Proactive Scheduling for Situated Displays

F. Reinaldo Ribeiro[1], Rui José[2]

[1] Departamento de Informática da Escola Superior de Tecnologia de Castelo Branco
Av. Do Empresário
6000-767 Castelo Branco, Portugal
fribeiro@est.ipcb.pt
[2] Departamento de Sistemas de Informação da Universidade do Minho
Campus de Arurém
4800-058 Guimarães, Portugal
rui@dsi.uminho.pt

Abstract. Public situated displays can represent an important mechanism to enhance users' experiences and reflect the information and the interactions associated with their environment and the people on it. However, this type of embodiment requires the display system to be able to perform unassisted scheduling based on the current state of the environment in which it is integrated. In this paper, we explore this problem domain and the various ways in which it can be approached. This analysis is organised around three main parts of this problem: which dynamic information to consider, how to introduce domain knowledge for enabling rational choices and how to model the scheduling process itself. We also describe our own approach to these issues, a context-sensitive MAUT scheduler that selects the next item to be presented by determining the one with the highest utility considering the current context of the display.

Keywords: Proactive Scheduling, Situated Displays, Context-awareness, Public Displays, Utility Theory.

1 Introduction

Even though large public displays have always been part of the ubiquitous computing vision (the "boards" [1]), their use has attracted considerably more interest in recent years, given the increasingly pervasive presence of plasma screens, projectors and smaller TFTs. These digital displays, public and semi-public, large or small, can play many different roles in ubiquitous computing scenarios, and their potential applications go far beyond today's traditional uses of public displays, i.e., advertising and dissemination of information to passive users. This work builds upon a vision of situated displays as public, shared, networked, and pro-active devices that are embodied into their environment and reflect the information and the interactions associated with that environment and the people on it. Such situated displays should enable new and more engaging user experiences by sensing their environment, giving users a more active role in the system behaviour, and providing people with brief encounters with information that is relevant for their specific situation. This could improve local awareness, enrich our perception of reality, promote information sharing within communities, and ultimately make situated displays an integral part of the social settings in which they are integrated.

The Challenges of Unassisted Scheduling

A scheduling function that determines what is displayed and when is a central feature for any multi-purpose display system. In traditional digital signage system, this is a well-known problem that is normally addressed with some type of authoring tool where the manager of the system can set a detailed timeline defining exactly what is going to be presented and when, thus leading to pre-arranged and static presentation cycles. A slightly more sophisticated approach may allow specific scheduling parameters, such as priorities, types or an expected time share, to be associated with presentation items and then generate on the fly an optimised schedule, but it is still an a priori definition.

The concept of situated displays as embodied artefacts that reflect their physical and social context introduces an additional requirement in the scheduling process that is not addressed by any of the previous approaches: the ability to make dynamic scheduling decisions based on the recent and current state of the system and its environment. Situated displays should thus be able to perform some sort of unassisted scheduling in which the system is continuously re-arranging the schedule in order to maximise the overall utility of the system under the continuously changing context.

In a scenario like that, the display manager, instead of setting a detailed timeline, could set what we will call a behaviour definition that essentially identifies a potentially very large set of presentation items and the criteria for choosing among those items under varying circumstances of its operating environment. When the display started executing that behaviour, it would take into account the current context and its previous presentation history to select which item to present next. Even though the manager would not be able to know the exact sequence of presentations that would be generated, he or she would know that the system would meet the criteria specified in the behaviour. From this perspective, the behaviour definition could be seen as a sort of genetic code that determines the nature of the system, but not exactly what it will do. In the end, the effective behaviour exhibited by the system will result from the combination between this genetic code and the stimulus received from the environment in which the system is operating.

In the remainder of this paper we will start by reviewing related work. In Section 3 will make a general analysis of context-aware smart scheduling and the various ways in which it can be approached. We will then describe in Section 4 our own model for smart situated displays and end with the final remarks and future work.

2 Related Work

Research on situated public displays has received considerable attention recently, with many projects addressing the issues of how to enable information access and share, and enhance collaboration within organizations or communal spaces [2-4]. However, there are other projects that have their focus on a proactive behaviour of the display with the goal of presenting the most relevant content for users.

Villar [5] describe a proactive environment in which information, which reflects the communal interests of current inhabitants, is proactively displayed on large scale public displays. A device, called pendle, is used to store user preferences and make them available to the environment. It also allows users to override the environment's proactive behaviour by means of simple gestures.

Proactive displays [6] are large computer displays augmented with sensors that can detect people nearby and show content relating to those people. The goal is to enable users to easily share the richness of digital lives with their neighbours in physical space, creating new opportunities for greater awareness and interactions by bringing the benefits of virtual communities into physical communities. Participants can create a profile that is associated with an RFID tag that they can carry.

The BlueScreeen project [7] is an intelligent public display, which selects and displays adverts in response to users detected in the audience. It utilizes Bluetooth-enable devices as proxies for identifying users and utilizes history information of past users' exposure to certain sets of adverts. Advertisements are preferentially shown to those users that have not seen them yet.

Muller [8-9] describes a mechanism to adapt advertisements on digital signage to the interests of the audience. Here, each advertisement has a set of keywords and the history of all advertisements a user was interested in is kept. It uses a naïve Bayes classifier to estimate the probability that a user is interested in a certain advertisement given the keywords and users history. It uses voucher collection information as a form of feedback for the system.

In this work we are addressing the general problem of smart scheduling, and considering that we may not have any a priori information about the value of specific presentations, as is the case in advertisement auctions. We also assume that we have no a priori knowledge about users and their interests, and that users are not carrying any dedicated device for the purpose of interest manifestation.

3 Context-sensitive Smart Scheduling

In general terms, a scheduler that operates without assistance and is able to select the best adaptation strategy for a particular set of contextual data, is what may be called a context-sensitive smart scheduler, and raises several research challenges related with pro-activity, adaptation, context-awareness, and machine learning that form the core of this work. In an attempt to clarify the respective design space, and thus contribute towards a general framework for the creation and comparison of multiple dynamic scheduling algorithms, we will now look more carefully at how we can approach each of the main parts of this problem: Modelling the environment to which the system must adapt, introducing "smartness" and perform dynamic scheduling.

Environment Model

The first issue regarding proactive scheduling for situated displays is how to model the display environment, i.e. which dynamic properties of that environment are going to affect the scheduling process. The overall input for the scheduling process is going to include both static and dynamic elements. The former are assumed to be part of the predefined scheduling information that characterises static scheduling processes such as priorities or presentation times. The latter, which are the focus of this work, correspond to the characteristics of the environment where the display is situated and cannot be determined before the system is actually deployed and running, i.e. they are dynamic or not known at the time of defining the system's behaviour. When considering the various types of dynamic stimulus that we could use for scheduling, we have identified the following key dimension types.

Table 1. Environment dimensions for context in situated displays.

Dimension	Overview
Environment Context	Context refers to the overall state of the environment where the situated display is integrated. This may include the time, nearby people or current weather. Changes in context can then be transmitted over the system components and be reflected in utility functions and context constraints.
Identification	Identification corresponds to the ability to identify the presence of unique entities, e.g. the presence of a specific Bluetooth device, the recognition of a specific tag, the reception of an SMS from a particular number. Entities may or not have associated profiles.
Job Relevance	Job relevance is a dynamic self-measure of relevance provided by the presentation item and its calculation must reflect the nature of the job and its information. For example, the relevance of an RSS feed may increase when it is updated and then slowly decrease as it approaches its expiry time.
User Hints	User hints are user interactions that can be used to infer interest in some type of application, topic or job. A user hint is not traceable to any specific job and therefore cannot be interpreted as a measure of success for any scheduling decisions. Examples may include a content request or the submission of new data for publication. We use the term user hint to broaden the set of user related events that may be considered, and which go far beyond explicit user requests.

Smartness

The second part of the problem is to take the possible inputs from the environment and use that data to make the scheduling process smart, i.e. with a behaviour that reflects rational choices in terms of what is appropriate at each particular context. For that, we need to incorporate into the system knowledge that creates associations between dynamic input data and specific behaviours. We will now look more closely at what we believe to be the key approaches for introducing this “smartness” in situated displays.

The first possibility is to try to embed into the behaviour definition (or directly into the scheduling algorithm itself) empirical knowledge that represents how we expect the system to be affected by changes in its context. For example, we may want to say that a particular piece of content should be shown only when there are Bluetooth devices nearby, or that another item should not be shown twice in the presence of any particular Bluetooth device, or that yet another item should only be shown when there is no one in the immediate physical proximity of the display. As this type of rules refer to contextual variables they can in fact enable dynamic scheduling in which the behaviour of the system is continuously adapting to the varying context of its operating environment. However, such rules are difficult to define and represent. Firstly, they are difficult to define because smartness is a rather vague concept that for most cases does not map directly into any specific type of reaction. Even though it seems intuitive to think that certain contextual variables can affect what is the most appropriate content to be presented in a particular context, it is very hard to specify the subtle influence of those factors in a way that is formal enough to be machine processed. The definition, based only on empirical knowledge, of high-level rules that represent the most efficient reactions that the system can have in multiple contexts is far from being an obvious task. For this reason, this approach is only efficient when the goal is to create a very direct association between a particular state and a very specific behaviour, normally something in the form of a trigger.

The second possibility is to try to create new domain knowledge by training the system according to some general notion of smart behaviour. This would typically include a training stage, in which a set of training cases of the expected behaviour would be generated, a process of supervised learning, in which new rules would be inferred, and then a dissemination stage, in which those rules would become domain knowledge ready to be embedded into scheduling processes. It is however unclear how much of that knowledge is generic enough to be applied to multiple situated displays, as the respective context and media items may be completely different.

Finally, situated displays may incorporate a generic feedback mechanism whereby its users are invited to express their opinion on the scheduling decisions, or more specifically about the relevance of what is being shown. This would then allow a process of learning in which the scheduler would be able to learn with experience and adapt to the specific characteristics of its environment. This learning process may lead to two different types of result. The first would focus on the popularity of individual items, whereas the second would focus on the effect of contextual data in user acceptance. The relevance of these two lines of learning would depend on the nature of the items to display and the frequency of the changes in the set. In either case, if we consider that each situation represents a possibly very specific combination of a physical and social setting, the entire learning process would be specific for each situated display.

Scheduling Model

The final part of the problem is to create a smart scheduling process based on the contextual input. Generally speaking, scheduling is a vast scientific field with a significant body of knowledge and applications in many different areas, from fleet management to operating systems. Therefore, the first step when approaching scheduling issues for situated displays should be to characterize this particular scheduling problem in order to identify related problems and solutions, and their applicability or adaptation to the specific scenarios of activity scheduling in situated displays.

In abstract terms, “scheduling is concerned with the allocation of scarce resources to activities with the objective of optimizing one or more performance measures” [10]. In a situated display, the scarce resources correspond to display opportunities and activities correspond to the presentation items that need display time. Performance measures are less obvious and more complex to define, but, for the moment we are considering that at a certain level of abstraction, the overall objective of this work is to optimize the utility of the system under the varying circumstances of operation.

Using the terminology and notation commonly used for scheduling problems, our goal is to process n jobs on m machines, where jobs correspond to atomic units of presentation that need on air time and machines correspond to the displays that provide those presentation opportunities. Displays correspond to machines because each particular display is only presenting one activity at the time, i.e. even though a certain number of jobs are associated with a display, they will never overlap in time and only one can be selected at any particular time. Only when one job is finished, can the next one begin its execution.

A scheduler implements a specific scheduling algorithm that allocates jobs to machines based on pre-defined criteria, and a schedule is for each job the allocation of one or more time intervals to one or more machines. Jobs may need additional resources for their execution. Resources are allocated to a job when it starts and freed when the job completes or is pre-empted. For example, the default sound speaker can be modelled as a resource that is appropriated by activities that need sound when being presented.

We thus have a set of m Displays D = {D1, D2,…, Dm}, and, for each, there is a set of n jobs J = {J1, J2, …, Jn} that are competing for on air time and may need a set of shared resources R = {R1, R2, …, Rs}. We assume that jobs are allocated a priori to a particular machine. i.e. there is a single presentation region, and therefore there are no spatial assignment decisions that need to be made by the scheduler. The problem is thus reduced to the selection of the next job to allocate to a given machine, i.e. the next job y to be activated for that display.

A particular characteristic of this problem is that there is no notion of job completion, i.e. items never finish. However, they may be added, removed or become temporarily unavailable due to time or context constrains. Together with variations in context, this means the scheduler must schedule jobs without knowledge of the future, thus precluding the scheduler from guaranteeing optimal schedules [10].

In order to support interactive applications, the scheduling process may also need to support asynchronous presentation requests intended to cause the immediate presentation of a particular job, and possibly the preemptive removal of the currently displayed job. A particular scheduling algorithm may choose to address this requirement by including timed requests in a general model of job selection in which jobs can have associated deadlines or some other type of priority attribute. However, this type of request imposes a completely different set of assumptions regarding the dynamics of the system and in fact represents an exception to the unassisted nature of the displays, because someone is actually telling the system what it should do, particularly if those requests override (not just affect) the current schedule. Therefore, while it may be important to provide support for these requests, there seems to be no reason why that should not be handled completely outside this particular scheduling process.

In a scheduling problem, the objective function defines the criteria that guide the optimization of the scheduling process. The definition of performance criteria is thus a crucial part of any scheduling problem, since it is that criteria that is going to make different scheduling approaches comparable. However, defining such criteria for situated displays must take into account the very specific nature of these jobs and is far from obvious. Traditional measures such as maximizing *throughput*, i.e. the number of completed jobs per unit of time, or minimizing *turnaroundtime*, i.e. the time to complete a single job, are not appropriate, given that the concept of job completion is rather unique. Instead, the objective function should be chosen carefully to represent the particular concept of smartness (as described in the previous section) enabled by the system. More specifically, if the approach is to embed a priori knowledge domain, empirical or inferred, that represents associations between environment states and specific scheduling decisions, then the objective function should be made in such way as to reflect those associations. The way that knowledge is represented should enable the scheduler to select from a set of possible alternatives the one that better satisfies the respective rules. If on the contrary, the objective is to make the system receive feedback and learn with its experience, then performance measures must be linked with some type of observable consequence of our scheduling decisions. In this case, the system must be able to collect data that directly or indirectly may be used as a measure for the performance of previous scheduling decisions and inform future decisions.

Direct feedback associates a particular user action with a particular scheduling decision in a particular context and has the advantage of being able to capture the strongly situated nature of these systems. There are essentially two forms of direct feedback. The first is to collect information about the audience, such as how many people are looking at the display at any particular moment, and trying to map that information into a measure of success for scheduling decisions. The second form of direct feedback, called actionables, can be used when presentation items are enticing users to perform some kind of action, such as sending an SMS that responds to a specific solicitation, collecting a specific voucher that was displayed, downloading a file through Bluetooth, or explicitly classifying the relevance of what is being shown.

If we are able to trace each of these actions to specific jobs, we can use that information as a form of feedback that can be used to measure the performance of the system and improve future performance.

Indirect feedback is not traceable to a particular scheduling decision and provides a measure for the performance of the system as a whole. An obvious example is the realization of user evaluation studies with interviews, questionnaires and users' observation to gather qualitative information on users' perception regarding the adaptation strategies employed by the system. While this form of feedback is not useful for the operation of a particular display, these studies may prove to be valuable sources to inform the design of scheduling rules at large.

4 MAUT Scheduling

We will know describe our own approach to the problem of context-sensitive smart scheduling for situated displays. The approach is based on Multi-Attribute Utility Theory (MAUT) and tries to determine a utility value for each of the active jobs, with the highest one being selected for presentation. The general model for our approach is depicted in Figure 1.

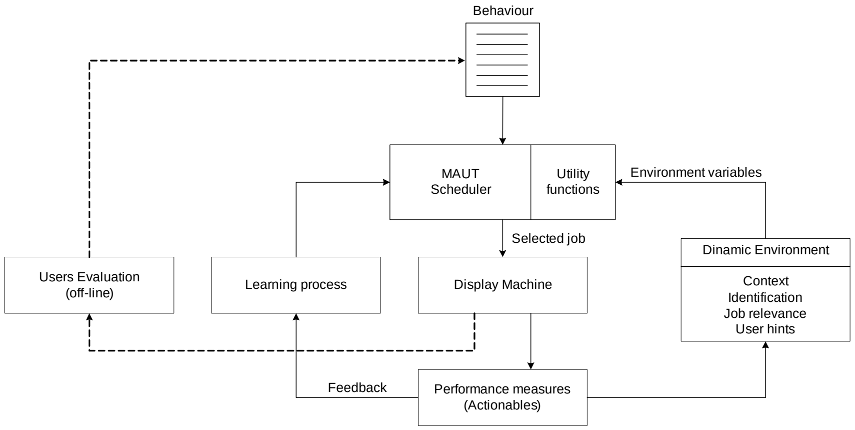

Fig 1. MAUT Scheduling model

The Display Machine at the centre receives instructions from the scheduler, which is responsible for selecting which item should the display machine present next. The scheduler receives as its initial input a behaviour file in XML that is a high-level definition of the expected behaviour of the system in the form of basic layout definitions and the identification of the presentation jobs that are going to be associated with the display. For each job there are static scheduling attributes, of which the main ones are a minimum, a maximum, and a base share that indicates what is the expected share of presentation time that this item is expected to have. Dynamic attributes will change the effective share time between the indicated minimum and maximum values.

When in operation, the scheduler also receives contextual data about its operating environment. In particular, our prototype display system collects information about the presence of Bluetooth devices and the number of faces facing the display. When an item is indicated in the behaviour file as being context sensitive, the respective contextual input is associated with a particular utility function that normalises the input from that source and creates a meaningful mapping of context values into utility values. For example, in the case of Bluetooth a particular application may need the presence of at least two Bluetooth devices. This would translate into a function that provided zero utility when the number of devices was below 2 and then jumped quickly close to maximum when there were two or more devices present.

Scheduling Algorithm

We will now describe the MAUT scheduling algorithm we have used. When there are no pending asynchronous requests, which are answered first and treated separately, this algorithm must select which of the jobs in a ready state can generate the highest utility. According to MAUT, the overall expected utility *EU(job)* of a job *job* is defined as a weighted addition of its evaluation with respect to its relevant value dimensions.

$$EU(job) = \sum_{i=1}^{n} w_i \times d_i(job) \qquad (1)$$

Where d_i is the evaluation of the job on the *i-th* dimension, and w_i the weight determining the impact of the *i-th* value dimension on the overall evaluation, with $\sum_{i=1}^{n} w_i = 1$. The vector of weights w_i defines the way in which the scheduler is affected by the surrounding environment and various combinations can be used depending on how one intends to balance between quick reaction to events and a steady scheduling program.

In the case of this particular function there are two basic dimensions, which are common to all schedulable items, and map expected share values with utility values. The first is calculated from the offset between the expected share and the effective share, and the second is calculated from the self-relevance measure provided by the item. In addition to these two dimensions that are essentially targeted at balancing long-time presentation schedules, each item may be associated with a particular utility function for mapping a particular environment variable and the way how that variable affects the utility of that item.

A further dimension associated with immediate presentation history is used to avoid the successive selection of the same job by strongly reducing its utility in the immediate period after its presentation.

Deployment

In our first prototype, we created a simulation to analyse the impact of context changes and rule definition on the expected job utility. This was useful in tuning the weights to be used for each of the dimensions and balancing the combined effect of context and scheduling history. Afterwards, we have created a public display using the above algorithm. In its current implementation this scheduler does not incorporate feedback, but it is being used to collect information about the number of faces looking at the display, which we expect to use as an off-line performance measure. In order to make this evaluation process more explicit, we added a second display next to the first one, that allows people to provide positive or negative feedback about the relevance of what is being shown.

5 Conclusions and Future Work

In this paper we explore the various development paths for enabling smart scheduling in situated displays. In particular, we have analysed the problem from the perspective of which dynamic information to consider, how to introduce domain knowledge for enabling smartness and how to model the scheduling process itself. We also described our own approach to these issues, which is based on the calculation of an expected utility for each presentation item. As part of our future work, we intend to develop the algorithm further by incorporating results from user evaluations, context-sensitive feedback and audience metering.

Acknowledgements

This work has been developed as part of the Situaction project and financially supported by Portuguese Foundation for Science and Technology under grant POSC/EIA/58832/2004 and is partially supported by FEDER through POS_Conhecimento. The first author was also supported by a Portuguese Foundation for Science and Technology scholarship (SFRH/BD/31292/2006). This paper has been extensively revised to reflect the comments of the anonymous reviewers to whom we thank for their comments.

References

1. Weiser, M., “Some Computer Science Issues in Ubiquitous Computing,” Communications of the ACM, vol. 36, issue 7, July, 1993.
2. D. M. Russell and A. Sue, "Using Large Public Interactive Displays for Collaboration," presented at Workshop on Collaboration with Interactive Walls and Tables, 2002.
3. S. Greenberg and M. Roudind, "The Nottification Collage: Posting Information to Public and Personal Displays," presented at Human Factors in Computer Systems, 2001.

4. D. Vogel and R. Balakrishnan, "Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users," presented at Symposium on User Interface Software and Technology archive at 17th annual ACM symposium on User interface software and technology, Santa Fe, NM, USA, 2004.
5. Villar, N., Schmidt, A., Kortuem, G., Gellersen, H.: Interacting with Proactive Community Displays. Computers and Graphics, Vol. 27, issue 6 (2003).
6. McCarthy, J. F., Nguyen, D. H., Rashid, A. M., Soroczak S.: Proactive Displays & The Experience UbiComp Project. 5th International Conference on Ubiquitous Computing, Seattle, Washington (2003).
7. Payne, T., David, E., Jennings, N. R., Sharifi, M.: Auction Mechanisms for Efficient Advertisement Selection on Public Displays. European Conference on Artificial Intelligence, Riva del Garda, Italy (2006).
8. Jorg Muller, J., Kruger, A.: How much to bid in digital signage advertising auctions. Advances in Pervasive Computing, Adjunct Proceedings of the 5th International Conference on Pervasive Computing, Toronto (2007).
9. Jorg Muller, J., Kruger, A., Kuflik, T.: Maximizing the Utility of Situated Public Displays. Adjunct Proceedings of User Modeling, Corfu, Greece (2007).
10. Leung, J.Y.-T., Handbook of Scheduling. Algorithms, Models and Performance Analysis. 2004: CRC Press LLC.