting events because large numbers of people can gather and watch them live. In addition, the screens are also known for interactive games that used the movements of individuals or groups of people as input. The persons' movements were detected using a combination of cameras that are installed on top of the displays.

Figure 3: King Edward Street with BBC screen, (By Nicholas Mutton, CC-BY-SA-2.0, http://creativecommons.org/licenses/by-sa/2.0, via Wikimedia Commons)

Towards the end, to better understand the case of digital signage systems, we provide three example state of the art digital signage solutions that can be used to build some of the types of systems we have just explored. We also indicate an international leading company that provides a platform for incorporating place-based social media with interactive digital displays.

Ubisign is a Portuguese private software company focused to offer state of the art digital signage solutions (Ubisign 2014). It provides an advanced and complete cloud service offering all the software tools needed to create and manage Digital Signage networks in a friendly cost effective environment. Ubisign solution allows from a central location the management, scheduling of multimedia content and distribution across a Digital Signage Network of one to thousands of display screens located across the world. Being offered as a Software as a Service (SaaS) business model, Ubisign eliminates also the hassle of running a dedicated IT infrastructure. The company uses professional providers to safely host at a central database, all media, playlists and schedule information. In order to use the services provided by Ubisign, a display owner would not need to install an application in a PC. The company offers a web service that can be accessed within a web browser, with a very friendly, rich, full-featured user interface, enabling the design of fully customizable screen layouts and the scheduling of all major types of media content as well as dynamic data driven by external sources like RSS/XML. It also includes support for popular web 2.0 services like Twitter, Flickr or YouTube as dynamic content providers. Ubisign includes application deployments in various venues such as stores, cafes, hotels, waiting rooms, and offer a diverse set of use cases including media-player applications and interactive location-based services where customers can search, select, view, post and share digital content, creating a much richer consumer-experience. The application domains covered by Ubisign range from retail, tourism, education, corporate and exhibition places, for companies that wish to achieve higher brand awareness and engage with their target audience.

Other two examples that are global leaders in digital signage solutions are Scala (Scala 2014) and BroadSign (BroadSign 2014). Such solutions can scale for hundreds of thousands of displays. Both solutions provide tools for content design and management using the same Software as a Service business model. Their content players support a wide range of media types including audio, media streams, and content for interactive touch-based screens. Their content managers support playlist generation, remote player monitoring and provision for emergency alerts. Scala powers the digital communications of companies like IKEA, Shell, Ericsson and Mercedes-Benz whilst BroadSign solution is used by JCDecaux for its new installations since 2011[9].

LocaModa is a place-based social media company that provides a software platform that facilitate the integration between social media and interactive large display applications (LocaModa 2014). The platform aggregates, curates and filters content from Twitter, Instagram, Facebook, Email, SMS or custom sources as a way of building value inside of digital place-based advertising networks and signage systems. Digital signage networks use LocaModa to turn their media communication into interactive dialogues, as well as monetize this value in the form of LocaModa advertising revenue. The company creates, produces and manages multi-channel marketing programs for brands and agencies. As a leader in multi-screen social integration, LocaModa anchor a significant part of its value proposition in the ability to support clients to engage users through cross-channel solutions and amplify their audience online, at live events, across the social web and even on the world's most recognizable digital billboards, e.g., Times Square[10].

2.1.1 Analysis

In Table 4 we highlight the possible behaviors of display signage systems. In our analysis we looked mainly to understand the operation behavior, content type and its origin.

Table 4: Analysis of different types of digital signage systems

<table>
<tr><th>Usage Domain</th><th>Behavior</th><th>Content Origin</th></tr>
<tr><td>Advertising</td><td>Broadcasting only advertisements. Display owners give control of the screens to advertisers.</td><td rowspan="4">System-delivered</td></tr>
<tr><td rowspan="2">Information presentation</td><td>Information dissemination serving various purposes, e.g., traveling</td></tr>
<tr><td>Kiosks functionality. Displays have two modes: broadcasting information and content browsing.</td></tr>
<tr><td>Signage</td><td>Replacement of analog signs by using (mainly) displays with low resolution. The information can be critical.</td></tr>
<tr><td>Art and entertainment</td><td>Entertain people through the usage of various interaction techniques. Displays have (usually) large form factors</td><td>System-delivered
User-generated</td></tr>
</table>

[9] http:/www.dailydooh.com/archives/53987 Accessed September 26, 2014
[10] https://www.youtube.com/watch?v=1Imd4D3_1lg Accessed September 26, 2014

Overall, digital signage systems are concerned to disseminate information to audiences, complementing mass media channels, e.g., television, radio and Internet. Although, the content employed in these installations could be targeted for specific communities and offer interaction capabilities, most of the systems do no offer strong consideration for their audiences. Opening these closed display networks to viewers by employing various interactive third-party display applications could become a source of value, which in turn may lead public displays to be more appreciated in their environment and capture more interests (at least for the advertising application domain). In consequence, new business models are envisioned in which applications can manage content by themselves and allow third-parties to create novel usage scenarios of such installations.

2.2 Single Concept Displays

In this section, we look at some of the most representative display research prototypes (15 systems) that were designed based on a specific theme. We call such systems as single concept displays, as each display is dedicated to a single service or application. Although some systems employ the notion of application and embed different software applications to enable different use cases of the display, they all serve the same concept and the system itself does not have support for allowing other applications, e.g., offering application browsing capabilities. While research on these dedicated displays is very rich, we just provide a reference set of systems covering multiple concepts and a diversity of behaviors. We describe the systems by exposing the benefits for their users and also highlighting the research objectives behind them.

2.2.1 Notification Collage

A seminal work on awareness systems is Notification Collage (Greenberg et al. 2001). The system is a bulletin board display that was designed to study how technology can support interpersonal awareness and interaction within small communities of colleagues. The system allows people to post potentially interesting information to others or share their work or topics of interest increasing awareness of co-workers’ activities.

The content employed by Notification Collage is displayed onto a large display installed in a semi-public setting where all members could see such as a meeting area. As well, people can access the system though their personal computers. The content is shown in a collage style in form of various media elements such as *sticky notes*, *videos*, *live video streams*, *desktop views*, *slideshows*, *webpages* and *activity indicators*. In this deployment, users play the role of content publishers, thus releasing the system for any system specific scheduling or automatic content generation tasks. New content items are randomly placed in the left side of the screen overlapping the old ones. Figure 4 shows a snapshot of Notification Collage system.

While only partially supported by other collaboration tools, user experiences with Notification Collage showed that people used the system as an awareness tool, an instant messenger, a media space, a MOO (multi-user text-based online virtual reality system), an asynchronous bulletin board, and as a public community board (Figure 4).

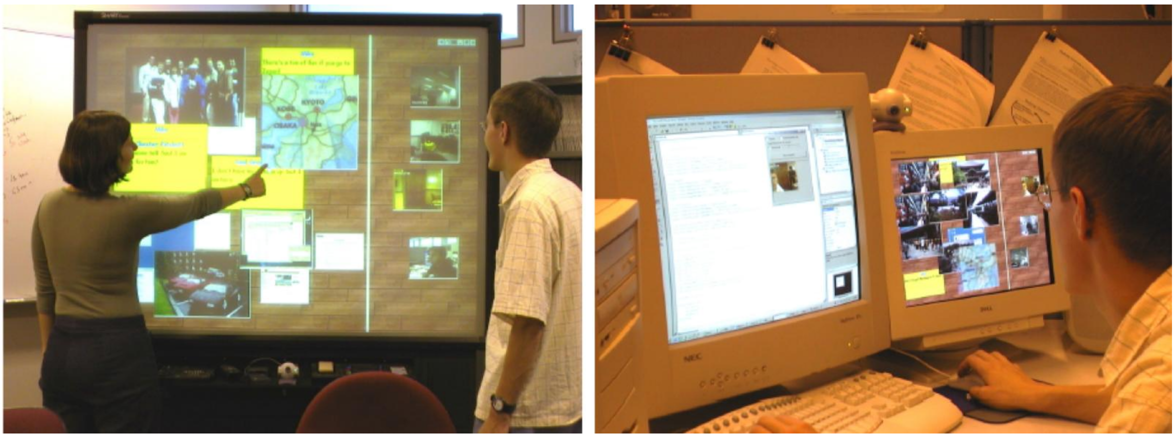


Figure 4: The Notification Collage system: left – public display; right – PC (Greenberg et al. 2001)

2.2.2 UniCast, OutCast & GroupCast

UniCast, OutCast & GroupCast systems (McCarthy et al. 2001) implemented three distinct content sharing mechanisms according to the type of space in which the systems were deployed. UniCast system was designed for personal usage within individual offices; OutCast system was designed for being used outside an individual office by potential co-workers and GroupCast system was designed for a common area where people could meet for occasional conversations. The research focus behind these systems was to understand what type of content users like to publish and how to personalize content in order to improve social awareness and foster the idea of a community.

UniCast application was designed as an individual content sharing mechanism that allows users to specify content (that is not important or critical) they would like to see on peripheral or ambient displays located within their primary workspaces. Using a profile, the user can specify via a web page the content modules to be displayed (15 classes of content modules in total), e.g., web pages, remainders, weather etc. By using the infrared badge system the application has the ability to detect if the user is in front of the computer or is away, which allows the content to be adjusted according to the user presence. UniCast enables people to control the content items displayed, e.g., pause, resume, back. User experiences with UniCast application revealed that all users run the application on a display that is peripheral to their primary workstation display. While the most popular modules among the users that were interviewed were the Web Page, Weather, Factoids, WebCams and the infrared badge-based In/Out List ("ActiveMap"), the least popular modules were those for Traffic and Reminders.

OutCast application was designed in order to publish personal information that is not only of interest to the owner, but also for others to view such as articles, cartoons, photographs, calendars, to the co-workers in the office environment using a display installed near the owner office door. Although UniCast application provides users minimal interaction with the content, the OutCast application exhibits two distinct behaviors: passive mode and active mode. While the passive mode is similar to UniCast, in active mode, the users can even navigate through and explore each module's content and even input text messages for the display or office owner. Informal feedback from OutCast users showed that people typically used the application when the owner (office occupant) was away. While the most liked features were the Location Information (finding owner's location when out of the office) and Calendar (finding an available time slot for a meeting), the least liked and used was the Text Message feature, because people were uncertain about the reliability of this feature and preferred using paper based Post It notes.

GroupCast application enables social interaction between passers-by, based on topics of interest according to a user's profile at the time. The application focus is to improve people awareness by informal interaction opportunities. GroupCast uses a large public display installed in a hallway with high people traffic and displays personalized content through the badge location system and users' profiles created for UniCast.

2.2.3 The WebWall

WebWall (Ferscha & Vogl 2002) is a media and service independent framework conceived to support a seamless WWW access in public settings, using large displays and mobile phones or handheld devices. The research goal was to investigate the usage of large, shared displays as novel and multi-user communication artifacts that stimulate social awareness and interaction. In particular, the focus is to understand the usage of visual displays in public spaces to allow for a ubiquitous WWW access for a broad, loosely related, non-determined and unstructured audience.

According to different classes of services, this work addressed the possibility of ad-hoc communication in public space based on a wall metaphor in which users are able to publish and retrieve various WWW media elements, such as, simple sticky notes, opinion polls, auctions, image and video galleries and even controlling the public display web browsing through mobile devices. WebWall system supports two interaction modes: direct manipulation and through a web-interface. Examples of services are: Notes, Gallery, Polls, Auction and Banner.

2.2.4 Semi-Public Displays

Huang et al. (Huang & Mynatt 2003) studied the benefits of an awareness system, i.e., Semi-Public Displays, which displays information within small, co-located groups, aiming to promote coordination, collaboration and to provide lightweight information about group activities. The academic context, in which the system is deployed, is characterized by members who already have some awareness of each other's activities.

The *Semi-Public Display* system displays four applications to support and enhance the interactions and information that group members utilize to maintain awareness and collaborate: *Reminders*, *Collaboration Space*, *Active Portrait* and *Attendance Panel* (Figure 5). These applications focused to improve the current collaboration methods by offering viewers a centralized source of relevant information in a more accessible way. The entire display system aimed at reducing the effort necessary for gathering this information through the current channels of email, instant messaging, calendars, and word-of-mouth.

The *Reminders* application aimed to easily remind the members' helping requests by posting them on a shared display. The content associated to these requests was retrieved from the users' email accounts. The *Collaboration Space* application aimed to provide a dynamic, captured space, in which users can edit comments per request basis. This type of application was envisioned for immediate and asynchronous collaboration. The *Active Portrait* application aimed to provide easy access to information about presences and recent activities in the lab space. The application offers low-fidelity presence information based on a group picture. The *Attendance Panel* application was conceived to show an abstract view of the popularity of upcoming events in a way that preserves the users' privacy.

Results from evaluation of *Semi-Public Display* system confirmed the initial perspective about the potential and effectiveness of the informative awareness applications designed to support the needs of small, co-located groups. In particular, this study highlighted that public display applications can be greatly beneficial within the context of small, co-located workgroups with shared interests, without incurring many of the difficulties of such applications when deployed within larger, less-connected groups.

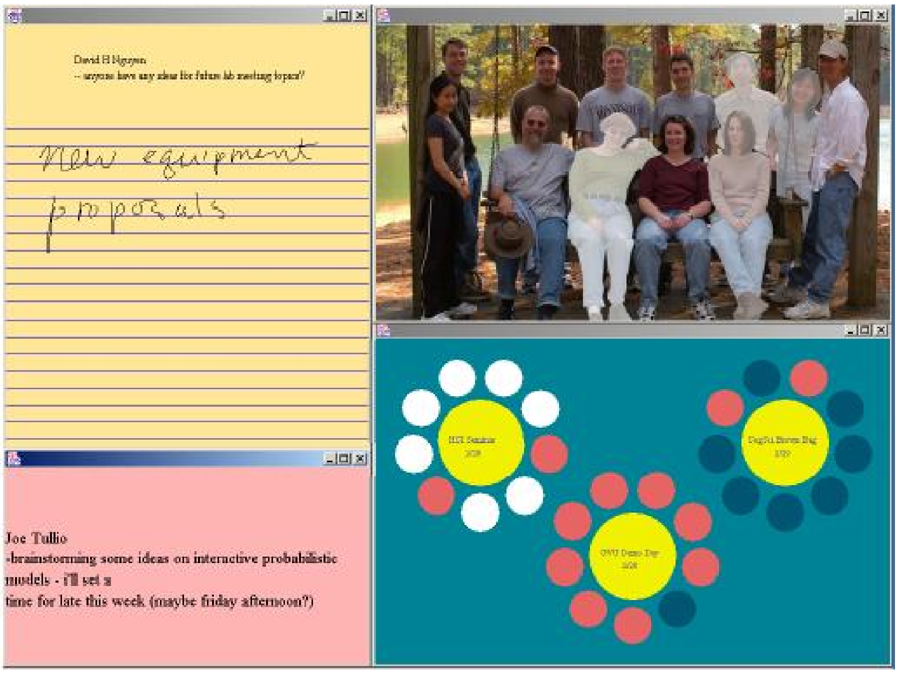


Figure 5: The Semi-Public Display prototype; clockwise from top left
a) Collaboration Space; b) Active Portrait; c) Attendance Panel; d) Reminders
(Huang & Mynatt 2003)

2.2.5 The Plasma Poster Network

The Plasma Poster Network (Churchill et al. 2003) is an infrastructure of three large screens and interactive digital bulletin boards. It was designed to encourage casual, unplanned, social interactions within organizations or semi-public context such as research labs.

The system addresses the gap between online asynchronous, community-based content sharing using personal devices and sharing of content in physical spaces using public displays. Displays are deployed in a portrait format and support direct touch interaction. The system software architecture is conceived as a content storage and distribution mechanism that supports content posting to all registered displays. The content employed comes from two sources: individually posted content by authenticated users through the use of email or a Web page (text, images and URLs), and content that is automatically sampled from the intranet such as announcing new technical reports and calendars of meetings. A content item may last for a certain time that is manually configured within each user's personal profile. As well, one user's profile may act as a history of old postings, which can be easily redisplayed.

The Plasma Poster Network employs four forms for the engagement with content: 1) *peripheral noticing* when the content is visually appealing and dynamic, and the system cycles through various posted items at an interval of 60s with one piece at the time; 2) *active reading* when the content can be paused, scrolled and printed; 3) *navigating and browsing* when the content can be retrieved from the database through content maps and overviews; and 4) *messaging* when the content items can be forwarded to others or the respective content' author can be emailed (Figure 6).

The evaluation of the system showed that people post, read and interact with content and the displays became an everyday part of life within the lab. While people were active readers there were seldom messages and replies to the authors of the posts. The overall experiences with the displays revealed that the system is a valuable addition to existing content sharing methods, and a driver for the social encounters.

Figure 6: Plasma posters in three different locations (Churchill et al. 2003)

2.2.6 Dynamo

The research by Shahram et al. (Izadi et al. 2003) aimed at promoting a novel way of building social interactions within occasional meeting settings, e.g., hotel, by allowing a lightweight, visible and fluid collaboration between users.

This work introduced the design and development of Dynamo system – that is a communal multi-user interactive surface that facilitates the cooperative sharing an exchange of digital media content, e.g., documents, audio and video files. Dynamo provided a public interactive surface for shoulder to shoulder collaboration in unstructured setting. It did not required networked connectivity and users were expected to bring media to the display using USB memory sticks or laptops. The system employed interaction mechanisms based on a combination of keyboard and mouse attached to the display or using laptops running instances of the Dynamo software.

Dynamo was evaluated in various stages of the project. The assessment of the systems consisted in using a projected display involving 30 users at a workshop held in a hotel, a lab-based study using a projected display that involved three groups of four people, and finally a 10-day deployment of Dynamo into the communal room at a high school using two 50” plasma screens mounted side-by-side as display devices. The evaluation results from the high school environment (Brignull et al. 2004), showed that the shared interactive surface was appreciated and used by students to display, share and exchange a wide variety of media. As well, people used the system for familiar activities, like sharing and exchanging information, spending time on their own and socializing with friends, in novel ways and with different media.

2.2.7 IM Here

Huang et al. (Huang et al. 2004) explored the benefits of Instant Messaging (IM) into a publicly accessible groupware display system. The aim of this work was to discover new value by breaking the paradigm of personally-owned IM and introducing IM as a shared public resource within other places where work occurs. The study presented the design and development of the IM Here system – an awareness and communication tool that uses IM type of communication for lightweight interaction within members of a group and large-scale displays that enable walk-up-and-use outside the personal working environments.

The IM Here system was deployed near the entrance of a conference room that was frequently used for formal and informal meetings. It included two components: The IM Here Event Display provides lightweight information about upcoming events and announcements; the IM Here Messaging Client allows users to send instant messages to workgroup members from the display in a walk-up-and-use fashion. The Event Display presented graphical and text postings regarding announcements or events on a continuous cycle with 25 seconds for each item. The content posted to the Event Display was generated using a web-based interface accessed on a personal machine. When the system was in the passive state or the messaging client was not used, besides the events announcements, the display showed the group member statuses, e.g., available, not available or busy. In the active state, through standard keyboard and mouse interaction, users could send messages both to individuals and all the members at once.

The system was evaluated during the first six weeks of deployment and the results showed its use within the workgroup type of environment. Given the size of the workgroup and the context of use, the system was successfully incorporated into everyday work activities. People opinions were generally positive regarding the both component of the system and they found IM Here to be convenient and useful, especially because of its proximity to the common meeting room.

2.2.8 UniVote

UniVote display prototype (Day et al. 2007) is a voting system developed in Lancaster University Campus, UK that explored the concepts of user engagement and human connectedness. The system aimed at eliciting user involvement and making students in the campus aware about the campus and world-wide events and news and thus breaking the concept of small isolated information islands, i.e., breaking the campus bubble.

The system uses mobile phones in order to enable the interaction with the large public display. It consisted in a web administration site in which the university staff can have access and post questions with the respective answers and a Java mobile client interface that can be downloaded to a Bluetooth enabled phone through which students could vote.

The system was not deployed in the wild, but it was assessed through a lab-based evaluation session and proved to successfully engage the users for interaction and human connectedness and the preliminary findings suggest that the campus "bubble" can indeed be broken (Figure 7).

Figure 7: UniVote display system (Day et al. 2007)

2.2.9 Context-Sensitive Public Display

The work by Morales-Aranda et al. (Morales-Aranda & Mayora-Ibarra 2007) describes a context-sensitive public display prototype that manages the presentation of personal and public information. The research aimed at providing a model for adapting display content according to contextual information such as the identity, location and temporal utilization of the display.

The system includes capabilities of personal and public content visualization for two users through self-configured layouts and hierarchical management of privacy-sensitive information. It was evaluated and tested in a home environment with two persons belonging to a family: mother and child. In supporting the adaptation mechanism, both users had to create a personal profile in which each of them configured what information is public and what is private. Therefore, the private information is shown only when one user, e.g., mother, approaches the display. If the other user, e.g., child, approaches the user that is in the proximity of the display, e.g., mother, the screen layout splits in three sections thus allowing both users to visualize shared information and each ones' personal content as well. When two users are approaching the display as seen before, the system hides private information automatically.

There are not many insights from the evaluation of the system and the results obtained so far inform the process of designing future usage scenarios and optimizing the user interface. Still, this work has limitations in supporting more than 3 users using the display in in order to have a usable system and in handling large number of information items simultaneously in the layout.

2.2.10 PEACH Display System

Stock et al. (Stock et al. 2007) described the main achievements of the PEACH project (Personal Experience with Active Cultural Heritage) regarding the development of adaptive intelligent user interfaces in museums for individual experiences. The project aimed to produce a multifaceted system that accompanies people and augments their overall visiting experience.

The system integrates seamlessly mobile devices (PDA) and stationary large screens and is able to support various presentation exchange sessions, e.g., users can request further information about a visited site through the usage of the fixed large display. Overall, the system is initialized with a user profile and in the course of a visit, adapts to the behavior of the visitor, proposing personalized, context-dependent presentations. Designed as a full-featured system deployed in a mobile setting, PEACH embodies built-in location detection, user modeling and adaptation mechanisms. Regarding adaptability, the system keeps track of implicit and explicit user interaction, e.g., user position, orientation, gaze and topic of interest (users could let feedback by liking or disliking the current piece of the presentation, with consequences for subsequent presentations). The system allows users to take away a personalized written report that summarizes the key aspect of the overall visit experience.

Although the system was not evaluated in a fully integrated version, the ideas underlying the many PEACH components were assessed through diverse user studies. One study showed that even older people are comfortable interacting with a major component of the system (Figure 8).

Figure 8: A museum site (left); b) stationary screen with welcoming characters (middle); mobile guide (PDA)(right) (Stock et al. 2007)

2.2.11 StoryBank

The StoryBank project (Jones et al. 2008) explored the uses of technologies within digitally poor communities by studying user-generated content practices. The goal of this work was to understand the impact of and improve the design of a situated display by evaluation and refinement of the system within a rural community context.

In this work, given the context of digitally impoverished communities, the system was conceived as a community resource (not personal) using a community-centred design approach rather than a user-centred one. The solution includes community-based mobile phones and a touch-screen display situated in the village’s community resources centre. Using Bluetooth-enabled devices, users can send and retrieve media content in form of archived audio and video programs from the community broadcasters, images, and audio-visual stories created by the villagers themselves.

The display was deployed in an Indian village under two system versions, which corresponded to different stages of the project. While the first version represented a system probe deployed to raise specific design issues according to the deploying site, the second one i.e., extended version, was the refined approach, which constituted the sole and main functional system for the community, as well the subject of further investigations.

2.2.12 Proactive Displays

The Proactive Displays system (McDonald et al. 2008) was designed to investigate the potential of large displays and sensing technology to augment fluid social settings such as a conference, in order to enhance the action and interaction of the participants.

The deployment involves three proactive large displays that can sense and respond to the physical presence of one or more people. The proactive behavior was implemented through web-based user profiles and associated RFID tag. When participants were near a proactive display, the system revealed information about their background and interests. The system employed three applications designed to augment an academic conference setting, within the formal paper sessions and informal breaks. The proposed applications are targeting multiple interaction patterns, such as one-to-one, one-to-many and many-to-many. As well, the deployment was characterized by the usage of an implicit interaction model, eliminating the direct manipulation of displays. In the following, we shortly describe the applications.

AutoSpeakerID application was conceived to augment the current practice of providing the name and the organizational affiliation of the persons asking questions during the conference paper presenting sessions. Instead of saying those details to the microphone, a secondary large display was showing the questioner name and affiliation during questioning. In this way the conference attendees (or audience) could easily understand the background of the person asking a question and as well facilitating further follow up.

TicketToTalk application was designed to facilitate potential discussion among conference attendees during an informal break, i.e., coffee breaks. One of the coffee physical space was augmented with a proactive display that showed professional interests, vacation pictures or book images, whenever the respective users approaches the coffee tables. Participating in this type of activity, people could foster social awareness and share common interests.

Neighborhood Window application installed in a side part of a lounge area, exploits the many-to-many interaction relationship, and it was designed to reach and promote group discussions. The application shows people and terms that are visually connected – culled from the users' homepages. This approach fostered conversations based on shared and unique interests among conference attendees.

All of the three applications were systematically evaluated regarding their social impact within the conference setting. While people reported about their increased awareness and interaction opportunities during the conference, they also identified distracting aspects of the proposed augmentation. Overall, the study revealed that social engagement can be positively or negatively influenced by the interaction design (control), the setting in which technologies are deployed (context) and the content on display (content). As a consequence, these three factors Control, Context, and Content, are critical to how these technologies impact the social environments and whether they are ultimately accepted or rejected.

2.2.13 Bluetooth Presence and Naming

José et al. (José et al. 2008) studied situated interaction around public displays as enabled by the use of Bluetooth presence and naming. Instant Places system served as an infrastructure for generating on a public display, the situation relevant content that can be implicitly and explicitly derived from Bluetooth presence. The implicit behavior was that users' name can be shown on the large display when their Bluetooth mobile phones were enabled. An explicit behavior was when the users were able to change their device name in response to the name already shown on the screen. By embedding specific commands in the device name, e.g., "my device flk:JohnSmith", users could request content, such as photos from Flickr.

The system has been deployed in a University Campus Bar for several weeks and the approach proved to be an effective way in sustaining the situated interaction around public display and enabled users for new forms of social practices such as using the display as a board for posting messages about the service or to other people in the room.

2.2.14 BiebBeep

BiebBeep is a large touchscreen that was conceived to augment the information and social function of a library (Kanis et al. 2012)(Veenstra 2011). It shows user-generated and context-specific information, such as the latest additions to the library collection, local news, facts and figures of (cultural) events and activities happening in the library and Almere region (in Netherlands). People can add information to the screen themselves, such as tweets and Flickr photos, so that the library and its visitors can inform and connect with each other. Aligned with the concept of Library 2.0 (Casey & Savastinuk 2006), the system drives its potential from the combination of interactive displays and social media such as Twitter, Flickr and YouTube, Figure 9.

The display in portrait orientation with 42" is mounted near the entrance and shows information objects that are obtained from the Internet, such as RSS news feeds and Tweets, which flow slowly in vertical direction. People can touch any content item to have a closer look. Library members create content by uploading information via Twitter (by using the tag @schermalmere in their Tweet), images via Flickr and movies via YouTube (by using specified tags such as schermalmere). As well, library employees can upload extra information via a custom made web-based Content Management System (CMS) tool.

BiebBeep was deployed and studied within the library context for more than one year. Based on a living lab approach, the design included several methods including a focus group with library members and several in-situ user studies. The evaluation of the system revealed that both users and staff considered BiebBeep a valuable addition as it enhanced the library information services and stimulated social interactions. The observations and interaction logs indicated that the most watched items on the display were the pictures and videos rather than text-based content. Regarding the social interaction, BiebBeep mediated discussion mainly between people who were already familiar to each other. Additionally, the Twitter functionality seemed to have potential of bringing people of the diverse Almere community together, which would normally not often engage or come in contact with each other.

Figure 9: Biebbeep display system (Kanis et al. 2012)

2.2.15 Proxemic Peddler

Proxemic Peddler (Wang et al. 2012) is a 52” multi-touch advertising display system that is able to tune its content in response to how passersby are attending them. It is based on the Peddler framework that captures fine-grained continuous proxemics measures by monitoring the passerby’s distance and orientation with respect to the screen at all times. Based on this information the system is able to infer passerby’s interest or digression of attention at any given time, and their attentional state with respect to their short-term interaction history over time. Finally, the attentional state is used to tune content to lead the passerby into a more attentive stage, ultimately resulting in a purchase (Figure 10).

The system was conceived for advertising a variety of products from a large online marketplace, where purchases can be made directly from it if desired. Proxemic Peddler uses book related materials from Amazon.com to investigate how books could be marketed in such a system. In their current implementation, the system tracks three important proxemic variables: 1) the person’s identity for showing products that matches users’ interests; 2) the position of the person in front of the display, and (3) the person’s orientation that shows the person’s direction of view. The implementation details is handled by Proximity Toolkit (Marquardt et al. 2011), which uses the Vicon motion capture system and provides both absolute position and orientation of people, the relative distance between the person and the display, and the orientation of a person’s head towards the display. The Proxemic Peddler prototype is limited to detecting one person at a time and it requires those people to wear markers. The system’ software was written in C# and Microsoft’s Windows

Presentation Foundation (WPF) framework, and runs on a computer attached to the display. The Proximity Toolkit sends events through socket connections.

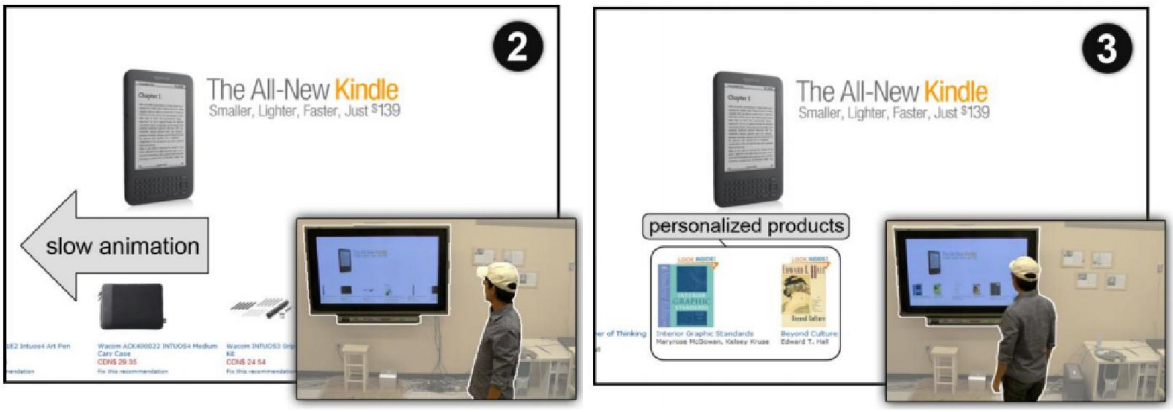


Figure 10: Snapshots from the interaction model enabled by Proxemic Peddler system; in gray boxes are the system actions in response to user's movement (Wang et al. 2012)

2.2.16 Analysis

Our analysis involved the examination of four main characteristics of these systems: the concept or main theme of the system, content origin, audience (viewers/users) and type of environment. In Table 5 we provide an overview of all the systems explored highlighting their concepts and individual behaviors.

Being in public spaces, outdoor/indoor or in a controlled setting such as a laboratory the various display systems that employed a single concept, informed researchers about their usefulness for the respective users. Comparing to digital signage systems, which largely ignored the viewers, the research prototypes do offer consideration for their audiences allowing them to engage in various forms. For instance, people can browse information, comment on that and even provide new content items. Displays are seen as information portals that dynamically change their behavior according to the local environment and people's preferences and they do not anymore employ static content playlists that are well-known before presentation and cannot be changed at playback time. Instead, displays adapts their content behavior based on users' interaction with them and the control over the content is decentralized allowing viewers to better engage with the overall system. However, the content in these research prototypes is described in the same way as encountered in traditional digital systems, e.g., graphical (pictures, videos) and textual data (emails, web pages).

This state of the art in public display research corresponds to a specific phase when display technology became more accessible as the costs decreased and researchers were able to explore its potential for particular purposes (usually a single theme), including deployments in public and semi-public spaces (Davies, Clinch, et al. 2014). While certain systems mention the notion of software applications as a mechanism to associate additional behaviors to specific content items, e.g., (Huang & Mynatt 2003), the overall focus of the system is mainly to serve one concept. These applications are very integrated with the system itself and there is not support for allowing other applications serving different themes on the same display. These systems were thus focus more on their utility towards one direction, i.e., dedicated displays.

2.2 Single Concept Displays

Table 5: Analysis overview of single concept displays

System	Concept	Behavior				
		Content Origin	User		Type of Place
			Who is the User?	Role	
Notification Collage	Awareness and collaboration	User-generated	Individual Group	Active Passive	Research lab
UniCast OutCast GroupCast	Social interaction and community	System-delivered	Individual Group	Active Passive	Hallway
The Web Wall	Social interaction and community	System-delivered User-generated	Group	Active	Public space
Semi-Public Displays	Awareness and collaboration	System-delivered User-generated	Group	Active Passive	Research lab
The Plasma Poster Network	Social interaction and community	System-delivered User-generated	Group	Active Passive	Research lab
Dynamo	Awareness and collaboration	User-generated	Group	Active	Occasional public meeting setting
IM Here	Awareness and collaboration	System-delivered User-generated	Individual	Active Passive	Near an entrance of a conference room
UniVote	Social interaction and community	System-delivered	Group	Active	Lab
Context Sensitive Public Display	Intelligent content presentation	System-delivered	2 People	Passive	Home environment
PEACH Display	Intelligent content presentation	System-delivered	Individual	Active	Museum
StoryBank	Social interaction and community	User-generated	Group	Active	Public space
Proactive Displays	Intelligent content presentation	System-delivered	Individual Group	Passive	Lounge area at conference
Bluetooth Presence and Naming	Social interaction and community	User-generated	Group	Active Passive	Café
BiebBeep	Social interaction and community	System-delivered User-generated	Individual Group	Active Passive	Library
Proxemic Peddler	Intelligent content presentation	System-delivered	Individual	Active	Lab

In our work, we consider that a key enabler for Open Display Networks would be the emergence of third-party applications development, which can transform these dedicated displays into general purpose computing artifacts. The plenty of concepts, use cases and behaviors we just explored are references to the many application classes and goals that can be exploited by independent or third-party developers to create meaningful experiences in which viewers would have more than a single choice. The main usage of this analysis is to understand and structure the design space around single concept displays that may inform the definition of possible design principles that third-party display application developers could benefit for.

2.3 Display Systems with Multi-Application Support

In this section, we describe display prototypes and real-world display deployments that anchor a significant part of their value proposition into the ability to provide people with many use cases or application alternatives. Rather than offering integrated and related application experiences as is the case with single concept displays, multi-application displays offer distinct, unrelated or weakly coupled interactive application experiences where the system behavior can be extended thus embedding various usage scenarios associated with different types of content.

2.3.1 iRoom

The interactive workspaces project (Johanson et al. 2002), i.e., iRoom, looked at new possibilities for enabling collaboration in technology-rich spaces equipped with several shared displays. This work aimed to find an approach on how to map a single defined physical location to an underlying system infrastructure and to a corresponding interaction model. The project stressed out the use of large interactive walk-up displays (some of them using touch interaction) while researching for real world applications atop iROS interactive workspace software infrastructure.

The iRoom computing infrastructure employed a varied set of usage models in which a set of local GUI applications could exchange data to achieve an integrated behavior across a multi-user, multi-application and multi-device environment. Examples range from moving data among the various visualization applications that run on screens in the room, and laptops or PDAs from the respective environment, moving control in which any user should be able to control any device or application from their current location and dynamic application coordination where applications that display data should coordinate with others for achieving an integrated behavior towards realizing a common goal.

The research conducted as part of the iRoom infrastructure is relevant for our work as it provides the theoretical background for the many general principles, computing metaphors and programing models employed in ubiquitous computing environments. This includes the boundary principle (Kindberg & Fox 2002) which states that ubiquitous computing infrastructure needs to frame the interaction between devices only within the bounds of the local physical space, in our case a place where a certain display is installed. Other general principles are related to: make application components interdependent, which in turn makes the entire system more robust, e.g., applications do not communicate directly with one another, but use indirection through the Event Heap; modular restartability that isolates any system failures so that applications errors do not propagate at bottom layers, e.g., any applications data may be stored in Data Heap, which is a storage for the local environment; leveraging the Web due to its popularity by utilizing browsers and the HTTP protocol.

2.3.2 e-Campus

The e-Campus is long-running public display research testbed at Lancaster University campus (Friday et al. 2012). The infrastructure was conceived as a "laboratory" for local researchers and artists, and it supports and encourages multi-disciplinary research in the fields of mobile and ubiquitous computing. The e-Campus testbed includes different types of displays such as LCD screens, small office signs and large projector installations. E-Campus has been fully operational for 9 years and at the moment, the main usage of the infrastructure (counting 30 displays), is for digital signage an emergency alert distribution. From the outset in 2004, the e-Campus system was used in numerous research projects exploring novel applications including interaction between mobile devices and public displays, sensing and accommodation of new content types. The main important research areas of the e-Campus infrastructure included system infrastructure and APIs for signage development (Storz, Friday & Davies 2006)(Storz, Friday, Davies, et al. 2006), user interfaces for decentralized content creation (Clinch et al. 2011), personalization and interactive applications (Davies et al. 2009)(Davies, Langheinrich, et al. 2014).

2.3.2.1 Signage functionality

The e-Campus LCD displays are not touch-based and cannot be directly reached. These displays use Mac Mini computers as media players. Regarding the software infrastructure, the e-Campus system includes an API for programming networks of public displays. The API provides developers with basic operations to control content on networked public displays and supports the notion of visual transactions to group arbitrary displays to control them as a single, atomic unit. The API performs only the most basic functions for scheduling content and lacks any notion of interaction, screen location, or spatial relationships (Storz, Friday & Davies 2006).

2.3.2.2 The e-Channels system

The main usage of the infrastructure is as an experimental digital signage solution for distributed control of displays and content – including images, videos and URLs (Clinch et al. 2011). In their e-Channels approach, content creators or providers (who could be both display viewers and display owners) could opt for publishing content through the usage of logical containers called channels. The e-Channels system is in contrast with traditional digital signage approaches, in which the display owner is in the position of centrally managing the content scheduling process. Instead, it separates the roles of content providers and display owners by focusing on opening the display network in such a way that any users can publish content on their own or each other's displays. Content providers or producers can manage their channels and content independently of owning a display. Display owners can use a web interface to browse the available channels and subscribe their displays to content from those channels. While the content may come from multiple sources as represented by particular channels, display owners are the only entity to manage what channels should be broadcasted in their displays deciding when and where content will be displayed.

2.3.2.3 App store system

While the usage behavior enabled by e-Channels implementation largely considered displays as end-points for content distribution without featuring any interaction support (showing university-wide and location-specific content), the latest development of the infrastructure considered the vision of an application store for interactive display applications (Clinch, Davies, et al. 2012) where content and applications can be published by third parties. In their Mercury implementation of the application store (Clinch, Davies, Friday, et al. 2014), developers contribute with content in form of web-based applications, display owners select and manages their subscribed content and the store itself provides RESTful APIs that allow integration with a variety of third-party services.

Showing content on the screens is achieved using an in-house multi-platform player and scheduler called Yarely (Clinch et al. 2013). The player is a software component for media playback including images, videos, web content, remote desktop, which integrates day-to-day signage schedules from multiple sources to result in a single pool of content to be shown at the display. Yarely interweave traditional static scheduling of content based on various constraints (e.g. what time an item should be shown at) with dynamic approaches enabled by sensor-based events (e.g. the arrival of a specific viewer). The core engine of the player is based on so-called Content Descriptor Sets that describe content items and instructions on how and when they should be displayed. These Content Descriptor Sets represented by a XML file is a new approach in content scheduling that replaced the APIs from earlier testbed implementations (Storz, Friday & Davies 2006). They provided similar functionality without the need for application developers to maintain persistent connections to a display in order to issue commands. An overview of the latest e-Campus infrastructure is depicted in Figure 11. The entity who produces Content Descriptor Sets is called Descriptor Factory.

2.3.2.4 Bluetooth-enabled system

The e-Campus testbed have been explored for supporting a range of interactive experiences in which viewers could control and indicate their preferences by using mobile phones clients and Bluetooth device names (Figure 12)(Davies et al. 2009). These interactive applications include maps, web queries and requests for web-based content such as from Flicker or YouTube. The content provided by these interactive applications is prioritized in the system and it can stop the running program of ordinary day-to-day content that is shown by default in the display network. When requested the content is displayed in a specific region of the display and consist in a slideshow of pictures, a video, an application or a web page. The system can process multiple request for a single public display by queuing them in a way that prioritize the requests originating from phones that have been served least recently.

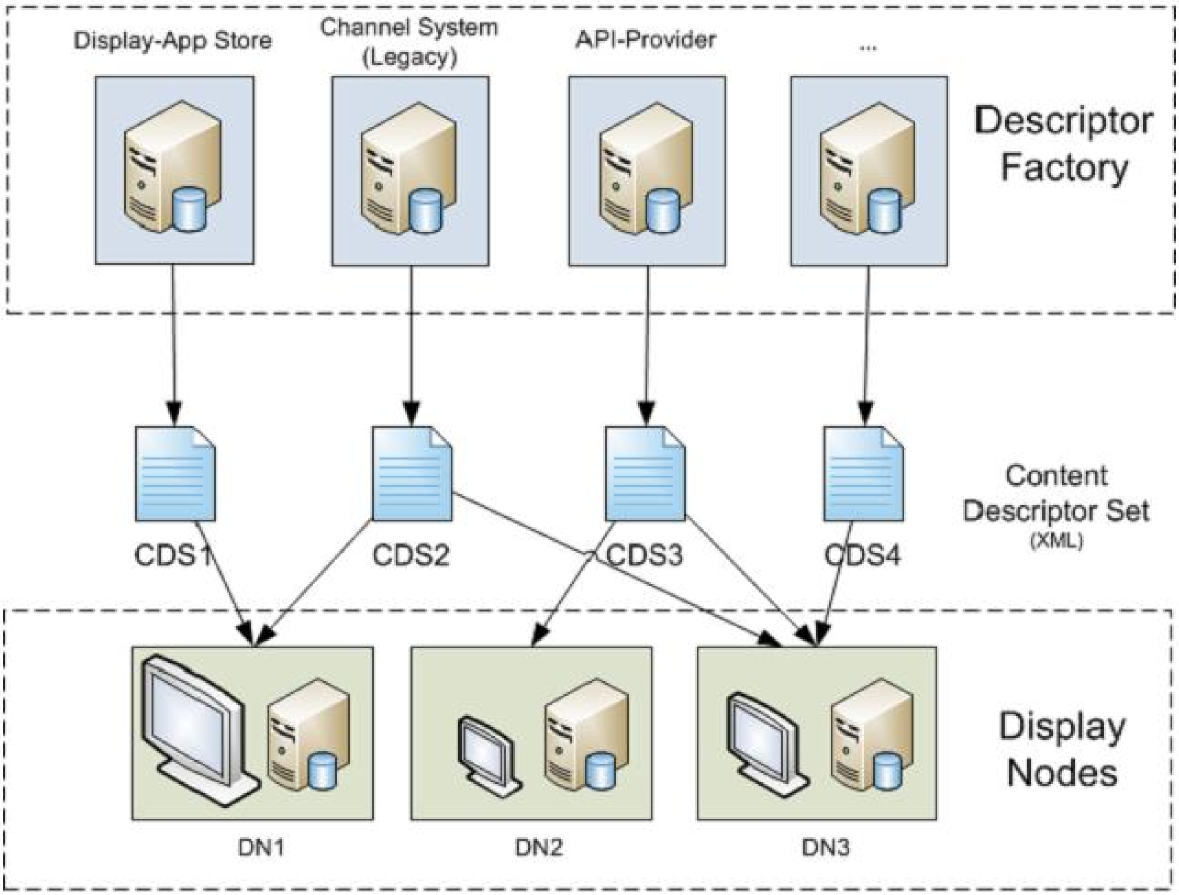


Figure 11: The e-Campus network infrastructure (Clinch et al. 2013)

2.3.2.5 Tacita system

A recent approach to personalize the display experience was conducted as part of Tacita system implementation (Davies, Langheinrich, et al. 2014). Tacita is based on a privacy-preserving architecture to allow display owners to open up their displays to content from passing viewers. By using an Android mobile client, viewers can browse for local displays and applications and send requests for screen real-estate. This approach leverage on the existing trust relationships between viewers and their cloud services as viewer requests are routed through the desired content providers themselves rather than broadcasted to displays. Similar to the Bluetooth-enabled interaction prototype, Tacita system uses a simple scheduling model in which user-requested content has a higher priority than scheduled content. In the case of multiple requests from different applications, the system splits the screen according to the number of simultaneous users.

Figure 12: Bluetooth-enabled interaction in e-Campus infrastructure (Davies et al. 2009)

2.3.3 Instant Places

Instant Places (José et al. 2013) is a display-centric platform for media sharing that handles sensing and interaction information associated with places where the displays are located and provides an integrated API from which subscribed applications can obtain information about the current circumstances around the display in which they are being used.

Instant Places bases its core design on three main concepts: *places*, *identities* that represent people and *applications* that generate content. A place in Instant Places is defined as a symbolic entity that connects one or more physical settings, providing a significant context for situated social interaction. Any information originated from places, e.g., sensing and interaction, is managed by Instant Places system, which supports related services needed in the development of situated applications. The applications running in various places are called place-based, and feature an adaptive behavior by conforming to the locally available resources and the current circumstances of a particular place. A place is thus an aggregator for people, interaction, resources and applications unifying them under a single coordination context. Every place specific practices and visiting patterns may constitute as a memory, which enables situatedness, i.e., that ability of an application to provide place-specific content. An identity in Instant Places embraces people representation that is built in a specific place, thus playing a major role in setting interaction expectation and context. Any identity may act as mask of a particular person that wants to interact with a display in a particular place. Applications are web-based software programs, which are published in the entire display network. The place owner is the responsible entity who selects, installs and configures applications as place-centric content generators, and permits applications to access the place information. An application can have multiple instances for the same place by handling distinct configuration settings.

The Instant Places core functionality is supported by a service platform composed by several web sites and various entry points. All the services depicted in Table 6 are taking part of a single integrated system and share the same design, look and feel and authentication mechanism.

Table 6: Instant Places service platform and the associated web sites

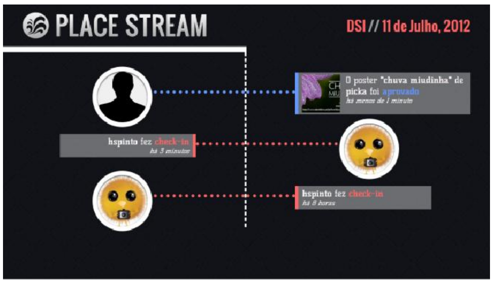
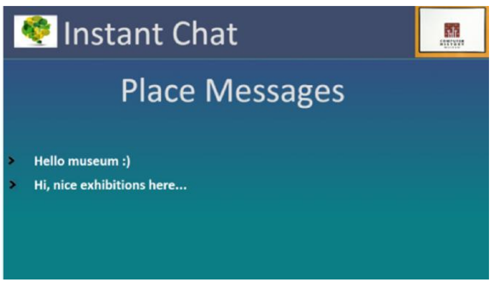
Service/URL	Description	Target users
Instant Places *www.instantplaces.org*	People can manage their identity presence in public places	Anyone
Instant Places API *api.instantplaces.org*	API for applications	Application developers
Places *places.instantplaces.org*	Manage places and applications subscriptions (including an experimental app store)	Place Owners
Developers *developers.instantplaces.org*	Registration and publication of applications for the system	Application developers

While for certain services the users are able to log in using an account that corresponds to an external Open Id provider, other services may require additional registration steps, e.g., application developer. Each service recipient has a concrete role in using the system, thus the Instant Places is offering appropriate views for them. Instant Places architecture is composed around four domains: the web-site infrastructure that represents the system core functionality, e.g., identity and place management, in-place sensing and interaction resources (sensors, displays, users' mobile devices), third-party services and applications (those that offer content and functionality to be used in places) and users, which can interact with the infrastructure regardless of their physical location (place owners, developers, display network operators, and people in general). The system contains also an Android-based mobile client, which enables people to manage their self-exposure in the visited places directly from their mobile devices. Currently the Instant Places system brings together nine places that correspond to different physical locations (Figure 13).

Instant Places display infrastructure includes a diverse range of applications embedding a variety of requirements. The most known applications are depicted in Table 7.

Figure 13: A display from Instant Places infrastructure

Table 7: The list of web-based applications developed as part of Instant Places

Small Picture	Title and Description
	*Posters (Instant Posters)* It cycles through a collection of posters (pictures) uploaded by people through Instant Places main website and approved by display owners.
	*Football Pins (Instant Pins)* When a user checks in at a place, it shows content related to soccer team from Portuguese Premier League fetched from third-party services such as Facebook, based on a collection of "pins" – which are references or tags associated with people profiles and reflect their display content preferences.
	*Place Stream* It shows the recent interaction and various events associated with a given place such as a poster has been accepted for publication, persons that have just checked in/out.
	*Presences* It represents the profiles of the people present in a place.
	*Instant Chat* This application was developed entirely by author of the thesis. It allows display viewers to send messages from a mobile web-based interface (mobile widget). The messages are aggregated based on each subscription of the application enabling a situated behavior. The main purpose of this application was not for real deployments. It was used as a prototype for demonstrating the situatedness of display applications as part of the 2nd International Symposium on Pervasive Displays, PerDis 2013 (Taivan, José, et al. 2013).

Other applications are:

1. **Video**: It shows YouTube videos and allows users interactions from their mobile devices in forms of comments and likes;
2. **Places:** It aggregates the contents of a place such as pins, checkins and has three views: a) check-ins and the current pins – shows the name of the place, recent checkins (for one week) and the place pins; b) Facebook account of the place. People can set the id of a place and see the related posts; c) Facebook albums – allows to present the place Facebook albums;
3. **Pins**: It shows all the pins from the place such as users driven pins from the checkins and the place specific pins;
4. **Dropbox** (Instant box): It allows place owners to present files from a Dropbox folder.
5. **Facebook**: It shows content from selected Facebook page walls;
6. **Twitter**: It shows discussions posted to a specified hashtag;
7. **RSS Feeds**: It shows selected news feeds;
8. **Media RSS**: It shows the feeds from a media RSS, targeted for images and videos.
9. **Polls**: It shows public polls on various topics.

These applications are presented using a traditional scheduling approach in which display owners firstly arrange applications in time sequences and then submit a fixed application "play-list" to the displays. Then, each application is shown using Internet Explorer browser which is directly controlled by a custom third-party player provided by Ubisign company. This mechanism enables only display owners to influence an application's start and presentation times. While display viewers can interact with individual applications and personalize the content shown, they have to wait for the applications to appear on the display and they cannot alter the application presentation time. A subset of these applications has been made available across multiple deployments where they have been used by local communities on a continued basis.

2.3.4 UBI Hotspot

The Oulu's UBI Hotspot infrastructure from Finland (Ojala et al. 2012) is a large scale public display network that currently counts 18 hotspots with twelve indoor and six outdoor locations embedded with computing and sensors capabilities. The network is available for the public use and it constitutes a key component of the Oulu's open urban computing testbed. A display (57" screen size) is equipped with a touch screen foil which allows users to trigger explicit actions. The display system integrates two cameras, a loudspeaker, a NFC reader for authentication and it may access multiple wireless networks like Bluetooth, WLAN and WSN.

The software architecture is based on a distributed model, where hotspots may interact with each other or with nearby mobile devices using a loosely-coupled network over an event-based communication overlay – initially implemented with FUEGO architecture and lately by using RabbitMQ (Heikkinen et al. 2014). Using this communication middleware, the hotspots can publish or subscribe to events related to their context and applications may reside on multiple levels from employing of one hotspot to the utilization of multiple hotspots simultaneously and to coupling with user devices. The hotspots employ a user interface based on Web technologies and it comprises a set of webpages rendered by corresponding webserver processes and managed by an in-house screen real estate management system (Lindén et al. 2010)(Heikkinen et al. 2011).

UBI hotspots have two operating modes, either passive broadcast or interactive mode. In the passive mode the whole screen is dedicated for broadcasting local advertisements in form of a customizable playlist of video, animation and still photographs, i.e., UBI-Channel. Based on a face detection from the video feeds of the overhead cameras or when someone touches the touch-screen foil the hotspot changes to interactive mode. In the interactive mode the hotspots provide passersby with a range (25-30) of interactive web-based services, i.e., UBI-Portal. A service consists in a web page that resides in the public Internet and can be configured locally at each hotspot. The system allows people the authentication and activation of the services through personal RFID UBI-key or personal mobile phones. Once the interactive mode is selected, the layout manger adapts the screen in such a way the users can interact with the display and benefit for the available functionalities without hiding the advertisement channel; when people are interacting with the hotspot, UBiChannel is squeezed to a quarter of the screen size and moved to the top-left part.

Oulu's display network supports a broad range of web-based applications or services. By April 2012 the system included 25 distinct applications in seven categories: News (3), Services (3), City (3), Third-party (4), Fun & Games (5), Multimedia (3), and New Cool Stuff (four apps developed within the 1st International UBI Challenge). Many of these applications are interactive, taking advantage of the touch-sensitive displays or the Bluetooth, RFID or Camera capabilities (Figure 14). Due to the high cost of realizing such an infrastructure and ensure its sustainability, more than a half of the services typically depend upon third-party content that is beyond the hotspots' administrative control. From their experience, the web paradigm has proven very efficient in implementing this approach, in which new services residing on any webserver in the Internet, can be easily integrated as long as they conform to certain minimal design guidelines. To achieve high execution requirements, services require to be easy to develop, robust, and be easy to deploy and maintain. In their approach, the hotspots considered the use of virtual-machines for both easing the overall display deployments and improve robustness against server and network problem (Lindén et al. 2012).

Besides interacting with the services using large display touch commands, UBI hotspot platform has support for distributed application user interfaces on interactive large public screens and personal mobile devices (Hosio, Jurmu, et al. 2010). In this sense, six distributed applications were deployed in a city center to explored distinct controlling modes of the applications: 1) mobile and public display control: *PlaceMessaging*; 2) no mobile control but with public control *BlueInfo* and UBI-Gallery; 3) has mobile control but without public control Social Surroundings, UBIRockMachine and *UBIPoker*. *PlaceMessaging* is an application that utilizes both the public and the private UIs and let users post content, such as, text, emoticons and pictures on a display, which can later be accessed and visualized. *BlueInfo* is an application that allows users to download various data to their mobile devices using the display touch screen control. *UBI-Galery* is an application that allows users to tag or delete their previously uploaded pictures. *Social Surroundings* is a social application that stimulates conversation in public spaces based on users' social media content from popular third-party services. *UBIRockMachine* is an audio community voting application. *UBI-Poker* is multi player poker game. The player uses the mobile UI to control it and the shared display for visualizing it.

Figure 14: Outdoor UBI hotspot in downtown Oulu (Ojala et al. 2012)

2.3.5 Screens in the Wild

"Screens in the Wild" (North et al. 2013)(Fatah et al. 2014) is a project deployed in UK that explored the potential of networked urban screens for communities and culture. The project addresses the specific challenge of how best to integrate urban screens, a radical and potentially disruptive new technology, into the urban realm. The urban display network is formed by four displays that were installed in two cities: two in London and two in Nottingham. Displays were located inside the front windows of public space venues, visible from the street or public areas including a public library, a community centre, an art space and a cinema.

The displays with 46''were vertically mounted behind glass and used capacitive touch foils for enabling touch interaction through single glazing windows (Figure 15). In addition, all displays had attached a video camera and a speaker. A special case for interaction was at one of the venue that had double gazing window and where an alternative solution for interaction was adopted – a light key pad (Ye & North 2014). Each display node had networked Windows PC as media players that were remotely administrated by team members via TeamViewer. As part of the project four interactive experiences or applications were developed. The applications were implemented in HTML5/JavaScript and run in a local copy of Firefox full-screen, with scheduler software switching between them. In order to provide real-time multi-user functionality for web applications across the screen network, UNION Client/Server infrastructure was used. The applications are described in the following.

Video Link is an interactive experience that enables synchronous four-way video communication (no audio). It was implemented using a set of four USB cameras installed in each display node (at a low height to encourage interaction from and with children) and a combination of YawCam and iSpy software. The corresponding video signals from each location were displayed at the bottom of the screens in four panels - the pane to the left always showed the local feed. To attract people for interaction the local video stream was also mirrored so it could be easily noticed by passers-by. The main purpose of the video link was to encourage synchronous multi-user interaction across the four screen nodes. In this way, people could see if someone is using any display at remote locations and manifest the will to interact with the other party by waving or through gestures.

Sound Shape is a networked musical instrument that allows collaborative music making between users in different physical locations. It is based on a client-server architecture in which each display node actively participate in creating a common sound track across different locations. A sound or a note is represented by a touch pad and there are 25 in total, displayed in a grid of 5x5. The central server keeps the state of the music and updates in each location the current status of the sounds pads according to their touch/untouched behavior. In this way, all participants see and hear the changes to the musical composition. Each active sound pad has an animated illumination that flashes in time with the musical sequence. Additionally, in one of the screen where the chosen touchfoil technology did not work with double glazing, users could scan an onscreen QR code into their smartphone enabling them to interact with the screen.

ScreenGram is a photo application that gets its content from tweets with a specific hashtags, which have associated Instagram photos. It then fetches periodically the photos from Instagram and shows them in a slide show fashion. Users can upload their images by sending them over Instagram, to Twitter, using a hashtag of those associated with the respective displays. Normally, uploading a picture to the displays takes between 30s and a minute. Through a set of radio buttons on the right-hand side of the screen, users can select a certain hashtag from which the content will be shown, e.g., #summer, #olympics. Researchers and venue stuff can also change the hashtags that are associated with those buttons and extend the content possibilities that can be shown. Overall, the main scope of ScreenGram is to encourage asynchronous multi-user interaction between the networked screens and people elsewhere interacting through their phones.

Moment Machine (Memarovic et al. 2013) allows people to simply capture their everyday moments by taking an image through a display and spreading it across the network. Its interface is very simple and contains a live video feed – which is a good mean for getting passersby attention. People can take a picture by pressing a button. In order to give users a time to prepare themselves the picture capture is delayed by five seconds, which is indicated through a countdown timer. Additionally, before taking a picture, people can change the “look” of their snapshot by selecting a filter before taking the photo. After an image was captured, users have thirty seconds to decide if they want to leave the moment on a display, otherwise the moment would automatically appear on the screen and across the display network. Display viewers can also browse through images captured at all locations.

All four applications were deployed in the wild as part of the Screens in the wild project. Various user studies were conducted willing to understand the extent to which people are engaged in the interactions and manifest interest in using the infrastructure. The findings suggested that building an artifact, i.e., urban screens network, for a community involved dealing with conflicting requirements or tensions and contradictory usage models. This is because the public space is subject to different views that multiple stakeholders might have about the use of such a screen network. The results indicate that it is not necessarily for an urban screens network to reflect all of the tensions in the totality from the inception of the system. Instead, all conflicting requirements might be incorporated gradually over the product lifetime. This is specifically highlighted in Screens in the wild project as each screen experience that is displayed across the four locations offered an opportunity to reflect some of the tensions. For instance, regarding the Moment Machine experience (Memarovic et al. 2013) some of the participants did not see the point of leaving the image on the network unless they could "take" the moment with them. However, people did engage the Moment Machine application although its purpose was questionable. Its evaluation results were promising and showed that such an application can be used by local community members to capture images "on the go."

Figure 15: SoundShape application (left) and a screen shot of ScreenGram (right); video links are displayed at the bottom of each experience (North et al. 2013)

2.3.6 Multi-Touch Display at a Conference

The research by Ardito et al. (Ardito et al. 2012)(Ardito 2012) focused on developing software applications for large multi-touch displays installed in public spaces that, besides being attractive and engaging, have a specific utility for their users, i.e. support them in performing tasks that go beyond mere entertainment. These applications provide information and/or services in different contexts, such as cultural heritage, tourism, public events, and education.

As part of the conference deployment, three applications have been designed to provide services to people attending the event: *Interactive Program*, *Taxi Sharing*, *Conference Photos*. The applications were made available through a 46-inches Full HD LCD multi-touch display installed in an appropriate location in which the environmental factors would encourage interaction and, at the same time, not embarrass possible users.

Interactive Program complements the conference program with various multimedia content items associated to each paper and author. By using the applications conference attendees can also create their own agenda whose events will be available in their google calendar. Taxi Sharing enables attendees to post a request for a taxi or reserve a seat on one of the shuttles provided by the conference organization. Subsequent people could opt to share the taxi reserved by a previous attendees, who will be notified of this by e- mail and SMS. Finally, Conference Photos allows users view and share photos uploaded into the system through Flickr (the photo needs to be tagged with the name of the conference), memory cards and USB sticks.

The evaluation of the multi-application display showed its success in attracting people's attention and it was used by some 82% of conference attendees often more than once. Although the displays were not too large, most of the people interacted individually or in dyads, which facilitated social interaction as conference attendees were likely to know each other and could alternate actions towards a common goal. The most used application was Taxi Sharing for booking transportation to the airport (44% of uses), followed by Interactive Program for consulting the conference program (30%) and Conference Photos for picture sharing (26%).

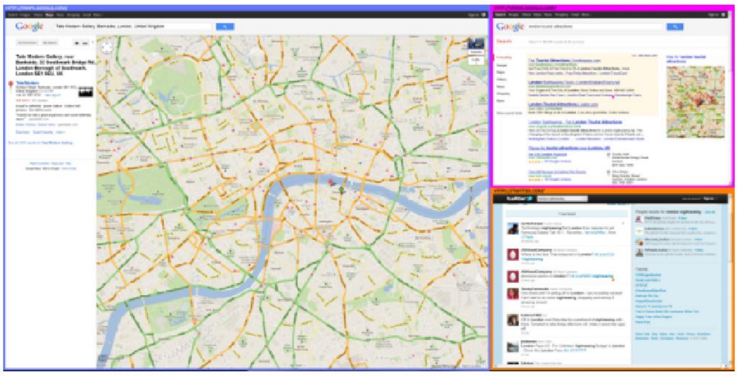
Overall, due to the social context specific to a conference venue (a semi-public space) in which people might know each other and have a strong, shared identity, the system proved to enable easy socialization in front of the device. While people firstly used the system just for curiosity and exploration, utility became a driver at a later stage as conference attendees returned to perform a specific task, like booking a taxi. In particular, Taxi Sharing application that supported people in performing the most needed tasks was used more frequently, regardless of its lower graphical appeal. Finally, this study suggested that application type is an important dimension that contributes in attracting people at different stages of the interaction (first contact vs. subsequent interactions) and due to the task variability elicited by the different applications, people could better engage and keep interest in using the system.

2.3.7 The LunchTable

LunchTable is a multi-display and multi-user display system installed in a semi-public setting (lunch space) that supports casual group interactions around a lunch table with the goal to access and share of information. The system uses a two-meter wide vertical display with 4.7 Megapixels for showing rich visual data and an interactive multi-touch display embedded within a regular lunch table for controlling the information on the vertical display. The design of the system enables equal, simultaneous access for all participants and provides simple and unobtrusive interaction mechanisms adapted to a lunch setting.

The system interface is implemented in C# on Windows 7 using Windows Presentation Foundation. In order to control windows on the large display as well as to replicate windows on the horizontal surface Microsoft's Desktop Window Manager API was used. The interface elements for the horizontal surface were implemented using SMART Table SDK. By interacting with the horizontal surface, users are able to manipulate application windows within the wall display space. The vertical display is composed by six back-projected screens in a 3×2 tile arrangement and it can present up to six windows. The windows are represented by web browsers that can be used to navigate to any web page or application. At a moment, one tile can display only one application window and the system allows windows to be configured to cover 3×2, 2×2, 2×1, 1×2, or 1×1 tiles. Overall, the system allows up to six users to manipulate up to six windows at the same time.

The horizontal surface interface is composed by a window manager showing a minimized representation of the application windows contents, an application launcher, and a space for virtual input devices. The window manager is shown at the center of the interface and shows a thumbnail of the content from the large vertical display updated in real-time. The Launcher is represented by the area surrounding the window manager and it contains a slowly revolving train of icons: nine application icons, a wastebasket icon, and a trackpad icon. Applications are activated and displayed on the large screen by dragging any application icon onto any of the regions of the window manager. Removing an application from screen can be achieved by dragging its thumbnail from the window manager onto the wastebasket icon. The launcher includes the following applications: Google search, YouTube, Gmail, Google Docs, Google Maps, Facebook, Twitter, Flicker, and Wikipedia. The applications were chosen to cover a range of activities that people might want to engage in either individually. Finally, users can enter text and control the application windows contents by using virtual input devices, which contain a trackpad and an on-screen keyboard. To allow identification of the relations between system interface elements on both displays, the application windows border, input virtual devices and respective cursors use the same color (Figure 16). A single virtual input device can be associated with a particular application by dropping the trackpad icon onto an existing window. Through the standard one- or two-finger rotate, resize, and translate manipulations, the virtual input devices can be can enlarged, shrunk, or moved to any position or orientation within the tabletop workspace surrounding the window manager.



(a)

(b)

Figure 16: The LunchTable system; (a) large vertical display (b) tabletop multi-touch display (Nacenta et al. 2012)

The system was evaluated for one week within the lunch space of a university research lab. The results showed a substantial use of the LunchTable for sharing visual information (rather than textual data) such as online maps and videos that are otherwise difficult to share in conversations. While the most opened application was Google Search, it was just used to navigate to various other information sources. On the second place people enjoyed Google Maps, which was shown in 10 of the 63 windows that were opened during the seven days. Overall, the observations of the real usage of the system revealed that equal simultaneous access from several users does not seem critical in casual group interactions due to the role of social protocols in these situations.

2.3.8 Analysis

As opposed to single concept displays, multi-application displays have the potential to address the interests of an extended set of stakeholders beyond the display administrative domain or place owners. We analyze these systems to understand the type of engagement that such displays are enabling and how the individual applications are presented to the audience. The analysis overview of multi-applications displays is given in Table 8 and Table 9.

Table 8: Analysis overview of multi-application display systems

System	System behavior regarding the applications employed	What is an application?	Application Presentation	Applications' Behavior (logic or functionality)	Content Origin
iRoom	No scheduling; By using a combination of devices, users explicitly select and activate applications	Web-based GUI application	Full screen	Explicit content presentation. Users can issue touch commands and access applications' functionalities	App-delivered content
e-Campus Bluetooth-enabled system	No scheduling; By using Bluetooth device names, users explicitly requests and personalize applications to be shown for a limited amount of time	System specific applications that renders specific types of content	A specific frame of the screen	Explicit content presentation. Users cannot interact directly with the screen.	System-delivered content (digital signage content); App-delivered content
UBI Hotspots	No scheduling; By using a service directory, users explicitly select and activate applications for an undetermined amount of time	Web-based GUI application	Right half of the screen	Explicit content presentation. Users can issue touch commands and access applications' functionalities. They can also use smartphones for certain apps	System-delivered content (digital signage content); App-delivered content
Instant Places	Application are selected and activated by the system (applications are scheduled) using a fixed amount of playback time, e.g., 1 minute; user do not have control over the schedule	Web-based GUI application	Full screen	Implicit content presentation; Users cannot interact with the screen directly. They can use a smartphone application to post user-generated content	App-delivered content The content is only handled by applications not by system; applications are responsible with the behavior (logic) to present their content.

Table 9: Analysis overview of multi-application display systems (continued)

System	**System behavior regarding the applications employed**	**What is an application?**	**Application Presentation**	**Applications' Behavior (logic or functionality)**	**Content Origin**
The Lunch Table	No scheduling; By using a launcher, users select and activate applications	Web-based GUI application	Tile-based presentation Full screen	Explicit content presentation. Users can remotely issue commands and access applications' functionalities	App-delivered content
Screens in the Wild	Applications are selected and activated by the system; (applications are scheduled); users do not have control over the schedule	Web-based GUI application	Full screen and specific region of the screen	Implicit content presentation; Users can issue touch commands and access applications' functionalities	App-delivered content
Multi-application display at a conference	No scheduling; By using a menu, users explicitly select and activate applications; The system has selected by default one of the available applications	Web-based GUI application	Full screen	Explicit content presentation. Users can issue touch commands and access applications' functionalities	App-delivered content
Tacita	No scheduling; By using a smartphone app, users explicitly requests and personalize applications to be shown for a limited amount of time	Web-based GUI application	Full screen or specific region of the screen	Explicit content presentation. Users cannot interact directly with the screen	System-delivered content (digital signage content); App-delivered content

2.3 Display Systems with Multi-Application Support

Since the concept of a display application is not well defined in the research community, each infrastructure adopts it according to the type of experience that can be afforded by the displays themselves. While multi-application displays tend to employ a diversity of applications, there is not a common way to represent these applications and sometimes it is not easy to understand where is the border between one application and another. This makes even difficult to conceptualize what an application might be and what might not. For instance, in single concept displays we have seen that applications can have different region on the screens and handle different types of content in a given way. However, such applications are closely related and the underlying system presents them in an integrated way. In the case of multi-application display the distinction between one application and another can be based on its content and related behavior, e.g., it is a gallery app or a map app. Alternatively, the display itself may have means to help users making this distinction e.g., offering browsing capabilities (service directory) or cycling with a given frequency (an application is shown at each 5 minutes). This analysis of multi-applications displays is important because it highlights the type of support that these systems need to have in order to enable a diversity of behaviors.

Based on our analysis, four main ideas emerged on what a display application might be and how it should be presented, i.e., system-activated, user-activated. In addition, not covered by the systems we have just analyzed, applications can have their own logic to activate and start to present content regardless users' explicit request or system initiative.

1) Non-interactive applications that pack content from various sources (display owner and audience) with various purposes and are system-activated (system-scheduled); in this scenario users can upload content to applications by using mobile phones and desktop computers.
2) Non-interactive applications that pack content from various sources (display owner and audience) with various purposes and are user-activated; in this scenario, users can personalize applications' content by using mobile phones.
3) Interactive applications that pack content and functionality and are system-activated. Users can interact with these applications using touch-enabled screens; users do not have control over the application schedule and they can use applications for a determined amount of time.
4) Interactive applications that pack content and functionality and are user-activated; Users can interact with these applications using touch-enabled screens for an undetermined amount of time.

While in single concept displays, the system itself was responsible for the behavior to present any content items (content sourced by display owners or audience), in the case of multi-application displays, the common pattern is that applications are responsible with what content to present and the logic associated with it. The system is thus released by any behavior (logic) to handle any content items. However, this distinction is not clear yet because displays must have a default behavior when no users are interacting with them. They cannot be turned off like desktop computers and in consequence such display systems have a default behavior in which some content items are shown to passersby or an application is system-activated by default. Therefore, from a computational perspective multi-application displays can be characterized using two main paradigms as shown in

Table 10. In consequence, a display system need to evolve from the simple provision of content playing functionality to behaviors that allow applications to present their own content and enable users to explicitly request for applications.

Table 10: Paradigms for multi-application displays

Paradigm	**Content**	**Experience**	**System behavior**
Everything is an application	Content belong to applications; applications are responsible with the logic to present their content	App-delivered content; Applications can be either interactive or non-interactive.	The system cannot control content items; It can only control when applications may be presented, e.g., choose a default application, allocate a determined amount of time for each one or let users to issue requests
Hybrid system – a mixture of a digital signage system and a desktop computer	Content belongs to the system; Content belongs to applications that can be explicitly requested and controlled by users;	System-delivered content; App-delivered content; Two modes of operation: 1) non-interactive signage mode; 2) interactive mode where applications cannot present their content by default; applications' content is driven by users	The system controls the behavior of a specific set of content items. Usually, these content items are managed by display owner explicitly.

In our work, we define the concept of display application and examine its potential principles and characteristics. This provides the general understanding of what a display application might be and in what ways it is different from applications in other computing environment, e.g., desktop and mobile computers. In this context, a display application would have a clear meaning with its properties and requirements well understood with a clear separate of concerns between display owners and content or application creators. This also allows different display infrastructures to explicitly employ the notion of an application as a class of software that is not tied to particular settings or assumptions and can be developed by third parties. In conclusion, none of these display infrastructures supports yet the type of open application development that we are researching in this thesis.

2.4 Toolkits for Interactive Display Applications

As described above, many researchers address the field of interactive display applications. These toolkits have the potential to transform the single concept displays into rich multi-application systems. This kind of work is also known as system software or middleware for public displays and represents specific studies on facilitating the development of interactive display applications. In the following, we provide a description of a set of four reference research toolkits in creating interactive display applications continued by an analysis of their relevance for our work.

2.4.1 MAGIC Broker

The MAGIC Broker middleware (Erbad et al. 2008) is a toolkit that enables rapid development of interactive public display applications with a focus on mobile device interactions. It provides a set of common abstractions and a RESTful web services protocol to facilitate and speedup the development process.

This middleware is focused on the following five design factors: 1) the use of events complemented with local state as the key interaction abstraction (applications and devices can retrieve the state of the environment instantaneously and restart where they left off). Using this approach in which event sources are decoupled from event consumers makes the system less sensitive to the order of actions and more resilient toward failures; 2) the use of mobile devices without the need for specific applications, i.e., client-less devices and support different forms of interaction, e.g., SMS, Voice and Web interaction. 3) the use of a lightweight, domain specific HTTP-based protocol for communication between the various components of the system; 4) the use of web-oriented tools and standards, such as HTML, JavaScript, Flash, PHP, Java applets and Java servlets to facilitate the creation of high quality content; and 5) support for flexible device groupings that can be re-arranged according to the current context and applications when the number of displays, users and applications increases beyond single display deployments.

MAGIC Broker derives its abstractions from the core group of abstractions of the Ubicomp Common Model (Blackstock et al. 2006) and the key abstractions required for interaction applications, events and state (Patterson et al. 1996)(Dix 1991). It includes the following abstractions: 1) *channels* to address groups of situated displays, individual screens, users, and the functionality supported by the screens; 2) *events* which represent the messages sent from event sources to sinks via the event broker; 3) *state* to designate the system persistent state as applications often need to store and retrieve the current state of the system, e.g., retrieve the last object selected on the screen; 4) *services* represent a way to support synchronous RPC-style two-way interactions with a service hosted outside the middleware e.g., Google map or Flickr; and 5) *content* is the data stored and retrieved by the middleware in forms of images, videos, text, and HTML documents within/from a channel.

The evaluation of the toolkit proved its usefulness in supporting a wide range of interactive display applications and interaction modes. Developers did not encountered difficulties in understanding the abstractions and the REST protocol was handy for rapid prototyping. Overall, this middleware is targeted to scenarios where the number of displays, users and applications increase beyond single display deployments and therefore a middleware layer may help to support rich interaction between components while keeping the system manageable.

A second version of this toolkit, i.e., MAGIC Broker2 middleware (Blackstock et al. 2010) addressed the need to provide a platform that offers a consistent model and interface for building Internet of Things (IoT) applications from cooperating things. This version of the toolkit consolidate the previous features of the middleware and highlights two important benefits: it offers a simple, uniform web- based API for building IoT applications and provides developers three built-in programming abstractions: publish-subscribe event channels, persistent content and state storage, and brokerage of services via remote-procedure call. As for the first version, the platform was used within the eCampus (Friday et al. 2012) and Magic (Finke et al. 2007) ubicomp testbeds to create several applications involving spontaneous device interaction such as between mobile phones and public displays and shared sensor/actuator networks.

2.4.2 OSGiBroker ("Dual display")

Kaviani and his colleagues (Kaviani et al. 2009)(Kaviani et al. 2011)(Finke et al. 2010) investigate the design of interactive large public display applications that support a "dual display" approach, which enables applications to execute across large displays and mobile devices utilizing the input and output capabilities of both device types. In their approach, some elements or widgets (in graphical user interfaces) of a user interface are placed on the large display and some elements on the handheld device. A widget can be either non-interactive or interactive. Non-interactive widgets are mostly used to present the system state or parts of the system state to users and interactive widgets accept user input to initialize system state changes (user interface). Additionally, the interactive widgets often provide feedback that an input has been received, e.g., a button widget changes it appearance every time a user clicks on it.

The infrastructure underpinning the "dual-display" approach is based on an updated of version of MAGICBroker (Erbad et al. 2008) called OSGiBroker. The new middleware provides a protocol-independent communication between senders of a message (message sources) and receivers of that message (message sinks) and supports faster and more spontaneous message exchange between sources and sinks than MAGICBroker.

While both versions of the middleware use publish-subscribe as a design pattern and channels as the main abstraction, OSGiBroker has several improvements and design considerations. In RESTBroker, events are sent to a channel using HTTP and are received by named subscribers registered to the respective channel and capable of receiving messages through HTTP. OSGiBroker architecture is different and is based on Open Service Gateway initiative middleware (OSGi) – which represents a centralized service oriented component model to modularize Java applications. The new architecture enabled the connection of any message source to any message sink regardless of their required communication protocols. This permitted to extend the communication protocols between mobile phones and interactive widgets on the large display including SMS, VoiceXML, SOAP and REST communications over HTTP, and direct TCP, UDP, or Bluetooth.

The "dual-display" paradigm, in which parts of an application interface are shifted down to a personal mobile device, was conceived as a solution that can solve a number of problems originating from limitations in large display real estate. These limitations present concerns regarding a sound and engaging interaction experience for both actors and spectators. While actors may control and/or manipulate the content on public displays through mobile devices and thus change the 'flow' and 'pace' of the presented content over time, spectators are just mentally engaged with the displayed content and do not have any control over the content. Similar to (Dix & Sas 2008), Kaviani et al. considered two types of conflicts when multiple actors attempt to access the same interactive widget or manipulate the same content simultaneously: conflicts in space and conflicts of pace/flow. Conflicts in space mainly originate from the limited screen space to provide visual feedback when executing a sequence of actions (interaction process) or providing information about a new system state (content presentation). Conflicts of pace/flow usually occur when users do not have any feedback on the system behavior, e.g., the time between consecutive system state changes is too short or there is no feedback on why changes on the display are happening. These conflicts impacts not only the actors as spectators require being able to follow and understand system state changes presented on a large display and have reasonable predictions about forthcoming state transitions.

In order to increase interaction engagement of public displays audience, i.e., actors and spectators, and to solve the emergent conflict situations the authors defined four design strategies: 1) *localized interactions* – that reduce conflicts of space by blocking the visualization of executing a sequence of actions on the large display (the interaction process will entirely take place on mobile devices and only new system states will be updated on the large display); 2) *distributed system state* – reveals only general parts of a new system state on the large display and keep detailed information on the small display (in this way the large display gets a larger presentation space for content, thus overcoming conflict in space); 3) *providing display focus* - provide the actor with a visual cue that indicates if an interactive widget (at the large display or mobile device) is in active or inactive state so that a user knows where to direct the input; and 4) *cause summary* – this strategy reduces conflict of pace/flow by providing visual cues on the large display summarizing the execution of a sequence of actions, e.g., displaying information about what causes the new system state. These design strategies have been examined for their effectiveness through the implementation of several different applications that were deployed on large interactive displays and on mobile phones at University of British Columbia. The "dual-display" approach proved to be useful and the application examples were effective in supporting both actors and spectators. In addition, the OSGi-based architecture provided real-time seamless information exchange between display types. Overall, the experiments supported the hypothesis that user interfaces can be distributed on large displays and mobile devices without a significant effect on user performance and the approach proved to be an improvement for the end-user experience.

2.4.3 RED Framework

RED Framework (Really Easy Displays framework) (Calderon et al. 2014) is a web-based platform to facilitate spontaneous interaction between devices and applications. It provides a single abstraction for content and interaction between display types, data streams and interaction modalities, and empowers developers to create multi-display applications through the sharing of web document object models (DOMs) across displays. In order to simplify the development of multi-display applications, the framework leverages Web technologies to allow interoperability between web page objects across displays, i.e., text, images, video, and interaction events. In addition the framework can access other non-web data sources such as accelerometers, gesture sensors with ease.

RED framework consists of three main components: event management, application virtualization, and data abstraction. Developers using RED are provided with a RESTful JavaScript DOM API that allows the sharing of interaction events using a RED backend server and enable the extension of common DOM elements to be shared transparently between interacting displays. Application Virtualization is composed of an application container that manages RED applications and provides a set of interaction services for a particular place, scenario or network of displays, e.g., user accounts, checkins. The framework is built on the top of the Thing Broker (Almeida et al. 2013) –which is a platform for the Internet of Things (an updated version of the MAGIC Broker 2) and provide a simplified abstraction for differing protocols and data sources. The data model of the Thing Broker is organized in three layers: (1) meta-data that describe the human-readable properties of an object, (2) streams of data or events consisting of typed name/value pairs, e.g., numbers, strings, objects or arrays, and (3) associated MIME-type resources, e.g., unstructured text, HTML, photos, video or audio clips. Through the usage of RED API, developers map DOM objects to Thing Broker "things", leveraging its ability to share state and meta-data and interaction events between shared DOM objects on different displays. Additionally, Thing Broker enable RED applications to access to data sources outside web documents, like sensors and actuators e.g. pressure sensors, relays, accelerometers or gesture interaction from devices such as the Kinect.

The framework was used by several internal and external developers to create diverse applications including information broadcasting applications (a presentation application that allows users to browse through presentation slides using their mobile phones); collaborative applications (a competitive memory game where a large display is used to display a pattern that users match on their mobile phones); mobile device applications (a visualization of private messages sent by people using their mobile phones within the RED container showing social interactions between members of a community); application to manipulate data (create and edit medical records using a large display). Overall, the development insights showed the success of the RED's web-centric approach in overcoming the complexity of spontaneous interaction with public displays, i.e., eliminating the need of specialized software or hardware, reducing the deployment and maintenance burden and providing a simple abstraction for data streams and interaction events.

2.4.4 PuReWidgets

PuReWidgets (Cardoso & José 2012)(Cardoso & José 2013) is a toolkit that provides interaction widgets for generalized interaction with public displays. Developers could integrate it into their public display applications to support the interaction process across multiple display systems without considering the specifics of what interaction modality will be used on each particular display. An interaction widget is an abstraction that provides developers with high-level interaction data, hides the specific details of the underlying input techniques and can have different graphical representations in different platforms. The toolkit targets web-based applications that can be hosted on third-party servers to serve content to many displays while benefiting of the locally available interaction resources. PureWidgets assumes that a public display will show content from multiple applications and will iterate through those applications based on some pre-defined scheduling criteria, e.g., time-based scheduling. If an application is not visible on the screen it can receive and process interaction events and produce display specific content that can be accessed in multiple ways including a web page or a custom mobile application. The main features of the PuReWidgets toolkit include multiple interaction mechanisms (SMS, email, Bluetooth naming, Bluetooth OBEX, QR Codes, mobile application, and desktop web application, automatically generated graphical interfaces, asynchronous events and concurrent interaction by multiple users.

The toolkit is implemented in a distributed fashion with two main components: a widget library and web service that handles interaction events. While the widget library (which can be included in a web application project) is used to instantiate widgets within the application and registering interaction event callback functions, the web service is used to store some metadata about the instantiated widgets across all possible applications thus effectively decoupling widgets from applications. The web service has a remote I/O infrastructure that accepts raw input events from users and stores them in a queue until the application is ready to receive them. The events are stored even if the respective application was not executing at the public display and the web service routes them to the application/widget that was addressed by the user.

The current implementation of the toolkit is done using Google's App Engine platform[11] and Google's Web Toolkit[12]. While the library is in the form of a GWT module that developers can include in their GWT projects, the web service is implemented as an App Engine application that exposes a REST API to the library. Depending on the application's data needs programmers can choose from a list of six widgets – each one having a specific type of high-level data. The widgets supports the most common interaction scenarios encountered in public displays: Button, List box, Text box, Upload (media files), Download (to personal devices), Check-in (users informs applications about their presence). The graphical components of the interaction widgets are implemented based on the standard GWT widgets.

PuReWidgets was evaluated along several dimensions including system performance, API usability and real-world deployment and the results were in general positive meaning that the toolkit reached its goal. It provided high-level controls or widgets that abstract the input from several heterogeneous interaction mechanisms, allowing programmers to focus on the interaction features of their applications, instead of on the low-level interaction details.

2.4.5 Analysis

Aiming to incorporate interactivity and to provide display viewers new forms of engagement, several research attempts tried to build infrastructure, middleware and toolkits that focused on making it easier to develop interactive public display applications. This trend included spontaneous interactions with public displays and various interaction patterns as required by multiple interactive applications. Still, due to the fact that applications for interactive displays are still a new topic with a lack of widely accepted and well-defined concepts, the most available toolkits or middleware are specific to certain display deployments and there is no common view about the key principles and characteristics of the emerging interactive applications. While all the toolkits contribute to open up the development of interactive applications and to attract interested developers to

[11] http://code.google.com/appengine Accessed September 26, 2014
[12] https://code.google.com/p/google-web-toolkit/ Accessed September 26, 2014

imagine novel usage case of public displays, these toolkits are based on set of assumptions and are limited to a restricted set of interaction scenarios. There is no unified understanding about what an interactive application is and what execution environment it should expect.

All the toolkits described in this section help in understanding the various software abstractions that foster the development process and build a map about different interaction scenarios with public displays. While the research on single concept displays and multi-application infrastructure mainly studied design challenges facing end-user or audience interactions with a particular application or a set of applications, the focus of these toolkits is to empower application developers with necessary abstractions and software to facilitate building interactive applications quickly and easily. Definitely, this body of research constitutes an enabler for the vison of Open Display Network in which public displays would be transformed into rich computing artifacts. A particular case is the PuReWidgets, which advocates for consistent interaction models across different displays with different interaction capabilities where users can easily recognize and understand the respective interaction features and controls similar to the experience offered by desktop computers, e.g., a button has the same representation and meaning.

In our research, we provide a structured description and analysis of the key principles for application development in public displays and build a case of developing web-based applications for this context by understanding the opportunities and limitations of Web technologies. Web and its enabling technologies and protocols, were seen as a general principle in the field of ubiquitous computing since the deployments of interactive workspace like iRoom from 2003. Since that time, Web was used in many system infrastructures and software applications, e.g., MAGIC Broker middleware in 2008. While these toolkits have their specific goals, not much is known in how to effectively use Web technology to create these interactive display applications. This means that despite the numerous toolkits and display deployments there is not yet a general understanding about the implications of display applications on the ability of Web technologies to serve as an appropriate technological framework for the creation of this type of application. On the contrary, our work considers the analysis of these implications and proposes a set of key development specificities and insights that developers should consider when creating web-based applications for public displays. In addition to the assessment of the Web technologies for creating display applications, we consider the extension of web development practices and expertise to support this type of development.

2.5 The Case of SmartTVs

Smart TV (also referred as connected TV or hybrid TV) is a term used to represent the state of the art about Internet integration into TV sets. The term comes from the analogy with smartphones and their characteristics towards more advanced computing and communication capabilities. Saying it differently, the concept of Smart TV reflects the technological convergence between computers and television sets. The existing TV sets designed as Smart TVs blend traditional broadcast media with Internet and app-delivered multimedia content and interactive services.

Given the advanced computing features, a Smart TV comes with a specific operating system that provides developers and end users with a platform and a Software Development Kit (vendor specific), to develop, publish, install and run interactive applications. For example, Google TV is platform for SmartTVs and allows developers to create applications using a specific version of Android OS (Android TV[13]). Another example is Samsung Smart TV, which allows the creation of application using Web technologies such as HTML, CSS and SVG, coupled with various device APIs i.e., JavaScript applications[14]. Given these software application possibilities, a television set would change its common way of presenting broadcasted content, being more targeted on how to seamlessly and viewer-friendly display various input sources, e.g., user generated content. Overall, the purpose of a Smart TV is to deliver content from multiple sources, e.g., local or cloud based, and supporting the integration of Internet-based services such as the traditional broadcast TV channels but as well more engaging and interactive multimedia applications.

Web and TV is an initiative to bring the potential of Web in TVs. The main discussion around this topic is organized by W3C[15] as part of the "Web of Devices" program (W3C 2014j). W3C is the organization focused on technologies to enable Web access anywhere, anytime, using any device. This means to access Web from mobile phones and other mobile devices as well as use of Web technologies in consumer electronics, printers, interactive television and even automobiles. Integrating Web into TVs has a history more than a decade. There have been many efforts from multiple device manufactures and software companies in this direction but the results were not as expected. The companies have gone through a lot of trial and error under various names: WebTV, Internet TV, Connected TV, Net TV, Apple TV, Yahoo TV, Google TV and Smart TV. TV makers have learned that the user experience of watching TV is radically different from that of a PC and trying to fit a PC into a TV set does not work. However, inspired by the success of the smartphones, TV makers are motivated to lunch Smart TVs on a very large scale.

2.5.1 Discussion

Although SmartTVs are not considered public displays, this class of devices presents a set of similarities with the type of the work conducted in this thesis. Being addressed by the W3C community, the topic of integrating Web technologies in TV sets has the potential to provide many benefits for the home audiences and, at the moment, constitute useful lessons learned for the context of public displays (W3C 2011).

The main similarity between a SmartTV and a public display is that they have the same value proposition, i.e., *content is king* or *the content is the system*. Both type of audiences (home and public) benefit and engage with the system if there are various content alternatives (Toeman 2010)(Storz, Friday, Davies, et al. 2006). While for the end user experience the content might represent the excepted result (which often engage people passively), for a long-run usage the way in which the content is handled and presented to the audience would also have a case. For SmartTVs, this means that the traditional television broadcasting is currently complemented by individual and autonomous

[13] https://developers.google.com/tv/android/ Accessed September 26, 2014
[14] http://www.samsungdforum.com/Guide/art00007/index.html Accessed September 26, 2014
[15] http://www.w3.org/ - The World Wide Web Consortium, Accessed September 26, 2014

mechanisms to present content from third parties such as YouTube or Flicker in the form of app-delivered content. For public displays, current research deployments are also embedding content as part of individual applications experiences. However, at the moment, the SmartTV end user experience enabled by more interactive applications that are not primarily conceived to deliver content in a passive way, e.g., Twitter, Facebook, is very poor, (actually being worst) when compared to other computing platforms including desktop or mobiles. While more innovation is needed for the field of SmartTVs, the industry efforts seem not to have assessed very well the type of expected user experience – "*the psychology of the couch*" (Toeman 2010).

Our research was primarily anchored in understanding the type of experience people prefer in public displays and, as part of the entire PD-Net project, several studies were conducted to assess the relevance of possible application behaviors. In consequence, we looked to the properties of public displays as a way to provide added value for audiences when designing for multi-application displays. This led to the formulation of a set of application principles that build the fundaments of a "*psychology of the public space*". While the community of users would decide the practical usage of interactive multi-application displays, we did not employ the notion of software application because everybody did. Building multi-application displays would require applications as software mechanisms to pack or access content and employ interactive features that are adequate for public circumstances.

2.6 Summary

In this chapter, we offered a comprehensive description, analysis and discussion of different types of related works. This can be summarized in five main strands: *current deployed displays*; *single concept displays prototypes*; *multi-application display prototypes*; *research toolkits for interactive display applications;* and *platforms for SmartTVs.*

All these insights from different classes of systems helped to build and consolidate the motivation of our research case. While the actual deployments of displays that we found in our urban environments are mainly described as closed systems being under the control of their owners with limited scenario of usage, the research efforts proposed a variety of systems that enabled open architectures and enabled new levels of engagement for audiences by employing a diversity of behaviors and use cases. We make distinction between dedicated displays or systems that are designed around a single concept and those that have support for many independent application experiences. Typically, a single concept display has an embedded value proposition and it is designed for a specific context of use. However, the developments in interaction techniques, ubiquitous computing infrastructures, services and mobile devices made possible extending the usage case of public displays by employing a diversity of behaviors and thus promoting different usage scenarios within a single display unit. This trend is also seen in the convergence between TV sets and computers. The emerging new computing devices or SmartTVs, conceived for personal and home shared environments, show the many possibilities in which one can extend the default behavior of a TV by employing third-party software applications.

Despite this multitude of display systems, the concept of a third-party display application for public display is a new concept without a common definition and meaning across display infrastructures. This thesis extend thus the work of multi-application displays embedding a diversity of third-party software applications by providing the lacking understanding of what display applications might be, how they can be described based on their properties and what implications they can raise on available technological options and usage scenarios in public spaces.

Chapter 3

Design Space

In this chapter, we aim to reach a consolidated view of the fundamental design principles and challenges that may drive the emergence of third-party application development for public displays. While many different directions have started being explored, there is not yet any systematic work of uncovering the general properties of this type of applications. This contribution frames the entire content of this thesis, and it should help to inform and guide the research agenda towards the vision of applications in Open Display Networks.

A key challenge for this research was the current lack of any established systems where display applications could be created and used in everyday life. The prevailing model in current Digital Signage networks does not consider applications and research efforts have typically focused only on specific parts of the problem domain. As part of PD-Net project (including Instant Places), we took a collective approach in which the entire team was involved in the analysis of multiple systems that have previously explored some variation of this concept of *third-party display application.* This includes a number of application development efforts that have been conducted within the team, but also a representative set of references from the research literature that addressed this particular domain. Even though previous work has not yet reached a global view of the common design principles that may guide the emergence of applications for display networks, we can expect that many of those principles may already be embedded in the many approaches that have previously been explored for application support in public displays.

3.1 PD-Net Application Experiences

The PD-Net project involved the design, development and deployment of a broad range of applications for public display infrastructures. These applications were developed independently for different display systems, and as a whole they provide a broad view of the range of approaches that can be used to create display applications and an important set of insights into the key challenges involved. In the following, we present a set of development and design considerations as informed by the development and deployment of display applications across four independent case studies. The first two, Digifieds and FunSquare, were developed by separate teams as a part of the 2011 UBIChallenge program[16] in city of Oulu, Finland. They represent application examples in which the developers had no control whatsoever over the infrastructure and its services. The other two case studies involve multiple applications developed across two distinct display infrastructures: Instant Places and e-Campus. In these cases, the infrastructures were owned by the application developers and, to some extent, there was a greater possibility to adjust

[16] http://www.ubioulu.fi/en/1st-UBI-challenge-2011 Accessed September 26, 2014

the responsibility between applications and infrastructure. Overall, the development of this diverse set of independent applications for different displays infrastructures with specific technological assumptions has provided us with a broad perspective of the emerging properties of display applications.

3.1.1 Digifieds

One of the applications that have been deployed in UBI-Oulu network (as part of UBI Challenge 2011) is Digifieds, a digital bulletin board application that allows passers-by to place and retrieve classified ads from the display using an on-screen keyboard or by means of a mobile app (Alt, Kubitza, et al. 2011)(Alt, Bial, et al. 2011). While developing and adapting the Digifieds platform for the UBI-Oulu environment developers faced a number of challenges (Figure 17).

Figure 17: Digifieds application running in a UBI hotspot
(Alt, Kubitza, et al. 2011)

Given that all displays are networked, developers first had to decide on which display posted content should appear. While making classified ads available on all displays in the network would have possibly targeted a larger audience, such an implementation would have ignored the opportunity to target particular communities (e.g., students) and a large portion of the content may not have been of interest to the majority of the users (e.g., student flats for residents in close proximity of the pedestrian area in downtown). Hence, application developers used a concept of display groups to which displays could be assigned. As a result, displays could be grouped based on arbitrary criteria, such as location, community, interests, etc. For example, all display on the University campus could be grouped, as could all displays located in the pedestrian area.

A second challenge was how to adapt the user interface to different display sizes. Hence, developers implemented the display client in such a way that it would automatically adapt to different screen resolutions. Thirdly, display owners could have very different requirements with regard to the look and feel of the display (for example, corporate design). As a consequence, the background of the display can be adjusted to the needs of the display owner by means of an administration interface. Finally, we had to find a way to connect the mobile client with the display client. The solution relied in hosting the Digifieds server on dedicated hardware and ignoring the communication through the UBI-Oulu network, but data exchange was realized over Internet, which did not require any particular setup by the users.

In summary, the application developers tried to overcome architectural conditions and constraints by designing for them in the client application rather than to make any changes to the host environment. This enabled a flexible setup that did not affect the overall infrastructure and thus other client applications running on the UBI-Oulu infrastructure.

3.1.2 FunSquare

Another application developed for the UBI-Oulu infrastructure (as part of UBI Challenge 2011) was FunSquare (Memarovic et al. 2011). This application explores the concept of self-generative content or autopoiesic content (Langheinrich et al. 2011), which is content dynamically produced by matching real-time context streams (e.g., the number of people within display proximity) with existing content fragments (e.g., the population of Pitcairn Islands) in the form of "fun facts" based on a set of matching templates. A "fun fact" is a new piece of localized content such as '*The population of Pitcairn Islands (150) is three times more than the number of people near the display (50)*'. Such self-generative content can be displayed by the FunSquare application in two modes on the screens: ambient mode that shows a sequence of "fun facts" (Figure 18) and game mode that shows "fun facts" as trivia questions (Figure 19). Developers faced two main challenges during the development and deployment of the FunSquare application on the display network: gaining access to local display sensors and gathering enough contextual data about the display environment.

Figure 18: FunSquare user interface in ambient mode (Memarovic et al. 2011)

Access to display sensors was possible, but there were often inconsistencies between their expected and their real behavior in relation to the surrounding physical space. The displays feature a 57" capacitive touch screen, a control computer, two overhead cameras, and an NFC/RFID reader. The displays that were used during the development phase were accessing the overhead camera in a different way than the displays used during the deployment, thus, requiring developers to maintain different versions of the code for different displays.

The FunSquare application depends on context streams to produce enough fun facts for each display setting. While integrating context streams was not particularly challenging, finding enough context streams and obtaining enough amount of context data per display was.

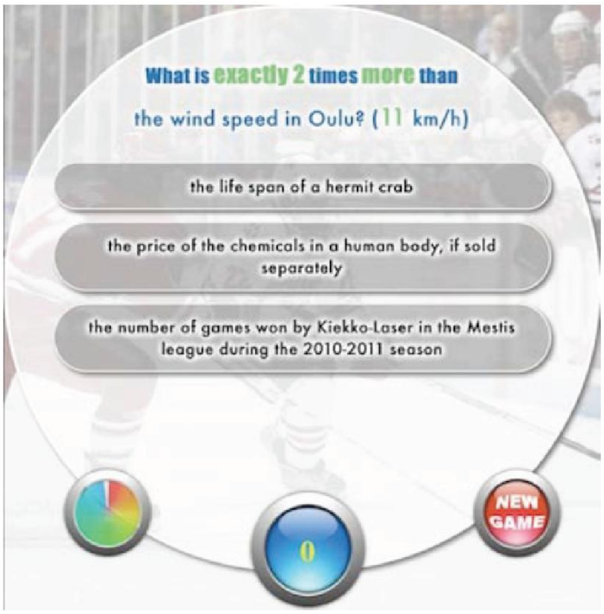


Figure 19: FunSquare user interface in game mode (Memarovic et al. 2011)

The display infrastructure provides three types of context data: Bluetooth information, presence information from overhead cameras, and touch interaction data. This was not sufficient to match with content fragments and create diverse and interesting fun facts. In order to increase the amount of context data developers incorporated data from third-party services, such as local weather, and data from online social platforms, such as Twitter and Facebook. In addition, it was used stream extrapolation and stream averaging techniques. Stream extrapolation converts raw data into other measuring units (e.g., number of present Bluetooth devices into Bluetooth power consumption). Stream averaging simply creates an average of context data over predefined periods of time (e.g., number of people using the display last week). With additional context sources and these two techniques we increased the number of context sources to 29.

3.1.3 Instant Places

Instant Places (José et al. 2013) is a display-centric platform for media sharing that handles sensing and interaction information associated with places where the displays are located and provides an integrated API from which subscribed applications can obtain information about the current circumstances around the display in which they are being used. As part of a Living Lab infrastructure, Instant Places counts a various set of deployments including a University, public cafés, schools and public libraries. Across these deployments, the platform supports a wide range of web-based applications (see Section 2.3.3 for further details); a part of them have been used on daily basis being the subject of various research topics, e.g., (José et al. 2013)(José et al. 2012)(Taivan, José, et al. 2013).

The main findings from Instant Places applications are essentially related with the sharing of place-based information with the applications to enable them to exhibit place-specific behavior. José et al. (José et al. 2013)(José et al. 2012) studied how to incorporate situatedness into the publication of user-generated content. A first example is the Poster application which determined users to reflect on the spatial and temporal characteristics of their posters, when published through the Instant Places online service. While certain users like to control the location/display where their posters will appear (and could control that based on the Android mobile client from where they check in at certain places and publish specific posters), others enjoy the idea of broadcasting their posters on all displays, regardless of location. This latter approach was also supported by the infrastructure in a subsequent implementation so that posters would become available in the entire infrastructure immediately after being created in the online service. Regarding the time dimension, people preferred to choose the maximum time of one month. For the cases when the posters were about scheduled events, users set the poster availability until the day of the event. A second example is the Pins application, which allowed users to control their self-exposure by selecting which pins to activate in a certain place. This allowed people to publish different content for the various places they visit. In conclusion, Instant Places applications allowed users to publish differentiated content across the entire display infrastructure. Situatedness is thus a main feature of Instant Place infrastructure and developers provided the necessary mechanisms so that applications could employ a global behavior, i.e., not being tight to a particular display, and in the same time being able to generate situated information on each of the displays where they are used (Taivan, José, et al. 2013).

3.1.4 E-Campus

The e-Campus infrastructure (Friday et al. 2012) is deployed at Lancaster University beginning in 2004 and has since grown to approximately thirty displays (LCDs and projectors) that support both day-to-day digital signage content and pervasive display research. The deployment runs Yarely (Clinch et al. 2013) an open source cross-platform scheduling and rendering software system. Content for each e-Campus display comes from a variety of content sources distributed through an experimental content distribution platform, e-Channels (Clinch et al. 2011). E-Channels groups content items into shared 'channels' implemented as network file shares. Content producers can manage their channels and content independently of owning a display. Display owners can use a web interface to browse the available channels and subscribe their displays to content from those channels. Study of use of e-Channels showed the need for simple interfaces when managing display content and demonstrates the willingness of display owners to schedule unseen content from trusted providers onto their screens (Clinch, Davies, Kubitza, et al. 2014).

A wide range of applications has been produced as part of the e-Campus deployment. The applications have been developed in a range of programming languages and included both web-based and native applications. Some applications were short-lived prototypes whilst others (e.g. e-Channels) have been deployed for many years. There are three such classes of applications:

Location-aware web-applications – These applications use data about the location of the display in the network and physical world in order to provide content tailored to that location. For example, *e-Posters* allowed students and staff to customize flyers for particular locations on campus. Other applications used the GPS location of the displays to customize feeds shown on the display – for example to show local weather, roadworks and emergency alerts.

Interactive applications triggered and controlled using personal mobile devices – These applications have used technologies such as Bluetooth to allow display viewers to interact with games executing over the display network. For example, a mixed-reality game `Capture the Flag' allowed groups of display viewers to capture displays by gathering together in front of them and having their devices detected by Bluetooth scanners at the screen. The winning team was the one who captured the most displays. Another Bluetooth-based application allowed individual display viewers to use the Bluetooth device name of a personal mobile device request content to be shown on the screen (e.g. 'ec flickr oranges'. A third application, HMessaging provided support for using public displays for person-to-person and person-to-place messaging; person-to-person messages were shown only when the recipient's personal mobile device was detected to be in proximity to one of the displays.

Complex application environments – These applications go beyond the traditional display media formats (images, web, video) using virtualization or native applications to provide highly customizable content. Supporting such media allows mobile users to forage for displays for any purpose. Within the e-Campus deployment, support for virtualization has also been used to explore the combination of cloudlets (Clinch, Harkes, et al. 2012) with display infrastructures and the impact of application location on interaction experience.

The e-Campus applications have provided a range of insights around the deployment of multi-display applications that can be customized based on their environment and the viewers located nearby. E-Campus studies of traditional content placed into e-Channels has shown that content is frequently tailored to the location in which it is shown, and the addition of sensors at the display allows the content to be further customized to improve relevance to content viewers. Usability studies of these various applications have shown that mobile devices can be a viable means of triggering customized content, but the application domain remains open with no obvious `killer applications' emerging. Cloud services could easily act as content sources and hosts for future display applications and the studies of users interacting with remote applications on public displays demonstrate that usability is not always compromised as a result of increased network distance (Clinch, Harkes, et al. 2012). Overall, the experiences of the e-Campus display infrastructure highlighted the potential for personalization and novel applications to add value to digital signage deployments.

3.2 Methodology

The approach to uncover the design principles of applications for public displays is based on a broad and systematic analysis of prior work and also a more distanced reflection on many of the experiences in application development and deployment conducted as part of PD-Net.

In regard to prior work, the team selected a representative set of previous research papers that somehow described displays systems supporting the execution of multiple applications, even if they were not explicitly referred to as software applications. 14 scientific publications were selected, covering different perspectives of the problem domain, from specific application examples to particular display infrastructures. This set of research papers constituted the first source of input for our analysis on the properties and challenges of multi-application display systems. The main purpose of this additional input was to complement the PD-Net application development experience and broaden the range of perspectives being considered for the formulation of the design principles.

To uncover the common design lessons emerging from PD-Net application experiences, the team run a joint workshop in which the developers of all the applications have been involved. The workshop started with the formation of break-out groups composed by 4 or 5 elements to discuss the common features and overall design approaches embedded in the set of applications that had been independently developed. This analysis was particularly focused on the dependencies between the applications and the display infrastructures in which they were based. The explicit identification of these, often embedded, assumptions was very helpful in clarifying some of the key abstractions that may be needed for the emergence of globally available applications that can be used on any public display.

The unique opportunity to confront requirements from multiple applications that were developed on top of multiple infrastructures was particularly important as it enabled a broader view of the application design space. The overall output of this workshop was a consolidated report describing the common requirements across the applications that have been developed for the different infrastructures. The results were also complemented by a set of focus groups conducted with application developers in the context of the entire PD-Net project. The focus groups were centered on the experiences of developers and explored the concept of analytics for display applications. The team conducted two focus groups with small groups of application developers to discuss which usage metrics can be more meaningful to the specific case of public displays and where should they be generated and processed. A total of ten participants, all with prior experience of developing public display applications, were included in our focus groups.

3.2.1 Coding and Analysis

The final step followed an approach based on the grounded theory methodology (Glaser & Strauss 1967), borrowing many of its phases: open, selective, and theoretical coding; memos; and sorting. For this phase of the study, it was used as input to a coding process the set of 14 scientific papers referenced as prior work, the PD-Net research work on applications, the final report from the development workshop and the report from the focus groups.

The coding was meant to identify any references to application properties across the various displays application concepts. Using coding software, the process started with an initial phase of open coding, in which was produced the first set of codes corresponding to specific application properties. Each property was marked with a code and also a short memo describing its meaning. Then, it followed an organizing of this initial set of codes into categories of properties, seeking for similarities and comprehensive aggregations.

We adopted the definitions of categories and properties from Glaser and Strauss (Glaser & Strauss 1967). A category stands by itself as a conceptual element of a theory. A property, in turn, is a conceptual aspect or element of a category. To identify and distinguish categories, the analysis involved the similarity between the properties and their ability to fit into a common one sentence description. The property categories and the resulting aggregations were then used as the starting point for a second round of coding in which the same papers were recoded. While categories were now the primary frame of reference for codes, still new codes could be created whenever an existing concept was being difficult to associate with them. In the end, the process originated a total of 16 concepts, corresponding to categories of properties that were referenced in 117 text segments across the 16 input documents. The respective text segments were then used to create the description of those properties. To further structure the results, the 16 applications properties were also aggregated by their affinity under 6 higher-level concepts corresponding to more generic design principles. The end result is a structure composed by 6 fundamental design principles that altogether comprise the 16 envisioned properties for applications in public displays.

3.3 Design Principles

The key result of this work is the enumeration of six fundamental design principles for display applications, along with the characterization of the 16 design properties, as summarized in Table 11.

3.3.1 Global Availability

A fundamental motivation for open display networks is to break the association between applications and particular displays. This would enable application developers to make content available anywhere across a global network of open displays and allow display owners to select applications from a wide range of globally available offers. The first design principle is thus the global availability of applications, in the sense that they should be usable in displays anywhere. We have identified two main design properties associated with the principle: a distribution service that allows display owners to discover and activate the applications they need and the existence of meaningful usage analytics that may assist them in the process of identifying the most relevant applications from a potentially very large pool of applications.

3.3.1.1 Distribution service

In a global application model, applications can be developed and hosted anywhere to be used on an open ended set of displays. However, simply making them available to the display network is not enough. Display owners need a way to find out about their existence, assess their suitability, and manage the association of the application with the displays, e.g. configuration or subscriptions. A key implication of global availability is therefore the need for a distribution service that can address the challenges of how to distribute applications to potential users and subsequently manage the relation between application providers and application users. This service should enable global access for both potential developers and users, reducing distribution effort and allowing even smaller development groups to launch application that may reach the same large audiences as those from more established companies.

3.3 Design Principles

Table 11: Key design principles and properties for display applications

Principles	**Key application properties**
Global availability The application is available for being used anywhere across a global and open network of public displays	[**Distribution service**] An application distribution service exists to enable application creators and applications users to gain access to each other and manage application usage. [**Analytics**] A system of usage analytics allows app creators to assess how their apps are being used and allows app users to make more informed decision on which apps to select.
Adaptability The application can be used across the many different types of display and execution environments of an open network of public displays	[**Portability**] The application is highly portable and can be run across a potentially very heterogeneous set of computing platforms. [**Visual adaptation**] The application can be conveniently presented across a broad range of displays with diverse size and form factors. [**Abstract interactions**] The application can be used through the diverse set of interaction techniques that may be available at each display.
Situatedness The same global application can exhibit a specific situated behavior at each of the many domains in which it may be in use	[**Explicit configuration**] Application behavior integrates local knowledge and preferences through an explicit configuration. [**Application local state**] Application behavior on a particular domain implicitly evolves with previous interactions and events at that domain. [**Environment model**] Application obtains information about the specific environment in which it is being used and adapts accordingly.
Content Management Application content is managed and exposed to optimize its use in public displays	[**Content placement**] The application content is placed in a way that addresses the specific requirements of public displays in regard to loading times and off-line behavior. [**Actionable content**] Application content is exposed as addressable and actionable resources.
Concurrent Applications At each domain, many applications are available and are being concurrently used by multiple users	[**Discoverability**] Users should be able to discover which applications are available at a particular display and possibly request their services. [**Coordinated scheduling**] The application coordinates with its scheduler to increase the overall value of scheduling decisions. [**Shared interaction**] Interaction channels on the public display, both input and output, may be shared among multiple active applications.
Display Ecosystems At each domain, the same application can be run across an ecosystem of large and personal displays	[M**ultiple concurrent users**] Multiple users are concurrently interacting with the various applications available in the environment. [**Multiple end-points**] An application is available in different ways across multiple devices in the environment, and not just public displays. [**Coordinated display groupings**] Applications can identify display groupings and treat them as a whole.

Clinch et al. explored the concept of application stores for display applications as a trustworthy method of distributing applications within an open display network (Clinch, Davies, et al. 2012)(Davies et al. 2012). An app store for public displays could play a central role in opening display networks to new applications from a wide range of sources, and eventually do for public displays something similar to what mobile app stores did for mobile devices, i.e. to open up what was before a closed model of development and deployment.

However, despite the many similarities, it has been identified some fundamental differences between the market of mobile applications and what could be a market for public display applications. First and foremost is the set of stakeholders; mobile devices are typically owned and managed by a single individual, with third parties providing communication services. In contrast, display networks feature a stakeholder set including display owners, space owners, viewers (users) and content (application) providers. In most cases, the owner of the display is different from the viewer of the display. For example, an advertisement outside a shop is managed by the shop owner, but the content is aimed at people passing on the street. Therefore, while application stores are primarily described as aiming to allow display owners to obtain applications for their displays, they could also allow users to subscribe to applications with which they could then appropriate some displays. It has been identified at least three different ways to combine the role of the various stakeholders involved (Davies et al. 2010): applications that primarily support the display owner, applications provided by the display owner for the display users and applications that are owned by the display users where the display owner provides only the run-time environment. The display owners themselves may be the persons buying applications from the store, in which case they will expect control over application scheduling, but, alternatively, applications may be purchased by users who expect them to be shown when they pass by a display. The system should thus address and provide appropriate interfaces for controlling scheduling (when/where an application is displayed) for display owners, application developers and users. In an open network, the process of selecting and scheduling content from an app store becomes more challenging as the user must describe an ever-changing set of displays. The consideration of cost/benefit exchanges of these multiple stakeholders raises considerable design challenges in regard to the nature of business models, which will need to reconcile the conflicting demands of those stakeholders, and more specifically who should pay for the applications and how such payment models impact on the acceptance by display owners of applications that primarily benefit the user. An early attempt at defining an API for these application stores, supporting services for the distribution or selection of applications is described in (Clinch, Davies, et al. 2012) and fully specified on the PD-NET website[17].

3.3.1.2 Analytics

In a world of many application offers, information about their usage is likely to play a key role for the various stakeholders. For application developers, this information will be crucial to improve the application and assess its success. For application users, usage data may provide a collaborative model for establishing the relevance of particular applications

[17] http://pd-net.org/api-docs/ Accessed September 26, 2014

for specific purposes or contexts. Display owners would also want to know what is happening on their displays and how much time people are spending on the various applications. Analytics may enable them to select the apps that are most sought-after by users, both from the set of apps supported by the display and from the (yet) unavailable ones. This may prompt display owners to subscribe to new apps, or give certain apps more or less screen estate or time.

Regarding usage metrics, we observed a broad range of preferences for different stakeholders. Usage data may include the number of distinct displays the application is shown on, the geographic location of displays, the hardware and software environment at a display, applications scheduled before or after this application, scheduling conflicts, details of application failures, contextual and demographic data about users, implicit interactions with the application (e.g. presence, attention changes) and explicit interactions with the application, such as the number of times the application is selected/launched, interaction patterns within the application, e.g. length of gesture sequences or scrolling events, and intentional reporting about the application, e.g. ratings, reviews.

Developers from the focus group have clearly expressed an interest in metrics that were primarily focused on optimization for the application itself, e.g. bug fixes or taking advantage of new hardware trends. They also mentions an interest in optimizing the application`s exposure, e.g. by looking at usage patterns to improve relevance or broaden the usage demographic, or by actively publicizing positive ratings, reviews and even the percentage of the population who spend time looking at the application. However, this level of analytics for display owners should not allow them to "spy" on individual apps and their users' interaction, e.g., while inputting personal information. Usage analytics also needs to take into account privacy, and particular take into account that not all viewers of the displays see themselves as users of the system. To comply with privacy requirements that data should be aggregated or altered to ensure that viewers are not identifiable.

Regarding the issue of where should those metrics be generated, multiple complementary approaches have been identified by developers. The first is for the app itself to generate and register its own metrics. This is the more flexible approach and may serve improvement goals for app developers, but it does not enable comparisons across applications, e.g., to assess overall app popularity. Analytics should thus also cut across different system elements from display nodes with privileged access to information about local context, to app stores with a global view of application usage, to third-party services (similar to Google Analytics) that would aggregate actual use across multiple displays.

3.3.2 Adaptability

A global application will be used across many displays with potentially very diverse characteristics in regard to the supporting hardware (computing platform, display size, type and resolution), the surrounding physical environment and, in general, the way they can be experienced by people. Some of these variations can significantly impact the behavior of an application or even the viability of its execution in a particular display setting. To overcome this challenge, global display applications should avoid major assumptions about specific properties of the target display settings. They should also include adaptation mechanisms that allow them to mask the uneven conditioning (Satyanarayanan 2001) that they are likely to find across multiple display settings, either by trying to hide those variations or by compensating them or explicitly stating minimal execution requirements. These variations, and therefore the respective adaption needs to occur at different levels. Within the PD-Net project, we have identified the following three main forms of adaptability: to multiple computing platforms (*portability*), to different layouts (*visual adaptation*), and to diverse interaction technologies (*interaction abstractions*).

3.3.2.1 Portability

The heterogeneity of the computing platforms associated with the public displays is a direct consequence of the openness of the display network. Portable applications should be able to work across multiple computing platforms. The use of web technologies as a technological framework for display applications is the most common approach for portability in the application frameworks that we have studied. The Oulu framework (Linden et al. 2010), Magic broker (Erbad et al. 2008) and Instant Places (José et al. 2012), they all resort to Web technologies as an approach to support seamless integration and deployment of third-party services residing anywhere in the public Internet, thus catering for openness and scalability, support for heterogeneous clients, cross domain interaction and web-oriented application design. Despite the obvious potential of web technologies, there are also some fundamental differences between the dominant usage model of the web and the usage model of public displays. Simply rendering content on a browser is not adequate and specific solutions that package web engines with the display-specific functionality may be needed to fully explore that potential (Taivan et al. 2014a). The use of virtual machines (Linden et al. 2010) or cloudlets have been explored as alternative approaches to overcome some of those problems, particularly in regard to easier deployment and reduced latency in interactive applications (Clinch, Harkes, et al. 2012).

3.3.2.2 Visual adaptation

Global display applications will have to render content on displays or display regions with very diverse properties and for which the size and form factor were not known a priori. This raises the need for visual adaptation procedures that can deal with such diversity while rendering appropriate visualization of the application content. This challenge is to a large extend similar to the same challenges that led to the principles of device independence and Responsive Web Design (Marcotte 2011), which have now become a de facto standard practice in web development. At least for Web-based applications, Responsive Web Design encapsulates a set of technologies that allow developers to create web applications that are independent of screen resolution and orientation and are able to adapt their web content to the display characteristics.

However, responsive design may not be enough to serve the visual adaptation needs of public displays because the range of content adaptation can be much broader for public displays than what is normally expected in smartphone/tablet-oriented development. Although display size and orientation generally fit into a well-known range of options (1920x1080, 1280x720, 1366x768, etc.), most digital signage services support display partitioning into multiple custom size containers (content presentation slots), e.g., horizontal bottom bars, or sidebars, leading to much more uncertainty about possible displays sizes and properties. Content may need to be rendered on small displays or small regions of a large display, but it may also have to fill an entire display wall.

Additionally, visual adaptation in public displays may have to go far beyond resizing adaptations to match the display size. Some displays may be interactive, touch-sensitive or even multi-touch surfaces, while others may simply be rendering content. Another key difference is that visual adaptation is not just determined by the display size, but also by the viewing conditions. The same content on the same type of display at two different locations may be best viewed with considerably different visualizations simply because of the nature of the space surrounding the display and how it affects viewing. Similarly, future displays may come in multiple shapes and form factors that will require specific forms of adaptation. Addressing this type of adaptations may thus require meaningful descriptions of the visualization contexts associated each public display that may also enable these other forms of adaptation.

3.3.2.3 Abstract interactions

Current display systems typically offer little or no interaction at all. Those that do, use a broad range of modalities including touch/multi-touch, video and interaction though use of mobile devices (e.g. via SMS, Bluetooth, or through a mobile application), Radio-Frequency Identification (RFID), gestures or face detection and we can only expect that still others modalities will emerge as possible contenders.

While interaction can easily be achieved for a specific display system with a particular interaction modality, the lack of proper interaction abstractions means that there is considerable work that needs to be done to support even basic forms of interaction. Additionally, this is an effort that must be replicated by each developer, representing wasted effort and leading to inconsistent interaction models across different displays. Users of these public displays are faced with very different interaction models and inconsistent interfaces across and, sometimes, within displays, which inhibits knowledge transfer about how to interact with different displays.

For applications developed for a global display network, this represents a major challenge, as the application would have to be used throughout the diverse set of interaction techniques that may be available at each of the displays in the network. In the same way that many mobile applications must handle a heterogeneous set of mobile devices (e.g. with different screen sizes, sensor configurations and cameras), we expect that display applications will need a degree of flexibility to make the best use of the interaction modalities available. Therefore designing interaction for applications to be launched on a public display network requires developers to consider the minimal requirements and how to create the best experience within the bounds of even a poor resource set. The problem of managing interaction is compounded when considering the creation of an application store because of the expectations generated when users pay for applications.

A possible approach is to describe interaction at an abstract (technology free) level to create an abstraction layer that can be mapped to multiple forms of interaction. With this type of abstractions, application developers would not need to worry about which mechanism is available at any given display. An application developed for a public displays should work transparently in places with support for different sets of interaction mechanisms. A first step in that direction is the work by Cardoso et al. and its PureWidgets toolkit that enables applications to focus on the type of data they need and then map that type of data entry into the multiple interaction modalities that can be available across displays or even on a single display (Cardoso & José 2012).

Similarly, the MAGIC Broker (Erbad et al. 2008) system also enables support for the various interaction patterns required by multiple interactive applications. Users can interact with the large screen by calling a Voice XML gateway, by sending SMS messages, or using a mobile web browser. Events are used to send information between the user's mobile device via SMS, the VoiceXML gateway, or the mobile browser directly to the large screen display channels.

Despite the obvious need to support multiple forms of interaction, an additional challenge will also be the emergence of consistent expectations on how to interact with applications on public displays. One of the primary success factors for interaction with mobile applications is that platforms typically provide a standard set of definitions for the look-and-feel of both applications and their interaction mechanisms (e.g. consistent touch-keyboard positioning and layout). The benefits of common interaction mechanisms for many different applications would lead to emerging practices and expectations in a way that would be central to sustained user engagement.

3.3.3 Situatedness

Situatedness is the ability of a global application to generate, on each of the locations where it is being used, content that is deemed specific to that location. While adaptability was concerned with the need to provide a similar service despite the differences between various execution environments, situated behavior is concerned with the need to generate differentiated and local content from a single global application. Without some form of situatedness, application content would be the same irrespective of the locations where it is being used or the people viewing it. Situatedness should thus enable a global application to exhibit a behavior that is specific to each of the many environments where it is available and the actions of the people on each of those displays. In PD-Net project, we have identified three major approaches to incorporate situatedness into global applications: explicit configuration by a local manager (*explicit configuration*), progressive adaptation to the environment based on interaction history (*local state*), and access to a formal environment model (*environment model*). It was also clear that these different strategies can easily be combined for maximum flexibility.

3.3.3.1 Explicit configuration

The most basic approach to enable situated behavior is by supporting some configuration mechanism through which the application users are able to explicitly change the behavior of the application for a particular display or group of displays. An explicit configuration is a simple process for enabling local knowledge to be embedded into the behavior of the application. To be combined with the global availability of the application, these configuration procedures should be independent per each of the displays where the application is being used.

For example, while the FunSquare application was highly situated, it was configured specifically for the specific display network where it was deployed. Each application instance would have a single state and a single set of configurations. For running in a global network and still be able to offer its situated content to each of the many displays where it could be in use, the application would have to support multiple autonomous instances and configurations that can apply the same application logic to different data spaces associated with each of the domains. This approach was partially explored in Instant Places, where applications exposed a configuration widget that enable place owners to explicitly configure how that application would behave when being used on their displays (Taivan, José, et al. 2013).

3.3.3.2 Application local state

Instead of forcing an explicit configuration process, applications may implicitly create and build their own state for each of the specific display domains in which they are being used. This app local state would be application specific and would emerge from the set of previous interactions with that application at that domain. This approach was used in the Bluetooth version of Instant Places (José et al. 2008), where applications were able to start with a default behavior and then progressively adopt a situated behavior that reflects the tags that have previously been shared by users.

3.3.3.3 Environment model

The third situatedness approach is to keep a formal model of the environment that aggregates data about the physical and social setting around the display and is shared with the applications. This way, application can continuously reflect that information on the content they present. For example, the FunSquare application (Memarovic et al. 2011) uses data from the immediate display vicinity as well as from city-wide sensors, to continuously generate locally relevant “fun facts" out of a set of non-localized fact sources. Transformation rules are used to find facts that match a particular piece of local context. For example, a rule might use the average number of Bluetooth-enabled devices in the vicinity of the display and combine it with both the average amount of power used by a Bluetooth module, as well as the power of a light bulb, to generate a statement like “*Did you know? The energy spent by all Bluetooth devices in the vicinity of this display in the last 24h is enough to drive a 100W light bulb for exactly 2 seconds*".

Instant Places supports the creation of places that may be seen as profiles of the displays environments. These place profiles combine explicit place descriptions with dynamic data implicitly generated by people who actively engage in interactions around the display (José et al. 2013). Based on the information from those profiles, applications on the display are able to generate content that is tailored to the circumstances around the display. For example, the system allows local viewers to manage the projection of their identity across multiple public displays using pins that are virtual representations of a lapel pin or campaign buttons, allowing the wearer to express support, for example to a sports club or political party. By detecting and aggregating the presence of such virtual pins in the vicinity of a display, any display application can reflect the interests and preferences of its viewers.

Elhart et al. describes the concept of Situated Memory (Elhart & Langheinrich 2010) as a profile built from the past actions of users in a place. At an individual level this may correspond to the actions the user has taken in that place. At an aggregate level, the combined profiles of place visitors could be seen as forming a "community fingerprint" of the place. This could include information generated implicitly as part of local actions or information explicitly contributed by users (e.g., ratings, reviews, mood messages). Together these fingerprints could create a narrative of a place showing the list of visitors, their activities, interests, and thoughts over the time.

These systems assume that a public display should exist as part of a social space which is explicitly represented by the system and can be used as a major driver for the system behavior. The system should thus be able to sense, process, and store information about a place and its visitors to provide networked displays with a sense of their environment, both physically and socially. An alternative approach is the one explored by Tacita (Davies, Langheinrich, et al. 2014), where the concept of display environment is explicitly avoided to prevent privacy concerns associated with the need to trust the display owners. Instead, Tacita advocates that participants should form a trust relationship with an application provider, which would then coordinate with the displays to offer personalized content without disclosing any personal information.

3.3.4 Content Management

To a certain extent, one of the key motivations to move from simple content to applications is the possibility to have application logic that can deal in a specific way with its own content. However, an important design lesson from PD-Net deployments is that applications for public displays need to consider the specificities of content management for public displays and optimize their operation accordingly. It has been identified two main properties associated with this principle: the way content is placed by the application and how it can be handled outside the specific scope of the application. A detailed analysis regarding the applicability of this principle on Web technologies as an appropriate technological framework for the creation of display applications is described in (Taivan, Andrade, et al. 2013)(Taivan et al. 2014a)(Taivan et al. 2014b).

3.3.4.1 Content placement

In a typical on-line interactive application, e.g. web browsing scenario, content is mainly user-driven, interactive and optimized for individual usage. The underlying assumption is that a user is controlling the flow of content requests, making situated decisions about where and when to go next, and will be able to deal with unexpected events such as broken links or failures. In a public display, most scheduling decisions will normally be done by the system itself, which thus needs to decide at each given moment, what resource to display. Optimal content placement, caching and delivery will thus become crucial for allowing the system to provide a positive user experience and function reliably during network disconnections, power outages and software failures.

Firstly, proper content placement is essential for ensuring that displays have the content in time to offer good aesthetic performance and low latency for interactive use (Davies et al. 2012). Any idle time during which a resource is being fetched from the server should never correspond to idle presentation time, as watching the system stop while content is being loaded completely destroys the user experience. Given their previous experiences, people expect the same smoothness and performance in the way applications present their content that they are used to see in traditional television broadcast and even in other existing display systems.

Secondly, proper content placement is also essential for fault-tolerance. In the traditional web browsing experience, when a content resource cannot be obtained, the result is a message error notifying the user about the problem and possibly giving additional indications on how to proceed. On a public display, content loading errors should never result in error messages being shown, because those people who would see the message might have not requested the content and probably cannot act to solve the problem. In addition, when a content source fails and a resource cannot be obtained, displaying error messages is not a solution as viewers do not really need to know that it was not possible to show a resource that was scheduled for presentation.It should thus be possible to detect that something went wrong when loading a particular resource and initiate some type of fault tolerance mechanism, possibly showing an alternative content or moving seamlessly to the next available item. Errors should be caught before they show up infamously on the screen and report them through some alternative channels.

Prefetch can play an important role in addressing both these issues and is thus an essential requirement for applications in public displays. Prefetch is the process of previously fetching resources from a server in anticipation that they will soon be needed. It significantly reduces resource presentation times, preventing people from seeing a white screen during application loading. Additionally, it also enables the system to know in advance that the resource is available, as we would not want the system to fail the presentation of content that it was not able to fetch. Prefetch may also enable the display to operate even when facing occasional network disconnections. However, it is likely that a display application would not able to cache all its data and components, but it should at least be able to support a graceful degradation mode based solely on resources that are already locally available (Lindén et al. 2012).

3.3.4.2 Actionable content

Even though display applications may be presented as an alternative to the simple distribution of content, this does not mean that applications should follow a black-box approach for handling their own content. It was identified several references to situations where it may be advantageous for applications to follow an open approach in regard to their own content.

Rather than being hidden inside the application, content resources in use by applications could be exposed to optimize their value to the overall system. The specific nature of what a content resource is may vary considerably between applications, but in general it should correspond to a content item that is addressable and actionable. This exposure of application content may facilitate multiple types of integrated behavior around the same data, such as external logging, social actions, content download or data exchanges with other display applications, as described for example in the iRoom scenarios, where the Data Heap facilitates data movement by allowing any application to place data into a store associated with the local environment (Johanson et al. 2002). It may also enable mobile applications to list and access the resources being shown on the display as described in Instant Places (José et al. 2013). In general, content addressability creates a content reference that can be used network wide and support a number of actions on content, including ratings, popularity assessment and other analytics.

This openness should also be extended to content creation. Instant Places deployments (José et al. 2013) have highlighted how people can resort to a very broad range of tools to create similar content and the importance of letting users leverage upon their favorite tools. It has also shown the importance of repurposing content from other services, especially social network sites. In these cases, the use of external references may also enable important forms of content addressability and cross-referencing. The Magic Broker deployments have shown how the system should provide easy and seamless tools for content providers and developers and highlighted how web-oriented tools and standards, such as HTML, JavaScript, Flash, PHP, Java applets and Java servlets could facilitate the creation of high quality content as many content providers and developers are familiar with such tools (Erbad et al. 2008).

3.3.5 Concurrent Applications

Previous work on public displays has often considered single-purpose displays representing a particular usage concept. However, a key motivation for Open Display Networks is the ability to allow display owners to benefit from a wealth of applications that may be globally available for being used on any public display. Displays are thus much more likely to exhibit a functionality that corresponds to a composition of applications rather than just a single one. These applications will be concurrently running on that display, and any particular application can only expect to be one of many trying to access the environment resources. This co-existence of applications raises challenges regarding application discoverability, the need to coordinate the scheduling of these various applications and also the need to share interaction channels, which can no longer be assumed as exclusive.

3.3.5.1 Discoverability

The availability of many applications associated with a single display necessarily ends-up leading to the issue that they will not all be relevant in the same way all the time. Instead, some mechanism should exist to support the launch or to give additional visibility to an application upon request. This, in turn, leads to the issue of discoverability, and how to make users aware of the applications available through the display so that they can request them (Hosio et al. 2013). Applications will have to describe themselves and be ready to accept requests, but the existence of a large number of applications will also raise specific discoverability issues. One of the challenges is how to convey the information about the available applications, as most public displays will not offer appropriate browsing affordances (Taivan, Rui José, et al. 2013). Another challenge regards customization in the way that is normally done at a mobile device where a single user owning the device can fully customize application presentation according to personal preferences. In a public display, the organization of applications cannot be based on personal preferences, and research has shown that the use of different layout schemes across different displays can lead to disorientation and an overall sense of being lost when searching for applications (Kostakos et al. 2013). Again, some form of coordination may be needed to present as a whole in a coherent way the application offers available at any particular display. A metaphor based on department stores, in which the store window displays only some products (applications) to draw customers in and then as people enter the store (begin interacting with the display), all the products are available in different departments (application directory) is suggested (Kostakos et al. 2013).

3.3.5.2 Coordinated scheduling

Having multiple applications competing for the space and time on the screen as well as for other resources in the environment, means that a scheduling service is needed to support the dynamic run-time partitioning of the screen real estate between multiple concurrent applications both in spatial and temporal dimensions (Linden et al. 2010). The scheduler service would control the way multiple concurrent applications can schedule content across the shared display network (Storz, Friday & Davies 2006), possibly enabling those applications to be spontaneously invoked by users. In this latter case, the possibility to switch between applications means that applications need to be ready to be halted and resumed at any moment, as requested by users. This behavior in which users can issue application requests, determines that the scheduling service should include both static timeline/playlist schedules and dynamic context-based scheduling.

While the concept of global application may somehow imply that application developers will not have any influence over where or when their applications will be used, developers in the focus group have expressed interest in retaining some control of the distribution by providing scheduling advice that would impact when or where their application would become visible and the degree to which it should be seen as relevant in a range of contexts (Clinch, Davies, et al. 2012). This scenario involves that applications should be able to exchange scheduling information with their execution environment. Rather than treating applications as black boxes, a scheduler would use these coordination possibilities to optimize the overall system behavior.

In addition to the collaboration between applications and display system, the potential in the interaction between the various applications operating within the same system might be a key feature in Open Display Networks (Davies et al. 2012). However, almost all systems in PD-Net study seemed to assume that such coordination will be mediated by a specific scheduling entity that will provide a centralized decision point. In general, applications are seen as autonomous entities (Taivan & José 2011) and a loosely coupled model that avoids strong dependencies between them was preferred.

3.3.5.3 Shared interaction

The other potential implication of a multi-application environment is the need to share the output and input channels. This is not normally an issue for displays based on a walk-up model, where a single user can normally get exclusive access of the display., e.g. Oulu touch-based displays (Ojala et al. 2012). In those cases, an application is first selected for interaction and then gains exclusive access to the interaction channels, including at least part of the display.

In other cases, however, the display may be running multiple interactive applications concurrently, all of which may at any time need to generate some sort of output or become the target of user interaction. Since the display is likely to be time-shared with other applications, no application can assume to be visible at all times. They should only expect to have a small slice of display time or even to be shown only upon explicit request. Still, they should be available for interaction at any time, regardless of whether or not they are currently being shown on the display. If that happens a feedback mechanism on the display may be helpful to acknowledge interaction and signal to others the existence of interaction events. Applications should also be designed to identify themselves when being show on the display. Regardless of the specific content being shown, the perception of which application is generating the content can be an important element to understand the context of that content and manage interaction expectations. This may include an initial splash screen clearly stating which application is now the active application and even some smaller identification visual cues that allows anyone occasionally looking at the display to understand what application is being shown (Taivan et al. 2012)(Taivan, Rui José, et al. 2013).

Input channels may also be shared with multiple applications and conveying interactive features may become much more complex. Not only, each application may have its own instructions and calls to action, but also any input channel may need to be disambiguated in regard to the application being targeted. For example, a single SMS number may exist to enable the sending of messages to the display, but, given that may applications can be targeted, additional targeting information may need to be included in the message to direct the input to the proper application. The same problem also applies to implicit input channels, such as those used to support audience models based on proxemics. In particular, it may have a significant impact on many of the audience behavior frameworks proposed in previous work, which essentially assume a single application display (Michelis & Müller 2011)(Brignull & Rogers 2003)(Prante et al. 2003). These frameworks aim to model the various phases of user engagement with displays, and they normally associate the ability of the display to react to audience changes. In a multiple application setting, many applications will be sharing the audience and therefore, rather than one audience, many different audiences may co-exist around the same multi-application display. The challenges of how to design engaging interactive experiences as a whole, attraction loops or conveying interactivities, may become much more challenging because no single application can assume full control of the process. As described for coordinated scheduling, this may also require some level of coordination between each individual application and a coordination entity that uses a broader knowledge of the situation to optimize the whole system behavior.

3.3.6 Display Ecosystems

Digital displays can be found in many different sizes and form factors, from very large shared displays to small personal displays on mobile devices. Using the concept of "*ecosystem of displays*" as introduced by Terrenghi et al. (Terrenghi et al. 2009), we could generally describe the public display environment as perch/chain sized ecosystems for many-many interaction, composed of displays of various sizes (from handheld devices, to medium/large wall mounted displays), and where "*many people can interact with the same* *public screens simultaneously*". We have identified three main implications for display applications that result from designing for this ecosystem of displays: supporting multiple concurrent users, supporting multiple end-points per application; and being able to address and coordinate displays groupings as a whole.

3.3.6.1 Multiple concurrent users

By its nature, public space will have multiple users, devices, and applications, all simultaneously active. In particular, we consider that more than one user may need access to the same application at the same time, and thus concurrent and shared interaction by multiple users should be possible. Previous work has not yet addressed the issues raised when users interact with multiple applications on a single public display (Hosio et al. 2013). Instead, it has mainly addressed the challenge that the display itself could be a shared device for which there was a need for floor-control mechanisms to manage conflicts, e.g. the PointRight device to gain control of displays in IRoom (Johanson et al. 2002) or the way in which the Dynamo system enables users to carve personal partitions of a public display for sharing their own media (Izadi et al. 2003). Work on single application displays with multi-user support includes the WebWall application, which presents its different services and the source of their input through the mobile phone of the senders (Ferscha & Vogl 2002). Similarly, applications like Jumbli (LocaModa 2014) allow various users to submit words at the same time and then show the scores of those concurrent users. A first implication from concurrent multi-user interaction would thus be the ability for each application to accept and acknowledge concurrent input.

For those applications that aim to personalize their behavior according to the audience, the simultaneous presence of multiple users to be served with personalized content will also become challenging. The existence of user profiles would enable people to systematically manage their exposure and content publication in public displays, and possibly indicate a number of preferences about applications behavior. This personalization to multiple users would have to find a proper balance between the specific preferences of each particular user and common preferences between all users. The adaptation process should consider the best strategy for dealing with the potentially varied interests expressed by those people. This generates a trade-off between the selection of content based on a profile combing the multiple interests of the multiple persons present and the selection based on the use of each individual profile, one at the time (Alt et al. 2009). The first is a balanced approach, but faces the risk of not really matching anyone's specific interests. The second approach can be targeted for each individual, but it raises additional privacy issues and may conflict with the idea of the public display as a shared and place-based medium.

Another key challenge for concurrent interaction in public displays is directing timely and appropriate feedback information to the correct user, on the correct channel, so that users understand the result of their actions. Given that public displays are shared devices that can be used simultaneously by multiple users and applications, it is not obvious what the feedback information should be and how it should be communicated. Most applications will need, at least, to be able to distinguish between different users in order to provide a shared environment where users know who's requesting what. In many cases, they should also maintain their interactivity independently of whether or not they are currently being displayed on the screen. In a multi-user, multi-application environment, being on the foreground in full control of the display cannot be expected to be the rule, and thus they should be prepared to maintain interaction regardless of their on-screen state of the application.

However, in a context where many people may be concurrently interacting with different applications on the same display, there is an additional layer of complexity resulting from the need to combine the conflicting goals of those users in a way that is fair and clear for all of them. At any moment, it is likely that there will be different types of users, and especially users at different stages of the interaction process, e.g. bystanders or participant. Dix and Sas (Dix & Sas 2008) analyze two main types of conflict that occur between the interacting users of the public display audience: conflicts of content that occur when there are different goals about what should be shown by the display and conflicts of pace, which occur when users have to wait to be served because other users are also present. Again, this is an issue that is hard to solve at the application level and for which some coordination between application and the display system would be needed to approach a balanced solution.

3.3.6.2 Multiple end-points

Large public displays will normally be surrounded by many smaller displays corresponding to tablets, mobile phones or other personal devices. For applications, this is not so much an adaptability challenge in the sense of trying to show the same content on any of the device types. It is mainly an opportunity to explore the different affordances of those devices to achieve the best possible user experience. These various types of devices can be in simultaneous use, with each being chosen for its efficacy in accomplishing some specific task (Johanson et al. 2002). The bigger displays (perch/yard sized) may be seen as the main information outlets of a place, providing a shared information and interaction point for the whole place – they are public, visible to everybody at all times (usually located in high-visibility locations), and can function as the reference display in a place. Medium-sized displays (yard/foot sized) may also be present and be used, for example, to provide an interaction point (using touch interaction) that shows some of the most important interactive features that are locally available. Finally, small displays (foot/inch sized) are typically the personal mobile devices such as smart-phones, tablets, or laptop computers owned by place visitors. Users that own these kinds of devices will want to take advantage of them to interact with their environment, including the available public display applications. They are normally seen as privileged input devices to the public display application, supporting a covert interaction model. However, they can also be used for output purposes, under a "dual display" paradigm that understands applications as executing across large displays and mobile devices and utilizing the input and output capabilities of both device types (Kaviani et al. 2009).

When we consider this display ecosystem, applications can no longer be seen as being tied to a single device/screen during execution, and running only on the public display itself. Instead, the entire display ecosystem becomes the execution environment. Whether an application is being shown on a particular display at a particular moment becomes irrelevant as they are expected to always be available across the whole environment with its possible multiple large displays and its many smaller devices. These multiple displays are not just close to each other. They share some information environment that enables them to coordinate their actions and maintain a shared state. This requires the actor's mental model to not only include knowledge about the actions the system supports but also to understand where to execute these actions and where to perceive the new system state (Kaviani et al. 2009). These multiple displays can thus be seen as multiple endpoints for the applications, optimized for the particular capabilities of those devices. For example, in Digifieds, the digified being shown on a display could also be available on a digifieds endpoint on the mobile device and allow the user to see complimentary information about it.

3.3.6.3 Coordinated display groupings

A unique feature of open public display systems is their ability to create experiences that span across multiple displays to achieve a combined effect. In this scenario, a group of displays collaborate and coordinate behavior offering to users an experience that is perceived as a whole, even if spanning across locations. For this to be possible, applications should be able to address multiple displays as a whole, based, for example, on location or proximity criteria. We identified three different forms of this type of coordination: synchronous, sequential and parallel.

Synchronous coordination assumes multiple displays at the same location offering a fully integrated experience. Davies et al. (Davies et al. 2012) describe the public display equivalent of a "bouncing ball" screen saver for a network of public displays in which the ball bounces from display to display within some specified geographic region. In this case, the experience is only effective if the displays are all nearby and supporting fine-grained coordination. Applications need information about the local displays and their physical relationships (where displays are relative to each other). They would also need a mechanism of scheduling the display of the ball in the context of other content that needed to be presented with very precise timings and they would need a way of addressing each screen so that the ball could be displayed appropriately. As the ball traversed between screens, other content would need to be pre-empted to satisfy the short-time scheduling requirements of the bouncing ball application.

The sequential coordination scenario assumes multiple displays at different locations. It is based on the assumption that a moving user will encounter sequential content as he or she goes across a number of displays one after another, e.g. while walking along a high street. The parallel coordination scenario corresponds to the case where displays at different locations are coordinating to provide a joint experience to the respective audiences. This might be a shared game or some form of window metaphor interaction in which people at both locations can perceive their mutual presence and interact.

What all these scenarios have in common is the need for some sort of abstraction for mapping content to particular displays in the real world and for addressing the displays or groups of displays that may be involved in a particular form of coordination. This type of fine-grained coordination will need new protocols that can enable large scale multi-screen experiences through third-party application deployment. Erbad et al. (Erbad et al. 2008) suggest a well-structured namespace that facilitates flexible groupings of devices according to the current context and applications and how this becomes increasingly important as the number of displays, users and applications increases beyond single display deployments. They use the notion of channels to address groups of situated displays, individual screens, users, and the functionality supported by these screens. Channels typically correspond to physical entities or groupings of entities and they can be grouped in a parent-child or containment hierarchy. Davies et al. (Davies et al. 2012) also mention the need to allow developers to address displays by knowing the specifics of their identity, and how they should access and control screen geometries and placement. They call for some type of programming models and protocols for supporting the synchronization of behavior between multiple displays. Storz et al. developed a scheduling API that supports a transaction like concept that ensures content only becomes visible if all needed display and content resources are available (Storz, Friday & Davies 2006). The atomicity provided by these transactions is especially valuable in multi-screen configurations and has been demonstrated as part of the e-campus display network in Lancaster to successfully coordinate the placement of content across displays. Hardy and Alexander (Hardy & Alexander 2012) associate names with projectable surfaces. These names are both human readable and descriptive and the assumption is that for relatively small systems such as those, this direct mapping between content and space would be more developer friendly than for example a large virtual canvas.

3.4 Summary

In this chapter, we addressed the challenges of building the design space of applications for Open Display Networks. Structured as a collective approach within the PD-Net team, the main contribution of this work is a detailed understanding of what display applications might be and how they should be designed. A set of six design principles have been identified also with sixteen properties that shape the emergence of future application models for future public displays and set them apart in terms of specific characteristics when compared to desktop or mobile computers. In particular, our contribution to this collaborative work lay in a number of specific studies and insights briefly reported throughout the chapter. Therefore, we see this work as a systematic and natural exploration of the application design possibilities – considerations that are valuable beyond any technological assumptions. This study constitutes the backbone of this thesis and the subsequent chapters reflect the implications of these principles for the Web technologies as an appropriate technological framework to create such applications as well as for understanding their impact on the user experience itself.

Chapter 4

Web Development

In this chapter, we propose to understand the development process of third-party display applications using Web technologies and how to effectively leverage on the Web community to support software development for multi-application public displays. We apply the design principles that were previously formulated to analyze the implications they might have on the ability of Web technologies to serve as the technological background for the creation of this type of application. We specifically address the use of Web technologies as the first approach to investigate the technological possibilities for *third-party display applications* development.

Web technologies can be particularly valuable in regard to openness, portability, widespread availability and easy to deploy in large scale (Pawan 2009). The main benefits lay in the wide usage of Web technologies and their ability to be supported across many platforms. A vast range of tools already exist and many people already have the competences to create all sorts of Web content. This includes also the emergence of new standards and specifications such as HTML5 and CSS3 which make it possible to deal with the properties and requirements of display applications. For our work, we considered that Web technologies would be the right context to study the properties of display applications and then, based on emerging limitations, identify more clearly the situations in which alternative technological frameworks could be adopted. This means that we are not claiming that Web technologies would necessarily be the best approach for all development and deployment scenarios as various alternatives could be explored. For example, it may be needed, depending on the application goals, to consider specific software components that would complement Web technologies such as the usage of display-specific virtual machines to improve robustness against server and network problems (Lindén et al. 2012) or use alternative architectures to deploy web-based applications or native counterparts within existing cloud/cloudlets infrastructures (Satyanarayanan; et al. 2009)(Clinch, Harkes, et al. 2012). However, since there are no widely accepted platforms for display applications, Web technologies are a better starting point than native applications.

Chapter 3 focused on understanding the key principles of third-party display applications, which might inform about the possible goals of this type of applications. In this chapter, we specifically address the Web application development perspective. Even if the research community is increasingly trying to offer insights on how display applications should look like, the required understanding, regarding the relation between what we know Web technologies are today and the requirements imposed by display applications, is still missing. Understanding how third-party Web developers would engage into this new type of development is important to help optimize this space – the Web intersecting public displays.

4.1 Challenges

The use of Web technologies in display systems poses many new challenges and simply showing normal web pages makes for poor signage content (Clinch et al. 2011). While the ability to present Web content from a specific URL is not a challenge in itself and is already an integral part of almost any display system, the overall context of how this content is selected, obtained and adapted to the circumstances of a particular display is something that is not well matched by prevailing approaches of web-based applications. Various display prototypes used Web for their infrastructure and applications (Erbad et al. 2008)(Memarovic et al. 2011)(Alt, Kubitza, et al. 2011)(Geel et al. 2013). The simple inclusion of an application URL is seen as a regular pattern to provide web content or interactive services to a public display. In particular, Social Networking Services (SNAs) are considered as a dynamic user contributed content source that can add more value for public displays (Hosio, Kukka, et al. 2010)(Elhart 2013). The integration of SNAs is also explored by Locamoda – the company that provides several place-based social media display applications focusing on enabling personalized and interactive experience with digital signage content (Locamoda 2010).

Very often researchers build display applications as distributed applications based on the Web paradigm. Common design goals include ease of deployment and content creation, maintainability and robustness. A reference example is the display infrastructure in Oulu (Ojala et al. 2012), with 18 interactive displays that support the deployment of services in form of web-based applications. Oulu's multi-application public displays based its design on the Web paradigm and enables content contribution from multiple third parties. Services may reside anywhere in the Internet under a simple URL. Their experiences over a period of three years have shown the many specificities of public displays, which mainly result from the public context of this type of installations. In their work (Lindén et al. 2012), an approach based on virtual machines and web technologies was suggested as an appropriate model for supporting application deployment. e-Campus public display infrastructure from Lancaster University (Friday et al. 2012) is another relevant example for using web applications as means to personalize user experience in front of large displays (Kubitza et al. 2012)(Davies, Langheinrich, et al. 2014). Based on a mobile Android application users can locate the nearby displays and configure what content to see as part of the associated display web applications. Memarovic et al. identified a number of challenges when moving from personalized Web content to personalized content for public displays including user identification, profile location, profile content, content tailoring, model refinement and applications that require personalization. In their vision, public display networks require novel approaches for personalization and existing web personalization solutions cannot be used as they are employed in desktop computing environments (Memarovic & Langheinrich 2010).

Overall, the Web and its set of enabling technologies are attractive for building displays infrastructures and applications but not much is known about the specificities of display applications and the implications they might have on these technologies. Even though the research community is already working on the concept of independent/third-party applications for public displays (Clinch, Davies, et al. 2012) and deployments of multi-application displays are starting to appear (Ojala et al. 2012), there is not yet a systematic analysis of what it means to create a web-based application for an open network of public displays. For these reasons, our work is the first to address in details the concept of display application from a Web development perspective, focusing on the extension of web development practices and expertise to support this type of development.

Building on the analogy with mobile ecosystem[18], in our work we identify and characterize what makes a web-based display application different from its desktop and mobile counterparts and build the case of web-based display application. Driven by the vision of third-party applications for Open Display Networks (Taivan & José 2011), and not by a platform or system specific incentive, which might limit the designs, perspectives and features, our main scope is to reach a generic understanding about development specificities of display applications that can frame the development of many different types of web-based applications across an unknown and diverse set of multi-application displays.

4.2 Research Design

Our research into understanding the development of display applications is mainly informed by the development activities conducted as part of Instant Places platform and two experiments with third-party developers: an one-day application hackathon and interviews with developers that created real display applications in long-term (months). The combination of these diverse perspectives has enabled us to consolidate the many issues and challenges that developers need to consider when creating Web-based applications for the execution environment of public displays.

As part of Instant Places infrastructure we had been involved in exploring the limits and opportunities of Web technologies for supporting the requirements of display applications. This constituted the main input for understanding the specific considerations that shape the development of web-based display applications and can be of value for the entire research community. From our analysis of all development efforts we formulated a number of key specificities that can be considered relevant for many development contexts and application models that use Web technologies. This was based on a close reflection regarding the extent of which various web-based applications matched the application design principles and properties identified in Chapter 3.

The goal of application hackathon was to introduce new developers into the development of display application by investigating the extent to which they could engage in this type of application development. In parallel, we assessed the effectiveness of a set of development tools that supported developers throughout the development process (Taivan, Andrade, et al. 2013). The assessment of our development tools was achieved by adopting an *informal and controlled laboratory evaluation* (Klemmer et al. 2004)(Heer et al. 2005). We invited five participants to create a given display web application by using our guidelines and tools and interviewed them about their experiences. We asked them fifteen questions about the entire development process, Instant Places application model and the degree of tasks

[18] W3C developed a number of technologies that explicitly address the specificities of mobile devices (e.g. network costs and delays, memory and CPU limitations, input differences, context-aware capabilities): CSS Mobile, SVG Tiny and XHTML for Mobile (W3C 2014f)

competition (Annex A). All of them had basic web development skills, e.g., JavaScript, HTML and CSS, and had never built a display application.

In order to get more detailed insights into the development experiences with display web-based applications we conducted semi-structured interviews with three third-party developers. Developers were researchers from our group that have not been involved the specification of Instant Places application model. They created nine real display applications using Web technologies (Section 2.3.3, applications from 1 to 9); all of applications have been created to be deployed in Instant Places infrastructure. In the interviews we asked them twelve questions (Annex B) about their overall experience in creating web-based display applications and how they tackled a number of specific problems, e.g., idle time, disconnected operation, communication between apps. The main goals of the interviews was to consolidate our findings as regards the key web specificities of display applications and to understand how we can effectively leverage on developers' web expertise to create display applications.

4.3 Web-Based Display Applications

Web-based applications are software applications that often run inside a Web browser and communicate with the user over a network connection using HTTP rather than existing within a device's memory. Examples of applications include web sites, light programs (web widgets) such as games, online calculators, calendars, as well as more intensive applications such as Gmail or Google Drive that provide users a more native-like experience.

Even though many of our findings are generic, most of our work was done under the assumption of a *single-page application* model (Mesbah & Van Deursen 2006)(Wikipedia 2014). We specifically consider the use of single-page application model as the primary technical approach to deliver content and behavior for multi-application public displays. The main reason for this particular approach consists in supporting a more fluid user experience similar to native applications. Single page applications can be created by using standard web technologies including HTML, CSS, and JavaScript. As opposed to the drill-down or page metaphors employed in web sites in which a click equals a refresh of the content in view, single-page applications allow users to interact with content in real time, where a click or touch performs an action within the respective view. In other words, single-page application is a type of web-based application that provides a more fluid user experience similar to native applications. For instance, the individual components that form a UI of a single-page application are updated and replaced independently in a way that do not require the entire page to be reloaded on every users' actions. An implication of the SPA model is that the logic from the server is moving to the client and web servers evolve into a pure data API or web service. Instead of thinking in terms of sequences of Web pages, Web developers build display applications based on a single-page user interface (UI), that is, a component-based fashion.

4.3.1 Display Application Model

This research was conducted as part of Instant Places (José et al. 2013), a Web-centric platform for place-based screen media, where we have been involved in the specification of a single-page model for web-based display applications. While we do no claim this to be the only model, Instant Places application model is a perspective that has evolved over the years with our ongoing research in this topic and there are several key reasons why we use it (previously highlighted). In the following, we clarify the main assumptions behind Instant Places' single-page display application model.

We consider a display application to be a web-based application whose primary goal is to render content on a public display. Like any other web application, display applications are based on Web technologies and standards, e.g., HTML, JavaScript and CSS. Display applications run on standard web engines or other types of specially tailored web stacks and they encapsulate both content and the means to render that content on screens. The need to support disconnected operation and specially tailored content management policies, led us to assume a rich client model in which the core of the application is running on the display node. Each application will have its own JavaScript code to handle, on the display side, issues such as obtaining and managing the content items that the application will need, caching and prefetching of content, or dealing with network disconnections.

We also assume that these applications entail a clear separation between content creators and particular displays, reflecting the need to develop applications that may potentially be used anywhere. Therefore, applications must be developed without any assumptions about their execution contexts. This implies dealing with the potentially strong variations in the resources that may be available across locations. Portability, in the sense of being able to work across multiple display platforms, it is the most obvious requirement, but there is also a need to accommodate other differences in the operational environment, e.g., display sizes or interaction modalities, as well as variations in the associated information space.

Regarding the user interaction model, we assume that multiple display viewers can interact with the applications only through mobile phones. Our software infrastructure enables viewers to personalize some of the content of the applications but do not allow them to influence the application presentation times or change the application schedule. Within our display infrastructure, these tasks are handled exclusively by display owners. Overall, the model of application presentation is driven by the content sliding by with a fixed interval for each application and without any direct interaction from viewers such as using touch or gestures, the only possible interaction being remote through the usage of mobile devices.

4.3.2 Development Tools

Beyond the varied set of web-based display applications (Section 2.3.3), Instant Places includes a set of development tools conceived to facilitate the application development process and a developers' web site[19] with key information on how to develop these apps. The main scope of these development resources was to support and foster third-party application development, as well as to assist team members in various development purposes. The development tools include an *Application Generator*, *Instant Places Library* and *Media Simulator*. The *Application Generator* provides developers with the possibility to generate a ready-made application structure. This considerably reduces the initial development effort and promotes the use of patterns and components that are known to

[19] developers.instantplaces.org – The thesis author was directly involved in the creation of this service. Accessed August 10, 2014

work better with this type of application. This was achieved by the generation of a Hello World display app, which constituted the skeleton for the creation of other apps. The *Instant Places Library* provides an abstraction layer for the Instant Places service that enables applications to integrate dynamic data into their content, more specifically place-based information about their surrounding settings, i.e. sensing and interaction information associated with displays. Finally, the *Media Simulator* allows display apps to be tested in their target execution environment, i.e., display nodes' player that uses Internet Explorer browser. Instead of deploying applications to the real display infrastructure, developers have the ability to use this tool to check in advance if a display app is ready to be shown on a public display. Based on a set of guiding reference tests, e.g., resizing the window of the application, unplug the network cable, a developer could observe the behavior of the app.

In addition to these tools, Instant Places offer a few supplementary guidelines on how to handle two key issues such as *network disconnection* and *visual adaptation*. Firstly, building a fault-tolerant app is essential to public display environments, because we do not have an end-user that is ready to solve the problem. Our application samples include a set of code blocks for the cases when no data was fetched or it took too much time to show up, e.g., splash screens routines for masking application startup delays or show something to its audience while external data is being fetched. For example, the Hello World application generated by the Application Generator already included a splash screen hiding the error of no connectivity. Secondly, to handle the diverse resolutions and orientations that public displays can have, there is a need to employ at least some basic techniques for making the application content look good and – especially – readable. The initial Hello World app already included a technique based on CSS media queries. It allows developers to add expressions to media type to check for certain conditions and apply different style sheets. For example, one can have one style sheet for large displays and a different style sheet specifically for mobile devices. The technique is really helpful because it allows adjusting to different resolutions and devices without changing the content. The condition that is often verified to trigger the changes is the viewport width. When the viewport is too narrow, applications can adjust the font and some box sizes.

4.3.3 Usage Scenario of Display Applications

To complete the description of Instant Places application model we describe how display applications were employed in the infrastructure (Figure 20). As represented in Table 6, Instant Places system contains a service for local display managers/owners to subscribe for display applications. There are four phases until an application is ready to generate content in a public display: publication, subscription, widgets creation and scheduling. In the following, we describe each phase and explain the functionalities of the infrastructure by giving various application examples including our developed application – Instant Chat (Section 2.3.3).

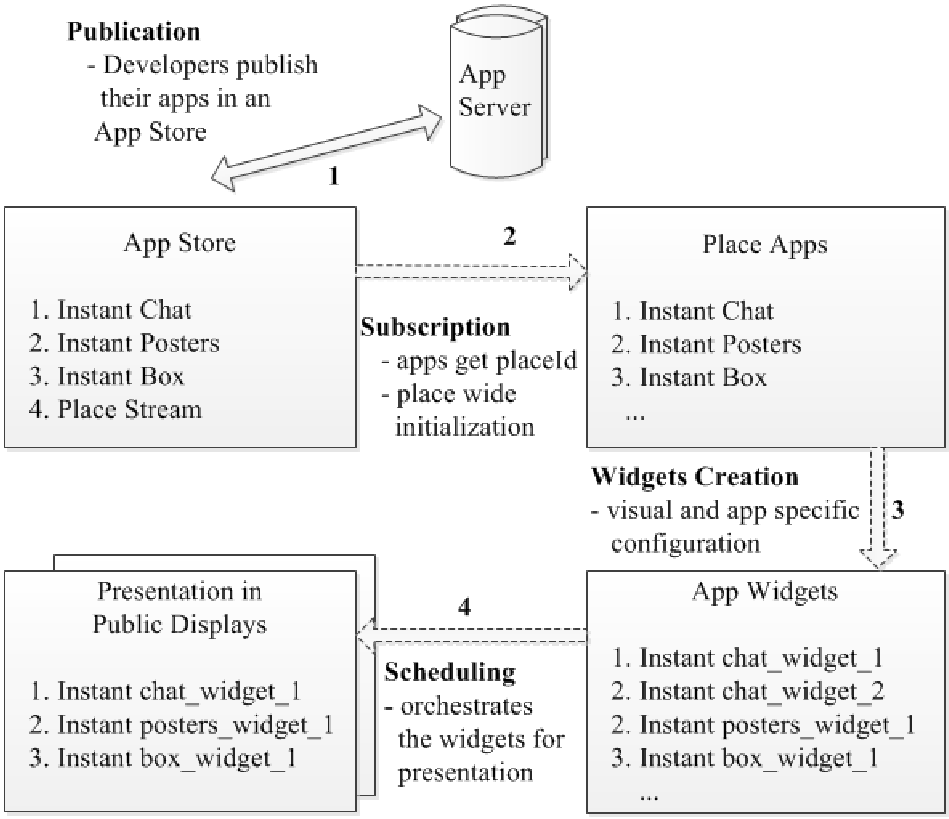


Figure 20: Instant Places usage scenario for display applications

4.3.3.1 Publication

Publication is the process of announcing the existence of an application, describing it and managing the administrative relation with potential users, e.g., buying options, usage restrictions or usage analytics. Regular web sites do not have to be registered somewhere in order to be used. Through search engines and links in other pages, traffic is expected to start flowing naturally. With web-based applications for public displays, there is always some explicit action of associating the application with a particular display environment. For this to be possible, applications need to describe themselves in a way that supports their selection for distribution purposes, very much like in mobile app stores. Therefore, deploying an application is not enough to make it available to potential clients. Even though the application may be completely ready to answer requests, clients need to find out about its existence and assess the suitability of the application characteristics to their needs. Web-based display applications thus require to support this distribution process by describing themselves and announce their existence so that any potential clients may find the applications they need.

In Instant Places developers are able to publish their work in an application store (a simple registry) from where applications can be found and subscribed for different displays within our infrastructure (Figure 21). The publication phase involved a simple deployment process in which applications become ready for execution by hosting them in our infrastructure servers. This has removed the need to handle cross-domain requests in handling the application configurations. As well, this approach allowed us inspecting third parties' application code for eliminating any security and offensive content risks.

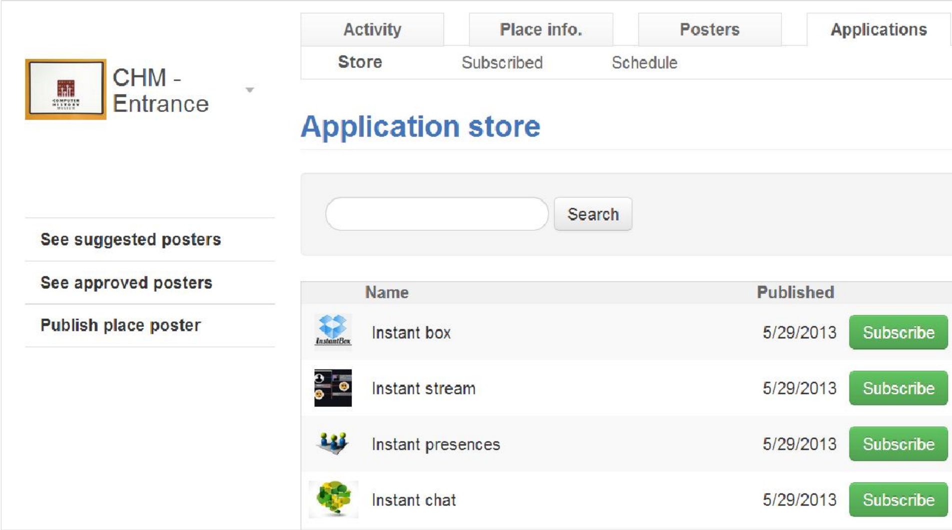


Figure 21: Instant Places application store as part of place management service

4.3.3.2 Subscription

A display application will be subscribed by someone interested in using it. This subscription may or not involve a payment, but it always represents a contract between the application developer and the application user that defines the terms in which the application will be used. After subscription, the application is available to be integrated into specific displays within that environment.

In Instant Places, the subscription is part of the application store functionality that is included within display environment management service where place owners have access. The subscription involves an instantiation process whereby an application is initialized to work in the specific display environment to which it is being associated, e.g., CHM (Computer History Museum) Entrance place as represented in Figure 21. This process also includes the association of the application with the particular display, where applications get access to the data in that environment. This is achieved by sending to applications the place ID as a query string parameter. The initialization parameters may be frequently changed after subscription, but at this point the essential is to guarantee that the application has the data and permissions it needs to work in the respective environment. We considered that whenever possible, applications should be able to work out-of-the-box. Even if no configuration is provided as part of the subscription process, they should still be able to generate adequate content if associated to a display. However, the subscription phase should be seen as an initialization step where the critical, place-wide parameters are set. For instance, Instant Chat application required to input the blocked words to be checked against all the messages (Figure 22).

While some applications fetch content from external sources e.g., Instant Box or Video apps, others fetch their content from Instant Places platform (through the use of place ID and public APIs) and show it as part of a given place, e.g., Posters and Presences apps. In the case of Instant Chat application, it was able to manage by itself the content (messages). Using the place ID received in the subscription, the application stores the messages according to each place where it was subscribed. Instant Chat application only fetches from

Instant Places infrastructure information regarding the place such as place name, logo and persons' identities that have checked-in the place at the moment of request.

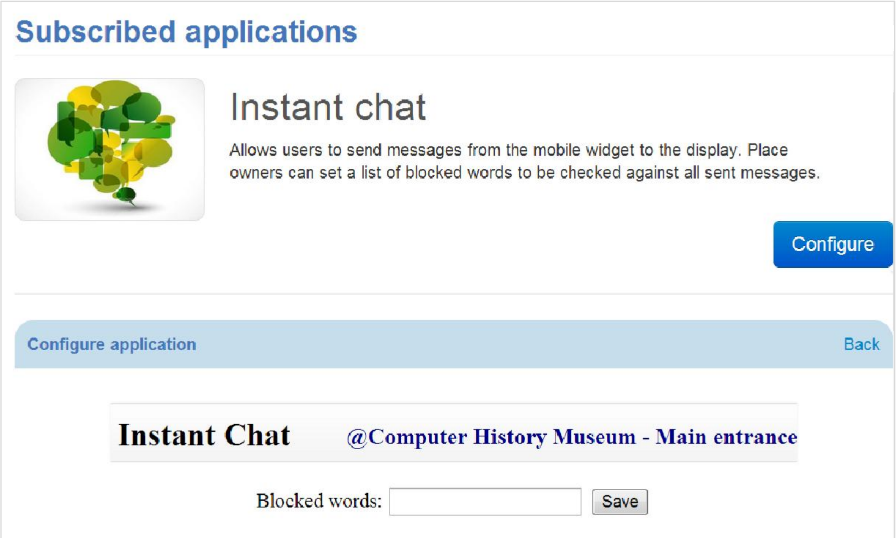


Figure 22: Subscription phase; the application gets the place wide parameters

4.3.3.3 Widgets creation

After an application is initialized to work in a specific display, display owners may require being able to set various application parameters including styling and layout information. For example, an application may employ different visualization themes according to seasons of the year or may support alternative views, e.g., full screen mode vs. status bar mode.

In Instant Places we are able to create widgets that are different representations of the applications. The outcome of this phase is a scheduling element or an application URL that can be distributed to a particular display node to generate application content using a given set of configurations. The current implementation passes the visualization settings as query string parameters (Figure 23, Figure 24).

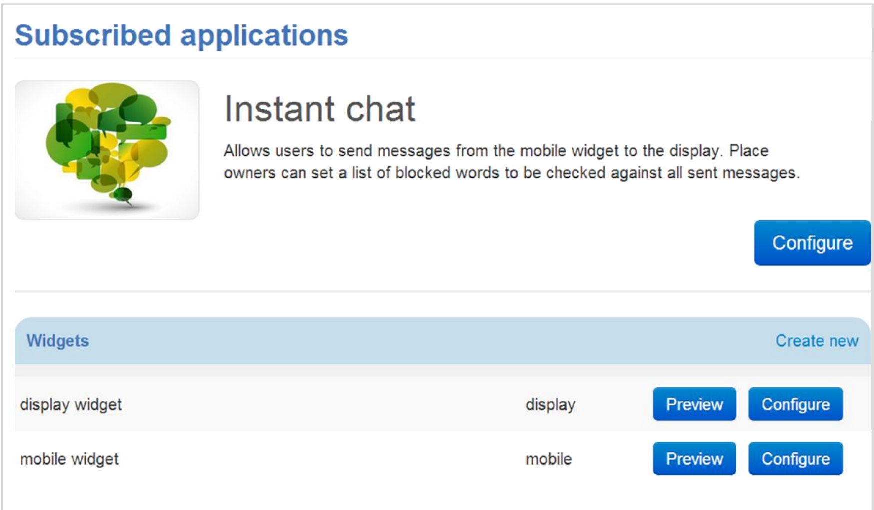


Figure 23: Available type of widgets (display and mobile)[20]

[20] Instant Chat application uses the default mobile widget. In this implementation, Instant Places do not offer support for creating multiple widgets for mobile; we consider the configuration only for display widgets.

Figure 24: Widgets creation

4.3.3.4 *Scheduling*

The scheduling process involve dealing with specific application scheduling requirements including the duration of content or application presentation and association of applications with presentation slots on the screen. In Instant Places, we consider a simplistic scenario in which all applications run in full screen mode without any specific scheduling requirements. After the widgets creation, applications are ready to be associated with specific displays and may dispose several URLs as scheduling elements corresponding to different settings. Display owners select manually from the list of possible scheduling elements that should be active in a particular display and allocates to each one a specific amount of time (Figure 25). We call a schedule the list of applications that are orchestrated for presenting content in a certain display. Each application presents its content by iterating over a predefined list of content items, e.g., messages, images. When the allocated time slot for an application expires, the application player, which directly controls a web browser, loads the next application from the schedule. While an application is presenting content it can also perform any request to external services or react to user input. For example, the Instant Chat application enabled users to post messages from their mobile phones. This was possible through the usage of a shared database between the scheduled application and its mobile web-based interface. All the messages were aggregated as part of the application data model according with each place ID or application subscription.

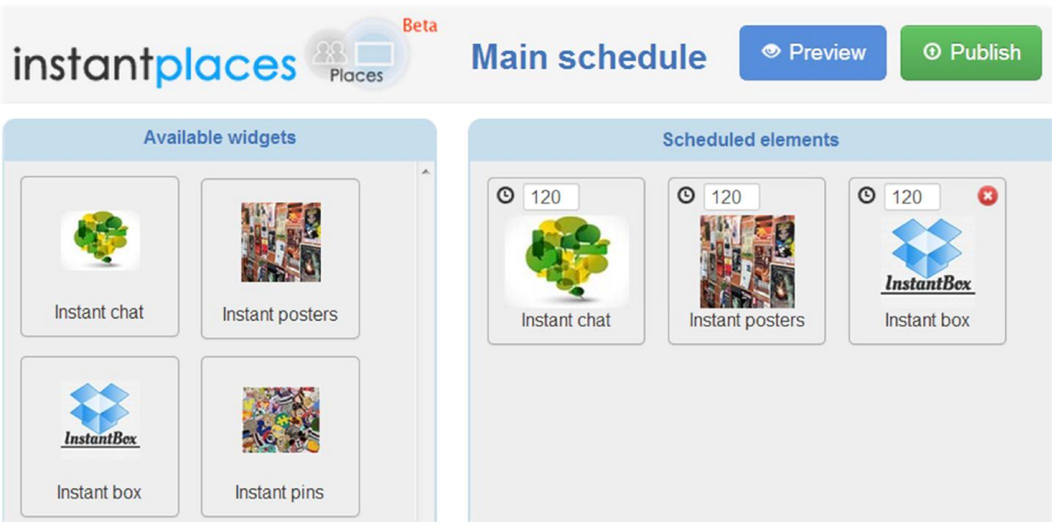


Figure 25: An example of application schedule

4.4 Hackathon Experiment

A week before the experiment, we sent participants the URL of the development web service so that they could learn the basics of the process. At the beginning of the experiment we gave them a brief tutorial of about 10 minutes in which we introduced the concept of display apps and explained the APIs. They were then asked to build a new display app, i.e., a poster grid app, based on the Hello World example. To do this, we formulated three development tasks that led developers to create the given app. The first task was to put the Hello World app running and test its execution. For this, they needed to install the App Generator and output an application example and Media Simulator for being able to test it. The second task was to use the Instant Places library for getting place related data, such as the *place name*, *place image* and *posters*. Finally, participants were asked to show the posters in a grid by using some CSS rules. In this step, developers needed to use splash screens and configure them to last for at least 3 seconds; support fault tolerance functionality (lack of data, lack of connectivity); prepare the app to be displayed correctly in an iPad or in another device of similar dimensions and test the application using a desktop web browser and Media Simulator tool. Throughout the experiment, participants were encouraged to raise questions and they had four hours to complete all the tasks. At the end, each of them was interviewed about their development experiences. The semi-structured interviews were audio recorded and the code produced by developers was kept for subsequent analysis.

4.4.1 Development Insights

The findings from this application hackathon show that the overall view of this experiment was positive, even for less skilled developers. Developers did not take part in the specification of our approach for creating display applications. All the participants have achieved the key development goals without wasting too much time in writing the code. They found our tools useful and necessary for the first contact with web-based display applications. Initially, participants had some effort to grasp the specific concepts associated with displays apps, but after that, they were quickly able to master the process.

Developers could easily follow the documentation provided by our development web site. Even though this was optional, all participants used the Hello World app generated by the App Generator tool as a template to start implementing the new display app. The participants didn't think very much about the structure of the application, which meant that the use of Application Generator was effective. When we asked developers how it would be to develop without this tool all of them responded that it would be difficult or even very difficult.

Developers had enthusiasm for this experiment despite their weaker experience with some of the required web technologies. This is demonstrated by the fact that all of them succeeded in applying their web development skills to develop a display app. However, a few of them experienced difficulties in understanding and using all specific development and testing scenarios, e.g., implementing the splash screens or providing the required code blocks for a fault tolerant display app. Only one of them could entirely test the app execution behavior.

Due to the fact that our display app was not too complex, it just required a set of API requests, the code source is quite identic among the participants and the final applications share the same structure and very similar lines of code. Having a previously scaffolded app structure proved to be comfortable to the participants and reduced the amount of code they had to write. Developers ended up not writing much code and not changing the application structure at all. Instead, their effort was mostly to combine various code blocks and configuring them appropriately. However, one participant noted that the integration of our code blocks was straightforward, while making various customizations was not so easy. Using the Instant Places API library was something that proved to be very handy. Although there were some initial problems in understanding the meaning of our API and the related code blocks, after getting the *place name*, they easily succeeded to get further data, such as *posters*.

Developers had difficulties when testing their apps because they weren't familiar with any tools to accomplish this task, e.g. Fiddler. Most tests were made using a common web browser while the Media Simulator tool was just periodically used to rule out eventual errors related to the different web engine of display players. Only one participant did not test at all the new application execution, neither in desktop web browser nor in Media Simulator tool. The others tested the application but encountered various difficulties.

Participants were really motivated by the innovative field of usage of Web technologies and recognized the potential of display applications when deployed in real world settings. They associated display apps with mechanisms to publish content, such as replacing the traditional paper based posters with digital forms of content. In their final comments, they all referred particular features for display web apps, e.g., a display app should provide content that is dynamic (changing at runtime), personalized and place-based.

4.4.2 Limitations

The focus of the application hackathon, was to understand the extent to which traditional web developers could approach the creation of a web-based display application. In this sense, we observed their overall programming behavior and assessed the effectiveness of our tools in supporting them throughout the development process. While we acknowledge that participants had basic web programming abilities, and more skilled developers could perform better for all the tasks, the scope of our tools was quite limited in terms of what developers could achieve by using them. For instance, in this experiment we have not insisted much on application architecture, lifecycle or their ability of to allow user interaction, e.g., by using mobile phones. Instead, the key objective was to identify what are problems or challenges that web developers may pose in general and to observe their first contact with a web-based model for display applications. For these reasons, we did not invite any additional participants as we could anticipate that they will not change too much what we could have learned within this early research phase.

4.5 Key Development Findings

In this section we describe our main findings in regard to key web development considerations of display applications as aligned with the corresponding design principles (Table 12). Since our research did not included any particular issues regarding application distribution services or application analytics we did not address further the principle of global availability. As well, the principle of display ecosystems has not been considered from a development perspective.

Our description includes a brief overview of the key findings, a set of technological considerations that highlight the opportunities and limitations of Web technologies as the technological framework for the creation of this type of application and insights from the interviews with long term application developers[21] (Taivan et al. 2014b).

Table 12: Mapping between development findings and the corresponding application design principles

Principles	**Overall coverage**	**Properties coverage**
Global Availability	Not covered	N/A
Adaptability	Covered	Portability, Visual adaptation
Situatedness		Environmental model, Explicit configuration
Content Management		Content placement, Actionable content
Concurrent Applications		Coordinated scheduling, Shared interaction
Display Ecosystems	Not covered	N/A

4.5.1 Visual Adaptation

We call visual adaptation the process of adjusting the content appearance of a web-based application to the browser screen dimensions. For instance, visual adaptation is performed when a web site adapts its text font size to be legible in a smartphone. While this need for visual adaptation is common in desktop and mobile web usage, the adaptability range in public displays can be much more extreme and the role that users can have in assisting the adaptation process is much more limited (Section 3.3.2.2).

4.5.1.1 Technological assessment

Responsive Web Design (Marcotte 2011) has become a de facto standard practice in web development targeted at the diversity of web devices. In addition to advocating the principle of device independence, Responsive Web Design encapsulates a set of technologies that allow web applications to automatically adapt their content to the display characteristics. For instance, a responsive web site can employ two different menu styles: when the site is viewed on a desktop computer it may have a horizontally arranged navigation bar, but when the same site is viewed on a mobile device, the style of navigation changes to a vertically organized set of links or buttons. Thus, the main idea of responsive web design is to provide users across a wide range of devices, e.g., smartphones, tablets and desktop, a single source of content that can be easily read and explored with a minimum of resizing, panning and scrolling. While responsive web design can be expected to be part of

[21] In the subsequent paragraphs we will use *developers* to represent any insights from third-party developers that we have interviewed for their long-term application development experience. We use *developers* to refer the overall development insights across a larger set of applications.

the visualization solutions for web-based applications for public displays, a few more techniques may be needed to deal with the great heterogeneity of potential presentation containers and with additional elements like viewing distance to the display.

When the level of visual adaptation may require more than simple resizing tricks, display applications may offer alternative views that are expected to be explicitly selected when the respective application is integrated into a presentation container. A view may be embedded with specific options in regard to the ideal displaying size, orientation and viewing distance. In addition to different viewing assumptions, alternative views may also encapsulate particular knowledge about the most appropriate data to be shown, offering different visualizations of the same data, or visualizations that focus different parts of the application data. A horizontal bar at the bottom of the display, for example, is not expected to simply squeeze the content of a full screen into a tight line. A bar of that type is expected to show some key headlines, possibly scrolling. Similarly, a small window designed for pop-up, is only expected to present short notifications. Each application may specify as many alternative views as suitable.

4.5.1.2 Development insights

All the applications developed as part of Instant Places infrastructure used responsive web design within a limited range of display resolutions or screen sizes. Our approach is mainly based on using percentages instead of fixed sizes for any visual elements in the application. We also used the *media* tags on the CSS that can control the sizes of elements in relation with the screen size. For example, for a given screen width developers gave a certain width to an element. While CSS media queries served very well for implementing the responsive web design, developers reached a limitation related to the sizes of images and the text length that comes from Twitter or Facebook APIs. The first assumption in their applications was that people cannot scroll text or images on the screen, so, all the content need to fit a single display. Due to the lack of support in availably plugins[22], they developed a custom text adaptor plugin that can enlarge text within a given min and max font size and ellipsis the rest. Still, they had the drawback of the text that required to be cut in order to fit the screen. As well, handling images involved some manual configuration depending of the availability of the image properties.

A second factor that influenced the visual adaptation solutions was the viewing distance. For the text, our approach was to experiment with different font sizes in a way that allow people to get the content from a few meters away. In particular, the Video application informed us that visual adaption may be different depending of the users engagement with the application. Users could see the videos on their mobile devices so, if they are in the back of a room this might not affect too much. They can come closer or just watch the videos on their mobile phones. Therefore, the viewing distance is very important for the first phases of the engagement. After the users have joined with the display application they may walk away from the display. In other words, visual adaptation is an interesting factor that may influence the displays' role to entice for interaction and to join the system. For example, in a big room an application may adapt the content (providing only keywords) in

[22] http://simplefocus.com/flowtype/ (just one example). Accessed September 26, 2014

a way that people at a large distance can notice what is going on and manifest interest in joining the system, while for small rooms it can further provide additional content having the same goal to entice people to get in. In conclusion, our experience shows that responsive web design technique should not only consider the screen size but also the space properties, e.g., maximum available viewing distance, in order to better adapt its content and entice interaction.

A third factor with a significant impact over the entire public display experience was the visual aspect of the applications. Developers struggled to design applications that do not have the look and feel of a traditional web-based application, avoiding elements like columns and menus. Instead, applications were designed to present content in a natural way so passersby could get that content easily understood. This was because our particular model for user interactions in Instant Places assumes that people interact with display applications through personal mobile phones.

4.5.2 Situatedness

Situatedness is one of the key design principle embedded in all the applications developed as part of Instant Places infrastructure (Taivan, José, et al. 2013). This decision is primarily sustained by the fact that public displays entail a set of characteristics that distinguish them from desktop, mobile or wearable computers. They cannot be separated from their physical, social and cultural setting. Therefore, the location, situation or the set of circumstances are key aspects that determine the design of potential applications and services involving public displays. While not all applications need to be situated and might be scenarios in which people would expect the same application experience regardless any specific circumstances, public displays make this distinction in how we perceive the consumption of information and entertainment, which is radically different from using a PC at home or carrying a smartphone wherever we go.

4.5.2.1 Technological assessment

The main challenge we had to solve when incorporating situatedness into web-based display applications was related to the mechanisms and decisions that need to be taken when splitting the responsibilities between display infrastructure and applications. Applications may offer multiple alternatives on the way they are used, from data related options, such as authorization procedures or the indication of external resources to visualization preferences, such as colors or alternatives views. However, display owners of those applications should not need to go individually to the multiple applications' web sites, making application-specific authentications in order to configure the various applications. Instead, we looked for a more adequate solution that would allow display managers to handle all the configuration procedures under the same framework of a display system (Section 4.3.3 provides details about the configuration procedures).

For example, in interactive web browsing, a user may at any moment go through any steps that may be needed to select the intended content and provide on demand any data that may be asked, including, if needed, authentication data. In a public display, there is no such possibility. The display system must be able to guarantee that any configuration or content selection options that may be needed have been done before the display starts accessing the applications and generating content for presentation. The system must be able to determine exactly what resources will be requested from the application, what configurations will be applied, and what type of authentication information will be needed to allow a particular display system to access a set of resources on the various applications it may be running. Display applications will thus need a set of integration procedures that enable them to serve the displays from a particular domain in a way that is specific to that domain. This location specific configurations and data access may also generate additional authorization and security requirements to create secure contexts for accessing local data and location specific content.

Our candidate technology that was explored for the communication between Instant Places infrastructure and display applications was the web mechanism of cross-domain messaging defined as part of HTML5 specification, i.e., Web Messaging (W3C 2014d). Web Messaging defines a messaging system that enables documents to communicate with each other regardless of their source domain in a safe way, without permitting any cross-site scripting attacks. This technique fills the gap of web browsers, which for security and privacy reasons do not permit any cross-site scripting or prevent any communication channel between two pages hosted in different domains. HTML5 Web Messaging allows embedded web pages to communicate with its container, e.g., *iframe* scenario.

4.5.2.2 Development insights

Our development considerations are related with the ability of display applications to provide situated content and functionality. In particular, we addressed two main approaches to incorporate situatedness into global display applications: *environment model* that provides information about local display context *and explicit configuration* by display owners/managers.

Environmental Model. Instant Places system is designed as an environment service that aggregates data about the local environment along the physical and social dimensions (Sections 2.3.3 and 3.1.3). It uses the abstraction of *place* to designate an ecosystem of people, displays, physical locations, applications and content. Every application that runs in Instant Places display infrastructure has the knowledge of a place ID that represents the connection with the system itself and its associated data space. Therefore, to different extents all the applications employ a type of situatedness that we describe in the following paragraphs. While many display applications would benefit if they would be able to adapt the content to a certain display and social interaction, we found our approach somehow limiting the introduction of other applications from other infrastructures that do not consider the environment model as a primary feature. However, our experiences conducted so far highlighted the benefits in doing so. We look forward to provide insights into a generalized environment model that can be used across different display domains to incorporate situatedness as a primary functionality for display applications.

Explicit Configuration. We used the following two main paradigms: *filter-based configuration* and *addressable configuration.* These two approaches allowed the content produced by an application to be specific for a particular place.

4.5 Key Development Findings

A filter-based configuration allows the application's generic content to be filtered in a way that makes it more specific. In the case of web-based applications, the specification of the filters can be done within the content request itself and therefore it does not necessarily generates a specific configuration state, i.e. the application does not necessarily needs to know which consumers are using which filters. For application that do not need to maintain server-side state for each of the potential places in which they are being used, a simple filter may suffice to specify how the application should answer the requests for a specific place. This filter will normally be formed through a set of properties in the URL that is sent to the application. For these applications, each request is a separate request and the result generated will not depend on any previous requests.

Addressable configuration refers to a pre-defined configuration that will enable applications to generate specialized content. Before application can be requested, a specific and identifiable configuration needs to be created and then all content requests will occur in the context of that particular setting. The content generated by the application will depend on the respective instance of configuration parameters. An application can have many addressable configurations associated to particular usage contexts. This indicates the various ways in which applications can embed a situated behavior by generating content specific to those embedded circumstances. An addressable configuration is a server-side concept that represents a separate scope for using the services offered by a display application. Such configuration instances must be created prior to content consumption, even though it may be possible to spontaneously generate new ones. The existence of addressable configurations should be transparent to application configuration and consumption. The differences will be in the type of configuration parameters and the generated presentation URL.

In our work, we have tested both types of configurations. While the addressable configuration clearly involved more development effort, we found it adequate when the configuration parameters are laborious to be stored in URL. On the other side, the filter-based configuration satisfied most of our configuration requirements in Instant Places infrastructure. However, in any case, the embedded application configuration page must itself generate a URL based on the provided configuration options. The generated URL needs to be communicated to display infrastructure, which can then be used to access the application content as part of display schedule. All the communication including sending the place ID to the application was handled using HTML5 Web Messaging without any specific accounts.

Overall, by studying the design principle of situatedness we have not reached any distinctive implications for the usage of Web technologies that might limit the design and require particular approaches as was the case, for instance, with content management and visual adaptation. Application developers should be responsible to handle their configuration procedures in a way that can integrate a third-party display infrastructure and allow cross domain requests in order to provide the necessary URLs for presentation in public displays.

4.5.3 Content Placement

In traditional interactive web browsing, content selection is assumed to be under the control of a single user, who may at any moment request a new content resource or be prompted to provide any necessary data, including, if needed, authentication data. In a public display system, content presentation can be mainly autonomously determined by the system itself, which must be able to guarantee that any necessary configurations or content selection options must have been done before the display starts presenting content (Section 3.3.4.1).

4.5.3.1 Technological assessment

Prefetch and cache are two web mechanisms that may be used to address these content management issues. In public displays, prefetch can be more necessary and also more viable because it is easier to identify the resources that may have to be prefetched. There are fewer potential resources to present and there is an application scheduler that will have at least partial information on what to show in the near future. The ability to prefetch content is thus an essential feature for display applications. Proper prefetch support may significantly improve the reliability of the system, provide better user experience, save communication costs, and improve the scalability of global applications. Currently, prefetch support is available in Firefox[23] and recently in Internet Explorer 11[24]. The Firefox prefetch mechanism uses the HTML <link> tags that instruct the browser to begin fetching a given URL. A site author explicitly defines what resources to be fetched in advance by using a relation type of either "next" or "prefetch" for the respective <link> tag definition. Based on these keywords, the browser will preemptively fetch and cache the respective resource. Standardization of this technique is part of the scope of HTML 5 specification – at present a working draft (W3C 2014b).

Additionally, Chrome and IE 11 browsers employ a distinct mechanism called prerendering. While Chrome has just support for prerendering, thus excluding prefetch support, IE 11 provides both features. At the moment, prerendering is an experimental feature in Chrome browser starting with the 13th release[25]. In Chrome, prerendering is triggered by an element added in HTML that tells the browser to fetch and render an extra page in advance of users actually clicking on it. Prerendering differs from prefetch in the way that a browser instead of just downloading the top-level resource (an HTML page), does all the work required to show the page to the user – without actually showing it until the user clicks. Prerendering mechanism behaves in such a way that the prerendered page is already loaded into a background tab, which is not shown to the user. Only when the user clicks on that page, its content is instantly shown in the current viewing page. Thus, from the user's perspective, the page is loaded much faster than before.

Cache can also be helpful in allowing applications to have local access to recently used resources. The caching properties on the web servers should be optimized to instruct browsers that resources are valid for a long period and should be kept in the cache. However, current web browsers offer limited control over their implicit cache mechanisms,

[23] https://developer.mozilla.org/en-US/docs/Link_prefetching_FAQ Accessed September 26, 2014
[24] http://msdn.microsoft.com/en-us/library/ie/dn265039(v=vs.85).aspx Accessed September 26, 2014
[25] https://developers.google.com/chrome/whitepapers/prerender Accessed September 26, 2014

which represents a major challenge to adapt cache behavior to the specificities of display apps, e.g. support for disconnected operation.

On the contrary, application cache is now well supported through the Offline Web Applications (W3C 2014c) technology introduced by HTML5 specification. Some application resources are modified rarely or infrequently, such as images, styles, JavaScript or static HTML. The technology brings the capability of a web application to be locally cached and still deliver its functionality while there is no internet connection. An Offline Web Application defines its application manifest file that specifies every resource that is needed to run locally. The first time the application is accessed it downloads its resources and will always use them, unless the application manifest changes. The manifest also specifies which resources must always use the network to be fetched and also fallback resources (resources to be used if non-cached resources cannot be downloaded). Supported by most modern browsers, this technique was designed to overcome the limitations of web browsers or client caching mechanisms. While HTML5 application cache mechanism is characterized by its simplicity, it is also the subject of technical shortcomings when employed in real world scenarios (W3C 2014i). For instance, a web browser is not aware of the modifications at the server side of cached resources. To trigger an update, the manifest file itself has to be changed. This limitation makes the caching technique not transparent for the developer. Even worse, in the case when the manifest itself is cached, then no update will be performed at all and this can lead to unpredictable behaviors.

4.5.3.2 Development insights

Avoiding idle times caused by fetching content from servers. Our applications used s*plash screens* as the first approach in dealing with this issue. A splash screen is a type of animation that informs users about the current application being loaded. While this approach is neither specific for Web technologies nor web-based display applications, it provides an effective way to notify users about what is going on. However, it has the drawback that if the idle times increase too much, users would see the same splash screen excessively. Developers considered a second approach in dealing with the idle times, which is based on a custom made pipelining technique called *in-app prefetch* employed by a set of applications called schedulers. The schedulers are able to present in full screen other applications (modules). For instance, a scheduler puts a module running while prefetching the next one. The prefetch of a module is implemented as a hidden request and as soon as that module is ready it triggers an event that is used by the scheduler to activate the next module. Developers stated the need for a better approach that would include a new type of player instead of the default Instant Places application player. Such a player will be responsible to prefetch, run the applications and coordinate them. For example, when an application will be ready it will inform the player that it is able to run and the player acts accordingly.

Since the default player could not coordinate the applications, e.g., the case of extended idle times from fetching content from servers, developers considered a new approach that will exclusively be based on Chrome browser without any additional and platform specific components. The decision for choosing Chrome was informed by the entire team attempts to find a browser that can run a web-based player in a way that allows overlapping operations like prefetching content while presenting other applications. Moreover, Chrome browser offers support for Cross-Origin Resource Sharing (CORS) technique – which is an essential feature of the applications within Instant Places development model. The iframe approach has the drawback that when the content from an iframe blocks or does not respond (e.g., Javascript errors), the others block as well. Instant Places solution to overcome this issue is based on developing a Chrome App[26] that can employ *webviews* instead of *iframes*. A *webview*[27] is a way to actively load live content from the web over the network and embed it in a Chrome app. The advantage is that a webview runs in a separate process from the main application and it does not have the same permissions as the main application. All interactions between the Chrome app and embedded content will be asynchronous. In this way, the main application is kept safe from the embedded content. Since the current player is platform-dependent, the new player based on Google Chrome browser is able to alleviate the restrictions associated with the portability and deployment of display infrastructure software.

Make any content fetching errors transparent to users. In our applications we tried to catch and hide any errors and inform users about those generated by content loading processes. Currently, the method used in all the applications that we analyzed is based on a very naïve approach that displays a funny custom message in form of a splash screen that redirects users to the main web page of the system. Since our current player does not have the possibility to switch between applications, users will be shown the same screen message for an extended amount of time.

However, all developers stated that this is not the intended approach and all of them are considering the functionalities of the new player (previously introduced) that will be able to solve this issue. An application may inform the player to show other application because its content is not ready or show another splash screen, or the player can detect in advance that the application is not ready and skip it from presentation. In general, to avoid any error messages, the application should recognize the problem before diving to it. This should also eliminate any idle times.

Support disconnected operation. Disconnected operation is a major problem within the majority of Instant Places applications. Currently, our solution is based on 3 technologies: App Cache (W3C 2014g), HTML5 Local Storage (W3C 2014k) and IndexedDB (W3C 2014e). Overall, a key aspect in offline behavior is to consider the frequency of data updates. Most of the applications may survive a few hours of disconnection. For instance, *Place* app is more dynamic and requires more frequent data updates in order to show relevant information; if there is no network the presences information is lost. A critical situation would be if an application does not have any content to show and in this case it should inform the player which might schedule or some predefined content items or other applications that do not change too frequently such as those with 2-3 updates a day.

In regard to local storage techniques, developers struggled to implement a functionality to store images locally. They have developed a wrapper around IndexedDB which allowed them to increase the local storage size to GB instead of MB used in HTML5 Local Storage

[26] https://developer.chrome.com/apps/about_apps Accessed September 26, 2014
[27] https://developer.chrome.com/apps/tags/webview Accessed September 26, 2014

mechanism. Right now, the only issue is that the development process is a bit more complex and harder to develop these applications because they have to store everything they have in the application even the content that comes from external servers. In particular, the challenge is that the images from external servers do not allow cross domain requests and they cannot retrieve the images for storing locally. A solution might be to use our servers to proxy and retrieve their content.

In conclusion, the solution for disconnected operation may cover the case with 2-3 updates per day and in this regard HTML5 Local Storage and App Cache work pretty well. For the case of applications that need data within seconds, developers noticed the aforementioned problems. Besides the local storage limits, we are also limited by the lack of available services that tackle this problem for other types of web-based applications.

4.5.4 Actionable Content

A key distinction between a desktop web-based application and a display application is that in the latter there is a much stronger need to systematically handle the data exposed by the application. In a normal user-driven browsing scenario, the issue is mainly about links and navigation menus that the user will invoke as needed. When content is being consumed by a display system, the issue is mainly about exposing and characterizing the application resources as content items that can be available uniquely within a display system (Section 3.3.4.2).

4.5.4.1 Technological assessment

The use of resources identifiers is already an integral part of web technologies. The exposure of web content in the form of multiple individual resources, each with its own identifier (URL) is even one of the essences of the popular resource-oriented architecture - RESTful mode (Richardson & Ruby 2007). A resource-oriented architecture (ROA) is a style of software architecture and programming paradigm for designing and developing software in the form of resources with "RESTful" interfaces. These resources are software components, e.g., pieces of code, data structures that can be reused for distinct goals.

Presentation units might play a similar role as the concept of permalink in blog posts. For instance in a blog scenario, with the emergence of permalinks (permanent links) posts can now have a specific URL that remains the same even when they are no longer visible in the blog front-page. This permanence of the links enables those posts to be linked by other sites and provide a reference that supports many other key web functions, such as searching, traffic measure and comments.

4.5.4.2 Development insights

The Instant Places system allows the scenario in which applications can expose their resources. Applications can expose their content items by exposing a URL to the resource. For instance, people may interact with application resources by getting their URLs in personal mobile devices. Then, users can have a closer look to a specific content item regardless of the content being shown on public displays. In this scenario, it is up to applications to decide which resources to expose, if any at all. For instance, the Video application has this feature and the system allowed the content of the screen to be available on people's mobile phones so that they could subsequently access the respective application resources independently. Other possible scenario envisioned (not implemented) would be for applications to be able to expose their resources in order that other applications could integrate them in different ways. For example, there might exist applications that aggregate content from many other applications and offer an integrated experience to the user such as a dashboard application that provide a content overview of the applications running in a display from a given place.

4.5.5 Concurrent Applications

An underlying assumption behind the notion that displays will be open to many applications from third-parties is the idea that any particular application is expected to be one of many that may simultaneously be running on a single display and requires sharing the display resources, e.g., screen real estate or interaction features. This means that a display system will employ optimization protocols between applications themselves and between applications and their execution environment. Application developers do not know a priori the conditions in which their apps will be running. Thus, applications could use these protocols to coordinate between themselves to exhibit an integrated behavior, e.g., avoiding contradictory presentation times.

The optimization protocols could also allow apps to have access to local machine resources, e.g. interaction techniques such as a Kinect device, or obtain information about the environment, e.g., display ID or presence of people in the vicinity of the display (Section 3.3.3.3). It may also help to coordinate the content scheduling process, by allowing the container to inform applications about the best moment to start, stop or prefetch content presentation, and also inform applications about the allocated presentation time. Likewise, applications may inform the container about internal events that are relevant for the scheduling process, such as content loaded or interactions received from users, or it may request additional presentation time, request to be removed from presentation or even take the initiative to request presentation when certain events occur.

In order to optimize network resources usage, display applications should report their possible errors to the execution environment, so that it can channel them more efficiently, e.g., to some application quality service that then informs developers. Ideally, developers should have access to libraries and tools for capturing errors and channeling them appropriately.

4.5.5.1 Technological assessment

Security restrictions may raise a few issues for the integration between applications and their execution environment, mainly because of the different usage assumptions between traditional web browsing and display applications. For security reasons, web browsers impose the restriction of Same-Origin Policy (W3C 2014h), which says that if a document containing a script is downloaded from a certain web site, the script is allowed to access resources only from the same web site and not from other sites. There are, however, some techniques and workarounds that are usually helpful in circumventing these restrictions, which are becoming easier to deal with in modern web browsers. These includes Cross-

Origin Resources Sharing (CORS) (W3C 2014a), JSON-P[28], cross-document messaging, i.e., Web Messaging (W3C 2014d), and the use of a proxy to the required external resource. Web Messaging is a HTML5 technology that allows web applications to communicate with embedded web content from external domains. It is a safe messaging system, which does not allow any cross-site scripting attacks. Web Messaging can also be used for communicating configuration data between the application itself and configuration container or display infrastructure.

The execution of a particular instance of an application on a specific web engine may generate local state that may need to be kept between subsequent invocations of the same application on the same browser. However, every time web content is loaded it will not have any information about previous loading events. With client side state, the web content could keep state between subsequent instantiations. For example, a slideshow application may start iterating photos from the point where it stopped the last time. In this regard, HTML5 specification provides a well-known mechanism, which is Local or Web Storage (W3C 2014k). Web Storage was firstly introduced as an HTML5 feature and is now a W3C specification by itself. It introduces two mechanisms to store structured data on the client side: *SessionStorage* and *LocalStorage*. The *SessionStorage* mechanism is conceived for scenarios where the user is carrying out a single transaction or multiple transactions in different windows or tabs at the same time. The data can be accessed by any page from the same domain. *LocalStorage* is designed for storage that covers multiple windows, and lasts beyond the current session. Using LocalStorage, web applications become capable to store megabytes of user data, such as user-authored documents or user's mailboxes.

Web Storage is an alternative to HTTP cookies storage mechanism (IETF 2014). However, cookies do not really handle well these two cases of client side storage. For instance, in the case of session storage the data can leak from one browser tab to another if the same web application is used, e.g., buying two flight tickets in two browser tabs. Moreover, cookies are transmitted with every request, which makes the storage capacity of cookies quite small and inappropriate for storing of large data sets.

4.5.5.2 Development insights

Communication between the player and apps and between apps themselves. Our applications do not include any integration with their current player. The main role of the new player is to act as a controller that is responsible for the execution of each application. For instance, the communication between applications and player is needed in order for the applications to inform the player when they are ready to be presented, i.e., control the application start/stop time. A further example for this communication includes managing of application errors, e.g., reschedule an application or remove it from the presentation. In particular for the Video application, if there is a video currently playing, instead of stopping the video when the time allocated to an application elapses (actual behavior) the application can ask for more presentation time (envisioned behavior).

Accessing local resources. So far, Instant Places system acted as the main channel for any data required for application to work and for this reason we did not considered any other

[28] http://json-p.org/ Accessed September 26, 2014

type of access to local resources. While developers presented interest into this topic, their focus was the use of Web technologies as the primary technological framework. For accessing the hardware resources, the usage of third-party libraries would be mandatory. DepthJS[29] is one of such library that is under development of MIT Media Lab. It is an open-source browser extension and plugin (currently working for Chrome) and allows any Web page to interact with the Microsoft Kinect using JavaScript. DepthJS provides the low-level raw access to the Kinect as well as high-level hand gesture events to simplify development. An alternative for interfacing a Kinect device could be based on a NodeJS[30] server that intermediate the communication between the sensors output data and web-based application events (Ribeiro & Duarte 2012).

Keeping state. Most of the applications keep state between subsequent application calls. The applications store the state of the last content item shown and next time when they run will not show the same item again. In the majority of the applications developers used HTML5 Local Storage without any specific considerations (only Video app used cookies).

4.6 Summary

The openness and portability of Web technologies are key properties when considering the development and usage of third-party applications in Open Display Networks. However, public displays represent a new frontier for Web technologies, with novel usage situations and technical requirements. This means that a Web-based development model for display applications would be informed by specific activities e.g., adaptation procedures, creation of new tools, complex configuring, redesign that make sense only for the context of shared, large pervasive displays and are not relevant for desktop or mobile computing cases. For instance, not having a scroll in a public display is something that impacts the development approach which might lead to the emergence of new techniques. Similarly to what has happened in the mobile landscape, there is a need for specific approaches that enable display applications to seamlessly integrate the content they generate on the presentation context of public displays.

As part of Instant Places urban deployment of public displays, a diverse set of applications has been created and deployed, embedding multiple characteristics and requirements. Based on these experiences, we have consolidated a generic view on how to leverage on existing web development practices and expertise to support the development of third-party display applications. Our approach led us to abstract from Instant Places' particular model of creating web-based applications and come out with a set of key development specificities that challenges third-party developers in different ways. We highlighted that while the Web has various building blocks that can serve our scope, display applications have a number of specificities with important implications on how web technologies can be used in this context.

Firstly, we found that in order to effectively leverage on developers' web development experience, clear development specifications, guidelines and tools are required for creating web-based display apps. Secondly, we provide a detailed description of a set of specificities

[29] http://depthjs.media.mit.edu Accessed September 26, 2014
[30] http://nodejs.org/ Accessed September 26, 2014

and insights of display applications together with the best ways to adapt Web technologies for the creation of this type of application. This contribution will determine and shape the emergence of new web-based models and wide available display application ecosystems. While many Web techniques are ready available for being used in development of display applications, the challenges arise from the particular usage scenarios and user experience offered by Open Display Networks.

During our work in Instant Places, we also encountered a set of considerations that limits the applicability of our research results. Firstly, Instant Places infrastructure with its web-based application model and the overall development experience built over the last four years should be seen as a tool or a starting point that guided this research and informed the emergence of the stated development specificities. The rationale behind these specificities is based on the set of application design principles, properties (formulated within the PD-Net project) and the specific assumptions underlying Instant Places system. For instance, we do not offer any detailed insights concerning the interaction model and in what ways the user interaction would be specific in display applications considering the Web approach. Secondly, our development experimentation groups had a small number of participants, i.e., five in the hackathon and three for the long-development activity. While this aspect could challenge our research results, our approach is motivated by the lack of any basic understanding of third-party application development for public displays and this work should be seen as the first approach within the research community that aims to define this specific case of web-based software development for multi-application displays. In consequence, our contribution would build the foundation of subsequent web application models as part of the overall vision of Open Display Networks.

Chapter 5

Application Diversity

The idea of third-party applications for ODNs has the potential to change the value proposition of the currently ignored public displays. Instead of having display serving just a single purpose as enabled by a default content playing application (e.g., most advertising displays), we rather envision the benefit of having a multitude of applications that may concurrently be running, and be able to handle user requests. However, this multi-application vision has several implications on how we are going to use such display technology (Section 3.3.5). Building on the *Concurrent Applications* design principle, we go further by challenging the range or diversity of applications that people might expect in public displays.

Building on the parallel with the mobile application landscape, application diversity is often presented as one key motivation for a similar model for display applications. While we can easily acknowledge the overall potential of the approach, we cannot base our entire expectations on what is today the success and the characteristics of mobile applications, as there are significant differences between the mobile ecosystem and that of public displays. Mobile applications are designed for specific and personal devices owned and used by a single person. Display applications target a shared environment, where they can impact multiple stakeholders, from display viewers to venue/display owners and to content or applications creators. One possible implication, for example, is that *common expectations* regarding the range of available applications may become more important than *distinct preferences* in which each display is entirely different in its application set than all the others. Thus, managing common expectations might be reasonable in public displays as they are inherently shared devices. This may also suggest that common application interests could determine more restrictive application sets for display systems, which is in opposition to the huge application diversity employed in personal devices such as smartphones.

In this chapter, we extend the work on people's perceptions towards public displays by *uncovering some of the limits of application diversity* in future scenarios where broad availability of third-party applications could offer people the opportunity to select potentially any content they would like to see in public displays (Taivan & José 2014). While previous work have revealed different behaviors on how people perceive the value of public displays in relation with the content shown as was the case of single-application displays (Huang et al. 2008)(Muller et al. 2009)(Memarovic et al. 2012), our study identifies a set of perceptions and expectations regarding the usage of a diverse set of applications for public displays. We focus on the concept of application rather than content, as it can offer audiences rich and custom designed experiences beyond the classical model of merely distributing content where people are just passive consumers of the respective information.

5.1 Research Design

A key challenge for this research was the current lack of any established systems where display apps are already being created and used in everyday life. The prevailing model in current Digital Signage networks does not consider applications, and research efforts have typically focused only on specific parts of the problem domain. This means in the first place that we do not yet have any real display application ecosystem from which to obtain data about application diversity. It also means that we cannot expect people to be able to envision the range of display applications that they are likely to have in the future.

To mitigate these challenges, we have devised a research methodology anchored on what is currently the rich and diverse range of applications in the mobile application market. Even though there are multiple differences between both systems, the mobile app stores provide an extremely diversified set of applications that are already part of people's lives and therefore an excellent sample to consider display-based versions of the same applications. By challenging people to think beyond the most common and obvious examples of display applications and consider scenarios that they would otherwise never envision, this approach has allowed us to significantly broaden the range of applications being considered in this study.

The experiment involved the creation of a representative application set and a questionnaire about the envisioned value of having those applications available on public displays. In order to study the application diversity for public displays we identified two dimensions: (1) the range of application categories seen as relevant for public display usage and (2) the set of application categories associated with different types of places.

5.1.1 Application Selection

The initial step was the selection of a representative sample of display applications. We used Google Play application store[31] as our source of applications. For creating a representative set of applications for our study, we selected applications from the whole range of application categories in the Play Store. However, the number of applications in each category was not uniform. Instead, we used data on mobile applications usage (Böhmer et al. 2011) to select from each category a number of applications proportional to the total number of applications in the same category. We came out with three levels of categories.

The first level involved those categories with more than 500 distinct applications, i.e., *Books & Reference, Business, Comics, Communication, Lifestyle, News & Magazines, Productivity, Social, Tools, Travel & Local*; in this level we chose 5 applications per category (50 in total). The second level involved categories with more than 100 distinct applications, i.e., *Health, Finance, Sports, Shopping, Multimedia*; in this level we chose 3 applications per category (15 in total). The third level involved categories with fewer than 100 distinct applications, i.e., *Education, Entertainment, Transportation, Medical, Weather*; in this level we chose 2 applications per category (10 in total).

[31] There was not any specific reason to choose Google Play store.

To maximize the diversity of applications, we selected within each category, applications that were as diverse as possible. For example, in the case of *Book & Reference* category, we selected apps ranging from dictionaries and information sharing to software for reading e-books. In the end, the selection process identified a set of 75 applications.

5.1.2 Application Categories

The second step was the categorization of the applications. The goal was not to envision what future application categories for display applications might be, as real display applications categories will evolve based on the dynamics of application usage over time. The goal was to provide a frame of reference to support the analysis of the results. For this, we considered two different types of categorization.

The first categorization was to simply use the categories from the Play store itself. These were already associated with the applications and their analysis provides an interesting path for comparing the diversity in mobile app stores with the potential diversity of public display applications. Since there was a disparity between current Play Store categories and those described in the study by Böhmer et al. (Böhmer et al. 2011) that we used for app selection, the Play Store categories Media&Video, Music&Audio, Photography were merged into a single one called Multimedia. The categories of Games, Widgets, Libraries&Demos, Personalization, Live Wallpaper were not considered because most of the applications in those categories were mainly aimed at specific features of mobile devices.

The second categorization was specific to public displays applications. For this, we analyzed the literature on display applications in search for different classification dimensions for those applications. To minimize the potential subjectivity involved in classifying applications that do not exist yet, we have only selected very high-level classifications and we have explicitly defined any additional assumptions that were needed to resolve ambiguities. The result was a set of 6 application categories, described next, that represent the main combinations of different categorizations from the literature. We then classified our sample of 75 applications according to these categories.

1. **Personal (11 apps).** This category includes applications that are based on content that is to some extent private as identified in (Morales-Aranda & Mayora-Ibarra 2007). Such applications are traditionally perceived as more appropriate for an individual usage. The content or services these applications provide can only be accessed by its owner and is less suited for public broadcasting e.g., *Private Diary*, *File Manager*.
2. **Informative (21 apps).** This category includes applications whose primary aim is to disseminate information through public displays, regardless of the specific type of content, as described in (Ojala et al. 2012). Content provided by this type of application can be presented in public circumstances where there is more than one person attending a display e.g., *IKEA Catalogue, Wikipedia*.
3. **Situated (8 apps).** This category includes applications that address the display context. We followed the description from (Langheinrich et al. 2011) where the authors make a distinction between content that is static, i.e. does not consider the display context and content that is dynamically assembled for each particular display. This is where we

included all types of location-based applications, e.g., *Where are you sweetie*, *GPS Navigation and Maps*.

4. **User Generated Content (4 apps).** This category includes applications whose primary aim is to support the publication of user-generated content, according to some particular publication paradigm (José et al. 2013). This is where we included most social media applications, e.g., *Facebook*, *Instagram*.
5. **Interactive Experiences (18 apps).** This category is based on (Ojala et al. 2012) where the authors describe that functionality, not information type, defines a display. It includes applications that involve rich user interactions leading to an engaging, and possibly playful experience where one is totally absorbed by the interaction. The goals of these applications vary from diverse communication practices, e.g., *Skype*, *Azores Cam 2*, to entertainment and artistic scenarios, e.g., *Real Piano*, *Fun Face Changer*.
6. **Other (13 apps).** This category includes applications that do not fit into the other categories, mostly because they do not have a specific type of content that could be shown on the display. They are rather seen as tools for specific goals, e.g. *Smart Compass*, *QR Code*.

5.1.3 Application Distribution among Places

Another dimension of application diversity is the extent to which different types of places may suggest different sets of application categories. We would like to identify particular applications or application categories that may be seen as universally relevant or places that stand out in terms of their unique application set. This would be an important hint for managing user expectations in regard to each new public display that they may find.

We defined a set of 8 types of places: **Parks:** city parks, children's playgrounds; **Shopping:** Malls, Hypermarkets; **Transport:** Airports, railway stations, bus/metro stations; **Squares**: Cities' square, Plazas; **Shop Windows**: Shop Window in Streets; **Bars:** Cafés and Bars; **Sports:** *S*kate parks, Football or Basketball Stadiums; and **Corporate:** Public and Private Institutions. Despite the potentially substantial differences between these places, we did not consider any potential limitations imposed by different types of display capabilities, e.g., touch or gesture interaction or display position, e.g., direct reach.

5.1.4 Experimental Procedures

To collect data regarding peoples' perception of the value associated with the applications in our sample, we created a mockup of a display app store populated with the 75 applications from our sample. They were described by title, a small application image and a short description as found in the Play store. The goal was to ask participants to use the mockup app store and select the applications they consider relevant for public displays. To manage the effort needed to answer the questionnaire, each participant was only shown 15 of the 75 applications. To guarantee an adequate distribution of the various application categories, we created 10 different combinations of 15 applications, each to be shown to each participant.

The procedure involved two consecutive steps. In the first step, we presented participants with 15 applications (a random group from 10) in a random order and asked them to indicate the relevance of each application in public displays using a 5 point Likert scale (1 – Not relevant, 2 – Slightly relevant, 3 – Not sure, 4 – Relevant, 5 – Strongly relevant). Applications were shown without any indication about their category. In the second step, participants were asked to associate each of those applications to the type of place, from our list of 8 they considered to be appropriate for a particular application. Participants could associate each app with more than one place.

The questionnaire was announced on several internal mailing lists at our University and also through flyers distributed at two university bars. During the one month period in which the questionnaire was open, we received answers from 72 different participants, mainly students, researchers and professors (most of them had background in computer science). An answer from one participant corresponds to the assessment of 15 distinct applications and overall, we had 1080 applications assessments. On average, each application was assessed 14.4 times, with every application being assessed at least 12 times. From all the assessments, 307 (28%) were for applications that have been marked as not relevant for public displays and 773 (72%) were for applications that presented some relevance between 2 and 5 values of the Likert scale.

5.2 Results and Discussion

In this section, we analyze the results according to each of the categories and also to the association to places.

5.2.1 Mobile Application Categories

The first line of our analysis is the relevance associated with the categories from the mobile app store, as represented in Figure 26. For this analysis, we aggregated the answers for each application category and calculated an average score. While the results for each individual category may raise interesting interpretation questions, the lack of qualitative data about the options made by participants do not allows us to take many conclusions on particular order of the categories.

We can however point that the most relevant one, *Medical* category, was actually composed of only two applications, both associated with emergency situations: *In case of emergency app* (location based listing for hospitals and doctors, SOS message) and *First Aid* app (helping people to follow the right procedures in a stressful situation). This seems to be aligned with previous work that considered the potentially strong role that public displays could have in emergency situations (Davies et al. 2012).

Similarly, the *Multimedia* category, was only composed of three applications, i.e., *Real Piano* (play piano), *Customizable Gallery 3D* (make 3D photo galleries), *Diptic* (tell a story by combining multiple photos to create a photo collage). The low rank of these creativity-oriented applications may suggest that people perceive the role of public displays as being mainly informative, and would probably not feel comfortable in exploring the entertainment and playful side of public display installations or expressing themselves through a medium that they do not yet fully understand. This would be in line with previous results on people's preferences of content by Müller et al. (Muller et al. 2009) and is also coherent with the observation that almost all categories in the first half of the diagram (from

Medical to *Weather*) are mainly about informative content. However, it may represent a key challenge for the many types of interactive display applications (e.g., (Memarovic et al. 2011)) that are increasingly deployed in urban environments, as people may not yet be prepared to understand the full potential of public displays as a highly interactive and public multimedia tool.

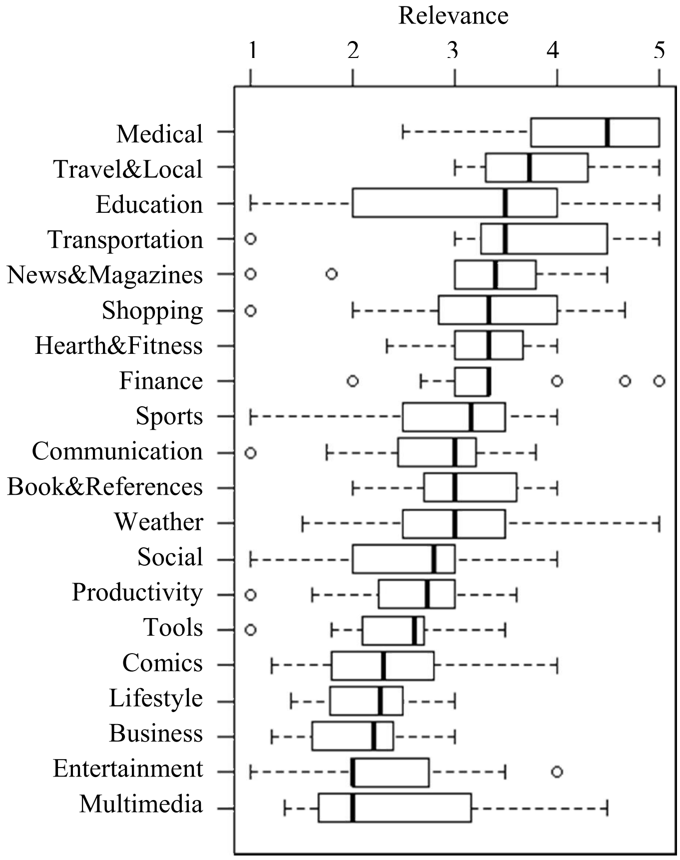


Figure 26: Application relevance by app store categories

5.2.2 Public Display Application Categories

The second line of our analysis is the relevance associated with the categories from the display application categories, as represented in Figure 27.

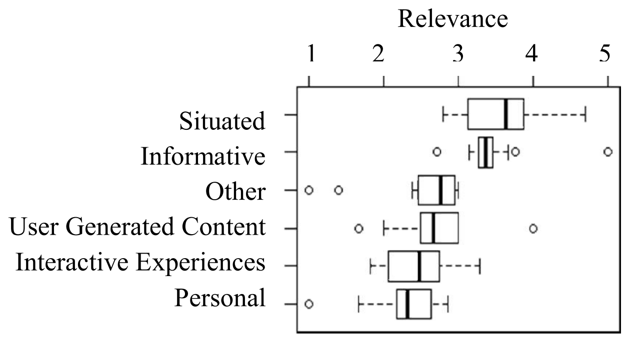


Figure 27: Application relevance by public display application categories

When we considered these categories, the preference for applications focused on informative content became much clearer. However, applications with the ability to offer situated content should be favored in regard to applications that merely distribute static content. This seems to confirm that situated content that is dynamically assembled at each public display is perceived as more relevant for public displays than their use as a mere replacement for traditional static digital displays (Memarovic et al. 2011). A Kruskal Wallis test was applied for the first three categories: *Situated, Informative and Other* and it revealed a significant effect of categories on application relevance ($\chi^2(2)=26.05$, $p < 0.001$). A post-hoc test using Mann-Whitney tests with Bonferroni correction showed significant differences between *Situated* and *Other* ($p < 0.001$, $r = -0.75$) and between *Informative* and *Other* ($p < 0.001$, $r = 0.79$).

The relatively low ranking of applications for user-generated content may represent a huge challenge for the many display deployments that are now exploring the intersection between public displays and social media. Even though most people are now social media users, they do not seem to understand the possible role of public displays as an additional channel for the expression of their identity. Previous work on sharing social media on public displays has also identified this problem. For example, Memarovic et al. (Memarovic et al. 2012) studied sharing practices of social media content and found that personal content, e.g. pictures from last night's clubbing, comments, personal photos or personal status updates is not desirable for publishing and viewing on public displays. Instead, people prefer to use Social Networking Services (SNS) to share this type of content. This also seems to confirm the need for new publication paradigms that enable people to publish on public displays while being in full control of the process itself and especially its social meaning (José et al. 2013).

Again, the low score of *Interactive Experiences* category seems to confirm the idea that people do not immediately perceive the creativity, playfulness or communication potential of public displays or at least may fear the "social awkwardness" that may still be associated with most such experiences (Brignull & Rogers 2003).

The case of *Other* category reveals that people do perceive a public display as a tool where different functionalities may exist to assist them in short-task based scenarios such as Google Search and QR code. Such applications, as opposed to those from *Personal* category, can be more appropriated in public circumstances as they do not employ too much consideration for people' privacy. In the case of *Personal* type of applications, people were more reluctant for expecting them in public displays and that's why there are situated at the bottom of the boxplot. These applications involve a more personal usage experience given the nature of the content they might employ. This is aligned with previous work that provided some evidence in this sense. The LunchTable is a large digital display system supporting group conversations (Nacenta et al. 2012). In a seven days study, participants used the display for sharing more graphical data (maps, pictures, videos) than textual data. *Google Search* was the most often launched to briefly search for and navigate to another page. On the contrary, *Gmail* application was opened only once, which tell us that this type of personal experience is less expected in public spaces. While this result depend on the social setting in which a display is installed, e.g., people might know each other or only a single person controls the entire display space at the time, the finding from our study suggest a weaker expectation about this type of applications.

5.2.3 Application Distribution among Places

The third line of our analysis is the association between applications and places. We conducted the analysis from the perspective of application categories (Table 13) and from the perspective of places (Table 14). To analyze the association between application categories and types of place, we first created a table with the scores of each category for each of the places. A score was how many times the applications from a particular category were selected for a specific place. We then converted the scores into percentage scores using the total number of application assessments within a category – value that shows the maximum number of selections that can be attributed for a place within a category. For instance, in the case of *Personal* category we had 171 assessments (resulted from 11 apps, one app being assessed in average of 15.54 times) and only 27 were for *Parks;* this gave us a percentage score of 15.79%. Next, using the percentage scores, we calculated the mean (M%), standard deviation (SD% not shown), the coefficient of variation (CV%). In the end, we aggregated the results based on the two perspectives. In the case of Table 13, we show the results by category and in the case of Table 14, the results were shown by places.

In the first perspective (Table 13) a category with a higher CV score is one that presents more significant differences in relation to the types of places where it is seen as appropriate. For instance, in the case of *Personal* category, the highest CV shows that people were much more sensitive to the types of place when considering the use of applications in this category. On the contrary, more generic applications, such as those focused on informative content, are seen as potentially relevant anywhere. This means that they are more likely to become part of the expectations people may have in regard to any public display they may find.

In the second perspective (Table 14) a place with a higher CV score is one that presents more significant differences in relation to the type of applications that can be expected. For example, in the case of *Shop Windows* the highest CV shows that there is a stronger focus on specific application categories, while for *Bars* the lowest CV would potentially represent a place with a broader set of application categories.

Overall, these findings suggest that people's expectations in regard to the type of applications change according to the characteristic of the place. This idea that different places call for different types of applications is in line with the long-term insights from real-world display infrastructure (Ojala et al. 2012)(Schroeter et al. 2012) that clearly observed the importance and effects of the location upon the content and applications usage. One such effect described in (Ojala et al. 2012) is that location might decrease interaction. The authors observed how a similar display deployed in a swimming hall and a business center generated much more interaction on the swimming center because people had more time to spend without being in hurry.

Table 13: Application distribution among places organized by categories

Public Display App Categories	M	CV%	Most / Least Relevant Place
Personal	24.56	39.39	Corporate / Shop Windows
Interactive Experiences	26.35	31.48	Bars / Shop Windows
Situated	48.84	31.46	Transport / Corporate
User Generated Content	33.90	27.90	Bars / Sports
Informative	41.64	25.85	Transport / Corporate
Other	34.92	17	Shopping / Corporate
Most / Least Relevant Place(s) across Categories			**Transport, Bars / Corporate**

Table 14: Application distribution among places

Places	M	CV%	Most / Least Relevant Category
Shop Windows	28.23	44.18	Situated / Personal
Parks	34.70	40.79	Situated / Personal
Squares	38.97	40.47	Situated / Personal
Sports	25.77	32.44	Situated / Personal
Transport	45.01	30.61	Situated / Interactive Experiences
Shopping	40.37	28.90	Situated / Personal
Corporate	26.46	26.80	Personal / Interactive Experiences
Bars	40.78	18.77	User Generated Content / Situated
Most / Least Relevant Category across Places			**Situated / Personal**

However, more importantly, these results also indicate that place types may not only influence the appropriateness of particular application categories, but also the range or diversity of application categories that could be found on public displays. This seems to suggest that in a scenario of plenty of application offers, there might be public displays with a more restricted usage and consequently a more restricted application set, e.g., *Shop Windows*, and other displays with a much broader usage scope and consequently also a potentially much broader set of available applications, e.g., *Bars*.

To a certain extent, these results reflect the same perceptions uncovered by Müller et al. (Muller et al. 2009). People judged the expected type of content in regard to the place in which the screen was situated and in particular, to display owner. Thus, for most of the displays people expected information (most serving advertisement) targeted for the respective display owners. A deeper understanding of our insights is given by Schroeter et al. (Schroeter et al. 2012). The authors discussed in what ways different external factors, independent of the application, such as the nature of the location where a display is deployed, the positioning of the screen as well as the selection of content influence not only the frequency of interaction with the screen, but also the quality of the interaction. Therefore, public display applications would need to adapt to different contexts employing different interfaces that in turn enable better user experiences. For instance, a *Personal* type of application might not be appropriate in *Shop Windows* given the intimacy of the experience, available time to use and public or shared nature of a large digital display. Instead, it may be more suited for a *Corporate* type of environment.

Again, these results consolidate the findings from the second line of analysis (public display application categories). While *Situated* applications were perceived more relevant for outdoor locations (the majority of places), *Informative* applications are more likely to be part of common expectation of any display. An interesting example is *Other* category which shows that people do perceive short task-based functionalities as universally relevant apps.

In Table 15 and Table 16 we highlight the strength of people's expectations as informed by place relevance analysis. We did this by sorting Table 13 and Table 14 by M value – which represents the amount of assessments attributed to application categories and types of places. By providing these additional representations of the results, new observations are emerging. A first note to be taken is that Table 15 is identic to the results obtained within public display application category analysis from Figure 27. This suggests that the relevance of an application is deeply connected with its environment where it is going to be used. A second note taken from Table 16 is that people perceive the value of public displays mainly from the urban spaces where most of them are currently in use, i.e., Transport, Bars and Shopping. The other types of places suggest weaker expectations regarding possible application deployments. This might be because people do not yet perceive the role of public display as an additional or complementary communication and multimedia tool (as stated before) or there is a strong value proposition on the primarily activities employed in these spaces. For instance, people might find difficult to imagine the benefits of digital displays within a Sport type of place.

Table 15: People's considerations regarding the type of applications

Public Display App. Categories	**M%**
Situated	48.84
Informative	41.64
Other	34.92
User Generated Content	33.9
Interactive Experiences	26.35
Personal	24.56

Table 16: People's considerations regarding the type of places as informed by place analysis:

Places	M%
Transport	45.01
Bars	40.78
Shopping	40.37
Squares	38.97
Parks	34.7
Shop Windows	28.23
Corporate	26.46
Sports	25.77

5.2.4 Limitations

A key limitation in this study is the way we relied on the ability of participants to envision how different types of mobile applications could be repurposed for the public display context. Being open in regard to how people perceived this adaptation was part of the methodology so that people would not be caught up in the details and could instead focus on the respective value propositions. However, we cannot fully account for the effect that these open interpretations may have had in people's answers and to what extent the results would have been different if participants had answered based on a frame of reference composed by known applications or new applications for which we could provide our own description.

5.3 Summary

In this chapter, we uncovered people's perceptions regarding the diversity of applications that may emerge in future application ecosystems for public displays. Open Display Networks constitute a new frontier for digital content and user expectations are going to affect the evolution of an application ecosystem in this area. Multi-application displays as novel computing devices need to be evaluated within their context by identifying perceptions and distinctive psychologies of potential usage. As well as a full desktop application would not make sense in a mobile device, it is mandatory to remind and understand ourselves that an application is a designed experience for a particular computing device, with particular interactive features used by people in certain situations. Therefore, understanding users' perceptions and expectations before a technology deployment is valuable as it frames our expectations as researchers to create more effective designs and deliver solutions that converge to how people perceive and use digital technologies. These insight are valuable as they may inform the design, development and deployment of multi-application displays that are tied to people's needs and expectations about the envisioned usage behavior (Veenstra 2011).

The purpose of our user study is neither to predict what will be the application categories nor to envision what will be the actual usage in future public displays. Such insights may come only through real world display infrastructures deployments that could reach a critical mass of users over time and inform with possible usage patterns and type of applications, as it is the case of Oulu's display infrastructure (Kukka et al. 2011). Instead of assuming that application diversity would have a similar evolution as seen in mobile landscape, we set ourselves to uncover some of the factors that might affect the range of display applications, which might be more focused on common expectations rather than personal choices.

In a world dominated by advertisements people do not expect much from public displays as they have built over time a strong negative perception about them (Muller et al. 2009). However, instead of asking people about their expectations with current and future public displays, we used a reference set of mobile applications, which constituted a positive starting point in broadening people to think of what might be an interesting application experience to have in public displays. This approach has allowed us to identify several expectations with their underlying perceptions, which can assist researchers in further, specific experiments.

Firstly, our study revealed that there are multiple types of functionalities or applications on which people would be interested in public circumstances ranging from dynamic and informative content to applications providing utilities for short task-based scenarios. It appeared that people do not expect to appropriate public displays for a more personal and individual application usage, given the fact that such experience might involve different privacy issues. Secondly, we observed that certain applications could be considered as core or universally relevant and should be available everywhere, while others are tightly connected to particular type of place.

Chapter 6

Application Control

Public displays are just a subset of the entire ecosystem of computers that feature visual feedback or visual information representations (Schmidt et al. 2012). Following the principle of *Display Ecosystems,* the emerging pervasive display systems can be described as perch/chain sized ecosystems for many-to-many interaction, composed of displays of various sizes including personal handheld devices, and medium/large, shared wall mounted displays, in which many people might concurrently engage in the interaction with multiple displays (Terrenghi et al. 2009).

A first implication emerging from the shared interactive nature of public displays embedding multiple applications (Section 3.3.5) is that passers-by should have access to appropriate techniques that would allow them to control the way applications are shown and used in the respective environment, e.g., can people identify, select, activate and use a specific application. Such techniques should enable each user to reason and express intentions about the system behavior, while also dealing with concurrent requests from multiple users in a way that is fair and clear. A second implication is that a pervasive display environment should be able to manage the temporal and spatial allocation of the displays between any available applications and inform users about any decision made.

Multi-application displays are mainly investigated from the perspective of single user interaction paradigm. In this model, one user at the time can appropriate a large display by accessing rich interactive applications and services (Ojala et al. 2012). Such applications can share the same display by allocating specific regions to different purposes (display space multiplexing) (Lindén et al. 2010). However, there are specific cases when users can access distributed parts of an application on their personal mobile devices (Hosio, Jurmu, et al. 2010)(Hosio, Jurmu, et al. 2010)(Clinch 2013) and may join a collaborative task mediated by the large screen. Alternatively, when users cannot directly reach the public displays or they are not able to launch a specific application, displays shows different applications based on a specific time slot (José et al. 2013)(North et al. 2013) (time-based multiplexing). A distinct system is presented by Davies et al. (Davies et al. 2009). In their e-Campus architecture, users can invoke various applications by using commands in their Bluetooth device names. If multiple application requests are issued, the system serves them based on a queue approach by prioritizing requests originating from phones that have been served least recently.

For the scenarios when just an application is employed, many users can have access to a large shared display and enjoy interaction depending on the available space and the right moment to take the turn (Peltonen et al. 2008). If the display is able to adapt the content on the screen based on the number of users, it will divide the screen according to the number of passersby (Morales-Aranda & Mayora-Ibarra 2007) (display space multiplexing). While social protocols may play an important role in shared use of public displays, this might not be the case for situations involving mobile phones interaction. In this case users may not be mutually aware of each other and, therefore, their perception of the control process may have to be entirely grounded on the information provided by the system.

Controlling a multi-application display is a new topic within research community. Despite the novelty of these infrastructures, we still can get various insights from display systems that already address similar issues when focused on content rather than applications. Dix et al. (Dix & Sas 2010) examined several synergies and opportunities between personal mobile devices and public displays, addressing issues such as the physical size of the situated display, the use and purpose of the mobile devices, the level of integration of the public and personal devices, the movement and physical contact within the interaction, the spatial context of the situated display, and the social context. They proposed a design framework for analyze potential issues, problems and requirements regarding the potential conflicts between individual interaction and audience experience (passers-by and bystanders). In their work, Dix et al. have identified two main types of conflict that occur between the interacting users of the public display audience: conflicts of content (what is seen), and conflicts of pace (when it is seen). Conflict of content can be of three types: *"(1) conflict between the use of the screen for displaying content and for displaying interactive feedback (menus, etc.); (2) conflict between different users wanting different specific content (3) conflict between the particular requirements of an individual and maintaining a content stream that is intelligible, useful and engaging for bystanders".* Conflicts of pace have two nuances: *"(1) users cannot always have things when they want due to other users requests (c.f. content conflict), the playing of media, etc. (2) users cannot speed-up, slow-down, stop or replay the flow of information because of the audience."* Resolving these conflicts is a design challenge for public displays and, in particular, for multi-application and multi-user displays. For the latter case, the main reason being the variety of applications embedding a diversity of behaviors, which might complicate users' perceptions towards identifying and understanding the multiple ways in which content can be presented.

In this chapter, we study how to inform the definition of novel techniques for application control in pervasive display environments that can address the above challenges. These techniques should enable multiple users to concurrently drive the selection of the applications being shown and control their behavior. We have devised a research methodology involving two phases. In the first phase, we investigate concepts of application control from GUI systems based on a mixed-initiative scenario in which a display system and viewers are both involved in the process of application presentation. This provided us with a set of well-established GUI techniques that constituted the basis for analyzing the similar issues in the context of public displays. In the second phase, we used the application control concepts to create an implementation of a set of potential techniques targeted for public displays and report about their first evaluation in a real setting.

6.1 Concepts of Application Control

In the first phase of our methodology, we investigated application control mechanisms in traditional GUI systems, more specifically, desktop and mobile devices, and analyze to what extent they could be repurposed for public display application interaction.

As the first step, based on WIMP style of interaction, we selected technical descriptions of application control mechanisms from desktop and mobile interaction GUIs. These were retrieved from various sources, including Wikipedia, books, and various web sites. We used a total of 31 descriptions referring to 20 different concepts, e.g., Alt-Tab, Taskbar, Tasks scheduler, Icons, from various operating systems (OS), e.g. Windows, UNIX, Mac OS, iOS and Android. A complete listing of the input sources can be found in Annex C. We then analyzed each description to identify the further ways in which applications could be controlled. For example, "*Taskbar*" concept got four codes: *Start Menu Button*, *Quick Launch Bar*, *Taskbar Buttons*, and *Notification Area.* Each reference to a control technique was coded using open coding and a dedicated coding software. For every code created, a small memo describing its generic meaning was associated. This resulted in 61 codes. At the end, we conducted several consolidation sessions to establish the main concept clusters in relation to application control, resulting in 5 main categories: *Controlling Application Life Cycle, Application Identification, Implicit Application Selection, Explicit Application Selection* and *Visual Layers*. Finally, using these categories and inspired by the approach described by Bellotti et al. (Bellotti et al. 2002) for analyzing the challenges of sensing systems, we examined the specificities of application control in public displays. For each category, we review the traditional GUI solutions and define a set of new challenges when applied for public displays. When considering those specificities, we assume in particular that there can be many concurrently interacting users in the environment and also that the execution environment of the applications is not a single display, but instead an ecosystem of displays with multiple distributed user interfaces that span across multiple devices, e.g., public displays, mobile phones, touch-enabled surfaces.

6.1.1 Application LifeCycle

The application life cycle embodies the sequence of execution states that occur between the launch and termination of an application, e.g., background execution, suspended, inactive, foreground execution. An application can change its state based on users' explicit actions, operating system or application internal events. For instance, in iOS the application life cycle is composed of five distinct states: *not running* – the state of a rebooted device, *active* – the application is displayed on the screen and receives inputs, *background* – the application may execute code without receiving inputs or update the screen, *suspended* – an application is frozen and its state is stored in RAM and *inactive* – a temporary rest between two other states, e.g., yielded by incoming calls or if the user has locked the screen. While in the active state, an application may require visual and input resources, in the background execution the application is running in a constrained behavior without requiring display real estate or user input.

For public displays, the execution environment should be seen as the physical environment of the displays, where potentially multiple displays may exist. Therefore we should separate application availability in the environment from its presentation on the displays or from its execution on any particular device of that environment. While applications may be expected to be always available and ready to produce content on any display, their normal execution mode may be a waiting mode in which they are ready to receive input signals and in appropriate moments generate content for presentation on the displays. The main challenge is modeling this combination between various execution states, e.g., full-screen content presentation, in a way that people can easily perceive and learn to control.

6.1.2 Application Identification

Application identification is concerned with ways in which users can recognize and distinguish the various apps. Normally, the applications from traditional computing platforms may be identified through icons – that are thumbnail photos briefly describing its functionality or, during execution, by using specific system level indicators of common app description fields, e.g., window title, favicon. For instance, in Windows and Mac desktop environments there is a system-based application that lets users identify and inspect all applications in execution and their respective processes or tasks. In particular, in Mac OS, the name of the foreground app is always on the menu bar. However, in mobile devices, many running applications do not have a clear application title.

In particular, application icons allow users to easily recognize and launch applications. It is represented as a small picture, which intuitively describes the function of the respective program. An application icon is designed to be language independent (does not contain any text) and it offers rapid entries in the system functionalities. Application icons may also be extended to present key application state. For example, Windows 8 features a user interface paradigm based on the concept of live tiles that are dynamic icons with a larger size that identify the respective app and shows app specific data at the same time, e.g., the number of unread email messages for a mail application.

For public displays, identifying applications is also important so that people may associate the content they see on the displays with the application generating that content. An adapted version of GUI concepts, such as application titles may be used in some cases, but may also be inappropriate in other cases because it may interfere with the rich visualization requirements of public displays. Alternative approaches may include a list of the applications that are currently available to be shown on the displays. This list may include the application id and a summary of its content, e.g. live tile, and may be available through mobile devices or occasionally shown on the display to prompt interaction.

6.1.3 Implicit Application Selection

Implicit application selection is initiated automatically by the system or by the applications themselves. The system-based activation is an additional way to launch applications as a result of various event triggers. This may include time-based events or certain system events, e.g. a device join or a change in network availability. When an event occurs it may trigger background or foreground application activation. In Windows, as a consequence of system-based scheduling, most processes are launched in background mode without any user interface, e.g. Server, Network Connections. The application based selection may also entail application specific logic that triggers its appearance, e.g., from background to foreground. This approach is very common in mobile devices where various applications can be triggered by external events, e.g. a phone call.

For public displays, implicit application selection may be part of a regular scheduling process in which the systems iterates over the multiple applications available, but it may also triggered by external events. In a mixed-initiative model, the system would need to implicitly call for specific apps, even if there is no activity from users. Additionally, some applications may only be relevant when particular contextual conditions occur. In such case the system may at any moment make selections based on the interpretation of the respective context, e.g. people presence and their preferences. Therefore, a challenge for pervasive displays is the ability to integrate this dynamic application selection into the application execution mode.

6.1.4 Explicit Application Selection

Explicit application selection refers to an action in which a user requests the activation of a particular application. This may correspond to the initiation of the application or to a change in its execution state, e.g., from background to foreground. Selecting applications from the whole list of available applications can be complex because of the potentially large number of applications. Most operating system offer some type of short-list of commonly used applications, either as more specific sub-lists or through particular GUI elements, e.g. application icons on a desktop environment. Selecting from a list of active applications is contextually very relevant and can be accomplished through specific tools, such as the taskbar, app switch shortcuts, or app docks. For example, in Windows, Mac, KDE, and UNIX a specific key combination, i.e., Alt+Tab, switches foreground execution between the most recent top-level application windows.

For public displays, viewers need firstly to be able to identify available applications and request them. The first challenge is which activation or selection techniques can be more adequate for public displays. Given the wide range of available interaction techniques people can employ very distinctive approaches. For instance, gesture-based techniques enable performative interactions and lower the entry barriers for engagement. Contrarily, mobile devices do offer a richer set of interaction features but they might be perceived as isolating people within the social environment. Subsequently, given the multi-user context of public displays, another challenge is the mediation between possible conflicting requests from multiple users or even between users and system goals.

6.1.5 Visual Layers

Visual layers enable multiple applications to be simultaneously active while sharing a single visual display. The existence of a single foreground layer coordinates where the current focus is and therefore to which application an input should be directed. Applications may trigger visual attention by using a special, always-on-top layer. This is often used for splash screens or in other cases to force users to attend an interaction request. A similar goal can also be achieved through notifications, which are a particular type of visual layer. In a traditional OS, a notification message warns users about application data updates or about system level issues. Mainly, the computer notifications contain two classes: a) one that calls for user attention, e.g., pop-ups and b) the other that does not call for explicit user attention, e.g., pop-under. A pop-under notification contains a non-intrusive content that resides behind scene. In Windows environments, non-intrusive notifications are shown in the notification area situated in the right side of the Taskbar.

For public displays, multiple visual layers can also be an important feature. However, considering that multiple people may be sharing the display, it becomes much more challenging to achieve a balanced combination between multiple layers and a good interaction experience. Still, well-designed notification layers that choose the best time to present themselves may provide an important alternative channel for presenting contextually relevant content outside the normal presentation cycles of the applications. In particular, these alternative visual layers may be important in generating feedback for users trying to interact with the system and support progressive interaction modes in which users and displays are increasingly aligned while minimizing interactions by accident, such as in gesture-based interfaces.

6.1.6 Analysis Overview

The techniques from traditional GUI systems provide a worthy starting point in considering specific solutions for public displays and particularly for grounding a systematic analysis of the challenges of application control. In Table 17 we offer a summary of the specificities of application control in public displays by explicitly stating the traditional approaches alongside the new challenges associated with this particular domain. These may also be formulated as five questions that public display designers should be able to answer to support appropriate application control mechanisms.

Table 17: Traditional GUI solutions and public display challenges for application selection and control

Basic Questions	Traditional GUI Concepts	Challenges for public displays
How can the system or the people using it control the application life cycle?	Users click on buttons; System priorities; Triggers; Execution as service; Execution states.	How to model the combination between various execution states in a way that people can easily perceive and learn to control?
How do viewers identify applications?	Static and dynamic icons, e.g., live tiles, windows title bar, favicon, etc.	How to raise awareness about installed, scheduled or running applications? How to associate content being presented with its application? How to address an application?
How does the system implicitly select applications?	System events, e.g., device join or network availability	How to integrate the dynamic application selection into the default application execution mode? How to select relevant apps based on current context?
How do viewers explicitly select applications?	Keyboard, mouse. Start button, Start menu, App Switch (Alt+Tab), app dock, folders or taskbar.	Which interaction techniques make sense in public displays? How to mediate between possible conflicting requests from users?
How to use multiple layers to enhance application control?	Applications windows and notifications; system level notifications, e.g., pop-up, pop-under.	How to effectively use multiple visual layers without disturbing the current content presentation and overall user experience?

6.2 Descriptive Field Study

In order to investigate how people would succeed in controlling a set of applications in a public display, we conducted an experiment in a university café setting. Our trial had two main objectives: 1) observing users behavior in regard to multi-user conflicts, e.g. engagement, embarrassment, frustration, concentration, enthusiasm and 2) assess the responsiveness of the system logic.

6.2.1 Experimental setup

The experiment involved the deployment of a single public display in a bar at our University Campus. The display was able to show a particular application when requested by users. In our display deployment we used Instant Places platform (José et al. 2013) as the underlying technology for creating and managing applications and also for supporting user interactions. For our experiment of application control, we used the four web-based applications that Instant Places disposes, i.e., *Posters, Football, Presences* and *Place Stream* (Section 2.3.3). These applications were conceived outside the scope of this paper.

In order to present the applications in full screen mode, we developed a specific player with the ability to combine implicit application selection with the explicit requests. While the display system was presenting the four applications in a time based schedule with 1 minute time slot, the requested applications were shown as part of the explicit presentation behavior – starting with the highest requested to the least one. To provide users with feedback about the number of requests, we specifically implemented a web-based application called ShowRequests that had 1 minute time slot to inform users about the ranking of the preferred apps. Based on this feedback, participants could understand how the decisions regarding applications presentation were taken. The ShowRequests app was presented between the default behavior and explicit applications sequences, Figure 28.

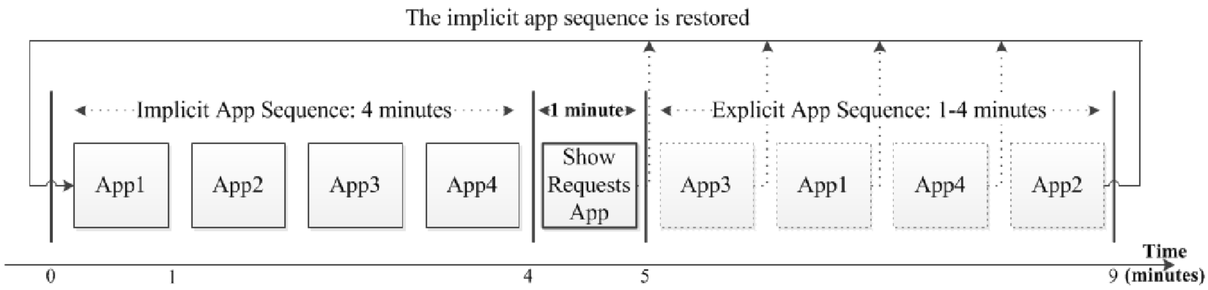


Figure 28: Application presentation logic

To issue application requests, participants used a mobile web application dubbed *MakeRequests*. The mobile app simply showed four icons of the applications that were running on the large displays. During the default sequence, participants could initiate any application requests by touching one of those icons. A confirmation pop-up with the server request acknowledgment was raised after an app selection. Table 18 summarizes the set of techniques that we have implemented in our experimental system to address the various application control challenges that were identified in previous section.

Table 18: Application control techniques in public displays

Basic Questions	Application Control Techniques
How can the system or the people using it control the application life cycle?	Users cannot directly control the application life cycle. They only indicate application' preferences through mobile phones, e.g., touching app icons; The system activates users' requested applications as soon as the implicit app presentation ends. Applications have just one execution state, i.e., full screen content presentation.
How do viewers identify applications?	Splash screens; App icons in ShowRequests and MakeRequests apps.
How does the system implicitly select applications?	Fixed timeline with 1 minute time slot; The implicit application sequence is repeated as soon as the system finishes presenting the users' preferred apps.
How do viewers explicitly select applications?	By using MakeRequests app, people touch application icons and the system stores the requests; Voting logic to deal with conflicts.
How to use multiple layers to enhance application control?	Notification messages in users' mobile devices; ShowRequests app provides users with the ranking of the requested apps.

6.2.2 Evaluation

For the evaluation of the proposed techniques, we invited a group of 6 participants to engage in a situation of concurrent application control in a real setting. Participants were master students and researchers that had no previous knowledge about the display system or the logic behind applications control. In our experiment, we concentrated on how multiple users succeeded to select an application from a list of four. They tested the application selection mechanisms by requesting an application to be displayed earlier than waiting for it to be shown as part of the implicit or default presentation list.

We briefly introduced participants into the applications' functionalities and invited them to make various application selections by using the MakeRequests mobile app. We chose not to provide any details on the system selection process or how the system combines implicit selection with explicit selection from multiple users. This way, the experiment could be closer to a real world situation, where people would not normally have the time or willingness for lessons about public displays behavior. People had approximately half an hour to test how the system worked. In the end, we conducted a semi-structured interview (5-7 minutes) in which we asked participants about the experience with the system. The interviews were audio recorded and transcribed in text. Overall, the experiment had two main data sources: 1) live observation of users' behavior while selecting applications and 2) users' opinions about the logic of application selection.

6.2.3 Results

Understanding the application presentation logic and making a clear distinction between the implicit and explicit applications presentation sequences was not so easy. In the beginning every participant requested at least one application and many requests were concurrently issued. This has considerably increased the complexity of the overall selection process and made it much more difficult for each individual participant to understand how the system was taking into account users requests. After roughly 15 minutes, participants started to focus more on making sense of system behavior rather than just issuing requests. They started to pay more attention to the notification messages provided by ShowRequests public display app. At a certain point, they agreed to make a combined vote that would enable them to uncover the application selection logic. They all voted for the *Presences* app. In this way, they understood how the system resolved the requests and succeeded to familiarize with our approach.

> *„In the beginning all of us selected all the apps and that's why it was difficult to clearly uncover its functionality."*

Participants remarked that the logic was simple, although it required close observation to figure out the content presentation order. The logic was categorized as fair by all the people involved and there was no lock on a certain application due to the implicit presentation behavior.

> *„What I know is, it took me a while to see what the system is and it takes really close observations".*

> *„The application selection logic is reasonably fair for everyone".*

Participants noted that the system had low responsiveness and they would like to have more immediacy in showing the apps. For instance, given the timeline of the implicit application behavior of 4 minutes, one could get his preferred app on screen after those 4 minutes for the best case, and 7 minutes for the worst case. This fact highlighted that implicit application presentation behavior could not be interrupted thus imposing a minimum delay of 4 minutes. Users clearly found this delay a limitation and a sign of low responsiveness. They expected more immediacy for presenting the requested apps, e.g., allow them to interrupt the default presentation order or schedule and get the apps displayed within 1 minute time span – for the worst case. Furthermore, for the case of many applications the solution was categorized as not being scalable – the lowest requested app would always be shown at the end.

> *"... I believe people want the requested apps to be presented immediately."*

Although our display system employed two types of notifications, participants gave us a lot of comments about the need for rich notifications, both for public display and mobile devices. They liked to be informed about when their particular requests were served, when was the appropriate time to vote or when the requested apps were being selected by the system. This importance of the feedback in the interaction process is in line with previous findings by Kaviani et al. (Kaviani et al. 2009).

> *"You can use more instructions in the MakeRequests mobile web app. You do not use the screen. The screen can be used for more information, e.g., nr. of requests. Would be better to inform people when a particular requested app will be shown and when is the default or implicit behavior."*

Participants further suggested that a well-designed set of notifications could keep users engaged without losing their interest for the desired application. For example, a good system improvement could be a status bar for both devices (public and personal displays) presenting what the next app is, which is the current applications sequence and when people can request apps. Moreover, the MakeRequests mobile app could have the ranking of the requested apps and even a history of the previously most selected apps.

> *"While we are using the MakeRequests mobile app, we may have more information, e.g., history of the previously requested apps. Also, would be nice to inform people when the time to vote is."*

Other notifications recommended for mobile clients were those that informed users about the presentation time of a requested application (e.g., "*Watch the screen, this is your preferred application*") or when exactly that will happen (e.g., "*In less than 1 minute your requested app will be shown*").

6.2.4 Discussion

An appropriate design for application control techniques is one that delivers acceptable responsiveness and rich and clear notifications in such a way that people may reason about system behavior and understand how they can control it. In the following we are discussing three design considerations that we reached as a result of our system deployment.

6.2.4.1 Making sense of the system behavior

Our multi-application display system presented applications based on a mixed initiative model in which the system and users influenced the applications presentation sequence. Our applications had a simple lifecycle with just one execution state, i.e., full screen content presentation. A distinct use case was mentioned by a user who would like to "*freeze*" an application presentation in order to get sense of its entire content. We believe that there are various similar scenarios in which the existence of a rich application lifecycle would be a strong requirement for future display systems. For instance, the *Presences* app might go from full screen presentation state to a background execution state from which it can pop up unobtrusive data notifications and release the full screen side for other applications that really need more screen space, e.g., *Poster* app.

6.2.4.2 Alignment

In our experiment, participants had difficulties in monitoring the system response to their actions. We only provided the minimal notifications mechanisms both for large display and mobile devices. For instance, our four applications could be identified based on their titles and associated splash screens. While the *Presence* and *Place Stream* apps could be easily identified by their titles in the top side of the screen, *Poster* and *Football* apps had rich visualization requirements and only employed splash screens as a technique to avoid information overloading and still deliver high resolution content.

The MakeRequests app somehow created an analogy of what is presented in the large display, though it had a simple interface of showing the application icons and titles. The clue here was that our web apps (display and mobile) used the same data source or API to get the application metadata and implicitly the graphical representation in the mobile devices was similar with the implicit presentation sequence in the large display.

Given the rich application lifecycle requirements of public display environments, we learn that people require various ways for app identification. The issue can become very complex if we consider that applications may be run in background mode. Still, even for our case, the simple and single visual layers we deployed were not sufficient and there is a need for a set of GUI elements and concepts that could complement each other and better characterize this particular interaction scenario, e.g., status bar with application names or rich information user interfaces for the mobile clients. One user clearly stated that our display system required "*close observation*" and this was because of the lack of visual layers we implemented.

The need for more notifications for both public display and mobile clients was also motivated by the real life scenario where one can be distracted and miss the notification delivered by the public display. To overcome this issue, participants suggested the implementation of similar notifications for the mobile clients, e.g., checking the applications requests on mobile devices. Whilst achieving a balanced combination between multiple layers and a good interaction experience constitute apparently a non-complex issue, our observations uncovered the opposite. Despite the fact we had only one full screen application presentation at the time, which meant simple display layout, people wanted to get personal notifications about the effect their commands would have on the current application presentation.

6.2.4.3 Fairness and responsiveness

Participants found difficult to grasp the overall content presentation experience given the existence of the explicit application presentation sequence. People often mentioned that "*The system does not seem to respond*". Therefore, the interviews reflected the need for a system that presents the users' selected applications in a more responsive way, i.e., interrupt the default behavior and have the requested applications presented in less than 4 minutes. Others believed that 1 minute could be a long time presentation for certain applications, e.g., *Presences* app.

Even so, the application selection logic appeared to be fair for all the users involved and enabled a systematic approach in resolving multi-user requests as long as people understood the system and set their expectations accordingly. Firstly, this means that participants' expectations had an important role to set their perceptions about the fairness and responsiveness of our system logic – a fact that we clearly observed and stated throughout the interviews. For example, one user said that could not find any other fair solution in that moment to improve the way applications are presented and still avoid the application locks.

The system responsiveness was clearly seen as a limitation, but user' expectations were very important in regard to this issue. We have not conceived the system to respond immediately and in most realistic scenarios, people would not get an immediate response from a display. However, proper feedback has shown to be crucial in leading people to understand this and accept that their request was being given attention and would be answered soon. If that was the case, perhaps responsiveness has been understood less negatively.

6.3 Summary

Future open display system will tend to increase their value proposition by allowing content from multiple sources and concurrent applications that are able to continuously react to multiple viewers. Such open display environments are facing radically new challenges for application control.

In this chapter, we first analyzed traditional GUI concepts for application control and discuss how they could serve as the basis for addressing similar challenges in multi-application display environments. In this sense, we came up with a systematic approach for identification and aggregation of challenges involved in the design of novel mechanisms for applications control in public displays. The results highlighted that there are many similarities and therefore many common solutions that should be easily adapted for this new application domain, but also identify a number of unique challenges that may require addressing the specificities of multi-user interactions with public displays and associated social protocols.

Secondly, we addressed the challenges of application control in multi-application and multi-user display systems by designing, implementing and evaluating a set of novel techniques. This was handled as part of an experiment in which we deployed a public display system in a café setting and asked six participants to evaluate the application selection experience. The system employed multiple applications and was able to receive explicit application presentation requests from multiple viewers. The results indicated some success in enabling viewers with the necessary techniques to influence the application sequence in a public display and to mediate the potential conflicting requests. While all of the participants reported the easiness and fairness of the application selection approach, they also noticed the need for more responsiveness and rich notifications for the various human-computer interaction stages, both for public and mobile devices.

Chapter 7
Conclusions

In this thesis we set ourselves to unpack the potential of third-party applications for public displays by understanding how such experiences could motivate both developers and content creators to create meaning when engaging a display system either passively or interactively. We did that by formulating the concept of *third-party display application* for Open Display Networks over three main dimensions: *design*, *development* and *usage.* By understanding the application design perspective we are able to imagine and specify clear scenarios in which public displays can be used. This knowledge provides a systematic exploration of public display design space as it enables researchers, users and display owners to reflect the technical possibilities of these systems and their real added value for the current state of our digital landscape. The feasibility perspective of our work is covered by the usage of Web technologies to create these applications, which present a series of benefits and opportunities that are ready to be leveraged. Finally, a closer focus of the end-user experience is considered as multi-application displays would require novel concepts and mechanism on how applications are presented and controlled.

The findings from this thesis help in shaping our expectations as researchers towards the usefulness and added value of multi-application displays. Instead of concentrating too much on interaction itself and make it a goal on its own, researchers and designers of multi-application displays need to pay very close attention to the user experience of public displays, including their social, cultural and physical meaning, i.e., why to interact? or in what ways an interactive public display would enrich one's experience in a public venue. As soon as we have clear scenarios with solid value propositions that can be tested in urban settings, we could employ them by using different technological approaches including relevant or appropriate interaction techniques. Therefore, our position is that third-party display applications developed using Web technologies are adequate to be repurposed for the execution environment of Open Display Networks. Web-based display applications are able to deliver a pleasant and engaging user experience both as content publishing mechanisms (a passive experience) and interactive tools that pack features and functions related with their content (an interactive experience).

7.1 Contributions

By achieving our objectives and conceptualizing the notion of a *third-party display application* under different dimensions, we have developed the *fundamental understanding* about what a display application ecosystem might be and what implications it might have on current Web technologies. This enables a better focus of our efforts (as part of the entire research community) towards the realization of what is called Open Display Networks. In the following, we provide the summary of how we pushed further the state of the art of public displays research.

7.1.1 Reframing the Scope of Public Displays

The first contribution is reframing the scope of public displays to employ many applications that can be created by third parties, i.e., display applications, instead of content items. We describe in details a parallel between the current closed displays and those with multi-application support. In this sense, we provide an understanding of the components and explain the difference between what we call "content" and what we call "application". Further, we provided a definition for display applications, describe potential usage models of multi-application displays and give examples from related display infrastructures. Overall, we highlighted the need for our work by unfolding the research on three main dimensions: design, development and usage and explains the arguments in doing so. This contribution will have direct impact on the emergence of multi-application displays by motivating researchers and designers about the benefits of display applications. While many other definitions might appear in the future, this work provides a fundamental understanding regarding the convergence of public displays and desktop computers and shape researchers' expectations about the usefulness of future multi-application displays.

7.1.2 Design Principles and Properties for Display Applications

The second contribution is an extensive analysis of multiple collective efforts in instantiating the concept of applications for Open Display Networks. This work enabled us to identify the key design principles emerging from the research community (including the entire PD-Net effort) and the main challenges involved. This contribution frames the content of the entire thesis, as well as it informs with new research opportunities towards the vision of applications in Open Display Networks. The role of these principles is to provide abstractions that orient designers to different aspects of their application designs and facilitate thinking about trade-offs between one design and another. They also provide a systematic approach to understand the design possibilities of display applications with their underlying properties and requirements, which are expected to complement and integrate the services and applications that are already deployed using personal computers (laptops, tablets, smartphones, and wearables). Therefore, this contribution would directly impact the shape of future application ecosystems for public displays.

7.1.3 Web Development Considerations for Display Applications

The third contribution is the definition of a set of development considerations that developers need to consider when creating web-based applications for the execution environment of public displays. We highlighted what are the opportunities and limitations of Web technologies in supporting those considerations and assessed the development process by third-party web developers. We make distinction between two types of development considerations: *specificities* that shape a new web usage model specific for public displays and *insights* that are results from our experiences in creating situated-display applications. We have identified key development specificities along four lines of the design space: Adaptation, Situatedness, Content Management, and Concurrent Applications. These specificities help developers to understand in what ways building a web-based display application would be different from an equivalent one for desktop computers. On the other side, our development insights introduce developers into the appropriate web mechanisms that can be leveraged for the development of situated display applications and do not have the meaning of achieving something specific for public displays. The same mechanisms can and are currently employed in various web-based applications in other computing platforms and we have not reached any implications when repurposing for public displays. Having these considerations clearly identified, it would shape the emergence of potential application ecosystems around public displays, in which developers may understand how to design various web-based models for this type of application.

7.1.4 People's Perceptions towards Display Application Diversity

The fourth contribution is the identification of people's expectations and perceptions about the diversity of applications that might be deployed in future public displays where broad availability of third-party applications could offer people the opportunity to select potentially any content they would like to see in public displays. Building on the fact that display applications are radically different from desktop or mobile counterparts, our position is that common expectations regarding the range of available applications may become more important than a huge diversity in which each display is entirely different in its application set than all the others. This work should be seen as a first step to understand people's interests towards display applications and not to predict what would be the future application usage behaviors. In this sense, we concentrated to uncover some of factors that may affect application diversity in future scenarios of display deployments. Thus, the main impact of this contribution is to frame our expectations as researchers towards the emergence of an application ecosystem in this area. These factors may directly aid developers as they could make more informed decisions of what types of applications to create. In addition, this work contributes in the designing of application stores for multi-applications displays by informing designers that people may value more common expectations rather than a huge diversity of personal application choices. Overall, the methodology employed in this study motivates and challenges people to imagine novel ways of using interactive public displays that can fit into their daily life activities by providing real benefits.

7.1.5 Concepts for Display Application Control

The last contribution is the specification of five concepts of application control for public displays: *Application LifeCycle*, *Application Identification*, *Implicit Application Selection*, *Explicit Application Selection* and *Visual Layers*. The main usage of these concepts is to provide the necessary understanding on how to approach the application control in a multi-application and multi-user public display. Designers of future public display that embed many applications should provide appropriate application control techniques that take into consideration these concepts. As well, independent or third-party application developers should reflect on these concepts as they enable to consolidate what display applications might be and how potential users are going to use them. We also offered first insights into a multi-application display deployment in regard to how people select a specific application. During the evaluation of the system, we identified several expectations and perceptions that are valuable feedback on how to design more effective application selection mechanisms.

7.2 Future Work

By defining the concept of *third-party display application* new research and development opportunities are emerging. Our contributions are thus essential in the development of future multi-application displays aligned with the vision of Open Display Networks[32]. However, during our experiments as part of the Instant Places infrastructure, we identified a series of limitations that deserve a special attention and constitute important building blocks of an application ecosystem for public displays.

7.2.1 System Software

Deploying a display infrastructure in a real world environment requires many software components to be in place. One such component that has a key role in the overall experience of multi-application displays is the player that runs the applications and is able to act as a scheduler. As we assume the provision of applications from many sources that can be presented in different ways, e.g., as users pass by or an application is not ready yet – a display system would need to take decisions dynamically by embedding specific controlling functionality. In our approach, we considered the player as a Web-based component, which runs in a browser and coordinates most of the application events.

When applying the identified concepts for application control, we faced several limitations in regard to the richness of system software support of our multi-user and multi-application public display. In order to support our scenario of controlling the sequence of applications, we needed to provide a more dynamic application scheduling support and the capability of viewers to issue application presentation requests. For these reasons, we built a specific web-based software component that had the role of playing or presenting a set of web-based applications, count users' requests and acknowledge them. However, the scenario described in detail in Figure 28 could be even improved and allow users' requests to stop the default application presentation and show up as soon as possible.

Future work on integrating these concepts of application control into new system deployments would require the player to communicate with applications and dynamically adjust the playback time depending on each application presentation requirements. Then, the player should be more responsive in handling users' application requests and inform people on how their requests would be served. HTML5 Web Messaging [30] will be a candidate technology for implementing the communication between the player and applications. As highlighted by third-party application developers, additional features for the player would include removing or adding applications to the schedule, logging errors and reporting them to the application developers. Having this player in future deployments will enable a more fluid application presentation without putting people to wait too much for the application to be ready.

[32] http://www.hindawi.com/journals/tswj/si/931658/cfp/ - *Application Models for Interactive Public Displays*, is an open call for a special issue in which Instant Places team members (including the thesis' author) have been involved in writing the proposal. The contributions of this thesis have direct impact on the topics covered by this issue and, it is expected, as part of this journal, a new publication that would aggregate our insights in a more compacted version. Accessed September 26, 2014

7.2.2 Further developments of display application concept

Further developments of the concept of third-party display application would include a detailed specification of an application lifecycle in public displays, e.g., background vs. foreground execution, insights also indicated in (Cardoso 2014). In addition, the mobile application landscape can be a source of inspiration regarding the multiple ways in which applications can be presented ranging from a simple task-based utility to an experience meant to consume the user's focus and attention. However, the research would come to analyze to what extent those mobile application experiences would make sense in public displays. The variety of ways in which applications can be presented should be seen as appropriate application contexts in which a user can perform effectively. Getting inspiration from the mobile application ecosystems we can understand that an application can employ many interfaces or presentation contexts that are relevant for particular situations (Fling 2009). For example, in his book, Fling presented how a mobile application can leverage on smartphones features to allow for an orientation change, so if the device is rotated to landscape mode, the app switches from an informative view to a utility view, or maybe from a location-based view to an immersive view. A similar insight is given by a research publication which aims at reaching the sweet spot for building more engaging and effective application design that matches the particular characteristics of public display contexts (Schroeter et al. 2012). Therefore, the concept of a display application should be perceived within its display environment and further research should analyze to what extent this approach can deliver better and pleasant user experiences.

Regarding people's perceptions on the diversity of public display applications, in our future work, we will try to uncover in more detail the set of applications that may correspond to what people may expect to find on most public displays, as well as other application aggregations that may correspond to specific display concepts. Alternative ideas can benefit of the research testbeds of real-display infrastructure (Oulu's infrastructure) and establish some collaborative studies in order to relate expected people's perceptions with actual usage patterns. Based on the emerging insights, we can then reshape the vision of possible application types and inform on how we should design more effectively multi-application displays in a way that maximize usage and provide clear benefits for people.

Studying further the situatedness design principle is also a good way to consolidate our efforts that were initiated as part of Instant Places infrastructure. In addition to show content based on users' requests (e.g., (Davies, Langheinrich, et al. 2014)), display systems may provide content appropriate to their situation, focusing on the physical dimension of displays' environment. Ensuring the appropriateness of displays in both fixed and dynamic situations is a key challenge and new studies are required to evaluate and elaborate further a possible architecture for situated display applications together with specific concepts and application design patterns. The target is to abstract away from specific infrastructures and provide knowledge on how to design global services that would give applications environmental data. Future work will address the need to evaluate our design decisions and propose a set of guidelines that help both designers of multi-application displays and third-party developers to effectively incorporate situatedness into their infrastructure and applications respectively.

7.3 Final Remarks

We expect that our work would be of interest not only for the research community, but also inspire digital entrepreneurs to give a more interested attention to the emergent field of interactive multi-application public displays and to discover novel operational approaches that would make this vision a reality. In the past five years the research community, including our particular efforts, has made important advancements in the notion of multi-application and multi-purpose displays and has deployed several long-term systems that experimented open platforms. The digital revolution we are facing today would definitely push the state of the art of public displays (Sellen et al. 2009). Considering pervasive displays as a communications medium that goes beyond the simple distribution of advertisements would be a must or we will strengthen our skills to ignore them. While the evolution of mobile application ecosystems is a proof of added value and creative engagement, we are looking forward to shape and witness this value change for public displays too.

Annexes

Annex A

The questions used in the Hackathon experiment interviews.

1. What are the differences between display applications and traditional web applications?
2. Could you apply your web development experience to create a display application?
3. How do you see the process of creating a web based display application, is it a lot different than what you have experimented before?
4. Have you used CSS media queries for data presentation?
5. Is it critical that you do not have these tools integrated into your preferred development environment?
6. What were the problems in using the App Generator tool to start building your app?
7. What were the problems in using our Instant Places API library?
8. Have you tested the poster application by using your desktop web browser or media simulator tool?
9. Did you have any problems in understanding the reference tests, for example changing the window sizes and using Fiddler?
10. Did you finish implementing all tasks?
11. How would it be to develop without our tools?
12. Did the tools meet your expectations? Would you like to have more features for the future?
13. Would you find any potential in developing display applications? Give an example.
14. Are you going to keep interest in being a developer for our platform, Instant Places?
15. What motivates you as a developer to keep enthusiasm in creating display applications?

Annex B

The questions used in the long-term application development interviews:

1. What applications have you developed?
2. What are the challenges of using web technologies for creating applications for public displays?
3. From these challenges, which one must developers solve first?

   Extra info: Content management policies, Fault tolerance support, Rich visual adaptation
4. How do you manage the idle times when fetching content from servers? Is your application stop presenting content while the next content item is being loaded?
5. How do you deal with the error messages resulted from content loading errors? Do they appear on the screen? Is the application showing alternative content? Does your app catch them and report to developers?
6. How do you manage disconnected operations? Is the display stops presenting content when there is no connection or the content items failed to be fetched?

   Extra info: prefetch, prerendering, cache, application cache
7. How does your application organize its content? Is your app expose content as content items or resources?
8. What is the range of visual adaptation techniques used in your applications? Examples, responsive web design (with a focus on the viewing distance), alternative views
9. Have you addressed the need of applications to communicate with each other and with the execution container/player? (for configuration, scheduling, integrated presentation behavior)
10. How do you deal with local machine resources, e.g., camera or a Kinect device? How do you access information about the display environment?
11. Does your application keep state between subsequent instantiations?
12. Does your application report the errors to the execution environment/player? Or, did you use any libraries or tools to capture application errors and report them to some entity?

Annex C

The coded text fragments referring traditional GUI concepts for the application control experiments are represented in Table 19. For the coding process it was used MaxQda software[33].

Table 19: List of GUI concepts

No.	Traditional GUI concepts	URL
1	Alt + Tab	http://en.wikipedia.org/wiki/Alt-Tab
2	Background execution	Lee, Wei-Meng. (2010). *Beginning iOS 4 Application Development* (p. 656). Wrox; 1 edition. (Ch. 21 – Background applications, p. 519-520)
		Mark, David., Nutting, Jack., & LaMarche, Jeff. (2011). *Beginning iPhone 4 Development: Exploring the iOS SDK* (p. 676). Apress; 1 edition (Ch. 13 - Grand Central Dispatch, Background Processing, and You, p.464-466)
		http://en.wikipedia.org/wiki/Background_process
3	Computer Icon	http://en.wikipedia.org/wiki/Computer_icon
4	Dashboard	http://en.wikipedia.org/wiki/Dashboard_(Mac_OS)
5	File Shortcut	http://en.wikipedia.org/wiki/File_shortcut
6	Keyboard Shortcuts	http://en.wikipedia.org/wiki/Keyboard_shortcut
7	Local Notifications	Lee, Wei-Meng. (2010). Beginning iOS 4 Application Development (p. 656). Wrox; 1 edition. (Ch. 21 – Background applications, p.530)
8	Manage of Apps	http://ipod.about.com/od/iphone3gs/ss/iphone-app-arrangement.htm
		http://www.dummies.com/how-to/content/how-to-manage-your-droids-android-market-apps.html
		http://www.pcworld.com/article/190266/install_and_manage_apps_in_android_market.html
		http://www.lostintechnology.com/software/two-applications-to-maintain-your-applications-on-android/

[33] http://www.maxqda.com/ Accessed September 26, 2014

Table 20: List of GUI concepts (continued)

No.	**Traditional GUI concepts**	**URL**
9	Metro UI	http://en.wikipedia.org/wiki/Metro_(design_language)
		http://www.microsoft.com/design/toolbox/tutorials/windows-phone-7/metro/
10	Pop-ups	http://en.wikipedia.org/wiki/Pop-up_ad
11	Print Screen	http://en.wikipedia.org/wiki/Print_screen
12	Splash screens	http://en.wikipedia.org/wiki/Splash_screen
13	Start Menu	http://en.wikipedia.org/wiki/Start_menu
14	Suspended State	Mark, David., Nutting, Jack., & LaMarche, Jeff. (2011). Beginning iPhone 4 Development: Exploring the iOS SDK (p. 676). Apress; 1 edition (Ch. 13 - Grand Central Dispatch, Background Processing, and You, p.464-466)
15	Task Manager MAC	http://osxdaily.com/2010/08/15/mac-task-manager/
16	Taskbar	http://en.wikipedia.org/wiki/Taskbar
17	Tasks Scheduler	http://en.wikipedia.org/wiki/Windows_Task_Scheduler
		http://en.wikipedia.org/wiki/Cron
		http://www.iopus.com/guides/winscheduler.htm
		http://support.microsoft.com/kb/308569
		http://www.codeproject.com/Articles/50140/Windows-7-Trigger-Start-Service
		http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=%2Fcom.ibm.mq.csqzal.doc%2Ffg13970_.htm
18	Windows Desktop Gadgets	http://en.wikipedia.org/wiki/Windows_Desktop_Gadgets
19	Task Manager Windows	http://en.wikipedia.org/wiki/Windows_Task_Manager
20	Title bar	http://en.wikipedia.org/wiki/Title_bar

References

Almeida, R.A.P. et al., 2013. Thing Broker : A Twitter for Things. In *UbiComp'13, Adjunct Proceedings, Fourth International Workshop on the Web of Things, WoT 2013*. pp. 1545–1554.

Alt, F. et al., 2009. Adaptive User Profiles in Pervasive Advertising Environments. In *European Conference on Ambient Intelligence*. Springe-Verlag Berlin, Heidelberg, pp. 276 – 286.

Alt, F., Bial, D., et al., 2011. Digifieds : Evaluating Suitable Interaction Techniques for Shared Public Notice Areas Digifieds. *Adjunct proceedings of the 9th International Conference on Pervasive Computing*, (1).

Alt, F., Kubitza, T., et al., 2011. Digifieds: insights into deploying digital public notice areas in the wild. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia - MUM '11*. New York, New York, USA: ACM Press, pp. 165–174. Available at: http://dl.acm.org/citation.cfm?id=2107596.2107618 [Accessed January 31, 2012].

Alt, F. et al., 2012. How to evaluate public displays. *Proceedings of the 2012 International Symposium on Pervasive Displays - PerDis '12*, pp.1–6. Available at: http://dl.acm.org/citation.cfm?doid=2307798.2307815.

Ardito, C. et al., 2012. A Field Study of a Multi-Touch Display at a Conference. In *International Working Conference on Advanced Visual Interfaces,AVI'12*. pp. 580–587.

Ardito, C., 2012. People ' s Reaction when Large Displays Go Beyond Mere Entertainment. *1st International Symposium on Pervasive Displays, PerDis'12, In Adjunct Proceedings, Poster Session*.

Bellotti, V. et al., 2002. Making sense of sensing systems. In *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02*. New York, New York, USA: ACM Press, p. 415. Available at: http://dl.acm.org/citation.cfm?id=503376.503450 [Accessed March 10, 2012].

Blackstock, M. et al., 2010. MAGIC Broker 2: An open and extensible platform for the Internet of Things. *Internet of Things (IOT)*, pp.1–8. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5678443.

Blackstock, M., Lea, R. & Krasic, C., 2006. Toward Wide Area Interaction with Ubiquitous Computing Environments. In *First European conference on Smart Sensing*. pp. 113–127.

Böhmer, M. et al., 2011. Falling asleep with Angry Birds, Facebook and Kindle. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI '11*. New York, New York, USA:

ACM Press, p. 47. Available at: http://dl.acm.org/citation.cfm?id=2037373.2037383 [Accessed July 8, 2013].

Brignull, H. et al., 2004. The introduction of a shared interactive surface into a communal space. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work - CSCW '04*. New York, New York, USA: ACM Press, p. 49. Available at: http://portal.acm.org/citation.cfm?id=1031607.1031616 [Accessed November 17, 2011].

Brignull, H. & Rogers, Y., 2003. Enticing people to interact with large public displays in public spaces. In *Proceedings of INTERACT'03*. IOS Press, pp. 17–24. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.129.603.

BroadSign, 2014. Digital Signage Software Solutions. Available at: http://broadsign.com/ [Accessed September 26, 2014].

Calderon, R. et al., 2014. RED : a framework for prototyping multi-display applications using web technologies . In *3rd International Symposium on Pervasive Displays, PerDis'14*. pp. 148–153.

Cardoso, J., 2014. *An Interaction Abstraction Toolkit for Public Display Applications*. PhD Thesis.

Cardoso, J. & José, R., 2012. PuReWidgets: a programming toolkit for interactive public display applications. In S. R. José Creissac Campos, Simone D. J. Barbosa, Philippe Palanque, Rick Kazman, Michael Harrison, ed. *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems - EICS '12*. New York, New York, USA, Denmark: ACM Press, p. 51. Available at: http://dl.acm.org/citation.cfm?doid=2305484.2305496 [Accessed June 30, 2012].

Cardoso, J.C.S. & José, R., 2013. Evaluation of a programming toolkit for interactive public display applications. *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia - MUM '13*, pp.1–10. Available at: http://dl.acm.org/citation.cfm?doid=2541831.2541834.

Casey, E.M. & Savastinuk, C.L., 2006. Library 2.0: Service for the next-generation library. *Library Journal*. Available at: http://lj.libraryjournal.com/2010/05/technology/library-2-0/#_ [Accessed August 4, 2014].

Churchill, E.F. et al., 2003. The Plasma Poster Network: Posting Multimedia Content in Public Places. In *Proceedings of INTERACT'03*. IOS Press, pp. 599–606. Available at: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.3606.

Clinch, S., Davies, N., Friday, A., et al., 2014. Demo : An Ecosystem for Open Display Networks. In *3rd International Symposium on Pervasive Displays, PerDis'14, Adjunt Proceedings, Demo Session*. pp. 198–199.

Clinch, S., Davies, N., et al., 2012. Designing application stores for public display networks. In *Proceedings of the 1st International Symposium on Pervasive Displays -*

PerDis '12. New York, New York, USA: ACM Press, pp. 1–6. Available at: http://dl.acm.org/citation.cfm?id=2307798.2307808 [Accessed December 30, 2012].

Clinch, S., Harkes, J., et al., 2012. How Close is Close Enough ? Understanding the Role of Cloudlets in Supporting Display Appropriation by Mobile Users. In *International Conference on Pervasive Computing and Communications, PerCom'12*.

Clinch, S., Davies, N., Kubitza, T., et al., 2014. Ownership and Trust in Cyber-Foraged Displays. *Proceedings of The International Symposium on Pervasive Displays - PerDis '14*, pp.168–173. Available at: http://dl.acm.org/citation.cfm?doid=2611009.2611010.

Clinch, S. et al., 2011. Reflections on the long-term use of an experimental digital signage system. In *13th International Conference on Ubiquitous Computing of UbiComp'11*. ACM. Available at: http://dx.doi.org/10.1145/2030112.2030132.

Clinch, S., 2013. Smartphones and Pervasive Public Displays. *Pervasive Computing,IEEE*, (1), pp.92–95.

Clinch, S. et al., 2013. Yarely: a software player for open pervasive display networks. In *Proceedings of the 2nd ACM International Symposium on Pervasive Displays - PerDis '13*. New York, New York, USA: ACM Press, p. 25. Available at: http://dl.acm.org/citation.cfm?id=2491568.2491575 [Accessed August 7, 2013].

Davies, N. et al., 2010. Challenges in Developing an App Store for Public Displays – A Position Paper. In *Research in the large: Using App Stores, Markets and other wide distribution channels in UbiComp research Workshop*. Copenhagen: UbiComp '10. Available at: http://large.mobilelifecentre.org/dotclear/public/Davies-in-the-large.pdf.

Davies, N. et al., 2012. Open Display Networks: A Communications Medium for the 21st Century. *IEEE Computer*, pp.58–64.

Davies, N., Langheinrich, M., et al., 2014. Personalisation and privacy in future pervasive display networks. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, pp.2357–2366. Available at: http://dl.acm.org/citation.cfm?doid=2556288.2557287.

Davies, N. et al., 2009. Using Bluetooth Device Names to Support Interaction in Smart Environments. In *7th international conference on Mobile systems, applications, and services, MobiSys'09*.

Davies, N., Clinch, S. & Alt, F., 2014. *Pervasive Displays:Understanding the Future of Digital Signage* 1st ed. M. Satyanarayanan, ed., Morgan & Claypool.

Day, N. et al., 2007. Breaking the campus bubble: informed, engaged, connected. In BCS-HCI '07. Swinton, UK, UK: British Computer Society, pp. 133–136. Available at: http://portal.acm.org/citation.cfm?id=1531407.1531442 [Accessed March 30, 2011].

Dix, A., 1991. Status and events: static and dynamic properties of interactive systems. In *Eurographics Seminar: Formal Methods in Computer Graphics*. Available at:

http://alandix.com/academic/papers/euro91/euro91.html [Accessed September 6, 2014].

Dix, A. & Sas, C., 2010. Mobile Personal Devices meet Situated Public Displays: Synergies and Opportunities. *International Journal of Ubiquitous Computing, IJUC*, 1(1), pp.11–28.

Dix, A. & Sas, C., 2008. Public displays and private devices: A design space analysis. In *Workshop on Designing and evaluating mobile phone-based interaction with public displays. CHI2008*. Florence. Available at: http://www.comp.lancs.ac.uk/~dixa/papers/Dix-Sas-PDPD-2008/Dix-Sas-PDPD-2008.pdf.

Elhart, I. et al., 2013. Key Challenges in Application and Content Scheduling for Open Pervasive Display Networks. In *Work in Progress Session PerCom 13*. IEEE.

Elhart, I., Langheinrich, M., Memarovic, N., et al., 2014. Scheduling Interactive and Concurrently Running Applications in Pervasive Display Networks. *Proceedings of The International Symposium on Pervasive Displays - PerDis '14*, pp.104–109. Available at: http://dl.acm.org/citation.cfm?doid=2611009.2611039.

Elhart, I., 2013. WE-BAT : Web Based Application Template for Networked Public Display Applications that Show User Contributed Content. In *Poster at PerDis 13*.

Elhart, I. & Langheinrich, M., 2010. Situated Memory for Situated Public Displays. In *Digital Object Memories in the Internet of Things, DOME 2010, Workshop at Ubicomp'10*.

Elhart, I., Langheinrich, M. & Memarovic, N., 2014. Integrating Interactive Applications with Digital Signage: Towards a Scheduling Framework for Pervasive Displays. In *PD-Apps workshop at PerCom'14*.

Erbad, A. et al., 2008. MAGIC Broker: A Middleware Toolkit for Interactive Public Displays. In *PERCOM '08 Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*. IEEE, pp. 509–514. Available at: http://ieeexplore.ieee.org/ielx5/4517340/4517341/04517447.pdf?tp=&arnumber=4517447&isnumber=4517341.

Fatah, A. et al., 2014. Screens in the Wild : Exploring the Potential of Networked Urban Screens for Communities and Culture. In *3rd International Symposium on Pervasive Displays,*. pp. 166–167.

Ferscha, A. & Vogl, S., 2002. Pervasive web access via public communication walls. In *First International Conference on Pervasive Computing*. Springer Berlin / Heidelberg, pp. 84–97. Available at: http://www.springerlink.com/index/3v9gu49ud35euu0d.pdf [Accessed March 16, 2012].

Finke, M. et al., 2010. Investigating distributed user interfaces across interactive large displays and mobile devices. *Proceedings of the International Conference on*

Advanced Visual Interfaces - AVI '10, p.413. Available at: http://portal.acm.org/citation.cfm?doid=1842993.1843091.

Finke, M., Blackstock, M. & Lea, R., 2007. Deployment Experience Toward Core Abstractions for Context Aware Applications. In *Smart Sensing and Context*. Springer Berlin Heidelberg, pp. 239–254.

Fling, B., 2009. *Mobile Design and Development - Practical concepts and techniques for creating mobile sites and web apps*, O'Reilly Media. Available at: http://shop.oreilly.com/product/9780596155452.do [Accessed February 11, 2014].

Friday, A., Davies, N. & Efstratiou, C., 2012. Reflections on Long-Term Experiments with Public Displays. *Computer*, pp.34–41.

Geel, M., Huguenin, D. & Norrie, M.C., 2013. PresiShare : Opportunistic Sharing and Presentation of Content Using Public Displays and QR Codes. In *Proceedings of the 2013 International Symposium on Pervasive Displays - PerDis '13, 2013, New York, NY, USA, ACM*.

Glaser, B.G. & Strauss, A.L., 1967. *The discovery of grounded theory*, Huber. Available at: http://www.ualberta.ca/~iiqm/backissues/5_1/pdf/mills.pdf.

Greenberg, S., Rounding, M. & Tn, C., 2001. The Notification Collage : Posting Information to Public and Personal Displays. In *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'01*. pp. 515–521.

Greenfield, A., 2006. *Everyware: The Dawning Age of Ubiquitous Computing*, New Riders Publishing. Available at: http://www.amazon.com/Everyware-The-Dawning-Ubiquitous-Computing/dp/0321384016 [Accessed February 7, 2013].

Hardy, J. & Alexander, J., 2012. Toolkit support for interactive projected displays. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia - MUM '12*. New York, New York, USA: ACM Press, p. 1. Available at: http://dl.acm.org/citation.cfm?id=2406367.2406419 [Accessed April 1, 2013].

Heer, J., Card, S.K. & Landay, J.A., 2005. prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*. New York, New York, USA: ACM Press, p. 421. Available at: http://dl.acm.org/citation.cfm?id=1054972.1055031 [Accessed September 7, 2012].

Heikkinen, T. et al., 2011. Declarative XML-based layout state encoding for managing screen real estate of interactive public displays. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, pp. 82–87. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=5766976&contentType=Conference+Publications [Accessed May 3, 2012].

Heikkinen, T., Luojus, P. & Ojala, T., 2014. UbiBroker : Event-based Communication Architecture for Pervasive Display Networks. In *PD-Apps workshop at PerCom'14*.

Hosio, S., Kukka, H., et al., 2010. Enhancing interactive public displays with social networking services. In *Proceedings of the 9h International Conference on Mobile and Ubiquitous Multimedia - MUM '10*. MUM '10. New York, NY, USA: ACM, pp. 23:1–23:9. Available at: http://doi.acm.org/10.1145/1899475.1899498 [Accessed November 4, 2011].

Hosio, S., Jurmu, M., et al., 2010. Supporting distributed private and public user interfaces in urban environments. In *Eleventh Workshop on Mobile Computing Systems & Applications, HotMobile'10*. HotMobile '10. New York, NY, USA: ACM, pp. 25–30. Available at: http://doi.acm.org/10.1145/1734583.1734590 [Accessed October 25, 2011].

Hosio, S., Goncalves, J. & Kostakos, V., 2013. Application Discoverability on Multipurpose Public Displays: Popularity comes at a Price. In *Proceedings of the 2nd International Symposium on Pervasive Displays, PerDis'13*. ACM Press.

Huang, E.M., Koster, A. & Borchers, J., 2008. Overcoming Assumptions and Uncovering Practices: When Does the Public Really Look at Public Displays? In J. Indulska et al., eds. *6th International Conference on Pervasive Computing, PerCom'08*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 228–243. Available at: http://dl.acm.org/citation.cfm?id=1532219 [Accessed October 7, 2011].

Huang, E.M. & Mynatt, E.D., 2003. Semi-public displays for small, co-located groups. *Proceedings of the conference on Human factors in computing systems - CHI '03*, (5), p.49. Available at: http://portal.acm.org/citation.cfm?doid=642611.642622.

Huang, E.M., Russell, D.M. & Sue, A.E., 2004. IM Here : Public Instant Messaging on Large , Shared Displays for Workgroup Interactions. In *SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. pp. 279–286.

IETF, 2014. Cookies. Available at: http://tools.ietf.org/html/rfc6265 [Accessed January 23, 2014].

Izadi, S. et al., 2003. Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST '03*. New York, New York, USA: ACM Press, pp. 159–168. Available at: http://dl.acm.org/citation.cfm?id=964696.964714 [Accessed August 30, 2011].

Johanson, B., Fox, A. & Winograd, T., 2002. The Interactive Workspaces project: experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1(2), pp.67–74. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1012339 [Accessed March 22, 2013].

Jones, M. et al., 2008. "Narrowcast yourself." In *Proceedings of the 7th ACM conference on Designing interactive systems - DIS '08*. New York, New York, USA: ACM Press, pp. 369–378. Available at: http://dl.acm.org/citation.cfm?id=1394445.1394485 [Accessed September 22, 2011].

José, R. et al., 2012. Beyond interaction: tools and practices for situated publication in display networks. In *Proceedings of the 2012 International Symposium on Pervasive Displays - PerDis '12*. New York, New York, USA: ACM Press, pp. 1–6. Available at: http://dl.acm.org/citation.cfm?id=2307798.2307806 [Accessed February 7, 2013].

José, R. et al., 2008. Instant Places: Using Bluetooth for Situated Interaction in Public Displays. *IEEE Pervasive Computing*, 7(4), pp.52–57. Available at: http://www.computer.org/portal/web/csdl/doi/10.1109/MPRV.2008.74 [Accessed May 30, 2011].

José, R. et al., 2013. Pins and Posters : Paradigms for Content Publication on Situated Displays. *IEEE Computer Graphics and Applications*, 33, pp.64 – 72.

Kanis, M. et al., 2012. Studying Screen Interactions Long-term : The Library as a Case. In *1st International Symposium on Pervasive Displays, PerDis'12, In Adjunct Proceedings, Poster Session*. pp. 4–5.

Kaviani, N. et al., 2011. Investigating a Design Space for Multi-Device Environments. In *Distributed User Interfaces*. Springer London, pp. 103–112.

Kaviani, N. et al., 2009. What goes where?: designing interactive large public display applications for mobile device interaction. In *Proceedings of the First international Conference on internet Multimedia Computing and Service*. ACM, pp. 129–138. Available at: http://portal.acm.org/citation.cfm?id=1734637.

Kindberg, T. & Fox, A., 2002. System Software for Ubiquitous Computing. *IEEE Pervasive Computing*, 1(1), pp.70–81.

Klemmer, S.R. et al., 2004. Papier-Mâché: Toolkit Support for Tangible Input. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*. New York, New York, USA: ACM Press, pp. 399–406. Available at: http://dl.acm.org/citation.cfm?id=985692.985743 [Accessed September 7, 2012].

Kostakos, V. et al., 2013. Multipurpose Public Displays: How Shortcut Menus Affect Usage. *IEEE Computer Graphics and Applications*, 33(2), pp.56–63. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6373665 [Accessed April 22, 2013].

Kostakos, V. & Ojala, T., 2013. Public Displays Invade Urban Spaces. *IEEE Pervasive Computing*, 12(1), pp.8–13.

Kubitza, T. et al., 2012. Using mobile devices to personalize pervasive displays. *ACM SIGMOBILE Mobile Computing and Communications Review*, 16(4), pp.26–27. Available at: http://dl.acm.org/citation.cfm?id=2436196.2436211 [Accessed February 12, 2013].

Kukka, H. et al., 2011. This is not classified: everyday information seeking and encountering in smart urban spaces. *Personal and Ubiquitous Computing*, 17(1), pp.15–27. Available at: http://link.springer.com/10.1007/s00779-011-0474-1 [Accessed July 23, 2014].

Langheinrich, M. et al., 2011. Autopoiesic Content: A Conceptual Model for Enabling Situated Self-generative Content for Public Displays. In *1st Workshop on Pervasive Urban Applications (San Francisco, CA, USA, June 12, 2011)*. San Francisco.

Linden, T. et al., 2010. Web-based framework for spatiotemporal screen real estate management of interactive public displays. In *Proceedings of the 19th international conference on World wide web - WWW '10*. New York, New York, USA: ACM Press, p. 1277. Available at: http://portal.acm.org/citation.cfm?id=1772901 [Accessed October 5, 2010].

Lindén, T. et al., 2012. Towards multi-application public interactive displays. In ACM, ed. *Proceedings of the 2012 International Symposium on Pervasive Displays - PerDis '12, 2012, New York, NY, USA, ACM*. New York, NY, USA: ACM. Available at: http://dl.acm.org/citation.cfm?id=2307798.2307807 [Accessed February 1, 2013].

Lindén, T. et al., 2010. Web-based framework for spatiotemporal screen real estate management of interactive public displays. In *Proceedings of the 19th international conference on World wide web - WWW '10*. New York, USA: ACM Press, p. 1277. Available at: http://dl.acm.org/citation.cfm?id=1772690.1772901 [Accessed February 6, 2012].

Locamoda, 2010. Fifteen Seconds or More. Engaging Audiences With Place-Based Social Media. In *White Paper*. Available at: http://locamoda.com/about/white_papers, 21.06.2012.

LocaModa, 2014. LocaModa — Place-Based Social Media. Available at: http://locamoda.com/ [Accessed August 31, 2011].

Marcotte, E., 2011. *Responsive Web Design by Ethan Marcotte*, A Book Apart.

Marquardt, N. et al., 2011. The Proximity Toolkit : Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. In *24th annual ACM symposium on User interface software and technology, UIST'11*. pp. 315–326.

McCarthy, J.F., Costa, T.J. & Liongosari, E.S., 2001. UniCast, OutCast & GroupCast: Three Steps Toward Ubiquitous, Peripheral Displays. In *UbiComp 01 Proceedings of the 3rd international conference on Ubiquitous Computing*. UbiComp '01. London, UK, UK: Springer-Verlag, pp. 332–345. Available at: http://dl.acm.org/citation.cfm?id=647987.741333 [Accessed November 1, 2011].

McDonald, D.W. et al., 2008. Proactive displays. *ACM Transactions on Computer-Human Interaction*, 14(4), pp.1–31. Available at: http://portal.acm.org/citation.cfm?doid=1314683.1314684 [Accessed January 21, 2014].

Memarovic, N. et al., 2012. Designing "interacting places" for a student community using a communicative ecology approach. *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia - MUM '12*, p.1. Available at: http://dl.acm.org/citation.cfm?doid=2406367.2406420.

Memarovic, N. et al., 2013. Moment Machine : Opportunities and Challenges of Posting Situated Snapshots onto Networked Public Displays. In *Human-Computer Interaction – INTERACT 2013*. Springer Berlin Heidelberg, pp. 595–602.

Memarovic, N., Elhart, I. & Langheinrich, M., 2011. FunSquare: First Experiences with Autopoiesic Content. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia - MUM '11*. New York, New York, USA: ACM Press, pp. 175–184. Available at: http://dl.acm.org/citation.cfm?id=2107596.2107619 [Accessed February 7, 2013].

Memarovic, N. & Langheinrich, M., 2010. Beyond Web 2 . 0 : Challenges in Personalizing for Networked Public Display Environments. In *Pervasive Personalisation Workshop at Pervasive 2010*.

Mesbah, A. & Van Deursen, A., 2006. Migrating Multi-page Web Applications to Single-page AJAX Interfaces. *11th European Conference on Software Maintenance and Reengineering CSMR07*, pp.181–190. Available at: http://arxiv.org/abs/cs/0610094.

Michelis, D. & Müller, J., 2011. The Audience Funnel: Observations of Gesture Based Interaction With Multiple Large Displays in a City Center. *International Journal of Human-Computer Interaction*, 27(6), pp.562–579. Available at: http://www.tandfonline.com/doi/abs/10.1080/10447318.2011.555299 [Accessed February 14, 2014].

Morales-Aranda, A.H. & Mayora-Ibarra, O., 2007. A Context Sensitive Public Display for Adaptive Multi-User Information Visualization. *Third International Conference on Autonomic and Autonomous Systems (ICAS'07)*, pp.63–63. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4437940.

Muller, J. et al., 2009. Display Blindness : The Effect of Expectations on Attention towards Digital Signage. In *Proc. of 7th Int. Conf. on Pervasive Computing*. Springer Berlin / Heidelberg, pp. 1–8.

Nacenta, M.A. et al., 2012. The LunchTable : A Multi-User , Multi-Display System for Information Sharing in Casual Group Interactions. In *1st International Symposium on Pervasive Displays, PerDis'12*. pp. 1–6.

North, S. et al., 2013. Tension Space Analysis : Exploring Community Requirements for Networked Urban Screens Networked Urban Screens : a Special Case When Capturing Community Requirements. In *Human-Computer Interaction – INTERACT 2013*. Springer Berlin Heidelberg, pp. 81–98.

Ojala, T. et al., 2012. Multipurpose interactive public displays in the wild: Three years later. *IEEE Computer*, pp.42–49.

Patterson, J.F., Day, M. & Kucan, J., 1996. Notification servers for synchronous groupware. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work - CSCW '96*. New York, New York, USA: ACM Press, pp. 122–

129. Available at: http://dl.acm.org/citation.cfm?id=240080.240232 [Accessed September 6, 2014].

Pawan, V., 2009. *Web Applications Design Patterns*, Morgan Kaufmann.

Peltonen, P. et al., 2008. It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, CHI '08*. CHI '08. New York, USA: ACM, pp. 1285–1294. Available at: http://portal.acm.org/citation.cfm?id=1357255.

Phillips, E. & Pugh, D.S., 2005. How to Get a PhD - 4th edition: A Handbook for Students and their Supervisors : Books. Available at: http://www.amazon.co.uk/How-Get-PhD-Handbook-Supervisors/dp/0335216846 [Accessed September 1, 2014].

Prante, T. et al., 2003. Hello.Wall - Beyond Ambient Displays. In *Video Track and Adjunct Proceedings of the 5th Intern. Conference on Ubiquitous Computing (UBICOMP'03)*. Seattle, Wash., USA.

Ribeiro, A. & Duarte, C., 2012. Move and Speak : Interaction with Multiple Surfaces Using Gestures and Voice. In *First International Symposium on Pervasive Displays, Poster Session in Ajunct Proceedings*.

Richardson, L. & Ruby, S., 2007. *Restful Web Services*, O'Reilly Media; First Edition. Available at: http://www.amazon.com/Restful-Web-Services-Leonard-Richardson/dp/0596529260/ref=sr_1_1?ie=UTF8&qid=1323729053&sr=8-1 [Accessed December 12, 2011].

Satyanarayanan, M., 2001. Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, 8(4), pp.10–17. Available at: file:///C:/Dados/ACTIVO/Teaching/Ubicomp-Teaching/Topicos/1-Introducao/FutureWork/recursos/00943998.pdf.

Satyanarayanan, M., 2005. Swiss Army Knife or Wallet? *IEEE Pervasive Computing*, 4(2), pp.2–3. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1427640 [Accessed September 11, 2013].

Satyanarayanan;, M. et al., 2009. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 8(4), pp.14–23.

Scala, 2014. Digital Signage Software Solutions. Available at: http://scala.com/ [Accessed September 26, 2014].

Schmidt, a. et al., 2012. Interacting with 21st-Century Computers. *IEEE Pervasive Computing*, 11(1), pp.22–31. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6081843.

Schroeter, R., Foth, M. & Satchell, C., 2012. People, Content, Location: Sweet Spotting Urban Screens for Situated Engagement. In *Proceedings of Designing Interactive Systems Conf. (DIS 12)*. pp. 146–155.

Sellen, A. et al., 2009. Reflecting human values in the digital age. *Commun. ACM*, 52(3), pp.58–66. Available at: http://portal.acm.org/ft_gateway.cfm?id=1467265&type=html&coll=GUIDE&dl=GUIDE&CFID=57069731&CFTOKEN=65354721 [Accessed March 19, 2009].

Stock, O. et al., 2007. Adaptive, intelligent presentation of information for the museum visitor in PEACH. *User Modeling and User-Adapted Interaction*, 17, pp.257–304. Available at: http://dl.acm.org/citation.cfm?id=1265324 [Accessed October 6, 2011].

Storz, O., Friday, A., Davies, N., et al., 2006. Public Ubiquitous Computing Systems: Lessons from the e-Campus Display Deployments. *IEEE Pervasive Computing*, 5(3), pp.40–47.

Storz, O., Friday, A. & Davies, N., 2006. Supporting content scheduling on situated public displays. *Computers & Graphics*, 30(5), pp.681–691. Available at: http://www.sciencedirect.com/science/article/pii/S0097849306001191 [Accessed November 11, 2010].

Taivan, C., Rui José, et al., 2013. Design Considerations for Application Selection and Control in Multi-User Public Displays. *Journal of Universal Computer Science*, 19(17), pp.2526–2542. Available at: www.jucs.org/jucs_19_17/design_considerations_for_application.

Taivan, C., Andrade, J.M., et al., 2013. Development Challenges in Web Apps for Public Displays. In *7th International Conference on Ubiquitous Computing & Ambient Intelligence, UCAmI'13*. Springer International Publishing, pp. 135–142.

Taivan, C., José, R., et al., 2013. Situatedness for Global Display Web Apps. In *Demo Session, Adjunct Proceedings of the 2nd International Symposium on Pervasive Displays, PerDis'13*.

Taivan, C. & José, R., 2011. An Application Framework for Open Application Development and Distribution in Pervasive Display Networks. In *On the Move to Meaningful Internet Systems: OTM '11 Workshops*. Springer Berlin Heidelberg, pp. 21–25.

Taivan, C. & José, R., 2014. Application Diversity in Open Display Networks. In *The 3rd International Symposium on Pervasive Displays, PerDis'14*. ACM, pp. 68–73.

Taivan, C., José, R. & Elhart, I., 2012. Selection and Control of Applications in Pervasive Displays. In *6th International Conference on Ubiquitous Computing & Ambient Intelligence, UCAMI'12*. Springer Berlin Heidelberg, pp. 165–172.

Taivan, C., José, R. & Silva, B., 2014a. Understanding the Use of Web Technologies for Applications in Open Display Networks. In *International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, pp. 500 – 505.

Taivan, C., José, R. & Silva, B., 2014b. Web-Based Applications for Open Display Networks: Developers' Perspective. *Special Issue, International Journal of Computer Systems, Science & Engineering (In Press)*.

Terrenghi, L., Quigley, A. & Dix, A., 2009. A taxonomy for and analysis of multi-person-display ecosystems. *Journal of Personal and Ubiquitous Computing*.

Thomson, C., 2014. BBC and the Big Screens. Available at: http://www.bbc.co.uk/blogs/aboutthebbc/posts/bbc-and-the-big-screens [Accessed September 25, 2014].

Toeman, J., 2010. Why Connected TVs Will Be About the Content, Not the Apps. Available at: http://mashable.com/2010/10/20/connected-tv-content-not-apps/ [Accessed September 1, 2014].

Ubisign, 2014. Ubisign Corporation. Available at: http://www.ubisign.com/.

Veenstra, M., 2011. Beyond advertising : Large displays for supporting people ' s needs and activities in public space. *SIGCHI Conference on Human Factors in Computing Systems (CHI '11), Extended Abstracts*, pp.1–4.

W3C, 2014a. Cross-Origin Resource Sharing. Available at: http://www.w3.org/TR/cors/ [Accessed January 23, 2014].

W3C, 2014b. HTML 5.1 Prefetch. Available at: http://www.w3.org/TR/2013/WD-html51-20130528/links.html#link-type-prefetch [Accessed January 23, 2014].

W3C, 2014c. HTML5 Offline Web Applications. *Working Draft*. Available at: http://www.w3.org/html/wg/drafts/html/master/browsers.html#offline [Accessed January 23, 2014].

W3C, 2014d. HTML5 Web Messaging. Available at: http://www.w3.org/TR/webmessaging/ [Accessed July 8, 2014].

W3C, 2014e. Indexed Database API. Available at: http://www.w3.org/TR/IndexedDB/ [Accessed July 8, 2014].

W3C, 2014f. Mobile Web. Available at: http://www.w3.org/standards/webdesign/mobilweb [Accessed July 8, 2014].

W3C, 2014g. Offline Web applications — HTML5. Available at: http://www.w3.org/TR/2011/WD-html5-20110525/offline.html [Accessed July 8, 2014].

W3C, 2014h. Same Origin Policy. Available at: http://www.w3.org/Security/wiki/Same_Origin_Policy [Accessed January 23, 2014].

W3C, 2011. Second W3C Web and TV Workshop - Berlin, Germany, February 8-9 2011. Available at: http://www.w3.org/2010/11/web-and-tv/ [Accessed September 1, 2014].

W3C, 2014i. The Future of Offline Web Applications - W3C Workshop. Available at: http://www.w3.org/2011/web-apps-ws/Report#fixingapplicationcache [Accessed January 23, 2014].

W3C, 2014j. Web and TV. Available at: http://www.w3.org/standards/webofdevices/tv [Accessed July 8, 2014].

W3C, 2014k. Web Storage. Available at: http://dev.w3.org/html5/webstorage/ [Accessed August 8, 2014].

Wang, M., Boring, S. & Greenberg, S., 2012. Proxemic Peddler : A Public Advertising Display that Captures and Preserves the Attention of a Passerby. In *1st International Symposium on Pervasive Displays, PerDis'12*.

Weiser, M., 1991. The Computer for the 21st Century. *Scientific Am.*, pp.pp. 94–104.

Wikipedia, 2014. Single-Page Application. Available at: http://en.wikipedia.org/wiki/Single-page_application [Accessed September 12, 2014].

Wing Kosner, A., 2012. The Appification Of Everything Will Transform The World's 360 Million Web Sites. Available at: http://www.forbes.com/sites/anthonykosner/2012/12/16/forecast-2013-the-appification-of-everything-will-turn-the-web-into-an-app-o-verse/ [Accessed September 11, 2014].

Ye, L. & North, S., 2014. Light-Keypad Interaction through Coated Double Glazing. In *PD-Apps workshop at PerCom'14*.