# Display-centred applications in Ubiquitous Computing

Rui José and Helder Pinto

Information Systems Department, University of Minho
Campus de Azurém, 4800-058 Guimarães
{rui, helder}@dsi.uminho.pt

**Abstract.** Public displays can play an important enabling role in ubiquitous computing environments. This paper describes an on-going work for a multi-purpose, multi-display infrastructure, designed to address the requirements of display-centred applications in ubiquitous computing environments. The system provides an infrastructure in which situated displays can act as portals to the physical space, allowing ubicomp applications to support their association with the physical world by providing them with display services and situation-specific user input and context information.

## Introduction

Large public displays have always been part of the ubiquitous computing vision (the "boards" [1]), but their use has attracted considerably more interest in recent years, given the increasingly pervasive presence of plasma screens, projectors and smaller TFTs in public and semi-public places, and the emergence of new enabling technologies, such as steerable projection [2]. Despite this ever increasing presence of digital visual information in our physical world, the use of displays in ubiquitous computing remains a complex and normally system specific effort, given the lack of any system support for generalising the association between ubiquitous computing applications and display resources. Furthermore, most research in this area has traditionally been focused on specific applications and their evaluation from an end-user perspective, without much attention being given to the use of public displays as a generic infra-structure for ubicomp applications.

We use the notion of situated display as a multi-purpose and strongly situated information artefact that acts as gateway to a virtual and physical environment, which we call situation, allowing applications not only to place information in that environment, but also to sense relevant input and context events. An open-ended set of applications can be associated with that situation, requiring display services, either pro-actively or as a reaction to events in environment, and it is the function of situated displays to manage those requests and arbitrate display resources. We believe that this type of system support can provide a simple, yet powerful, path for the development of a broad range of ubicomp applications, which we call display-centred applications, that albeit combining multiple input and output modalities and being strongly reactive to their usage context, pose as their main requirement the ability to generate situated content for public displays.

This paper reports on an ongoing work on an emerging middleware for display-centred applications on ubiquitous computing environments. The proposed infrastructure builds on web standards and sets a few common abstractions that allow applications to be developed without much knowledge about the environment for which they are being developed and thus be more easily ported between environments. We first describe the type of system support provided by the situated displays and how it can be used to enable a significant range of applications. We then describe some applications and present our evaluation of the approach before the concluding remarks

## System support for display-centred applications

Our infrastructure is composed by situated displays, operating in isolation or strongly coordinated, that provide support to multiple ubicomp applications. The core functionality supported by situated displays is an universal display service that allows multiple applications to express their display needs. Our abstraction for this service is based on web standards, allowing applications to indicate an URI and the respective MIME Content-type. The URI will normally correspond to some remote resource, but URIs corresponding to local applications or content are also possible.

The association between applications and displays can be based on two rather different models; the first, targeted for user-centric applications, enables the spontaneous use of the display by nearby Bluetooth devices, allowing them to request the display of a particular resource; the other model, primarily targeted for environment-centric applications, assumes that someone is managing the situated displays and setting high-level definitions on how they should behave, i.e. programming the space. A long-term schedule is thus used to define the set of activities to be presented together with their scheduling properties.

Scheduling the allocation of display regions to applications is therefore another core functionality of a situated display. Applications from the long-term scheduling are modelled by the system as jobs with specific properties, such as an expected presentation frequency and context rules that define their sensitivity to context conditions. The scheduling algorithm is normally iterating through those jobs and providing them with display time. If there are no dynamic requests, jobs are evaluated to determine the current utility of their presentation, and the job with the highest utility value is presented. The utility function combines multiple context variables and is constantly evaluating them to assess their impact on the current utility of particular jobs. For example, a Bluetooth game may not be very relevant if there are no Bluetooth devices around. This type of constraint or rule is supported through the use of first order logic predicates embedded in the scheduling properties and their value can be determined without involving the applications.

At any moment, the display may receive unscheduled display requests, which will normally be served before the next scheduled presentation. If the request is preemptive it is served immediately, which is useful for interactive applications that need a quick reaction to user stimulus. If the request is non-preemptive it will be served shortly, normally after the current display service is finished. For example, a photo application can be routinely presenting its photos, but when someone sends a new photo via MMS, it tries to make sure that the new photo is displayed shortly after it is received, albeit not necessarily immediately.

Context sensing and interaction is supported through a sensing layer based on the concept of contextor [3]. Context and input information is available to the display itself and to applications that may need it through an Equip tuple space [4] that is shared between the display, other devices in the same environment, applications, and possibly other displays. Multiple interaction modalities are supported, such as RFID tags, SMS, MMS, touch-sensitive screens or mobile phones, but since they can also be used by multiple applications, interaction within the situated display is mediated so that it can be interpreted within the current context and redirected to the appropriate destination. Interaction with an application can always be available in the space, even if the target application is not currently being displayed.

## Integrating applications and situated displays

We will now describe how applications can combine their functionality with the generic services provided by situated displays to achieve an integrated functionality. To begin with, it is worth mentioning that we see display-centred applications and situated displays as independent entities, possibly managed in different domains by different administrative authorities (most of our applications were web applications running at some remote site). This means that, even though we aim to enable strongly situated applications, in most cases display-centred applications are not expected to have been developed for any situation in particular.

A situation specific behaviour is accomplished through the dynamic creation of a relationship between displays and applications that will enable coordination between them. However, we assume that applications may at any time become unavailable or available again and displays may also become temporarily unavailable to applications, meaning that relationships must be managed dynamically. Our approach is to assume that situated displays, based on their long-term schedule, are the primary source for the establishment of relationships. Applications have no a priori knowledge about the displays in which they are going to be used, and their situation-specific state is entirely transient and built on the interactions from displays.

When a display is initiated, it creates a relationship with the applications in its long-term scheduler by sending HTTP messages that support the exchange of configuration data with the application. The application replies to confirm that it has accepted the request and provides the display with additional information about its usage. After this initial relationship is established, applications and displays may participate in various collaborative processes, depending on the needs of the application. One of those processes is utility negotiation, which determines the current utility of a particular application in the current context of a situation, and is largely based on a relative relevance factor provided by the application itself. Only applications can reliably provide this information, since it can strongly depend on the nature of the application, its information domain, and its current data. Displays are expected to use it as an input into their scheduling process. For example, a situated blog should be displayed more often when there are recent posts.

All these interactions are supported by an HTTP-based coordination protocol that supports initialisation, utility negotiation and extraordinary display requests, preemptive and non-preemptive. In our Java-based implementation, access to context-aware information is achieved through the Equip tuple space. Since, we are currently migrating the system to the .NET platform we are also exploring alternative ways to support the tuple space paradigm.

## Applications

The main advantage of an infrastructure as the one described in this paper is to enable a new type of environment-centric application that, while being able to provide functionality that is specific to a particular situation, is in fact shared by multiple environments. Using an infrastructure of situated displays deployed in our University campus we have developed multiple applications with various degrees of integration with the displays. For example, one of those applications was a photo display application that would subscribe to events referring to the presence of Flickr ids on a particular situation (possibly shared by Bluetooth, USB pens, SMS, or some other locally available mechanism). Whenever on of those ids was detected, the application would generate a display of the photos from that user. Even though this was a single web application, any display could include its functionality and have its own slide show of the locally present users. This application demonstrates several features of the infrastructure. Firstly, the display initiates its interaction with the application. As a result, the application builds its own situation-specific state, which includes a subscription of the respective tuple space events, in this case for Flickr ids events. When a new id becomes available the application will try to generate a slide show with the photos of that user and generates a non-preemptive display request. Shortly after, the situated display requests from the application the respective photos which are then shown on the display.

## Related Work

The use of public displays to support individual access to digital services has been explored elsewhere, e.g. Dynamo [5] or BlueBoard [6]. They allow an individual to approach the display and use it at its own convenience for the time needed to complete a particular service, supporting direct manipulation of digital data, possibly in shoulder-to-shoulder collaborations between people sharing a common task. While providing multi-purpose support, these systems are explicitly controlled by users and are not designed to pro-actively address the display needs of applications associated with the respective physical environment. Many other systems support interactive content, possibly associated with multiple services, e.g. the webwall [7], the Aware Community Portal [8] or the Proactive Displays [9], but are not designed to provide the type of generic system support that we have described.

## Evaluation

The evaluation of an ubicomp infrastructure is very challenging given the lack of widely accepted metrics (the lack of benchmarks for display infrastructures has also been discussed and identified as an open issue at a recent workshop on the topic [10]). Furthermore, the fact that it must be evaluated indirectly through applications built on top of it raises the risk of getting distracted by the demands of application development and to lose sight of the real purpose of the effort, which is purely to evaluate an infrastructure [11]. We thus focused evaluation on limited prototypes in which we sought an adequate mix between issues related with the infrastructure and those related with application usability and usefulness. We have developed a diverse range of applications, from already existing web applications to more complex interactive ones that exercised the various features of the infrastructure. We have found that one of the most interesting features of the system is the flexibility on how the system's control logic can be distributed between applications and displays, depending on the requirements of a particular application or scenario. For example, application-specific scheduling decisions could easily be passed on to applications, which would do their own part of the scheduling process instead of blindly relying on the scheduling process supported by the situated displays.

From the perspective of the infrastructure itself, and considering some generic middleware design principles, we can stress the key role of using web technologies. Firstly, it enables a scenario-driven design, i.e. the system is optimised for the most common scenarios. Despite all the sophistication that ubicomp applications may address, we believe that the most common ubicomp scenarios will be simple ones and based on proven technologies. By targeting our infrastructure to web applications, we are leveraging on the huge potential of the web technologies. This also enables one other important principle, which is a low barrier to entry, as any web application can be used. Even an existing web application, by simply adding a relevance field to its HTTP response header, can immediately participate in the utility negotiation process. More sophistication is progressive and used as needed. Interactive applications will need the ability to send HTTP messages corresponding to display requests, and context-sensitive applications will need access to the Equip tuple space. The design principle of having a layered architecture capable of providing adequate abstractions that allow applications to access the powerful features of the infrastructure without handling all the complexity is not yet addressed, as developing interactive or context-sensitive applications still implies handling all the complexities of the tuple space and coordination protocols. The need for a development toolkit has been identified as future work. Finally, from the perspective of self-configuration and graceful degradation, we have been careful in guaranteeing the independence between applications and displays, with displays assuming that applications may fail on them at any time (which means the respective job becomes not ready) and applications using a soft-state approach in which their situation-specific state is always seen as transitory.

## 7 Conclusions

This paper described an on-going work on the definition of an approach to enable situated displays to provide adequate system support for display-centred applications. The situation abstraction and the dynamic creation of relationships between displays and applications enables applications to be reactive to the context of a particular environment without being designed specifically for that environment. We expect that this type of support, together with the use of a web-based approach will enable many new applications that were too costly and complex to develop for specific display systems, and may constitute an essential step in moving away from designs specialized for each particular environment to reusable building blocks that provide common infrastructure support for multi-purpose pervasive display systems.

## References


1. Weiser, M., *Some Computer Science Issues in Ubiquitous Computing.* Communications of the ACM, 1993. **36**(7): p. 75-84.
2. Pingali, G., et al. *Steerable Interfaces for Pervasive Computing Spaces*. in *IEEE International Conference on Pervasive Computing and Communications (Percom 03)*. 2003. Dallas-Fort Worth, Texas.
3. Coutaz, J. and G. Rey. *Foundations for a theory of contextors*. in *4th International Conference on Computer-Aided Design of User Interfaces*. 2002. Valenciennes, France.
4. Greenhalgh, C., *EQUIP: a Software Platform for Distributed Interactive Systems*. 2002, Department of Computer Science, University of Nottingham, UK.
5. Izadi, S., et al. *Dynamo: A public interactive surface supporting the cooperative sharing and exchange of media*. in *UIST 2003*. 2003. Vancouver.
6. Russell, D.M. and A. Sue. *Using Large Public Interactive Displays for Collaboration*. in *Workshop on Collaboration with Interactive Walls and Tables*. 2002. Goteborg, Sweden.
7. Ferscha, A., G. Kathan, and S. Vogl. *WebWall - An Architecture for Public Display WWW Services*. in *The eleventh International World Wide Web Conference*. 2002. Honolulu, Hawaii, USA.
8. Sawhney, N., S. Wheeler, and C. Schmandt, *Aware Community Portals: Shared Information Applicances for Transitional Spaces.* Personal and Ubiquitous Computing, 2001. **5**(1): p. 66-70.
9. McCarthy, J.F., et al. *Proactive Displays & The Experience UbiComp Project*. in *First International workshop on Ubiquitous Systems for Supporting Social Interaction and Face-to-Face Communication in Public Places*. 2003. Seattle, Washington, USA.
10. José, R., et al. *Workshop on pervasive display infrastructures, interfaces and applications*. 2006. Dublin, Ireland.
11. Edwards, K., et al. *Stuck in the Middle: The Challenges of User-Centered Design and Evaluation for Middleware*. in *Conference on Human Factors in Computing Systems (CHI 2003).* 2003. Fort Lauderdale, FL.