Integrating Interactive Applications with Digital Signage: Towards a Scheduling Framework for Pervasive Displays

Ivan Elhart, Marc Langheinrich, Nemanja Memarovic
University of Lugano (USI)
Via Giuseppe Buffi 13
6900 Lugano, Switzerland
{ivan.elhart, marc.langheinrich, nemanja.memarovic}@usi.ch

Abstract—Today's digital signage systems typically show content that has been scheduled well in advance by their respective owners. However, we envision that displays in our environment will soon be able to dynamically adapt to their surroundings and allow viewers as well as display owners to appropriate them by actively selecting and contributing content through interactive public display applications. Such multi-user and multi-application display environments require new forms of application and content scheduling that go beyond a predefined sequence of content. In this paper we present a scheduling framework for public displays, adapted from general scheduling theory, as a common notation that can be used to describe different application and content presentation requirements posed by both display owners and display viewers and define the overall behavior of public displays.

I. INTRODUCTION

Public displays are becoming a ubiquitous resource in the urban environment: we can find them at airports as conventional information displays, in bars displaying events and menus, and on the streets showing advertisements. Most of these displays are singular or standalone installations designed around a digital signage model that provides content control and scheduling mechanisms only to display owners - individuals or organizations that physically own public displays. In this common model, display content is first scheduled in a well defined sequence of content items and then distributed to end-point terminals, i.e., screens for the final content presentation. Such end-point terminals are usually isolated from their surrounding environment and often ignored by display viewers - their intended audience [9].

Research in public displays, however, has proposed many much more interactive systems. These systems are often completely open to display viewers allowing them to not only influence content scheduling, but also to instantaneously upload and show their own content. However, these are mostly single application and standalone installations that give no control over content scheduling to display owners.

In order to increase the overall utility of currently closed public displays and make them more appreciated in their environment, we have to open them up to a wide range of content sources in the form of display applications, and provide appropriate control and scheduling mechanisms to fulfil usually conflicting requirements of different stakeholders, in particular display owners and display viewers [2], [8]. While some researchers approach this by separating concerns of different stakeholders, we aim in providing new forms of shared control over interactive public displays to both display owners and display viewers. Therefore, our main research question is: how can we open today's closed and isolated public displays to third party applications and user-contributed content, but still keep their owners in the overall control? While there are different social and economic issues surrounding this vision [2], here we are looking into technical challenges of controlling and scheduling applications and their content on public displays [3], [4].

Application scheduling on public displays can be defined as a decision making process that deals with allocating display applications to one or more displays or one or more display regions as well as presenting a particular content item within the application. This decision making process directly controls when and what content is shown on the screens. Application scheduling is thus one of the central challenges for opening public displays to interactive applications, and can be seen as the point where different stakeholder requirements come together. Also, application scheduling is of high relevance for multipurpose displays, where different applications and their purposes form a unique display ecosystem.

In this paper we present a scheduling framework for public displays adapted from general scheduling theory where a scheduling problem is described through machine environment, tasks and constraints, and optimization function [11]. We propose to describe the scheduling of applications on public displays in the same way as a set of three elements that describe: a) display environment extended with the screen real-estate division, b) application environment and associated application presentation constraints, and c) heuristic rules that define the overall display behaviour instead of optimization function. The contribution of this paper is twofold:

1) The paper provides a notation and framework for describing scheduling problems in the area of interactive public display systems.

2) Using the framework we provide examples of how common scheduling scenarios of future public display
networks could be described.

II. Related Work

A number of projects have been instrumental in advocating the vision of opening public display to applications, such as e-Campus [1], [13], [14], Instant Places [6], [12], and UBI-hotspots [5], [7], [10]. However, these projects mostly focus on simplified schedulers and a smaller set of scheduling constraints.

Instant Places is a display platform that features a number of web-based display applications such as a presence application that shows the profiles of users around the displays, a poster application for publishing user contributed posters, and a pin application that shows content associated with the profiles of display users, among others [6], [12]. These applications are scheduled on the displays in a traditional digital signage way through well defined sequences. The platform provides a very powerful interface for display owners to arrange applications in time sequences and distribute the sequences to the displays. However, this approach has two possible limitations. First, if display owners want to change how applications are played, they have to update sequences manually and distribute the updated sequences to the displays. Second, only display owners can influence an application’s start and presentation times. While display viewers can personalize individual applications, they have to wait for the applications to appear on the display according to the sequence and their interactions can be interrupted after predefined amount of time.

UBI-hotspots is a network of indoor and outdoor public displays installed in the city center of Oulu, Finland [5], [10]. The displays combine standard digital signage content (images and videos) with interactive applications through a state machine with two defined states: passive and active [7]. In the passive state the displays show digital signage content arranged in a sequence and shown in a full screen mode called UBI-Channel. In the active mode the screen is divided into two areas, one showing UBI-channel and one offering a number of interactive applications through a touch screen called UBI-portal. This approach completely separates concerns of display owners, giving them full control over UBI-channel, and display viewers, giving them full control over UBI-portal. While there are certain advantages of this separation, there are also some possible limitations. First, display owners do not have full control over their displays such as to show high priority and emergency related content in full screen over certain periods of time. Second, display viewers cannot gain control over the entire screen area even for short interactions. Finally, the size of the channel and the portal is fixed and cannot change during runtime.

e-Campus is a university wide installation of public displays also featuring a number of interactive applications such as interactive map, Flicker photo view, YouTube video view, art installations, and application that shows university-wide and location-specific content, among others [1], [13], [14]. The e-Campus platform provides flexible ways for developing independent constraint-based schedulers. However, constraint-based schedulers are problem specific, requiring developers to develop and implement new schedulers for each scheduling problem and specific display setting. Also, the platform supports the existence of multiple schedulers at the same time, but it says nothing about the actual behaviour of such schedulers on the final application and content presentation times.

III. Scheduling Requirements

In scheduling applications and their content on public displays, a distinction can be made between traditional (static) content and interactive applications.

A. Traditional Content

In digital signage, content is typically scheduled well before the final presentation time. Traditional content is usually associated with scheduling requirements in the form of scheduling constraints such as:

- "show item1 from 14:00 to 15:00"
- "show item2 n number of times"
- "do not show item3 after 20:00"
- "do not show item4 more than m number of times"
- or "show item2 immediately after item4"

The scheduling constraints are processed by a constraints solver and the output of the solver is a sequence of content items, usually called a "playlist". The items in a sequence can be additionally orchestrated, i.e., fine-tuned to produce the desired display behaviour. Such "playlists" are usually produced at a central location and distributed to displays, endpoint terminals for the final presentation.

Having the finite set of content items arranged in a well defined sequence has at least two important properties. First, the start times of all content items are known at any moment of the presentation time. Second, the duration of all content items is constant or fixed during the entire presentation time. In contrast, interactive applications usually do not have these properties and traditional scheduling approaches cannot be applied to them.

B. Interactive Applications

Interactive applications are more complex to schedule than traditional content. They do not have the properties of static content which creates two additional problems:

1) Arbitrary application start time: How can we schedule applications which start time is not known in advance? Scheduling content in traditional systems is based on the assumption that the start times of all content items are known in advance, e.g., show a specific content item at 6 pm, or show the item every 2 minutes. However, in open display networks applications can be triggered at arbitrary times, for example when display users step in front of the screens or touch them. Scheduling algorithms in future public displays should support arbitrary start times.

2) Arbitrary application duration: How can we schedule an application on a display when durations of previously shown applications is not known? Traditional signage systems assume that content items are
of fixed duration. However, interactive applications can be of arbitrary duration depending on the user interaction. Removing an application before the users are done using it, or without a notification why it was removed, can negatively influence the user experience and the overall usability of the system.

Such a dynamic environment requires new approaches to scheduling interactive applications on public displays that go beyond a predefined sequence of content items [3].

IV. Scheduling Framework for Public Displays

Digital signage usually defines a scheduling problem with a set of scheduling constraints associated with content items. In order to integrate interactive applications with traditional content and to describe dynamic behaviour of future public displays, a more general scheduling framework is needed.

A. General Scheduling Theory

In general scheduling theory, scheduling can be defined as "a decision making process that deals with allocation of resources to given tasks over a certain period of time with the main goal of finding a feasible or optimal solution that is subject to a given set of constraints and one or more objectives" [11]. A mathematical representation of a scheduling problem is usually given as a sequence of three elements:

\[ \alpha | \beta | \gamma \]

where \( \alpha \) describes the machine environment and available resources, \( \beta \) provides information about tasks and associated constraints, and \( \gamma \) describes the objective to be optimized [11].

Machine environment describes the number of available machines and their mutual relations and possible processing constraints. Some of the common machine environments include parallel machines with different processing speed, flow shops, job, shops, or open shops. In public display domain this field could be used to describe the display environment, i.e., number of displays and their mutual relation. This field could be extended to include information about the screen real-estate division.

Tasks and constraints field holds information about individual tasks and constraints associated with them. A distinction could be made between hard and soft constraints as well as static and dynamic constraints. In a scheduling problem, hard constraints have to be satisfied at all cost. In contrast, soft constraints are usually described as preferences of lower priority. While static constraints are constant during the processing time, dynamic constraints can change over time and depend of other constraints. In public display domain, tasks can be public display applications with associated presentation constraints.

The overall goal of scheduling is to optimize one or more objective functions given the machine environment, tasks, and associated constraints. The common optimization techniques are concerned with finding an optimal or feasible solution that minimizes the objective function such as total completion time. In public display domain, there might exist objectives regarding the profit from advertisers or the user satisfaction, but they might be very simple or very complex to construct.

Instead, the third field could hold one or more heuristic rules, e.g., "service in random order with pre-emptions", that can define the display behavior and lead to desired objective.

B. Notation and Framework

Following the notation from general scheduling theory, a scheduling problem in public displays is defined as a set of three elements, \( \alpha | \beta | \gamma \), describing the display environment, applications and associated constraints, and overall display behavior.

The \( \alpha \) describes the display environment and division of the screen real-estate including the number of displays and their mutual relation. The relation among public displays can be:

- Independent displays \( (D) \) - Display behaviour is independent from the other displays.
- Displays in series \( (S) \) - Display behaviour depends on scheduling requirements of other displays allowing display applications to "travel" from one display to another in a time defined sequence. A simple scenario is information that follows users as they walk through a city and pass public displays.
- Displays in parallel \( (P) \) - Displays connected in parallel show the same application in specific time intervals. An emergency information would be a good example, when the displays show the same application at the same time.

In addition to describing the display environment, this field could contain information about the screen real-estate. Most traditional digital signage systems show content only in fullscreen mode. However, there are also many systems that provide space multiplexing allowing different screen regions to show different information. The screen real-estate division is especially important for multi-user environments where information tailored for different viewers can be shown in different screen regions. Typical screen regions include:

- Fullscreen \( (f) \) - an application takes the entire screen area.
- Sidebar \( (s) \) - an application has access to a small area to show a preview of its content or provide short but important information to the users until it gets access to the fullscreen mode.
- Ticker tape \( (t) \) - an application can show content in a small area at the bottom of the screen.

The screen real-estate division is not limited to these three elements. It is not hard to imagine that display owners may want to define different screen regions and create unique layouts for their displays. The number of displays, their relationship, as well as the screen division do not have to be static and can change over time. In addition, the first field of the scheduling problem can include possible application transitions from one display region to another.

The second element of the scheduling problem \( \beta \) can be used to describe the application environment and constraints associated with it. This includes a list of applications that can...
be scheduled on the display, constraints about the application presentation and duration times as well as associations of applications with display regions. Some of the most common scheduling constraints that can be applied to public displays are:

- Processing time \((p)\) - represents the limits of the application presentation time. It can be used to set a minimum presentation time required by the application, or the maximum presentation time after which other applications can take the screen resources.
- Release time \((r)\) - represents the start time of an application. It can be expressed in absolute times, relative times to other applications, or it can occur in regular time intervals.
- Due time \((d)\) - represents committed completion time of an application. It is possible that an application completes its presentation before of after its due time.
- Weight \((w)\) - is a weighting factor denoting a relative importance of an application comparing to other applications. This is also called a priority factor that gives a higher priority of presentation to applications with higher weights. However, this can lead to the application starvation problem, where low-priority application cannot access the screen due to the constant presence of high-priority applications.
- Pre-emptions \((prmp)\) - allows the scheduler to interrupt and stop the presentation of currently running application at any point in time and instead allow a different application (usually with higher priority) to access the screen. Pre-emptions are central point in allowing scheduling interactive applications with arbitrary start times. It is important that display applications start their presentation when users request them with a minimal delay. However, when an application is substituted with the other application, the system should provide visible clues for the cause of change.
- Precedence \((prec)\) - denotes the requirement that one or more applications have to be completed before another application can start its presentation.

The number of applications as well as scheduling constraints can change over time allowing for describing a more dynamic behaviour of displays. For example, the weights associated with displays applications can change according to the time of day or presence of display users.

The third element of the scheduling problem \(\gamma\) describes heuristic rules that define the overall display behaviour. In general, the rules can be static or dynamic [11]. While static rules are constant, dynamic rules can change during the processing time. The rules can also be elementary and composite. Composite rules are combinations of elementary rules in the form of a ranking expression. A simple rule is the "Service in Random Order" that takes applications from the list and schedules them in a random fashion.

V. EXAMPLES

The simplest scheduling problem in the domain of public displays can be described with one display and one application always present on the display:

\[
D | a_1 | -
\]

Another example, the local walk-to-school programme scenario described by Davis et al. [2] can be described using the same notation:

\[
D_{1,i=1...n} | \hat{a}_1, a_{j,j=2...m} | RR
\]

In this scenario, Jack, a six year old, participates in an interactive game on a network of public displays designed to increase fitness among school children and address childhood obesity. As Jack passes a number of displays \((n)\), the game \((\hat{a}_1)\) updates him on his own progress and allows him to collect "golden leaves" on his mobile phone. The hat symbol \((\hat{)}\) indicates an interactive application that has arbitrary start time and arbitrary duration and can pre-empt any other application when the user approaches the display. Other \(m-1\) applications are scheduled in Round Robin (RR) fashion until pre-empted by the game \((\hat{a}_1)\).

More complex scheduling problems can also be defined using the framework. For example,

\[
P_1, P_2, D_{i,i=3...n} | \hat{a}_1, \hat{a}_2, a_{j,j=1...m} : a_1p, \hat{a}_2p(t), a_3p, a_3r, a_4r | SIRO, PRMP(t)
\]

describes the problem of scheduling \(m\) applications on \(n\) displays according to the combination of two heuristic rules. While displays \(P_1\) and \(P_2\) are connected in parallel showing synchronized content, other \(n-2\) displays are independent. All applications can be scheduled on all displays. Applications \(\hat{a}_1\) and \(\hat{a}_2\) are interactive with limited processing times. Display owners can control processing times of applications by setting a constant \((a_1p, a_3p)\) or a time dependent function \((\hat{a}_2p(t))\). Applications \(a_3\) and \(a_4\) have constant start (release) time, e.g., start \(a_3\) at 14:00 for 5 minutes and \(a_4\) every 30 minutes. Applications are scheduled according to the Service In Random Order (SIRO) rule with pre-emptive behaviour that display owners can control over time \(PRMP(t)\).

A. Examples from Current Display Systems

The same notation can be used to describe already existing display systems and installations. For example, application scheduling in Instant Places can be described as:

\[
D_{i,i=1...n} | a_{j,j=1...m} | RR
\]

In this system a number of \(m\) applications are scheduled on \(n\) independent displays in Round Robin fashion. While applications and their content can be highly personalized and tailored to specific display situation using personal accounts and mobile phones, the applications are presented in a predefined sequence. The sequence of applications cannot be influenced by display viewers during the presentation time.

UBI-hotspots schedule digital signage content and interactive applications in two different screen areas, channel and portal. Therefore, the scheduling problem can be described with two distinct models running in parallel:
In this public display network, each of \( n \) displays shows the channel and the portal areas. The channel area shows \( j \) content items \((c_{ij})\) in Round Robin fashion. The content items shown in the channel are usually professionally crafted images and short videos arranged in a sequence. The portal area presents \( p \) interactive applications \((a_k)\) that viewers can interact with. While viewers cannot influence the content presentation in the channel, they have full control over applications and their content in the portal and can start and stop applications at arbitrary times. In the passive mode of operation UBI-hotspots show only the channel in full screen and in the active mode the hotspots shows both the channel and the portal in parallel.

The e-Campus system separates the roles of content providers and display owners by organizing content in logical containers called channels. Similarly to UBI-hotspots, channels contain individual content items such as images, videos, web pages, or live video streams. Channels can be scheduled on one or many displays within the network, depending on the display owner subscriptions, together with interactive applications and emergency alerts. Therefore, the scheduling of channels, applications, and emergency alerts can be described as:

\[
D_{i(j)} = \{c_{ij} \mid c_{ij} \in 1...m \} \cup \{a_k \mid a_k \in 1...p \} \\
\text{RR}(ci), \text{PR}(w)
\]

Each of \( n \) independent displays can show a number of channels \((ch_j)\), applications \((a_k)\), and emergency alerts \((e_l)\). Channels, applications, and emergency alerts can have associated weights, or priorities, \( w_j, w_k, \) and \( w_l \) that can influence the order of their presentation. Channels are usually weighted equally, but the weights can also be used to dedicate more airtime to content items from a certain channel. Each channel consists of a number of content items \((c_{ij})\). Content items from all channels are shown in Round Robin fashion as a combined set of content items \((c_i)\) from all channels \((ch)\) - \( \text{RR}(ci) \). By gaining a higher priority certain applications and emergency alerts can be shown on the screens using the priority rule \( \text{PR}(w) \).

VI. CONCLUSION

Traditional signage content and interactive applications pose different scheduling requirements for future public display systems. While it is highly important to allow display viewers to appropriate the screens by actively selecting and contributing content through interactive display applications, it is essential to keep display owners in the overall control of their resources. In this paper we have presented a framework and notation, adapted from general scheduling theory, for describing scheduling problems for future public display systems. We believe the framework will help the community conceptualize and construct scheduling policies and appropriate scheduling mechanisms for integrating interactive public display applications with digital signage and supporting display personalization by both display viewers and display owners.

REFERENCES


