

# Scheduling Content in Pervasive Display Systems

SARAH CLINCH, The University of Manchester, United Kingdom

MATEUSZ MIKUSZ, Lancaster University, United Kingdom

IVAN ELHART, Università della Svizzera italiana (USI), Switzerland

NIGEL DAVIES, Lancaster University, United Kingdom

MARC LANGHEINRICH, Università della Svizzera italiana (USI), Switzerland

Digital displays are a ubiquitous feature of public spaces; London recently deployed a whole network of new displays in its Underground stations, and the screens on One Time Square (New York) allow for presentation of over 16,000 square feet of digital media. However, despite decades of research into pervasive displays, the problem of scheduling content is under-served and there is little forward momentum in addressing the challenges brought with large-scale and open display networks. This paper presents the first comprehensive architectural model for scheduling in current and anticipated pervasive display systems. In contrast to prior work, our three-stage model separates out the process of *high level goal setting* from content *filtering* and *selection*. Our architecture is motivated by an extensive review of the literature and a detailed consideration of requirements. The architecture is realised with an implementation designed to serve the world's largest and longest-running research testbed of pervasive displays. A mixed-methods evaluation confirms the viability of the architecture from three angles: demonstrating capability to meet the articulated requirements, performance that comfortably fits within the demands of typical display deployments, and evidence of its ability to serve as the day-to-day scheduling platform for the previously described research testbed. Based on our evaluation and a reflection on paper as a whole, we identify ten implications that will shape future research and development in pervasive display scheduling.

CCS Concepts: • **Information systems** → *Multimedia content creation*; • **Human-centered computing** → *Ubiquitous and mobile computing systems and tools*.

Additional Key Words and Phrases: Digital signage, digital out-of-home advertising, multimedia scheduling

## ACM Reference Format:

Sarah Clinch, Mateusz Mikusz, Ivan Elhart, Nigel Davies, and Marc Langheinrich. 2019. Scheduling Content in Pervasive Display Systems. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4, Article 129 (December 2019), 37 pages. <https://doi.org/10.1145/3369826>

## 1 INTRODUCTION

Technology advances and falling hardware costs have led to a proliferation of digital displays in public and semi-public spaces. Such displays have traditionally been used for advertising and simple information presentation, with the result that many viewers perceive public displays as being of very low value and have become highly skilled at ignoring the content shown. This phenomenon is typically referred to as “display blindness” [72].

---

Authors' addresses: Sarah Clinch, Department of Computer Science, The University of Manchester, Manchester, M13 9PL, United Kingdom; Mateusz Mikusz, School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, United Kingdom; Ivan Elhart, Università della Svizzera italiana (USI), Faculty of Informatics, Lugano, 23185, Switzerland; Nigel Davies, School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, United Kingdom; Marc Langheinrich, Faculty of Informatics, Università della Svizzera italiana (USI), Lugano, 23185, Switzerland.

---

This paper is published under a Creative Commons Attribution International 4.0 License (CC BY 4.0). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2019 Copyright held by the authors.

2474-9567/2019/12-ART129

<https://doi.org/10.1145/3369826>

However, in recent years researchers have begun to explore entirely new forms of pervasive display systems that have the potential to transform the way that users engage with displays and media in public spaces. Such display systems are typically:

**Context Aware** New displays frequently include sensors that enable them to adapt to their physical environment (e.g. their location) and to the presence or absence of viewers. For example, video analytics systems enable displays to determine the number and demographics of viewers and tailor the content shown accordingly [46]. Such displays are often termed context-aware [9] or situated [53], while the ability for an individual viewer to influence a displays behavior is referred to as display personalization [24], display- or cyber-foraging [20] and appropriation [17].

**Interactive** Support for user interaction with public displays has been the subject of intense research and solutions have been developed that enable interaction through touch [48], gaze [28], gesture [76] or via user smartphones [16]. Interactivity may range from simple selection operations through to complex implicit interactions based on eye-tracking.

**Networked** Existing display systems tend to view displays as isolated entities or, exceptionally, part of choreographed configurations in locations such as train stations. However, the creation of display networks provides new opportunities for the way in which content is disseminated and presented – for example enabling “follow me” content that tracks individual users as they navigate the display network.

**Open** Traditional display networks operate as closed systems with no easy mechanisms for third parties to develop and deploy content for presentation. However, as the mobile phone market as demonstrated, innovation occurs most naturally in open systems and hence researchers have begun to articulate a vision of global, open display networks [26] that enable content to be freely created and shared.

In parallel with research into new display capabilities, researchers have also begun to explore the creation of new types of content for display networks [26]. These include highly interactive content [71], applications designed to promote ambient awareness or behavior change [22, 49] or social interaction [7] and content from autopoiesic display systems [60] that are able to autonomously generate content items for display. This wealth of content may be made available through large-scale brokerage networks or research platforms such as display application stores [21] that encourage sharing of content.

One key challenge for these increasingly complex display systems, is that of *scheduling*, determining which content item should be presented at which point in time, and on which display (or part of a display). This is a particularly challenging question as algorithms for scheduling onto pervasive displays must take into account numerous factors including, but not limited to: (i) the expectations of a large number of human/organisational stakeholders (including display owners, content producers, and multiple viewers and bystanders); (ii) the environmental context of displays and their viewers; (iii) the selection of content from a potentially large and diverse pool of available items; and (iv) determining appropriate schedules for both individual displays and across entire display networks.

As a research community, we have an extensive body of systems and deployments, none of which would have been realised without some fundamental decisions about scheduling. However, despite several decades of experience, researchers lack an accepted vocabulary for talking about the unique challenges and solutions for pervasive display scheduling. There are no common models, patterns or practices and relative to other challenges for pervasive displays (e.g. interaction), the problem of scheduling is under-served and immature.

The **primary research contribution** of this paper lies within its articulation of *a three-stage architectural model for scheduling content in pervasive display systems*. Although primarily targeted to emerging networks of open pervasive displays, this model is firmly grounded in experiences with traditional digital signage and can be applied to a wide range of systems. In contrast to prior work, our three-stage model separates out the process of *high level goal setting* from those of content *filtering* and *selection*. In so doing, we *provide a new*

*vocabulary* for talking about pervasive display scheduling. Such a vocabulary is important as it allows researchers to explore and innovate in three distinct but newly-articulated subproblems, potentially catalysing an era of new research in the domain.

In developing the scheduling architecture at the heart of this paper, we make **two further contributions**. Firstly, we **provide a comprehensive description of requirements for scheduling in pervasive display networks** based on a literature survey, use cases and real-world experiences. We identify eleven requirements that reflect the diverse set of challenges and expectations emerging from *traditional signage, interactivity, context* and *emerging notions of openness*. Secondly, since our architecture is conceptual and thus independent of any specific display system implementation, we provide a **description of the design, implementation and evaluation of key elements of this architecture in the context of a large-scale display testbed** – in so doing, we demonstrate the feasibility of realising what might otherwise be an entirely abstract model. Conducted entirely within the existing software of the e-Campus signage network, our implementation illustrates the applicability of the architecture to existing deployments without the need to re-engineer established signage software.

To the best of our knowledge, this work represents the first comprehensive architectural model for scheduling in current and anticipated pervasive display systems. In the subsequent sections of this paper, we summarise the scheduling problem and approaches taken in pervasive display scheduling to date (Section 2). We then identify eleven requirements (Section 3) that build on prior research (including both deployed systems and articulated use cases) and our own experiences acquired over twenty years working in the field. The core of this paper is the architectural model presented in Section 4 and realised with an implementation in Section 5. We evaluate the architecture (Section 6) first by taking a theoretical approach that compares against the presented requirements, then by exploring the performance of our implementation in a controlled setting, and finally by describing the performance of this same implementation when in daily use on ~100 displays. This evaluation confirms that a three-stage architectural model meets the needs of current and future display systems, and can be used as the basis for a concrete implementation that meets real-world expectations; these and other implications are discussed in Section 7.

## 2 RELATED WORK

### 2.1 Scheduling

Scheduling has been an important research topic for many years and has been explored both from theoretical perspectives [77] and in the context of applied fields such as operations research [42] and operating systems [51].

In general scheduling theory, *scheduling* is defined as a decision-making process that deals with the allocation of *resources* to *tasks*. Its goal is to find a *feasible* or *optimal* solution that adheres to a given set of *constraints* and one or more scheduling *objectives* [77].

In applied scheduling problems, *resources* are often equated with, for example, machines in a workshop or processing units in a computing environment. In such cases, *tasks* represent jobs or processes that have to be completed on the given resources (e.g. number of available machines). Requirements (*constraints*) governing the allocation of jobs to resources may be expressed in the form of hard limitations or preferences and can be associated with both resources and tasks. Hard constraints usually prevent the execution of tasks on machines that do not possess certain capabilities; these constraints have to be satisfied at any cost. Preferences, on the other hand, may indicate a choice to execute a task on a particular machine (e.g. faster or newer) but if needed the task can be completed on any available machine. Some of the most common scheduling constraints include processing times, release times, due times, weights, sequences, and preemption.

A common objective when scheduling is to find a solution that minimizes or maximizes an objective function representing the mathematical relationship between a variable (e.g. cost or reward) and the task's execution. A solution that achieves the theoretical maximum (or minimum) is called an optimal solution. In contrast to those

that seek an optimal solution, some scheduling techniques are concerned with simply finding a feasible solution that satisfies all constraints but does not necessarily minimize or maximize an objective function.

Development of mathematical models to find optimal solutions for practical problems has been the focus of operational research [42]. Initial operational research techniques can be dated back to the start of the industrial revolution, where technical advances created opportunities for optimization of increasing resources and tasks, and later for optimization of activities within emerging organizations. During World War II, the focus shifted towards the optimization of scarce resources and strategically moving them around. Many optimization models developed during this time have since found practical applications in the commercial and public sectors for the management of personnel and activities. Depending on the properties of the mathematical model and objective function, optimization problems can be deterministic - when properties of the model are known and constant, or stochastic - when only probabilities assigned to resources and tasks are available [77].

## 2.2 Scheduling in Digital Signage

Pervasive display research dates back forty years [22]. The very earliest systems focused on the multimedia capabilities of displays, exploring their use as media links that connected together physically separate spaces [6, 36]. Deployed to show just a single media item, these platforms had no need for scheduling techniques to determine media selection. However, more recent systems encompass multiple screens, content items and/or viewers, introducing a need for an active media selection (scheduling) process.

Over the course of decades of pervasive display systems and deployments, researchers have explored a number of distinct approaches to scheduling content items to displays. One recent paper classified these approaches as either “procedural” and “declarative” based on whether they required a display owner to “specify the exact schedule ahead of time” (procedural) or to simply “specify some notion of importance” and leave exact timings for the algorithm itself to determine (declarative) [8, p1]. This is a useful distinction, but abstracts over the variety of approaches used (in particular, the diversity of approaches used in declarative display scheduling). We therefore report prior work based on its use of four distinct scheduling approaches: *round robin and declarative playlists*, *interactivity*, *context-based methods*, and *auction-based or utility-driven algorithms*.

**2.2.1 Round-Robin and Declarative Playlists.** For small display installations that do not justify complex signage software, lightweight round-robin approaches to scheduling are commonplace. In such systems, scheduling consists of simply cycling through all available content items sequentially—once all items have been shown, the scheduler begins again with the first item). Widely used in commercial settings, such approaches have also been commonplace in research systems. For example, the Campus Coffee Display [13, 55], an indoor deployment, consisted of a single display sited within a café in Newcastle, UK. Scheduling was deliberately non-interactive and “slow-paced”, providing café visitors with a slideshow-style presentation of information on local cultural events. The display remained in place for over two years, acting as “a low-key technology probe” [13] that allowed researchers to explore display engagement behaviors. Round-robin approaches were also used in the early Hermes Photo display [10–12] to show user-contributed photographs.

In contrast to small-scale display deployments, large-scale commercial display deployments often feature in excess of 1,000 screens. Notable deployments of this kind (each featuring thousands of screens) include the INFOSCREEN networks in Germany [84], Austria [44] and Malaysia [45], the Adspace Digital Mall Network [1], the JCDecaux out-of-home advertising screens [50], and the GST displays embedded in petrol station pumps across the United States [27]. Such deployments often extend simple round-robin schedulers with the ability to create playlists for individual or groups of displays. These playlist-based systems typically allow users to create timelines that specify the content to be shown on single or groups of displays. In some cases support is also provided within the schedule for changes to the content shown in response to external events such as user interaction. For example, platforms such as Scala [80] and Sony Ziris [82] provide tools for content design,

management and playback, and both support a playlist-based model of scheduling. In addition to playlists, Scala provides remote player monitoring and provision for emergency alerts, whilst Sony’s application suite includes tools for the creation of playlists for the entire display or sub-portions thereof (e.g. to maintain a separate schedule for a sidebar or ticker tape).

**2.2.2 Interactivity.** Pervasive display research systems often support interactivity, using (typically touch-) input from users as a mechanism for controlling the content schedule. For example, the Plasma Poster Network [15] consisted of three touch-enabled plasma displays located in the FX Palo Alto Laboratory that were oriented in a portrait format to reflect the typical layout of traditional paper posters. Content was generated from the laboratory’s intranet pages and from items submitted by users via email and the web. Like the round-robin approaches previously described, a “PosterShow” interface cycled through content items, but touchscreen buttons were also provided for navigating between content items, printing, forwarding and responding to items; over ten months of use, the number of touch-interactions each month increased from 1,309 to 1,2733.

Combining user-driven and round-robin scheduling has been a common pattern for pervasive display systems (e.g. [14, 31, 33, 59]), and was an important feature of the UBI-Hotspots deployment in Oulu, Finland [41, 74], one of the largest city centre research display deployments. Twelve hotspot sites operating between 2009 and 2017 included large touch-sensitive LCD panels that enabled users to select from an array of media and applications; when the screen was not in active use, a default cycle of content was shown in order to attract attention.

As an alternative to explicit interaction, many display platforms have looked to adapt their schedules based on the mere presence of users (e.g. [31, 33, 59]). The Flexible Ubiquitous Monitor Project (FLUMP) [33] provided personalized content on two CRT screens to users identified by the presence of an Active Badge. Users configured pages for the screens ahead of time, assembling them from a variety of content widgets including unread email messages for the user, upcoming appointments, a cartoon, and the opening status of a local coffee bar. When no registered users were detected to be in the vicinity of display, the FLUMP screen would carousel (round-robin) through a sequence of default webpages. As an additional input to the content schedule, limited interaction with the FLUMP stations was possible: users could page through their content using two control buttons on their Active Badge. This type of scheduling behavior has also been seen in the Outcast and Groupcast displays [59] displays and the Plasma Poster Network [14]. Recent research has explored the potential to display user-specific social media feeds on shared displays (the WE-BAT system [31]) and the use of mobile phones to support display personalization in a privacy preserving fashion [25]. In commercial systems, a similar move has seen the use of video analytics [46] (cameras attached to public displays) to automatically determine the age and gender of proximate viewers and adjust the content shown accordingly. These various approaches share a common overall goal of responding to users, a specific case of contextual scheduling.

**2.2.3 Context and Constraints.** More complex contextual models have looked to account not only for the presence of users in content scheduling for digital signage, but also to account for place, time, changing place use, and previous engagement with a content item, e.g. in Ribeiro and José’s description of “highly situated displays that reflect the expectations... and practices associated with the people in a particular place” [79, p1]. Ribeiro and José’s content scheduler integrates multiple aspects of display context (referred to as the “place”) including long-term characteristics and goals (defined by the place owner) and more transient features influenced by place visitors. Place descriptors are represented as tags within an evolving tag cloud based on ongoing interactions (e.g. Bluetooth presence [53], SMS messages). As tags are updated, related content can be selected from a large set of Internet content sources. To select an individual item from related content sources, the place-based scheduler first identifies the most pertinent tag for scheduling (based on direct user requests, overall popularity of the tag over time, and excluding recently selected tags). Once a tag has been selected, the scheduler then identifies the most timely content feed of those appropriate for formatting on the target public display. The notion of ‘situationalization’ was also advocated by Lasinger and Bauer [58], who suggested that selecting content based

on its relevance to “an individual or group of individuals in the present situation, based on information about the current situation” [58, p2] would eliminate the need for personal (privacy-infringing) data use.

The importance of context in both analogue and digital display content scheduling was also identified in Müller et al.’s ReflectiveSigns [66]. They proposed that adaptive digital signs could be implemented as an architecture composed of two ‘loops’: a *scheduling loop* that identified current context and selected an appropriate item, and a *learning loop* that measured audience reaction and provided future input for the scheduling loop to learn from its previous behavior. Although the theoretical platform accounted for a wide range of contextual data (time, weather, location, audience profiles), an initial prototype using time and location demonstrated that time, location and content were all significant in determining the amount of viewer attention, with location having the largest impact on view time.

A number of researchers have noted the usability challenge presented by content management and scheduling systems that incorporate context into their schedules [35, 66]. For example, although commercial signage systems have evolved to incorporate conditional statements and constraints, such systems typically feature sophisticated user interfaces that may be overly cumbersome for non-expert users. Similarly, research platforms such as e-Campus have provided constraint-based scheduling for expert users (in this case, taking the form of an API for Python programming [83]), but the subsequent user interfaces have abstracted away much of this fine-grained control [35].

**2.2.4 Auction-Based and Utility-Driven Approaches.** In addition to the above models, the strong advertising focus of commercial platforms have led to the development of scheduling models inspired by the auction approaches used in other advertising mediums (e.g. web-based banner advert selection). Payne et al. [75] first proposed the use of auctions to select advertisements on public displays – their goal was to maximise each audience exposure to unique advertisements (i.e. to avoid repeated sightings of the same content item by any given individual). Their ‘BluScreen’ system used Bluetooth scanning to count users in front of the display and build up user history records. A repetitive second-price sealed bid auction was used to determine the winner from a set of advertising agents, each placing a bid on behalf of an advertisement to be shown. Each agent used the same valuation function and bidding strategy to determine their bid, this strategy used historical data representing previously successful advertising cycles and the devices present (i.e. which users had seen this advertisement before) and the currently detected Bluetooth devices (i.e. which users would be likely to see content if the bid was successful). For a small-scale simulated deployment (one display, ten content items, no more than 100 unique devices), BluScreen’s auction-based scheduling was shown to require an average of 36% fewer scheduling cycles to expose users to all content items than a round-robin approach.

The use of auctions for the scheduling of commercial content to digital signage was subsequently explored by Müller and Krüger [67, 68] whose ‘actionable advertisements’ were selected using a similar combination of user profiling, Bluetooth sensing and auction-based selection. By building user profiles, their display systems could calculate the probability that showing an advertisement would result in a desired action, informing auction bidding. Using data from previous actions also allowed their system to infer user interests and integrate recommendation algorithms into their bidding agents [68]. The use of user profiling for public display advertisements was explored further by Alt et al. [3], who created a system in which user preferences and behavior (detected through use the user’s mobile device) were combined to create a user profile. To select appropriate advertisements at a display the system either selected the optimal advert for each viewer in turn, or adverts that were a good fit for the majority of present viewers. Similarly MobiDiC [69], a prototype coupon-system, aimed to provide input for auction-based signage systems, by allowing displays to track shopping behavior that followed the display of advertising content. However, a twelve-month deployment on twenty screens yielded just thirty-seven coupon redemptions meaning that the system never reached a point where there was sufficient data to inform scheduling.

Recently, Bauer et al. [5] have suggested that auction-based systems are unlikely to be the primary method of pricing and content selection in large digital signage networks. They instead propose a hybrid solution as the most likely outcome, in which the majority of timeslots are ‘fixed’ (e.g. allocated based on a pre-defined playlist), whilst the remaining options are available for sale by auction.

Bushman and Labrinidis [8] proposed an alternative means of considering value in pervasive display scheduling. Their utility-based scheduler builds on the observation that the value of scheduling any given content item to a display changes over time. Bushman and Labrinidis divide media into anytime- and deadline-content, and use different utility functions for each. Their evaluation uses a simulated environment to compare the effect of different variables on ability to meet deadlines and overall utility.

### 3 REQUIREMENTS FOR SCHEDULING IN PERVASIVE DISPLAY SYSTEMS

The perceived value of a pervasive display is entirely dependent on both the availability of a pool of high-quality content items, and the selection of appropriate content items to display for any given set of viewers. Significant attention has been given over to techniques for developing high quality and engaging media (e.g. [40, 43, 54, 57]), compelling narratives (e.g. [2, 88, 92]) and interactive content (e.g. [37, 39, 56, 65, 90]). However, systems to support scheduling are less frequently at the foreground in pervasive display research (for exceptions, see [8, 31, 62, 70, 78, 83]), and requirements for selecting items for presentation are implicit rather than overtly expressed (for one notable recent exception, see [8]). In order to understand the generalised set of scheduling requirements for future display networks we draw on three distinct sources:

- (1) a study of prior literature including: (a) those reported in Section 2.2, (b) a recent survey of content scheduling [86], and (c) prior requirements elicitation initiatives [8, 83],
- (2) use cases for future display networks [22, 26], and,
- (3) experiences of operating the world’s largest research testbed for display networks [35].

We group the resulting requirements into four main categories based on the need to support traditional *digital signage* functionality, *interactivity and personalization*, *contextual awareness*, and *openness* (see Table 1 for an overview).

#### 3.1 Supporting Traditional Signage

Traditional display systems typically make scheduling decisions based on three factors: *content metadata*, *display features*, and *temporal and ordering constraints*.

**3.1.1 R1: Scheduling Based on Content Metadata.** The majority of digital signage systems support a wide range of multimedia content including still images, web pages, and videos. The most simplistic scheduling approaches simply play out all available content items (e.g. all files in a directory), no criteria is used to reduce the available content down to a smaller subset for playback. However, it is not uncommon for signage platforms to schedule based on descriptive features of the content such as format constraints (must have an audio soundtrack or portrait orientation) [86], content type (e.g. show adverts or news), media type (e.g. show only videos) or content creators (show content only from this company) [35].

**3.1.2 R2: Scheduling According to Display Features.** While research systems typically feature only a small number of displays (often one), most realistic deployments will feature tens or hundreds of displays [1, 27, 44, 45, 50, 84]. The selection of the display(s) on which to present content is a key choice for any scheduling algorithm. It is often necessary to schedule content to specific displays or groups of displays [86] (e.g. to show a conference schedule outside a meeting room [83], or an advert on all displays in part of a shopping mall). Further, in the case that a single physical display is sub-divided into smaller conceptual display areas (e.g. in the case of ticker-tape or sidebars) then scheduling will need to account for these sub-divisions. Prior work has also highlighted a wide

Table 1. Summary of scheduling requirements for pervasive display systems.

<b>Traditional Signage</b>		
R1	Section 3.1.1	Scheduling Based on Content Metadata
R2	Section 3.1.2	Scheduling According to Display Features
R3	Section 3.1.3	Scheduling According to Temporal/Ordering Constraints
<b>Interactivity, Personalization, and Priorities</b>		
R4	Section 3.2.1	Supporting Dynamic Scheduling that Accommodates Interaction
R5	Section 3.2.2	Supporting the Full Range of Personalization Models
R6	Section 3.2.3	Supporting Priorities, Policies, and High-Level Goals
<b>Context</b>		
R7	Section 3.3.1	Scheduling Based on Sensed Audience Characteristics and Behavior
R8	Section 3.3.2	Scheduling based on (Local) Sensor Values
<b>Openness and Control</b>		
R9	Section 3.4.1	Supporting Applications, Content, and Scheduling Requests from a Wide Range of Sources
R10	Section 3.4.2	Supporting Diverse Moderation Approaches in Scheduling Decisions
R11	Section 3.4.3	Support the Introduction of new Scheduling Policies from Multiple Stakeholders

range of display factors that may influence the scheduling decision, including their ability to support specific content types, their shape and layout, their ability to support presentation of concurrent content items (i.e. by spatially multiplexing content on the display [74]), and the display’s location [85].

**3.1.3 R3: Scheduling According to Temporal/Ordering Constraints.** Many signage systems use temporal or ordering constraints as factors when making scheduling decisions. Perhaps for this reason, the need to meet temporal constraints has been prominent in both early [83] and recent [8] attempts to describe requirements for pervasive displays. However, there is considerable variety in temporal constraints. These include requirements to show (or prevent the showing of) specific items at a given time (“Show *the lunch menu* from 11:00 to 14:00”, “Do not show *the movie trailer* after 20:00”) but also requirements that control the number (“Show *the movie trailer* twelve times today”, “Do not show *the movie trailer* more than five times this morning”) or order (“Show *local weather station report* immediately after *the national weather forecast*”) of media playback.

## 3.2 Interactivity, Personalization and, Priorities

The needs of traditional signage systems can typically be met using a static approach to scheduling, i.e. the system or the display owner can construct a static, deterministic schedule of content presentation that specifies precisely when any given piece of content will be shown and on which displays. Such an approach (characterised by Bushman and Labrinidis as procedural scheduling [8]) is epitomised by systems that provide users with interfaces based on producing timelines (a classic example of this type of approach is the Sony Ziris system [82]). However, the introduction of technologies such as touch screens that support interactive content make static, deterministic scheduling impossible. Similar problems occur when considering live content where the duration is not known at the start of presentation.

**3.2.1 R4: Supporting Dynamic Scheduling that Accommodates Interaction.** Interactive content does not share key properties of traditional signage content, where the start time and duration for each content item can be computed in advance. Interactive content may be requested at any moment and will often need to be visible during the entire interaction with the display. Supporting truly interactive content therefore requires support for three key scheduling features [29]:



Table 2. Summary of prior research and notable commercial deployments discussed in Section 2 with respect to the scheduling provision made by each display systems. Provisions are reported against the eleven requirements identified in Table 1 (see Section 3). A checkmark indicates that the platform provides the functionality described by the requirement, an asterisk indicates partial fulfilment.

	Traditional Signage			Interactivity, Personalization, Priorities			Context		Openness, Control		
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
<b>Round Robin &amp; Declarative Playlists</b>											
Campus Coffee Display [13, 55]	*										
Hermes Photo Display [10–12]	*			✓							
<b>Interactivity</b>											
Plasma Poster Network [14, 15]	*			✓	*						
FLUMP [33]	*			✓	*						
Outcast and Groupcast [59]	*			✓	*						
WE-BAT / USI Displays [31, 32]	*			✓	*			✓	✓	✓	
UBI-Hotspots [41, 74]	✓		✓	✓		✓		✓	✓		
Intel AIM [46]							✓				
<b>Context Constraints</b>											
Situated Displays [53, 79]	*			✓	*	✓		✓	✓		
ReflectiveSigns [66]								✓			
e-Campus API [83]	*	✓	✓						✓		✓
e-Channels [35]	✓	✓				✓			✓		
<b>Auction-Based &amp; Utility-Driven Approaches</b>											
BluScreen [75]								✓			
Actionable Advertisements [67, 68]								✓			
Alt et al. (2009) [3]					*			✓			
Bauer, Dohmen & Strauss (2011) [5]			✓					✓			
Bushman and Labrinidis (2019) [8]			✓					✓			

**Arbitrary content presentation start time:** Public displays should be able to react to user input and show interactive content in an immediate fashion. Any latency in reacting and showing interactive content can affect overall user experience with the system.

**Arbitrary content presentation duration:** The duration of user interactions (and thus the actual content presentation duration) is difficult to predict in advance. Removing interactive content or switching to static content during interaction can negatively influence the user experience and impact the overall usability of the system.

**Relative and absolute presentation time limits:** While interactive content can increase the utility of public displays, display operators need to be able to limit the duration of interactive content and applications to prevent them from monopolizing the screens. Such limitations can be described either in absolute terms (e.g. number of minutes or time of day) or relative (e.g. percentage) to other applications. In such cases, displays may need to provide appropriate transitions (e.g. using overlay warnings, or moving the interactive content to a smaller screen region) and show notifications to users explaining the display behavior.

**3.2.2 R5: Supporting the Full Range of Personalization Models.** In addition to interactivity, changes to predetermined schedules may also need to be accommodated in order to cater for personalization of a display to reflect the viewers present. While most display personalization work has focused on adapting immediately to viewers that pass by a screen, Davies et al. identified three distinct classes of display personalization [24]:

**Walk-by personalization** in which viewers passing by a single display see content that is relevant to them.

**Longitudinal personalization** in which viewer preferences for personalized content are realised as a shift in content on multiple screens in a given geographic area, typically over an extended period of time.

**Active personalization** in which users (inter-)actively engage with a display system to control personalized applications on a nearby display, e.g. to extend a mobile phone display for better viewing of a complex interface.

Each of these forms of personalization places different demands on the underlying scheduling system – from rapid response to viewer proximity, to the adjustment of long-term high-level goals and policies (see R6 below) on a per-user basis.

**3.2.3 R6: Supporting Priorities, Policies, and High-Level Goals.** In addition to both the static and dynamic scheduling constraints in R1 through R5, one can imagine more generally the specification of high-level goals or policies for a set of displays. Examples of the need to support high-level scheduling policies can be found in [78] and [35]. Such policies may include prioritising new content items [78], maintaining a particular ratio of content types (e.g. 80% local events, 20% advertising) [35], or guaranteeing certain exposure times for advertising content in the face of user interactivity. Such goals or policies have previously been referred to as long-term application scheduling in the literature [31], but in reality goals and policies may operate over a range of time-scales. For example, a policy that allows the display to be utilised for emergency announcements [22, 26] may give such content short-term but absolute priority over any content, irrespective of the long-term policy in place.

### 3.3 Context

Public displays are inherently situated objects – they exist within the physical, social and cultural setting of their deployment. This focus on the situated nature of displays has been an important part of pervasive display research since its inception. In their early edited volume, O’Hara et al. highlighted the bi-directional nature of this relationship: “...they inform us about places, amenities, and events of interest and reflect the activities of others... They act as important cultural reference points in the construction of shared meanings, beliefs, desires and the memories of groups and communities.” [73, pxvii]. In [52] José et al. used interviews and workshops to derive seven “dimensions of situatedness” that impact on pervasive display content: *location, spatial association, activity, community, perceived ownership, and place identity*.

**Location of the Display** Viewers often expect displays to show information relating to the location of the display. This can be as simple as a “you are here” annotation onto a larger map, but also extends to showing local weather and other content for which geographic location is a key determinant (as opposed to other location-related attributes covered below).

**Spatial Association** The spatial arrangement of the display relative to its surroundings was also found to impact on the content that viewers expected to see (e.g. wifi information close to a wifi router, or directional arrows that are correctly oriented).

**Activity** Displays may be associated with a specific activity – for example screens positioned in an urban travel hub should anticipate commuter needs by providing relevant travel information.

**Community** This dimension relates to the extent to which a display becomes part of, and draws content from, a community. Typical examples include coffee shop notice boards [13, 55] but other examples include

systems designed to encourage interaction between colleagues in a workplace [14], attendees at social gatherings [7] or in a school [47].

**Perceived Ownership** Displays are typically considered to have an explicit owner. Users may or may not be able to easily identify the owner but often assume that it is the same as the owner of the physical space in which the display resides [20]. Clearly the ownership of a display will have a direct impact on the choice of content.

**Place Identity** This relates to the public perception of the physical location within which the display is situated – and the expectation that this might put on the content being shown [52]. For example, one might expect high-quality content for prestige brands to be shown on large screens in airports or shopping malls while lesser brands might appear on screens in low-key locations.

Several of these dimensions are already subsumed within other, more general scheduling requirements. In particular, scheduling according to location and spatial association are part of R2 (Scheduling According to Display Features); perceived ownership is part of R10 (Supporting Diverse Moderation Approaches in Scheduling Decisions) below. However, the dimensions *activity* and *community* require the system to schedule content based on a display’s audience and their behavior – similar in principle to the longitudinal personalization of Supporting the Full Range of Personalization Models yet with a view towards preferences of multiple viewers. Similarly, environmental context can provide further scheduling requirements. For example, Memarovic et al. [60] describe a display system that is designed to support autopoiesic content, i.e. content that is generated by the display itself as a result of input from sensors in the local environment. We thus derive two further context-based requirements.

*3.3.1 R7: Scheduling Based on Sensed Audience Characteristics and Behavior.* In order to provide content suitable to the identity of a place, and or the dominant audience (both over time or at a particular moment), a system should be able to schedule content based on viewer demographics (average or current) and their behavior (visible and invisible, e.g. their walking speeds or combined WiFi downloads) [46].

*3.3.2 R8: Scheduling based on (Local) Sensor Values.* Beyond taking the presence, preferences, and behavior of a display’s viewers into account, one can also schedule display content based on other, environmental data, (e.g. light levels, temperature and humidity, noise, but also WiFi traffic or number of Bluetooth devices in vicinity). Individual sensor values or aggregation over multiple sensors can thus influence (and dynamically change) parameters of the scheduling logic over time, allowing the display to adapt to its changing environment over time. For example, a display may respond to increasing noise levels by reducing the priority of content that includes an audio soundtrack.

### 3.4 Openness and Control

While traditional display networks have operated as largely closed systems (managed by a single organisation), there has been increasing interest in stimulating innovation by creating “open display networks” that can accommodate content and control from multiple sources [26]. Such networks are characterised by open APIs and software components that enable developers to create new signage applications [26, 83], techniques for enabling content ingestion from multiple sources [21], and recognition of the complex set of stakeholders that exist in such networks [4, 21, 63, 89]. In [26] the authors provide three example scenarios of potential applications for open display networks:

**Emergency Services** In this scenario, displays in a shopping mall are used to distribute images of a missing child using images supplied by the child’s parents. From a scheduling perspective the important aspects of this scenario are that (a) the displays are owned by different organisations (e.g. multiple shops, the shopping mall); (b) the content appears on screens in an increasing geographic radius from the point at which the

child was last seen – reflecting the distance the child might have travelled; and, (c) the content is clearly of a higher priority than other content on the displays and hence preempts existing content (see also R6).

**Influencing Behavior** In this scenario displays situated within a town are used to encourage children to exercise on their way to school by providing a distributed treasure hunt. Key issues for scheduling in this scenario are (a) as in the previous scenario, content is scheduled across multiple displays, (b) touch interaction enables children to “collect” items from the screen, and (c) personalization is used to ensure that the application is only displayed when the target demographic (i.e. children) are present.

**Self Expression and Personalization** While much personalization work has focused on delivering targeted adverts to increase revenue streams for display networks, this scenario highlights the use of displays for self expression by those nearby, changing the display to reflect a passer-by’s music tastes. In this case the requirement is to support both (a) walk-by personalization, and (b) content from new sources.

The scenarios above illustrate many of the challenges that arise as a result of a move to open display networks. The problem is compounded by the diverse set of stakeholders involved. Prior research has identified four stakeholders in typical public display deployments: display owners, space owners, content providers, and viewers [4, 21, 63, 89]. In traditional display systems many of these stakeholders are represented by a single organisation (e.g. an advertising company may simultaneously act as both the display owner and content provider). However, display networks are increasingly featuring a more diverse set of stakeholders. For example, in shopping complexes many hundreds of displays may exist in close proximity yet be situated in spaces owned by different stakeholders (shops or concession owners), operated by different signage companies, and show content from multiple distinct content providers.

Allowing a wider set of stakeholders to introduce both content and applications to displays introduces a need for content moderation [34]. Research with display owners has shown that while most agree that an open display network can bring added value, they still prefer their own displays to remain firmly under their control [20]. When asked, most articulated a compromise in which passers-by could at best select from a set of pre-vetted content and applications, rather than calling up content from unknown sources. Should unknown content be introduced into the network, all expected that *moderation* would be required. Greis et al. [38] found that passers-by also expect content pre-moderation, and would be prepared to wait for up to 10 minutes for their content to be approved. However, their in-the-wild deployment found that pre-moderation significantly lowered the number of content items posted. Post-moderation approaches, such as the system by Elhart et al. [32], which allows display operators to instantaneously remove unwanted content from a display through an in-situ authentication (using a key card) and a touch screen are also possible.

Given the above, we formulate three further requirements to support open display networks:

**3.4.1 R9: Supporting Applications, Content, and Scheduling Requests from a Wide Range of Sources.** Supporting articulated visions for future open display networks requires new collaborations between stakeholders – these collaborations have a direct impact on signage scheduling systems. For example, architectural support will be needed to connect displays to a range of different sources, as well as to signal the availability of content (or the need to display content) downstream to individual displays or groups of displays.

**3.4.2 R10: Supporting Diverse Moderation Approaches in Scheduling Decisions.** In closed signage systems moderation is typically not critical as the content is derived from a closed eco-system of trusted content providers and display owners. However, in open systems in which content can come from a wide range of sources moderation becomes a more critical requirement. Elhart et al. [32] discusses options for moderating user-submitted content in open display networks, noting that pre-moderation is often not possible due to the heightened need for timeliness. Greis et al. [38] proposes community-based flagging of inappropriate content, or at least to provide information on how content is moderated and how long the average wait should be. As an alternative, Davies et al. [24]

propose the use of trusted applications that offer their own moderation. Beyond the support of moderation at the point of ingestion, a moderation-aware scheduling system would require appropriate metadata to support context-dependent moderation.

*3.4.3 R11: Support the Introduction of new Scheduling Policies from Multiple Stakeholders.* While R9 and R10 are primarily concerned with supporting new applications and content, a truly open display network would allow developers to extend the scheduling system itself. A variety of approaches exist in the software systems communities to facilitate extensible system development (e.g. component-based software), and signage platforms such as Yarely [18] have shown that these can be applied in the digital signage domain.

## 4 A MULTI-STAGE SCHEDULING ARCHITECTURE

### 4.1 Overview

Section 3 demonstrates the breadth of scheduling requirements faced by future pervasive display systems. Likewise, our exploration of related work has shown the wide variety of approaches taken to schedule content items in existing signage platforms. For example, commercial systems often use static playlists to schedule content onto digital signs (e.g. [80, 82]), and research platforms typically opt either for lightweight round-robin approaches (e.g. [13]) or complex constraint-based schedulers (e.g. [35]).

Despite their utility in systems to date, Table 2 clearly shows that none of these approaches come close to fully addressing the requirements of future multimedia signage. Playlist-based models and round-robin schedules ignore context (including potential audience members) completely, whilst more complex approaches typically focus on a single sub-component of the scheduling problem. For example, constraint-based systems tag each content item with a series of constraints (e.g. only show a specific content item between 9am and 5pm) that help determine when each item can be presented, but do not provide a definitive item schedule for any given point in time as there are likely to be multiple content items that satisfy the current constraints. Similarly, constraints are not well-suited to expressing the rich mix of priorities and policies that may further influence content selection. By contrast, auction-based systems (e.g. [75]) may result in a single winning item but need integration with other approaches to address availability restrictions, and have typically been restricted to advertising content in which it is possible to ascribe a financial value to presenting a content item.

Consequently, we believe that future pervasive display systems will require scheduling techniques that consider both:

- (1) the process of reducing available content to a pool of items that are legitimate for playback in the current setting, *and*,
- (2) the selection between multiple items that meet this description.

In both cases, scheduling should be conducted within the context of a set of expected behavior or goals for the multimedia signage system. The presence of interactive content and context triggers mean that precise schedules cannot generally be computed in advance and hence scheduling in display systems is not a simple static optimisation problem.

We propose a generalised architecture for a digital signage scheduler that is based on three distinct stages [Figure 1] – a *high-level goal setting stage* in which the expectations of stakeholders are collected and evaluated in order to generate a set of content items and their associated constraints, a *constraint filtering* stage that is responsible for filtering out those content items that cannot be displayed at a given point in time, and an *item selection* stage that selects the appropriate element to show from the remaining set. Unlike previous approaches for scheduling in digital signage, our architecture is designed to support not only hard constraints (e.g. this item cannot be shown in the current context) but also to comply with stakeholder goals and policies that operate over long periods of time (e.g. show newer content more frequently than old content). In order to achieve this, we

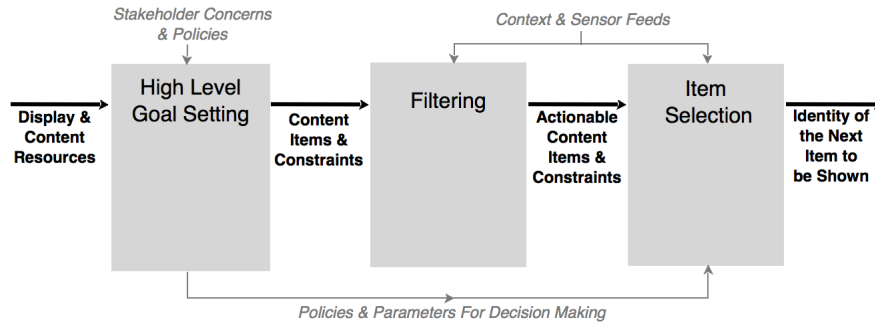


Fig. 1. A three-stage architecture for digital signage scheduling. Stakeholders set high-level goals for the use of display and content resources, based on these content is filtered down to a pool of actionable items, and then a single item is selected for each schedulable display resource.

ensure that item selection is informed by both the high level goal stage (through the communication of policies and parameters) and the filtering stage (through the communication of actionable content items).

#### 4.2 Stage 1: Describing Constraints and Goals

*During High Level Goal Setting, the scheduling process takes stock of the resources available (displays and content) and supports the expression of overall constraints and goals for the use of those resources.*

Our staged model begins with the gathering of stakeholder expectations and the representation of those expectations as a set of policies and constraints. Core entities within the signage ecosystem (content and displays) are inventoried, and stakeholders are provided with a set of *user interfaces* to populate these inventories and provide the policies that describe their expectations. These policies may take the form of *constraints* which must be respected by scheduling processes at every given moment (e.g. advertising content can only be shown between the hours of 12 noon and 2pm), and some higher level *goals* that describe the general expected behavior of the scheduler over time (e.g. the news bulletin will normally be shown once per hour). Whilst the preceding examples described policies that operated over content, policies may also related to specific displays (e.g. in the case of a display that will not normally show content that has recently played on a nearby display).

The goal management component integrates policies expressed by stakeholders together with the inventory data that describes the signage ecosystem and then exports these in a format to be processed at subsequent stages. For architectural purposes we distinguish between goals and constraints at the point of output generation, although in practice some implementations may provide an integrated export of these two datasets.

#### 4.3 Stage 2: Filtering

*In the Filtering stage, the scheduling process resolves constraints on content items and displays to determine the set of valid media and display resource pairings.*

During the Filtering stage, we attempt to restrict the content pool by removing items that are inappropriate for playback in the target context. In general scheduling theory and constraint programming, constraints are expressed as linear equations that limit the space of possible solutions. For example, end processing time  $e(a_i)$  of a task  $a_i$  can be expressed as a sum of start time  $s(a_i)$  and its processing time  $p(a_i)$ :  $s(a_i) + p(a_i) = e(a_i)$ . These constraints are used to optimize (minimize or maximize) the scheduling criterion or optimization function. However, in public displays high-level goals are usually not expressed using mathematical equations that are subject to optimization, and constraints typically take the form of statements that are either met (true) or not met

(false) at any point in time. For example, a content constraint that indicates availability of the content can be expressed as “*content item  $a_i$  is available at  $t_j$* ”. Further, as our exposition of requirements has already shown, constraints for content scheduling may relate to a wide range of factors including content metadata, screen capabilities, time and dates, environmental factors and the presence (or absence) of viewers.

Conceptually we can think of the Filtering stage as taking only two inputs: *the content items and associated constraints* (i.e. the primary output from Stage 1), and *a set of contextual data including sensor feeds*. The dynamic nature of display systems and their context mean that it is not possible to statically determine the scheduling for a display network using constraints. As a result we propose that Stage 2 is used only to filter content items at the point at which a scheduling decision needs to be taken. If the constraints associated with a content item clearly cannot be met in the given schedulable unit (display plus context) then the item will not progress any further through the scheduling process (it will be filtered out). During this second stage stakeholder concerns and policies are put aside (they will be revisited during the final stage).

Once the filtering process has been completed for all incoming content items, the final pool of *eligible content items* can be forwarded to Stage 3 (selection). Whilst in some settings, the final set of *eligible content items* may consist of a single item, it is likely that in the majority of deployments multiple items remain possibilities for playback. The resolution of these to a final decision is made in the Selection stage.

#### 4.4 Stage 3: Selection

*In the Selection stage, the scheduling process narrows a pool of equally-valid resources down to a single item for playback on the display. This process is conducted in accordance with previously expressed goals and policies, and may leverage current context.*

The Selection stage operates over a set of *eligible content items*, each of which has been determined to be valid for playback in the target context. During selection, the goal is to reduce this set down to a single content item (assuming scheduling is taking place for a single display region). To achieve this, the selection process revisits the policies and parameters specified during Stage 1 (High Level Goal Setting). Such policies may be highly variable, and may often depend on some contextual factors (including environmental context such as time, location, weather; but also playback context such as recently displayed content items).

Trading off the various policies and constraints is arguably the most challenging problem in display scheduling. In our architecture [Figure 2], this process is handled by a *selection manager*, receiving the eligible items and reducing them down to a single item. Whilst the selection manager itself could be considered a black box, we incorporate into our architecture recent work by Mikusz et al. [62], who propose that lottery scheduling approaches may be well-suited to this problem. Section 5.3 explores lottery scheduling as a solution for Stage 3 in more detail, but our overall architecture folds in core components for lottery ticket storage and allocation. Specifically, a ticket allocator is provided to handle each of the possible policy specifications (e.g. a goal to achieve proportionate scheduling of two distinct content types). Each ticket allocator may query the context store for data points related to the policy it is intended to uphold, leveraging this data in order to shape its distribution of tickets between content items.

## 5 EXAMPLE IMPLEMENTATION

The e-Campus system provides an example of a partial implementation of our scheduling architecture (Figure 2). Instantiated in 2004, the e-Campus system is the world’s longest running and largest public display research testbed, and is in daily use as the main digital signage system at Lancaster University. While aspects of the e-Campus scheduling system have previously been described in [18, 21, 35, 62], this represents the first attempt to describe the components of the e-Campus system in the context of a unified scheduling architecture.

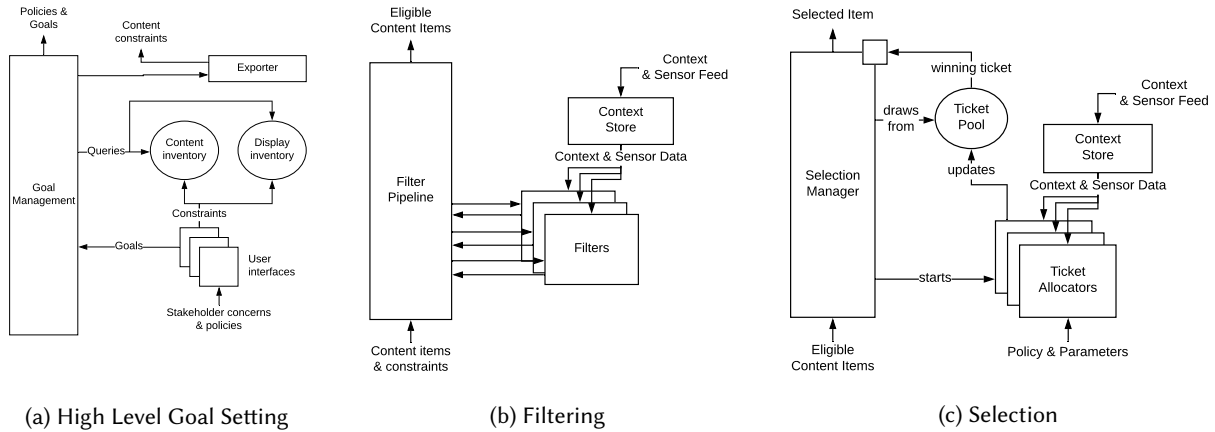


Fig. 2. Detailed architectural breakdowns for the three scheduling stages as realised in e-Campus: (a) High Level Goal Setting, (b) Filtering, and (c) Selection.

### 5.1 Stage 1

In e-Campus display owners and content creators can choose from two web-based systems for distributing their content and configuring display schedules: e-Channels [35] and Mercury [21]. The e-Channels system focuses on organising content items (e.g. images, slides or videos) into logical groups (“Channels”) of items provided by content providers. Time constraints can be placed on the availability of the channel and content providers can choose whether the channel is available only to the displays they themselves own or is to be shared publicly (i.e. available on any screen on campus). It is important to note that while content providers can determine content availability by adding or removing it from the channel, they cannot determine the order in which content is shown within a channel. Display owners can configure the operating times for their displays and can subscribe each display to content from one or more channels. They can also specify a policy for content playback ratio on a per-channel basis (e.g. request content from a certain channel to be shown 75% of the time whilst other channels are allocated a lower playback ratio).

Mercury is an application store for public displays and allows the distribution of content modelled as an application, providing support both for traditional signage content (such as images and videos) and new types of interactive applications. Display owners can use Mercury to choose from a large set of applications (content) and add these to their displays. The e-Campus version of Mercury provides the same basic capabilities as e-Channels in terms of adding and removing subscriptions to content channels. While Mercury was originally conceived as a successor to e-Channels, the enduring popularity of e-Channels with users has meant that the vast majority of users choose it as their preferred management platform.

Both e-Channels and Mercury export display content subscriptions (i.e. the set of channels and applications that were assigned to a display by the display owner) including constraints (e.g. display on and off times or content play constraints) using an XML-based Content Descriptor Set (CDS) format described in Clinch et al. [18]. The use of CDSs provides a consistent means by which processes in Stage 1 can communicate their results to Stage 2.



## 5.2 Stage 2

Each e-Campus display node runs the Yarely [18] software stack, an open-source modular digital signage player<sup>1</sup>. Yarely accesses display schedules from a combination of e-Channels [35] and Mercury [21] and is designed to periodically pull CDSs from a combination of local and/or web-based sources (e.g. e-Channels). The CDS is then first parsed by Yarely's Context and Constraints Parser and then converted into an object-based format that can be read by other system components.

In parallel to receiving the CDS, Yarely also provides an interface that enables capturing and storing of contextual events, e.g. from sensors connected to the display player or from remote sources (e.g. Tacita [24]). Contextual events arrive in the form of a combination of metadata (expressing the type of event) and a CDS (should the event include a content schedule request). The incoming event is parsed and stored within Yarely's Context Store – a lightweight SQLite database on the local file system [62].

Yarely initiates its content scheduling process with its implementation of the Filtering stage. This is realised in the form of a processing pipeline (previously described in [62]): the parsed CDS is passed through a pipeline of independent filtering components which each remove items that are not eligible to be played in the target context. The filters are not executed in parallel, instead each filtering operation generates a CDS result that consists of equal or fewer items than it received as input; this new CDS is passed on to the next filter in the pipeline. In the context of e-Campus, we developed the following set of independent filtering components:

**Cache Filter** Yarely has been designed to cache static content such as images and videos locally to reduce network traffic and to mitigate the consequences of network failures. For each content item that can be cached, the cache filter accesses the local file system to ensure that the individual content item is available locally. For content items that cannot be cached (e.g. live media streams), the cache filter skips the check and does not access the local file system. The overall effect of this filter is to exclude content that is considered to be cacheable but that is not yet available locally.

**Constraints Filter** Content Items (and entire CDSs) can be subject to temporal constraints that limit availability based on date, day, or time. For each content item, the constraint filter accesses constraints associated with a content item (and its parent CDSs) and ensures that these are met in the target context (typically now).

**Priority Filter** To support the prioritization of content (e.g. in emergency situations) content items specified in the CDS can consist of a priority flag such as “normal”, “high” and “highest”. The priority filter iterates over the available set of content items to identify the highest priority available, and conducts a second iteration of the available content to remove content items with a lower priority.

**Touch Input Filter** To support interactions with touch-enabled displays, Yarely can display an application dock that allows users to select content they wish to see on the display via touch input. Requests made in this way are stored in the local Context Store. The touch input filter queries the Context Store for requests made within the last five seconds; if multiple requests have been made, then only the most recent is used. In the event that a recent request is found, the touch input filter verifies that the requested content is eligible for playback in the target context (i.e. did it form part of the original CDS) and then removes all other content to ensure that only the requested content remains eligible for playback on the display.

**Tacita Filter** The Tacita filter supports remote display personalization requests in a similar manner to the touch input filter. The Tacita filter queries the context store for the recent personalization requests, verifies the eligibility of the content item for the target context, and then removes all other content to ensure the removal of all content not described in the personalization request.

<sup>1</sup><https://github.com/opensdisplays/yarely>

Each filtering component has been implemented as a separate Python module, making it straightforward to introduce new constraint types and associated filters. After completing the filtering process, the resultant set of eligible content items (a CDS) is passed on to Stage 3.

### 5.3 Stage 3

The selection of which specific item to show from the set of all eligible content items has been implemented in the form of a lottery scheduler [62]. The scheduler consists of three main components: a *selection manager*, a set of *lottery ticket allocators* and the *ticket pool*. The selection manager distributes available lottery tickets to individual lottery ticket allocators and orchestrates the overall content scheduling process. Each lottery ticket allocator operates independently to distribute its lottery tickets to content items in accordance with its own policies and algorithms. Dividing allocation in this way allows each to implement different scheduling policies and requirements, and for these to co-exist in the same system simultaneously.

Once a lottery ticket has been allocated to a content item, the assigning allocator places the ticket in a common ticket pool. Upon completion of the ticket allocation processes (which, unlike the filtering stage, execute in parallel), the selection manager draws a lottery ticket from the common ticket pool. The winning content item is then scheduled to the display. Our current implementation provides the following lottery ticket allocators:

**Ratio-based Ticket Allocation** The ratio-based ticket allocator implements the high level policies that describe preference for content playback ratios (as described in Stage 1, above, and in Friday et al. [35]). It distributes lottery tickets to individual content items based on the playback ratio of the channels in which the content items have been placed.

**Recency-based Ticket Allocation** Display owners may choose to prioritize content that has not been shown recently (perhaps as a result of not having been the subject of personalization or interaction requests). The recency-based allocator prioritizes content items that have not been played recently over other content items. This is achieved by allocating a higher proportion of lottery tickets to least-recently played items as follows. First, the allocator requests an ordered list of the  $n$  least recently played content items from the Context Store, it then assigns  $2^{n-1}$  lottery tickets to the  $n$ -th content item in the list of recently played content until no more tickets are left.

Whilst the above allocators take into account only limited contextual data (the ratio-based allocator uses no context at all, the recency-based allocator needs only to know about the items most recently shown on the target display), other ticket allocators could query the Context Store to inform the allocation of tickets based on external or contextual events.

Our implementation is structured similarly to the previously described Filtering stage, with each ticket allocator realised in its own Python module. Each allocator registers its presence with the Selection manager, and can communicate with other components of Yarely via the Context Store and ØMQ interfaces. While conceptually lottery tickets are allocated to content items, in the underlying implementation each ticket allocator receives a set of pre-generated empty ticket instances that can each hold a reference to one content item. The ticket allocation process is threaded: ticket allocators run in parallel and are managed and monitored by the Selection manager. Ticket allocators can then assign content items to lottery tickets until all available lottery tickets are assigned – or the ticket allocation process is stopped by the Selection manager to ensure a prompt scheduling decision. For more details on the motivation for using lottery scheduling in public displays see [62].

## 6 EVALUATION

Our evaluation considers the validity of the proposed architecture from three distinct angles of inquiry. **Firstly**, can the proposed architecture adequately support the wide range of requirements associated with current and envisaged pervasive display systems, and to what extent are these requirements met in our current implementation?

**Secondly**, can the architecture meet the performance demands of a modern signage system and, in particular how well does our implementation perform on representative demands made in controlled conditions? **Finally**, how feasible is the architecture in practice – can our implementation meet the needs of a real-world deployment? We address these questions using a mixed methods approach. We begin, in Section 6.1, by revisiting our eleven requirements and critically analyse both our architecture and the implementation with respect to these. We then turn further attention to the implementation as a measurable artefact that realises the proposed architecture – Section 6.2 presents lab studies that quantify the performance of key aspects of our implementation in controlled settings, whilst Section 6.3 quantifies experiences gained from long-term usage of the implementation in an in-the-wild testbed. Collectively we believe these evaluation methods provide evidence of the utility and feasibility of the proposed architecture.

## 6.1 Addressing Scheduling Requirements

In Section 3 (summarised in Table 1) we articulated eleven requirements based on a survey of systems reported in prior literature, published use cases for future open display networks, and own experiences of operating and studying a long-lived research testbed. Here we revisit those requirements reflecting on the described architecture and implementation. In so doing, we aim to establish if the many and varied needs of pervasive displays can be met using the proposed multi-stage model. We revisit the requirements in their original four groupings; our findings are summarised in Table 3.

*6.1.1 Traditional Signage Functionality.* The first three requirements were concerned with providing traditional digital signage functionality, that is, the scheduling of media based on inherent metadata (R1), to specified displays (R2) in accordance with some temporal or ordinal specification (R3). Addressing these requirements for traditional signage (R1 - R3) is supported across multiple stages of the architecture and implementation, but is principally expressed in Stage 1 and enforced at Stage 2.

Specification of policies related to all three requirements is handled at Stage 1 (High Level Goal Setting) as represented by the incoming arrow "Stakeholder Concerns & Policies" shown in Figure 1. Representative constraints are then passed through to Stage 2 (Filtering). At this stage items that do not possess the target metadata (R1), are not intended for playback on the target display (R2) or are not eligible because of timing or order restrictions (R3) will be removed leaving a set of actionable content items. It is evident that compliance with R1 - R3 may still leave a choice of content items at any given point in time. For this reason, a complete implementation to meet these requirements will still require a Selection stage (Stage 3) that determines the single most-appropriate item from all remaining actionable content items.

Our implementation largely follows the above pattern. Both e-Channels and Mercury provide user interfaces (Stage 1) that allow display owners to specify the content providers from which they will accept content (R1) and the operating hours for a given display (R3). Likewise, content providers are provided with user interfaces that allow them to indicate the set of displays (R2) and time periods (R3) in which specified content items can be shown. Note that content source is presently the only supported content metadata, (R1) and that whilst temporal control is supported for both display owners and content managers, ordering constraints (R3) are a notable omission as part of our attempts to keep the user-interface as simple as possible [35]. Constraints specified through the user interfaces are exported, together with the content items, as a CDS that forms the input for Stage 2. Enforcement of constraints related to R1 and R3 are implemented using Yarely's constraints filter. Given the relatively static nature of R2 constraints (displays do not often change their name or identity) we support this requirement by filtering at the CDS production stage – thus displays only ever download CDSs with content that is potentially applicable to them. Filtered CDSs then progress to Stage 3, where content metadata is used as part of our implementation of R6.

*6.1.2 Interaction, Personalization and Priorities.* Requirements four through six concerned support for systems that responded to viewer interaction (R4), presence and preferences (R5), and with the meeting of high-level goals (R6). Two of these requirements (R4 and R5) essentially demand a shift away from pre-specified deterministic scheduling approaches to more flexible models that can adapt to changing situations; R6 may invoke this same pattern, but also allows shaping of scheduling over the longer-term (e.g. to meet a desire for 60% of content shown to be video). Requirements for interaction, personalization and priorities (R4 - R6) are supported across multiple stages of the architecture and implementation, with Stage 3 having greater importance in enforcing R5 and R6.

Specification of R4 - R6 are handled at Stage 1 (High Level Goal Setting). For R4 and R5 this corresponds to whether or not interaction and personalization are supported, and the forms it may take. Specification of priorities, policies and high-level goals (R6) is likely to be more heterogeneous and could take a wide variety of forms – some examples are given in Section 3.2.3 but these are not intended to be exhaustive. Enforcement of an authorised interaction (R4) or personalization (R5) request takes place at either Stage 2 or Stage 3. In the majority of cases, the preferred behavior will be to show the content that is subject to an interaction or personalization request *at the expense of any other content*; to ensure this behavior, enforcement would take place at Stage 2 when all other items will be filtered out. The resulting set of actionable content items will only include those associated with the current interaction or personalization request(s); these will then progress to Stage 3 where selection will determine the most-appropriate item should multiple options still remain. In other cases (e.g. for longitudinal personalization), it may be the case that an interaction or personalization request does not preclude other content from showing. To enable this behavior, no action may be needed at Stage 2, but instead Stage 3 will need to take into account relevant requests (e.g. by weighting requested items more heavily in the selection process). With regard to R6, again Stage 2 or Stage 3 enforcement may be more or less appropriate depending on the policy/goal type; as a general rule, short-term policies are more likely to be enforced at Stage 2, whilst those operating over the longer-term will be enforced at Stage 3.

Our implementation largely follows the two patterns articulated above for R4 and R5 (enforcement at Stage 2), and a mixture of both patterns (enforcement at Stage 2/3) for R6. E-Channels allows display owners to specify whether or not their display permits personalization requests (R5) and the desired ratios (R6) between channels that a display is subscribed to (this functionality is not supported in the Mercury user interfaces). Our implementation does not presently allow stakeholders to specify whether or not interaction is permitted (R4); we have a small number of touch-interactive displays and these are assumed to always permit interaction. Constraints specified through the user interfaces are exported, together with the content items, as a CDS that forms the input for Stage 2. R4 and R5 are implemented using Yarely's touch input filter and Tacita filter respectively – these filters exclude all items that are not currently subject to an interaction or personalization request. Our specific implementations of these filters select only the most recently occurring event such that only one actionable item will progress to Stage 3. R6 is implemented at Stage 3 (content and policy-relevant descriptions pass through Stage 2 without transformation) with the ratio-based ticket allocator. Evidence of the generality of this approach is seen in the provision of an additional (but presently unused) Stage 3 ticket allocator that supports a policy of prioritising rarely seen content.

*6.1.3 Context.* The next two requirements relate to the needs of systems that respond to sensed audience (R7) and environment (R8). Specification of both requirements is handled at Stage 1 (High Level Goal Setting), enabling users to specify constraints or policies for content presentation relating to audiences and their behavior (R7) and the environmental context of the display (R8). Enforcement can again be distributed over Stages 2 and 3. Consider, for example, the following requirement that combines both audience- (R7) and environment-focused (R8) contextual information: “*show adverts for ice cream when the local environment has high levels of ambient light and exceeds 19°C, but only do this if the majority of proximate users are children*”. In this case, the architecture

will use Stage 2 to filter out the ice cream adverts when they are not eligible for playback (i.e. when either the temperature, light or audience constraints are not met), but may rely on Stage 3 strategies to increase the probability of playback when the constraint is met.

Our implementation does not presently provide an interface for users to specify either audience- (R7) or environment-based (R8) constraints and it is not currently possible to adjust scheduling based on audience or environmental factors.

*6.1.4 Openness and Control.* Our final three requirements relate to support for openness to novel content (R9) and scheduling policies (R11); this novel content may need to be moderated (R10). Although the requirements are conceptually related, they are addressed at very different points in the architecture and implementation.

R9 is the exclusive concern of Stage 1 (High Level Goal Setting). Openness to new content relies firstly on the availability of that content, and secondly on the agreement of stakeholders that policy permits presentation of that content on a given display. These are both represented in Figure 1 by the incoming arrows for "...Content Resources" and "Stakeholder Concerns & Policies" respectively. R10 specification is dealt with at Stage 1 (as described for prior requirements), and enforced at Stage 2 through the removal of unsuitable content from the set of actionable content items. Note that moderation policies often overlap with other forms of scheduling constraints, typically R1, R3 and R7 (e.g. "do not show 'R' rated content before 9pm or while children are present") and so the moderation UI may map onto the same underlying technology as is used to address these requirements. Further, while moderation decisions are typically binary in nature (yes/no) it is possible for there to be a degree of uncertainty in the context in making the decisions and this uncertainty could be reflected in Stage 3. R11 is supported through specification at Stage 1, the resulting implementations of these policies would be applied at Stage 2 or Stage 3 as appropriate.

Our implementation's use of CDSs [18] makes it straightforward for any display to receive content from new sources (R9); multiple CDSs can be received at any display node and will be aggregated into a single schedule. Further, the Mercury app store was explicitly designed as a platform for the easy distribution of novel content from new parties to be distributed to a large pool of existing displays (subject to stakeholder policies). Our current implementation does not provide a UI for expressing moderation preferences (R10, Stage 1) and these are not enforced at the later stages; however, an implementation based on constraints (Stage 2) would require only minor additions to our current codebase. Finally, our implementation of the lottery scheduler is explicitly designed to be extensible (R6) – provision of new constraint filter modules (Stage 2), and ticket allocators (Stage 3) allow for easy addition to existing scheduling policies. At present these need to be manually added to the correct execution path; discovery of new policies is left as a potential area for future work.

## 6.2 Lab-based Benchmarking

*6.2.1 Methodology.* In order to explore whether it was possible to create a performant implementation of our architecture we conducted a set of controlled lab-based experiments capturing the delays incurred during Stages 2 (Filtering) and 3 (Selection). We acknowledge that the specific performance of these phases is implementation-specific. However, these statistics are not intended to measure the absolute performance of the architecture, or to compare against existing solutions (for which benchmarks are largely non-existent). Instead, our goal is to provide a general indication that the proposed multi-stage architecture can be realised, and that its performance in controlled settings is such that one might consider the system practicable for real-world signage scenarios. Note that our benchmarks focus purely on Stages 2 and 3 – Stage 1 (High Level Goal Setting) is primarily centred on allowing stakeholders to inventory resources, set goals, and express policies – providing a set of performance measures for Stage 1 interfaces would not be appropriate.

Our benchmarking setup is intended to reflect typical use of the e-Campus implementation introduced in Section 5. We select two representative machine configurations, one based on a Mac Mini and macOS, and

Table 3. Mapping of requirements to architectural stages (plain text) and example implementation (italics).

Category	Req.	Stage 1	Stage 2	Stage 3
Traditional Signage	R1	Allows users to express constraints such as "only show this specific content item" or "only show content from this source". Both Mercury and e-Channels allow stakeholders to specify constraints on individual content items; display owners can also limit content based on the content source.	Supports selection of only those content items that meet the specified constraints. The Yarely scheduler is able to filter content items according to a wide range of constraints.	Supports item selection based on content meta-data. No special provision made. However, the Yarely Stage 3 lottery scheduler ticket allocators do allow specification of metadata-based policies (R9).
	R2	Allows users to express constraints relating to the displays on which content is scheduled. Mercury and e-Channels allow stakeholders to specify displays that content can be shown on. Mercury permits content providers to describe features of the displays to which their content can be scheduled (e.g., orientation, interaction) – this is not supported by e-Channels.	Supports selection of only those content items that meet the specified display constraints. CDSs are only served to matching displays so no additional filtering is needed.	Item selection based on display characteristics. No special provision made.
Interaction, Priorities & Personalisation	R3	Allows users to express constraints relating to time and ordering. e-Channels allows stakeholders to specify the times that channels and displays are active. Ordering constraints are not supported.	Supports selection of only those content items that meet the specified constraints. The Yarely scheduler filters items based on temporal constraints specified in the CDS. Ordering constraints are not supported.	Item selection based on temporal and ordering characteristics. No special provision made.
	R4	Allows user to specify whether interaction is supported. Not currently supported in Mercury or e-Channels.	Filters out other content items if interactive content is to be shown. Context store used to ensure interaction events persist. Yarely touch input filter removes other content if interaction is taking place.	Select interactive content according to policies specified. No special provision made.
	R5	Allows users to specify whether personalisation is supported. e-Channels allows display owners to specify whether a display can show personalised content and which personalised content is permitted.	Filters out other content items if personalised content is requested. Context store used to ensure personalisation requisites persist. Yarely's Tacila filter removes other content if personalised content is available.	Select personalised content according to policies specified. No special provision made.
Context	R6	Allows users to express high-level goals and policies. e-Channels enables display owners to specify the proportion of screentime given to content from different channels.	Policies can be encoded in constraints. CDS parsed to identify policies that can be applied at the filtering stage (presently no such policies exist).	Item selection based on high-level policies and goals. Supported by the Yarely scheduler's ticket allocators, e.g. the ratio ticket allocator.
	R7	Allows users to specify constraints on content presentation in terms of e.g. the number and demographics of viewers in front of the display and their behavior. Not currently supported in Mercury or e-Channels.	Filters out items that don't meet constraints in terms of sensed audience or behaviour. Audience characteristics are sensed and monitored [64], but are not utilised in scheduling decisions.	Item selection based on sensed display context. Not currently supported.
	R8	Allows users to specify constraints on content presentation in terms of external events. Not currently supported in Mercury or e-Channels.	Filters out items that don't meet constraints described in terms of external events. Yarely context store allows persistence of sensor data. However, no sensor data is captured at present.	Item selection based on sensed display context. Not currently supported.
Openness & Control	R9	New content sources can be added from multiple stakeholders. e-Channels allows users to specify multiple content sources. Mercury is explicitly designed to promote distribution of a wider set of content items across displays.	Allows processing of constraints on third-party content. Supported through the use of a standard approach to expressing constraints (CDSs).	Allows selection of third-party content. Supported through the use of a standard approach to expressing constraints (CDSs).
	R10	Allows users to moderate content as part of ingestion process, or to attach metadata to the content to assist in context-dependent moderation. Enables the specification of rules and policies for moderation. Not currently supported.	Filters out items based on moderation rules and current display context. Not currently supported.	Item selection adjusted in proportion to the perceived uncertainty in determining context to reduce the probability of scheduling inappropriate content. Not currently supported.
	R11	Allows the specification of new scheduling policies. Not currently supported in Mercury or e-Channels.	Allows addition of new constraint types and appropriate filters. Modular code design enables easy addition of new filters and context specifications into Yarely scheduler and CDSs.	Allows addition of new policies through the creation of new ticket allocators. Modular code design enables easy addition of new ticket allocators into Yarely scheduler.

Table 4. Summary of hardware and software specifications used in benchmarking experiments. These two configurations were selected as representative based on the existing e-Campus deployment.

	Hardware Specification	Operating System	Yarely Renderers	Yarely Scheduling
Mac Mini	2.6 GHz Intel Core i7 processor 8GB DDR3 memory, 1 TB HDD	macOS 10.10.3 (Yosemite)	Native renderers	Platform independent
Intel NUC Mini PC	2.3 GHz Intel Core i5 processor 8GB DDR4 memory, 256 GB SSD	Kubuntu 19.04 (Disco Dingo)	Platform-independent Qt5-based renderers	Revision: 26883f

	Filters (Stage 2)	Lottery Ticket Allocators (Stage 3)	# of Tickets (Stage 3)
Mac Mini	Cache Filter, Constraints Filter,	Ratio-based Ticket Allocation,	5,000 per allocator
Intel NUC Mini PC	Priority Filter, Touch Input Filter, Tacita Filter	Recency-based Ticket Allocation	

the other using an Intel NUC Mini PC and Kubuntu (full specifications are given in Table 4). We used the five Stage 2 filters previously described in Section 5.2 and the two lottery ticket allocators described in Section 5.3. We benchmark using randomly-generated image files as content items (each with 1,000 x 1,000 pixels and a file size of approx. 0.6 MB); our benchmarks execute with a varying number of content items from 1 – 500 (30 repetitions of each). We use 5,000 lottery tickets per ticket allocation component, this equates to at least ten-times the number of content items, allowing each ticket allocation component to conduct a fine-grained distribution of lottery tickets to content items.

In order to accurately determine the delays for individual stages and components, we captured timestamps within Yarely for the following events: start of the overall scheduling process; start and end of the Filtering pipeline (Stage 2); start and end of each individual Filtering component; start and end of content Selection (Stage 3); start and end of individual lottery ticket allocation (Selection) modules; and start and end time of loading and rendering content items until becoming visible on the display. For each number of content items (1 – 500), we created a CDS in which all content items were eligible to play – allowing us to benchmark each filtering component and lottery ticket allocation module using the complete set of content items. In addition, for each CDS generated we pre-populated the context store with 1,000 entries representing previous scheduling events for items within the CDS – this was important for testing the recency-based ticket allocator. We note that all images were cached before conducting the benchmarking to ensure that the caching filter would not remove any items (which would in turn reduce the number of content items considered by the remainder of the scheduling process). In order to ease reproducibility of the benchmarks reported in this subsection, the scripts used for the experiments are freely available in an open source repository alongside the Yarely signage player<sup>2</sup>.

**6.2.2 Stage 2: Performance of Constraints Processing.** Figure 3 provides an overview of the mean processing times of individual constraints components for both the Mac Mini and Intel NUC Mini PC deployments. The differences in processing times across the filtering components is caused by different types of implementations and operations. Both the Tacita and touch input filters make a single query to the Yarely context store for recently requested content items (received via Tacita and touch respectively) and, if any items have been requested, only keeps these requested items in the content set. For the purpose of the benchmark the context store did not contain any Tacita or touch events and hence returned an empty content set in both cases. The constraints filter iterates over the entire CDS once and validates constraints for each content item (e.g. date and time) and hence we observe a linear increase in the processing time with the number of content items to process. The cache filter also iterates through the CDS and for each item conducts a (costly) file system operation to ensure that the file exists (i.e. has been cached). Finally, the priority filter iterates over the CDS twice – once to identify content items with

<sup>2</sup><https://github.com/opendisplays/yarely-benchmarking>

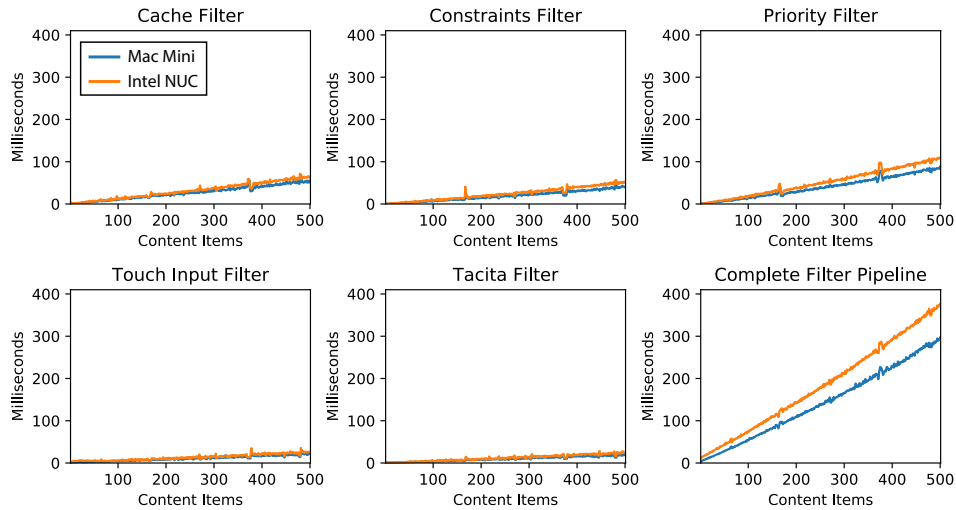


Fig. 3. Mean processing time (based on thirty iterations) for Stage 2 (Filtering). Blue: Mac Mini; Orange: Intel NUC.

the highest priority and a second time to remove lower priority items from the content set. The performance of the complete Stage 2 (Filtering) pipeline exhibits the same runtime complexity, but includes processing overheads associated with the initialisation of the filtering components and copy operations over the initial CDS. Both setups (Mac Mini and Intel NUC) perform similarly with the slight differences reflecting the different storage hardware and implementations of platform-dependent Python modules. Overall we note that while different filters have different performance characteristics, our test configuration is able to process large CDSs within reasonable time bounds (~300-350 milliseconds). Given that our configuration is composed of five separate filters designed to address most common signage needs, we believe that this provides a good indication that timely Stage 2 operation should be well-suited to real-world signage scenarios.

**6.2.3 Stage 3: Performance of the Lottery Ticket Allocation and Overall Performance.** Figure 4 shows the mean processing times for the execution of the ratio ticket allocation and recency ticket allocation components together with the overall lottery scheduling process. Whilst the runtime complexity for both ticket allocation modules grows linearly, the differences in absolute processing times are likely caused by an increased number of operations of the recency-based allocator. The recency-based module queries the context store for recently-played items and performs a comparison with the current set of actionable content items in order to prioritize appropriately. In contrast, the ratio-based allocator simply calculates the number of tickets to allocate based on metadata (i.e. ratios) that are provided as part of the content schedule, no additional querying or read/write operations are required. We observe marginal differences in the performance of the Mac Mini and Intel NUC setups, similar to those identified at Stage 2. Again, we note that while our allocators have different performance characteristics, our test configuration is able to select from up to 500 items within reasonable time bounds (< 200 milliseconds).

Considering the overall processing delay including filtering, lottery scheduling, content rendering and the overall processing time including any additional processing overhead (e.g. multi-threading management) we observe both similar growth patterns and processing times for both the Mac Mini and Intel NUC setups (Figure 5). Comparing the overall processing time with the processing times of individual components of the scheduling system suggests that the implementation of our scheduling architecture incurs minimal processing overhead.



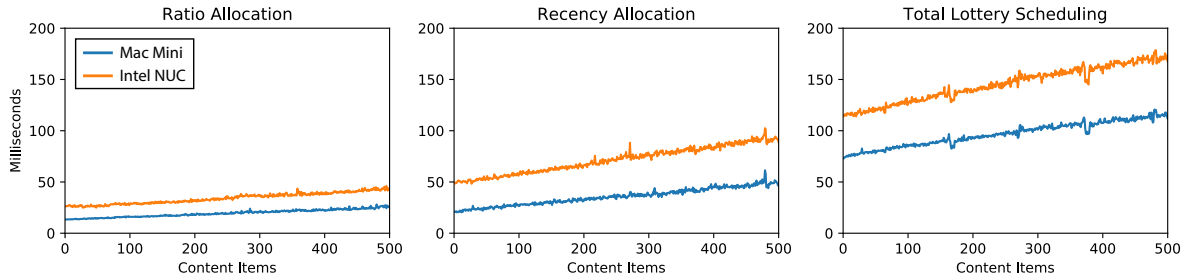


Fig. 4. Mean processing time (based on thirty iterations) for Stage 3 (lottery scheduling). Blue: Mac Mini; Orange: Intel NUC.

Indeed, for 120 content items (the average size of a display’s schedule in our e-Campus deployment<sup>3</sup>) the overall mean cost of conducting Stage 2 and Stage 3 scheduling on the Mac Mini is 151.8 milliseconds (median: 152.0; standard deviation: 20.2) and overall mean for the Intel NUC is 216.3 milliseconds (median: 217.0; standard deviation: 22.9) – both comfortably within reasonable bounds for a typical signage deployment.

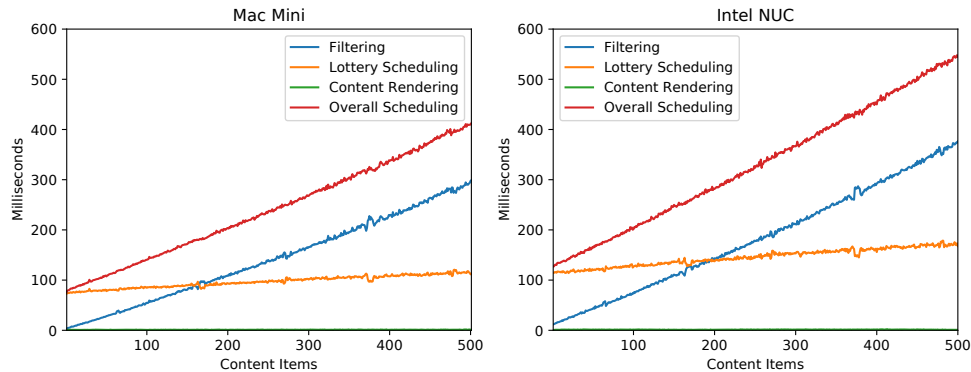


Fig. 5. Mean combined processing time for Stage 2 (Filtering, blue) and Stage 3 (Lottery Scheduling, green) based on thirty iterations. Left: Mac Mini; Right: Intel NUC. Note that the green line (Content Rendering) is sufficiently minimal to be indistinguishable from the  $x$ -axis.

### 6.3 Experiences from a Long-Term Deployment

Our final piece of evaluation uses the full multi-stage implementation described in Section 5. We build upon findings from Section 6.1 and 6.2 to ask how feasible is the architecture really, can our implementation meet the needs of the e-Campus testbed, a real-world deployment of approximately 100 digital signs? Again, we acknowledge limitations – our evaluation takes place in the context of one specific deployment with a particular set of design decisions. However, we argue that design decisions taken in e-Campus are by no means unique to the deployment and there is value in the demonstration of a real-world live implementation of the proposed architecture.

As described in Section 5, the e-Campus deployment is a research testbed that comprises mostly of thick-clients driving 40” to 65” LCDs. Screens are located across the Lancaster University campus including outdoor locations,

<sup>3</sup>Based on a snapshot taken in May 2019, for more details see Section 6.3.1.

departmental buildings, colleges and student bars. Specialist installations are also supported including a “Big Screen” (5 by 3 metres in size) situated in the University’s central square. A small number of displays are also located in the nearby city centre including the central bus station.

**6.3.1 Stage 1: e-Channels.** Stage 1 of our architecture is concerned with High Level Goal Setting; that is, taking stock of the resources available (displays and content) and expressing preferences for the use of those resources. As described in Section 5.1, in the e-Campus deployment the primary tool in use for these activities is e-Channels [19]. Usage patterns for e-Channels (2008-2011) have been published previously [19, 35] but the e-Campus network has grown and developed substantially since this time; new applications and content have been developed (e.g. Tacita [26]) and the number of displays has increased three-fold. We therefore summarise current use of e-Channels as an indication of the level of engagement with Stage 1 of our scheduling model.

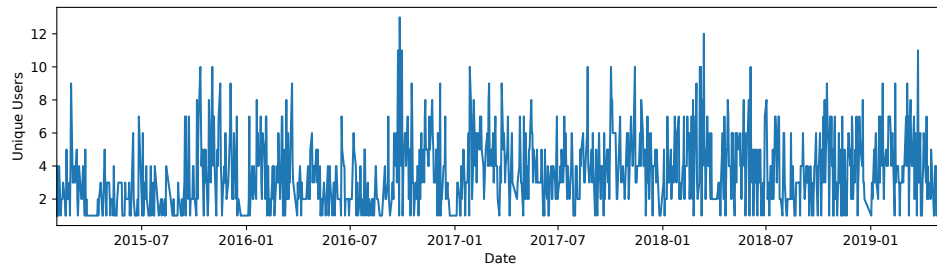
To provide an overview of the usage patterns in the context of our scheduling architecture, we study logs captured on e-Channels from 1 February 2015 (the initial release of our scheduling architecture) to 13 May 2019 (most recent snapshot of the usage logs). Throughout this period, we observed a mean of 4.17 logins per day (median: 3, standard deviation: 11.29) and 2.52 unique users (median: 2, standard deviation: 2.46) [Figure 6a], suggesting that users often accessed e-Channels more than once, e.g. to revise previously specified display or content constraints. We observed users frequently accessing e-Channels to adjust both channel- (Figure 6b) and display- (Figure 6c) related attributes such as content scheduling constraints. In particular, we observed a mean of 2.19 display-related activities per day (median: 0, standard deviation: 6.35) and 2.45 channel-related activities per day (median: 1, standard deviation: 3.30).

Figure 6d highlights that while a small number of users interact with the system on a regular basis many interact only occasionally – and that all users often have extended periods between interactions. This confirms findings reported in [19] and highlights the need for simple UIs for Stage 1 that support non-expert, occasional users.

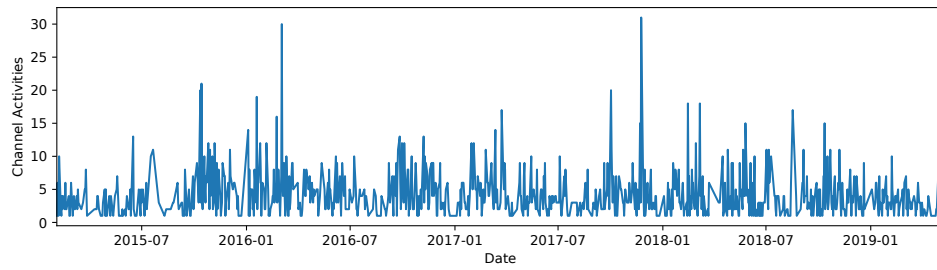
To examine typical display and channel configurations we took a snapshot of the e-Campus deployment on 10th May 2019 that consists of all the CDSs for each display (total number of displays online that day: 81). We observe a mean of 11.16 channels per display (median: 14, standard deviation: 8.02) and a mean of 9.66 content items (median: 4, standard deviation: 14.12) per channel. A mean of 119.83 content items (median: 187.50, standard deviation: 96.83) are scheduled for any given display – emphasising the need for efficient and accurate content scheduling mechanisms. The content types scheduled on displays consist of 8,983 images, 488 videos, 128 PDFs, 86 URLs pointing at websites and 260 incompatible content items (e.g. Microsoft PowerPoint and Microsoft Word files that cannot be shown) – as an indication of growth, our 2011 analysis reported a total of 1,796 unique content items in the system over a period of three years [19].

**6.3.2 Stages 2 and 3: Lottery Scheduler.** Our implementation of Stages 2 and 3 (described in Section 5.2 and 5.3 respectively) has been in use on all e-Campus displays since 2015. Using PHEME [64], our analytics platform deployed in 2014, allowed us to capture insights into the number of filtering content scheduling decisions conducted in the context of our deployment and following analysis has been first reported in [61]. The lottery scheduler typically conducts approximately 3.80 content scheduling decisions per second. Between August 2017 and August 2018, we captured a daily mean of 16,411 content scheduling decisions (median: 16467, standard deviation: 5291.78) – the high standard deviation is likely caused by an increased number of displays throughout the period considered and the fact that a number of displays are turned off during weekends.

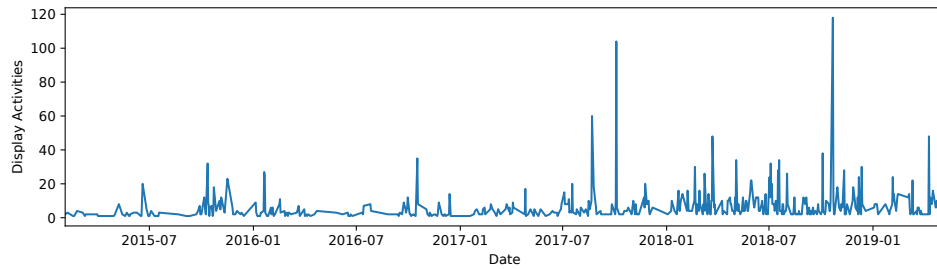
Table 5 shows the number of content items scheduled by the lottery scheduler and reported to PHEME and their playback durations. In total, the lottery scheduler showed 2376 unique content items that were visible on displays for a mean of 21.60 seconds (standard deviation: 25.97). Typical content types include images, video streams, PDFs, videos, websites and other content item. E-Campus displays have been configured to show static content for a duration between 15-20 seconds and videos for the entire duration of the video file – leading to the



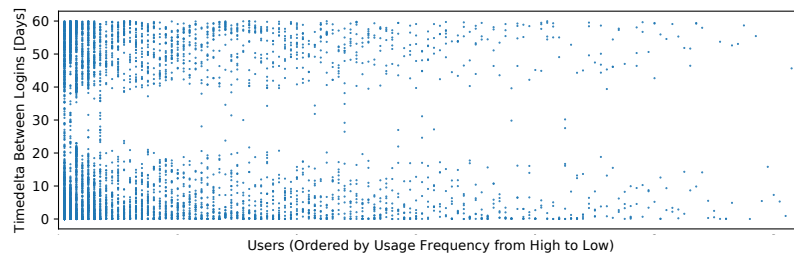
(a) Unique users per day.



(b) Channel-related activities per day.



(c) Display-related activities per day.



(d) Time deltas between log-ins to e-Channels per user.

Fig. 6. Usage logs from e-Channels.

Table 5. Number of reported content requests to PHEME [64] between August 2017 and August 2018 by content type (adapted from [61]).

Content Type	Request Count	Mean Duration [s]	Standard Deviation [s]
All	2376	21.60	25.97
Image	2098	16.77	4.11
Video	155	87.30	67.87
PDF	54	16.07	1.66
Website	53	16.47	6.98
Stream	9	80.33	112.97
Other	7	19.44	3.34

higher mean play time and standard deviation for different video lengths. Static content items (images and PDFs) have a low standard deviation of the playback time reflecting the configuration of e-Campus displays.

## 7 DISCUSSION

Scheduling software is critical in allowing displays to meet the needs of their deployments, making the fundamental decision about which of the available content items to be shown on which display (or part thereof), when and in response to what. Prior attempts to understand requirements for, and develop, these schedulers has typically occurred on a case-specific basis and has ignored the emerging challenges associated with future open display networks [26]. We argue that the complexity of new display deployments requires a thorough exploration of the scheduling problem. In this paper we aim to provide that exposition; considering the scheduling problem through from requirements, to conceptual architecture, and execution.

### 7.1 Requirements

Section 3 provides eleven requirements intended to describe the complex needs of current and envisaged pervasive display networks. These requirements are substantially more expansive than those expressed in previous research. As noted earlier, previous requirements have often been targeted to a specific deployment context (one notable exception being those from Storz et al. [83] who reflected on their experiences with three heterogeneous technology probes). By contrast, we deliberately sought out these previous requirement sets, others' experiences developing and deploying displays (Section 2.2), a recent literature survey [86], recent use cases for future display networks [22, 26], and our own extensive experiences of fifteen years operating the world's largest research testbed for display networks.

We compare our eleven requirements with three representative sets from the literature: Storz et al.'s [83] (2006) requirements based on experiences with three technology probes, used to inform the design of a computational model and API for the early e-Campus software; Elhart et al. [31]'s (2014) requirements for scheduling interactive and concurrent applications in display networks, and Bushman and Labrinidis's [8] (2019) requirements for a utility-based scheduler to drive a set of bus stop displays. Further discussions of requirements are present in the literature, but many take the form of general discussion rather than itemized requirements (e.g. [30]). In light of these three requirement sets, we note the following similarities and differences. Firstly, we identify significantly more requirements than previous work – Storz et al., Elhart et al., and Bushman and Labrinidis list five, three and four requirements respectively. This is reflective both of our attempt to generalise beyond a single deployment setting and an increasingly complex problem space (multiple stakeholders, more content, interactivity, personalization, context...). Secondly, we identify specific differences in requirement coverage. Considering in turn each of our four broad classes of requirement, we find that:

**Traditional Signage** Prior requirements elicitation have significant overlap with our three "Traditional Signage" requirements. All three of our identified prior requirement sets recognize a need for temporal constraints (R3). Storz et al. [83] additionally recognize a need to schedule across multiple displays (R2), and Elhart et al. identify a requirement to schedule concurrent applications "not only on one or more displays but also on one or more "screen zones"" [31, p3]. However, metadata-based scheduling (R1) is not present in any of these preceding requirement sets.

**Interactivity, Personalization, and Priorities** Coverage of these is also largely covered in the union of existing requirements sets – Storz et al. highlight the importance of support for "the rapid introduction of interactive content" [83, p9] (R4), and Elhart et al. [31] also note a need to support interactive content, whilst Bushman and Labrinidis [8] note a need to prioritize content scheduling in accordance with the value attached to it by passersby (R6). Neither Storz nor Bushman identify requirements for personalisation (R5), presumably as this had not been a focus for their deployments.

**Context** Requirements related to audience and environmental context feature only minimally in these two previous requirements sets: Storz et al. [83] mention responding to user presence (R7) as part of their requirement for interaction, and Bushman and Labrinidis's [8] motivation for being able to "handle deadline-driven content" related to the detection of proximate buses (an example of environmental sensing, R8).

**Openness and Control** Our final three requirements also have relatively limited coverage in these existing sets. Elhart et al. [31] are motivated by the introduction of novel applications as content for pervasive displays – the requirement to support variety in content and sources (R9) is therefore implied but not explicitly stated. Bushman and Labrinidis [8] include a need to deal with short-notice changes to the content pool. However, they make no mention of novel applications, nor of the sourcing of content items (and corresponding scheduling requests) from diverse sources (R9); moderation (R10) and openness to new scheduling policies (R11) are also omitted. Despite the relative recency of visions for open display networks (2012), Storz et al.'s 2006 requirements anticipate a need for "numerous independently developed domain specific schedulers" [83, p9] – something not too far from this final requirement to "support the introduction of new scheduling policies from multiple stakeholders" (R11).

Storz et al. [83] do include one requirement that has no equivalent in our requirement set; however, this requirement is focused on the logistics of managing hardware and data, and not any part of the scheduling problem itself. We therefore consider this to be a product of Storz et al.'s intention to use the requirements in the design of a complete scheduling and playback solution, rather than an omission within the present requirement set.

The above comparison suggests that our requirements extend but do not contradict prior understanding of the scheduling in pervasive displays. It also provides clear evidence of an increasingly complex scheduling problem, indicating that *emerging display platforms will require new forms of schedulers that go beyond those that are currently available*. Further, given the complexity of many recent scheduling systems, and the evident lack of coverage for the articulated requirements (see Table 2), we argue that *implementing new schedulers that provide complete coverage of the specified requirements is becoming an insurmountable barrier for individual research teams*. Instead, we suggest that *subdivision of the scheduling problem provides more accessible entry points for research* allowing the research community to co-operatively innovate to meet the scheduling needs of both current and emerging display systems. We believe that this paper provides that subdivision and vocabulary, marking a critical turning point in scheduling research for pervasive displays.

## 7.2 Architecture

In response to the requirements articulated in Section 3, we have presented a novel multi-stage architecture that characterises three distinct phases in the scheduling of content for pervasive digital displays. Unlike prior research into display scheduling [8, 30, 31, 62, 70, 78, 83, 86], we do not prescribe a specific algorithm or formula for optimization or constraint resolution. Instead, we seek to subdivide the problem of scheduling into three constituent challenges that are distinct from any algorithm or approach. This seemingly small step has significant impacts for the development of signage systems. By teasing apart problems that have previously been considered a single area of challenge, we encourage researchers to focus their attentions rather than attempt the simultaneous resolution of the entire scheduling problem (thus avoiding the development of highly complex mathematical models whose application to real-world systems is unclear).

Our architectural model identifies three discrete processes in determining what items should be scheduled to any given display resource(s): (1) High Level Goal Setting, (2) Filtering, and (3) Selection. Our description focuses on the inputs and outputs at each stage, and the intent to be met when transforming inputs to the specified outputs. Our architecture explicitly avoids advocating in favour of any given implementation of these stages. Instead, we leave this deliberately open with the explicit goal of encouraging further research and innovation. We see our architecture akin to design patterns in software engineering: providing a new vocabulary for discussing challenges and solutions for pervasive display scheduling.

Finally, we aim for a degree of completeness previously absent from discussions of pervasive display scheduling. Whilst we acknowledge that it is not the case that our stages represent three completely new problems in scheduling, prior research has typically sought to address one, or perhaps two of our articulated stages. For example, constraint-based approaches [30, 31] focus largely on the articulation of timing, ordering and other constraints (Stage 1) and formal expressions that enforce these (Stage 2), preventing items from being shown when their constraint would prohibit it. By contrast, auctions [3, 5, 67–69, 75] and utility-based approaches [8] focus the expression of overall values (Stage 1) and processes for maximising delivery on those values (Stage 3).

Given the above, we suggest that the presented architecture has clear implications for the research community. We argue that *pervasive display researchers should consider scheduling as an end-to-end problem in which High Level Goal Setting, Filtering, and Selection have equal importance*. Omitting any stage will result in an incomplete scheduling solution. However, we also encourage a separation of concerns, in which *an established set of interfaces are needed to encourage individual developers to focus their attention on innovating within the constructs of the three articulated stages rather than trying to address all stages with a single approach*. Further, not only is it the case that any given scheduling stage is unable to meet all of the identified requirements, it is also true that no stage is able to address any single requirement completely. We therefore note that *developers working on solutions for High Level Goal Setting, Filtering or Selection must concern themselves equally with a wide range of scheduling requirements*.

## 7.3 Execution

Our implementation and deployment of the proposed scheduling architecture allows us to realise the three-stage model, demonstrating its viability as a scheduling solution that can execute and operate over real world problems. In so doing we note the following.

**Suitability for a thick-client configuration** e-Campus and Yarely follow a thick-client design in which each display node is a fully-resourced processing and rendering unit that can operate autonomously even during extended periods of disconnection. Some centralised remote services are still provided (most notably, those associated with Stage 1, High Level Goal Setting), but execution of both Stages 2 and 3 take place at the display node itself. Thus the suitability of the architecture for thick-client systems is evident. In alternative implementations such as a fully cloud-based signage system (e.g. [81, 87]), placement of all

three scheduling stages is also clear – they must be supported server-side in the cloud. However, for hybrid approaches, careful consideration is needed to determine the appropriate location for implementation of each stage. This placement is likely to take into account a range of concerns including deployment and maintenance costs, autonomy and resilience, performance, and responsiveness.

**Just-in-Time vs. Lookahead Scheduling** The Yarely scheduler uses just-in-time decision making to determine the most appropriate item to be shown in a given moment; this is a stark contrast with systems such as Sony Ziris [82] whose playlists are dictated ahead of time, and with constraint-based systems that treat constraints and available resources as a complex equation to be solved. Playlists, constraints, and other lookahead systems are often an attractive choice for cloud-based systems (pushing a simple set of playback instructions to display nodes), but these also struggle with dynamic schedules (e.g. as a result of context-dependence or interaction). By contrast, just-in-time scheduling can be responsive to the needs of dynamic schedules and can reduce or remove the mathematical challenge associated with optimising over an extended period. We believe that both approaches are compatible with the presented architectural model but that the interactive, personalised nature of future signage systems dictates that look-ahead horizons are likely to become increasingly limited.

**Flexibility of Lottery Scheduling** As noted above, our architecture is implementation-agnostic, allowing developers to combine their preferred scheduling approaches (constraints, auctions, utility-functions, lottery etc.) at relevant stages. In our specific case, we used a lottery scheduler at Stage 3. Whilst this approach lacks the determinism of a utility function, our results with a multi-year deployment indicate that it works well for meeting the day-to-day needs of a moderate-scale deployment (~100 displays). Moreover, our lottery-based system has proved highly adaptable to new scheduling requirements suggesting that lottery scheduling may provide a suitable starting point for a wide-range of new scheduling policies and, crucially, offers an architecturally elegant way of integrating and arbitrating between multiple competing or complimentary policies.

**Performance Considerations** Our current implementation of Yarely largely relies on components actively polling for changes to both CDSs and context. With regards to scheduling, this means that a new scheduling decision is prompted at regular intervals, regardless of whether current playback has expired, or a new update has actually been received. The primary implication of this decision is a need for an efficient scheduling process that will not consume excessive resource (time, processing). Our benchmarking data (Section 6.2) indicates that for an average display in our deployment, the combined cost of Stage 2 and Stage 3 scheduling is between 152 and 216 milliseconds, meaning that there is no discernible impact on overall system performance. This suggests that for future signage deployments, implementations based on our architecture are viable on typical display clients.

**Support for Occasional Use** Our implementation provides a number of important lessons for designers of future signage systems. Perhaps most crucial is that (at least in our deployment) many users interact with the scheduling system only sporadically. During our ten years of engagement with e-Channels, users have consistently exhibited the same engagement patterns suggesting that the design of scheduling UIs must be optimised for non-expert users.

In light of the above, we suggest that, although platform-specific, our implementation and evaluation provide some valuable takeaways for the broader pervasive displays research community. Firstly, we extend existing considerations when determining the appropriate partitioning of signage software. Our multi-stage model is compatible with thick clients, fully cloud-based platforms, and hybrid approaches. However, *design of hybrid approaches needs to take into account placement of user interfaces and algorithms to address each architectural stage*. Secondly, we suggest that the separation of concerns presented in our architecture provides *a new opportunity for the application and combination of multiple scheduling techniques*, and that *lottery scheduling offers an architecturally*

*elegant way of allowing multiple competing or complimentary policies to co-exist in a single deployment.* Further, benchmarking and real-world deployment experiences of our lottery scheduling implementation provides evidence that both the overall architecture, and our specific realisation are well-suited to current signage needs. Finally, those same deployment experiences confirm prior observations that, for mid-scale digital signage with multiple non-expert stakeholders, *complex user interfaces for High Level Goal Setting are unlikely to be appropriate*; users engage too infrequently to build up expert knowledge.

#### 7.4 Future Research Directions

The scheduling architecture described in this paper is designed to provide a flexible platform upon which new scheduling approaches can be implemented. Our experiences with using this platform in the context of our research testbed have highlighted a number of future research directions for content scheduling in pervasive display networks.

**Resolving competing scheduling requirements for multiple stakeholders.** In striving for an open approach to pervasive display networks there is a clear need to support content and scheduling requirements from a wide range of sources (and hence stakeholders). While the proposed architecture provides support for this it does not provide guidance on *how to resolve competing scheduling requirements* originating from different stakeholders. For example, what should happen when two viewers approach a display with conflicting personalization requirements? Many personalization systems ignore this issue (e.g. [23]) but as the scale of personalization and openness increases, understanding how competing scheduling requirements can be addressed will become more critical. One particularly interesting avenue to explore in this domain is the creation of scheduling systems that are able to accommodate multiple competing requests by multiplexing content in multiple dimensions including space (e.g. by dividing a screen’s real estate), time (e.g. showing each content item for a small time slot) or by distributing competing requests across a network.

**Analytics driven scheduling.** Contemporary scheduling systems use essentially fixed schedules with, in advanced systems, some minimal tailoring of content or scheduling decisions based on the presence of viewers in particular demographic categories as identified by video analytics [46]. However, recent research [63] suggests that there are opportunities to combine much richer sets of analytics data to drive scheduling decisions. For example, combining analytics of human movements through a shopping mall with signage analytics could enable multiple new types of scheduling including selecting content items specifically to influence mobility patterns — perhaps to guide viewers to specific stores with sales events or to slow down or re-route viewers for safety reasons.

**Viewer-centric scheduling.** In the long-term we predict that scheduling in pervasive display systems will undergo a fundamental revolution — shifting from display-centric scheduling to *viewer-centric* scheduling. Such a change would see scheduling constraints transition from statements such as “show this content item ten times on a specific display” to “make sure that viewer Jo sees this content item at least ten times as they walk home”. In terms of the architecture proposed in this paper we believe that such a change would primarily impact on Stage 1, though clearly supporting viewer-centric analytics would also increase the level of support required for personalized content in Stages 2 and 3.

## 8 CONCLUDING REMARKS

Pervasive displays have played an important role in visions of ubiquitous computing since Weiser’s early descriptions of tabs, pads and boards [91]. As research has matured, such displays have become a pervasive feature of public spaces: shopping malls, transport hubs, offices, and healthcare buildings are all densely-populated with



displays in a wide variety of forms. Research into pervasive displays has spanned decades, but has gained momentum in recent years with the emergence of novel technologies and new visions for open display networks [26]. However, research on the fundamental issue of how to schedule content onto these devices is fragmented – although almost all display systems have relied on a scheduling technique for selecting from available content, these techniques are very rarely the subject of any serious consideration. In this paper, we turn our attention directly to this problem of deciding which content item to schedule for presentation in a given target context (including time, display, audience and environment). This is a particularly challenging question as it requires addressing the concerns of a large number of stakeholders (including display owners, content producers, and multiple viewers and bystanders), understanding the context of the display and viewers, selecting or generating content items from a very large pool and determining appropriate schedules for both individual displays and across entire display networks.

Our paper makes three distinct contributions. The **primary research contribution** of this paper is **a novel multi-stage scheduling architecture for scheduling content in pervasive display systems**. The key insight in this architecture is to explicitly separate the scheduling problem into three distinct stages: *High Level Goal Setting*, content *Filtering* and item *Selection*. In so doing, we help reduce the complexity of the problem, allowing developers isolate issues specific to each stage and to utilise different scheduling techniques to overcome them. Perhaps more crucially, this architecture also provides a new vocabulary for describing challenges and approaches to scheduling in pervasive displays. Akin to design patterns in software engineering, we believe that this new vocabulary is an important step in fostering new research and innovation in a previously under-explored domain.

Our second and third contributions emerge as part of our efforts to achieve the first. We **provide comprehensive description of requirements for scheduling in pervasive display networks** based on an *extensive survey of deployments reported in the literature*, a set of future-looking *use cases* that have emerged in recent papers advocating a transition to open display networks, and our own experiences developing and deploying systems in-the-wild. These eleven requirements capture the scheduling functionality necessary to support a broad range of scheduling applications including *traditional signage*, *interactivity*, *context-dependence* and *open networks*. While requirements for pervasive displays have been presented in the past, we believe that this is the first attempt to produce a comprehensive set that addresses concerns arising from the diverse set of applications and settings to which pervasive displays are now deployed. Further, we provide a **description of the design, implementation and evaluation of key elements of this architecture in the context of a world's largest testbed for digital signage**. This implementation highlights the flexibility of the proposed architecture. Taking place entirely within the existing constraints of e-Campus software, substantial re-engineering was *NOT* required. Instead, we can easily map existing elements of software to our conceptual architectural model. We believe that this feature is not a peculiarity of the e-Campus network, but is a reflection of the universal nature of our model. Evaluation of this implementation shows that the architecture supports the requirements identified, and that a performant implementation is both feasible and deployable.

## ACKNOWLEDGMENTS

This research was partially funded through the Future and Emerging Technologies (FET) programme within the 7th Framework Programme for Research of the European Commission, under FET grant number: 612933 (RECALL), and by the UK EPSRC under grant numbers EP/N023234/1 (PETRAS) and EP/N028228/1 (PACTMAN).

## REFERENCES

- [1] Adspace Networks, Inc. 2014. Adspace Digital Mall Network. <http://www.adspacenetworks.com/> [Last accessed: April 2014]. (2014).
- [2] Paula Alaves, Alexander Samodelkin, Esa Jääskelä, Riku Tanskanen, Bo Li, Timo Ojala, and Hannu Kukka. 2016. Campus Knights: Situated Pervasive Display As a Window into Pseudo-immersive Game World. In *Proceedings of the 5th ACM International Symposium on Pervasive Displays (PerDis '16)*. ACM, New York, NY, USA, 132–139. <https://doi.org/10.1145/2914920.2915011>

- [3] Florian Alt, Moritz Balz, Stefanie Kristes, Alireza Sahami Shirazi, Julian Mennenöh, Albrecht Schmidt, Hendrik Schröder, and Michael Goedicke. 2009. Adaptive User Profiles in Pervasive Advertising Environments. In *Proceedings of the European Conference on Ambient Intelligence (AmI '09)*. Springer-Verlag, Berlin, Heidelberg, 276–286.
- [4] F. Alt, J. Muller, and A. Schmidt. 2012. Advertising on Public Display Networks. *Computer* 45, 5 (May 2012), 50–56. <https://doi.org/10.1109/MC.2012.150>
- [5] Christine Bauer, Paul Dohmen, and Christine Strauss. 2011. Interactive digital signage-an innovative service and its future strategies. In *Proceedings of the 2011 International Conference Emerging Intelligent Data and Web Technologies (EIDWT)*. IEEE, 137–142.
- [6] Sara A. Bly, Steve R. Harrison, and Susan Irwin. 1993. Media Spaces: Bringing People Together in a Video, Audio, and Computing Environment. *Commun. ACM* 36, 1 (January 1993), 28–46.
- [7] Harry Brignull and Yvonne Rogers. 2003. Enticing People to Interact with Large Public Displays in Public Spaces. In *INTERACT*.
- [8] Kristi Bushman and Alexandros Labrinidis. 2019. Utility-based Scheduling for Public Displays with Live Content. In *Proceedings of the 8th ACM International Symposium on Pervasive Displays (PerDis '19)*. ACM, New York, NY, USA, Article 11. <https://doi.org/10.1145/3321335.3324940>
- [9] Jorge C. S. Cardoso and Rui José. 2009. A Framework for Context-Aware Adaptation in Public Displays. In *On the Move to Meaningful Internet Systems: OTM 2009 Workshops*, Robert Meersman, Pilar Herrero, and Tharam Dillon (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 118–127.
- [10] Keith Cheverst, Alan Dix, Daniel Fitton, Chris Kray, Mark Rouncefield, Corina Sas, George Saslis-Lagoudakis, and Jennifer G. Sheridan. 2005. Exploring Bluetooth Based Mobile Phone Interaction with the Hermes Photo Display. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services (MobileHCI '05)*. ACM, New York, NY, USA, 47–54. <https://doi.org/10.1145/1085777.1085786>
- [11] Keith Cheverst, Alan J Dix, Dan Fitton, Christian Kray, Mark Rouncefield, George Saslis-Lagoudakis, and Jennifer G Sheridan. 2005. Exploring Mobile Phone Interaction with Situated Displays.. In *PERMID*. 43–47.
- [12] Keith Cheverst, Faisal Taher, Matthew Fisher, Daniel Fitton, and Nick Taylor. 2012. *The Design, Deployment and Evaluation of Situated Display-Based Systems to Support Coordination and Community*. Springer Berlin Heidelberg, Berlin, Heidelberg, 105–124. [https://doi.org/10.1007/978-3-642-27663-7\\_7](https://doi.org/10.1007/978-3-642-27663-7_7)
- [13] Keith Cheverst, Nick Taylor, Mark Rouncefield, Areti Galani, and Christian Kray. 2008. The Challenge of Evaluating Situated Display based Technology Interventions Designed to Foster a Sense of Community. In *Proceedings of Ubiquitous Systems Evaluation Workshop at the 10th ACM International Conference on Ubiquitous Computing (UbiComp '08)*. ACM, New York, NY, USA.
- [14] Elizabeth F. Churchill, Les Nelson, Laurent Denoue, and Andreas Girgensohn. 2003. The Plasma Poster Network: Posting Multimedia Content in Public Places. In *Proceedings of the 9th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT '03)*. IOS Press.
- [15] Elizabeth F. Churchill, Les Nelson, Laurent Denoue, Paul Murphy, and Jonathan Helfman. 2003. The Plasma Poster Network. In *Public and Situated Displays*. Springer, 233–260.
- [16] Sarah Clinch. 2013. Smartphones and Pervasive Public Displays. *Pervasive Computing, IEEE* 12, 1 (2013), 92–95.
- [17] Sarah Clinch. 2013. *Supporting User Appropriation of Public Displays*. Ph.D. Dissertation. Lancaster University.
- [18] Sarah Clinch, Nigel Davies, Adrian Friday, and Graham Clinch. 2013. Yarely – A Software Player for Open Pervasive Display Networks. In *Proceedings of the 2013 International Symposium on Pervasive Displays (PerDis '13)*. ACM, New York, NY, USA, 25–30.
- [19] Sarah Clinch, Nigel Davies, Adrian Friday, and Christos Efstratiou. 2011. Reflections on the Long-term Use of an Experimental Digital Signage System. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*. ACM, New York, NY, USA, 133–142. <https://doi.org/10.1145/2030112.2030132>
- [20] Sarah Clinch, Nigel Davies, Thomas Kubitzka, and Adrian Friday. 2014. Ownership and Trust in Cyber-Foraged Displays. In *Proceedings of The 3rd ACM International Symposium on Pervasive Displays (PerDis '14)*. ACM, New York, NY, USA, 168–173. <https://doi.org/10.1145/2611009.2611010>
- [21] Sarah Clinch, Mateusz Mikusz, Miriam Greis, Nigel Davies, and Adrian Friday. 2014. Mercury: An Application Store for Open Display Networks. In *Proceedings of UbiComp '14*. ACM, New York, NY, USA, 511–522.
- [22] Nigel Davies, Sarah Clinch, and Florian Alt. 2014. *Pervasive Displays: Understanding the Future of Digital Signage*. Morgan & Claypool Publishers, Williston, VT. USA. 1–128 pages.
- [23] Nigel Davies, Adrian Friday, Peter Newman, Sarah Rutledge, and Oliver Storz. 2009. Using Bluetooth Device Names to Support Interaction in Smart Environments. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*.
- [24] Nigel Davies, Marc Langheinrich, Sarah Clinch, Ivan Elhart, Adrian Friday, Thomas Kubitzka, and Bholanathsingh Surajbali. 2014. Personalisation and Privacy in Future Pervasive Display Networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2357–2366. <https://doi.org/10.1145/2556288.2557287>
- [25] Nigel Davies, Marc Langheinrich, Sarah Clinch, Adrian Friday, Ivan Elhart, Thomas Kubitzka, and Bholanathsingh Surajbali. 2014. Personalisation and Privacy in Future Pervasive Display Networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2357–2366.

- [26] Nigel Davies, Marc Langheinrich, Rui José, and Albrecht Schmidt. 2012. Open Display Networks: A Communications Medium for the 21st Century. *Computer, IEEE* 45, 5 (May 2012), 58–64.
- [27] Destination Media. 2014. Gas Station TV. [www.gstv.com](http://www.gstv.com) [Last accessed: May 2014]. (2014).
- [28] M. Eaddy, G. Blasko, J. Babcock, and S. Feiner. 2004. My own private kiosk: privacy-preserving public displays. In *Eighth International Symposium on Wearable Computers*, Vol. 1. 132–135. <https://doi.org/10.1109/ISWC.2004.32>
- [29] Ivan Elhart. 2015. *Scheduling Interactive and Concurrently Running Applications on a Network of Open Public Displays*. PhD thesis. Università della Svizzera italiana (USI).
- [30] Ivan Elhart, Marc Langheinrich, Nigel Davies, and Rui José. 2013. Key Challenges in Application and Content Scheduling for Open Pervasive Display Networks. In *Adjunct Proceedings of PerCom'13: Works-in-Progress*.
- [31] Ivan Elhart, Marc Langheinrich, Nemanja Memarovic, and Tommi Heikkinen. 2014. Scheduling Interactive and Concurrently Running Applications in Pervasive Display Networks. In *Proceedings of PerDis '14*. ACM, New York, NY, USA.
- [32] Ivan Elhart, Nemanja Memarovic, Marc Langheinrich, and Elisa Rubegni. 2013. Control and scheduling interface for public displays. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication - UbiComp '13 Adjunct*. ACM Press, New York, New York, USA, 51–54. <https://doi.org/10.1145/2494091.2494106>
- [33] Joe Finney, Stephen Wade, Nigel Davies, and Adrian Friday. 1996. Flump: The FLEXible Ubiquitous Monitor Project. In *Cabernet Radicals Workshop*.
- [34] Claude Fortin, Carman Neustaedter, and Kate Hennessy. 2014. Posting for community and culture: considerations for the design of interactive digital bulletin boards. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. ACM Press, New York, New York, USA, 1425–1434. <https://doi.org/10.1145/2556288.2556970>
- [35] Adrian Friday, Nigel Davies, and Christos Efstratiou. 2012. Reflections on Long-Term Experiments with Public Displays. *Computer, IEEE* 45, 5 (May 2012), 34–41.
- [36] Kit Galloway and Sherrie Rabinowitz. 1980. Hole-In-Space. <http://www.ecafe.com/getty/HIS/> [Last accessed: October 2012]. (1980).
- [37] Kazjon Grace, Rainer Wasinger, Christopher Ackad, Anthony Collins, Oliver Dawson, Richard Gluga, Judy Kay, and Martin Tomitsch. 2013. Conveying interactivity at an interactive public information display. In *Proceedings of the 2nd ACM International Symposium on Pervasive Displays*. ACM, New York, NY, USA, 19–24.
- [38] Miriam Greis, Florian Alt, Niels Henze, and Nemanja Memarovic. 2014. I can wait a minute: uncovering the optimal delay time for pre-moderated user-generated content on public displays. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. ACM Press, New York, New York, USA, 1435–1438. <https://doi.org/10.1145/2556288.2557186>
- [39] John Hardy and Jason Alexander. 2012. Toolkit Support for Interactive Projected Displays. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia (MUM '12)*. ACM, New York, NY, USA, Article 42, 10 pages. <https://doi.org/10.1145/2406367.2406419>
- [40] John V Harrison and Anna Andrusiewicz. 2003. Enhancing digital advertising using dynamically configurable multimedia. In *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*, Vol. 1. IEEE, 1–717.
- [41] Tommi Heikkinen, Tomas Lindén, Timo Ojala, Hannu Kukka, Marko Jurmu, and Simo Hosio. 2010. Lessons Learned from the Deployment and Maintenance of UBI-hotspots. In *Proceedings of the 4th International Conference on Multimedia and Ubiquitous Engineering (MUE '10)*.
- [42] Frederick S. Hillier and Gerald J. Lieberman. 2010. *Introduction to Operations Research (Ninth International Edition)*. McGraw-Hill, Columbus, OH, USA.
- [43] Marius Hoggenmueller and Alexander Wiethoff. 2015. Blinking Lights and Other Revelations: Experiences Designing Hybrid Media Façades. In *Proceedings of the 4th International Symposium on Pervasive Displays (PerDis '15)*. ACM, New York, NY, USA, 77–82. <https://doi.org/10.1145/2757710.2757725>
- [44] INFOSCREEN Austria, Gesellschaft für Stadtinformationsanlagen GmbH. 2005-2009. INFOSCREEN - your city channel | Infoscreen. <http://www.infoscreen.at/www/homepage.php?lng=EN> [Last accessed: January 2013]. (2005-2009).
- [45] INFOSCREEN NETWORKS PLC. 2011-2012. Infoscreen Networks. <http://www.infoscreennetworks.com/> [Last accessed: February 2013]. (2011-2012).
- [46] Intel Corporation. 2018. Intel AIM Suite. <https://aimsuite.intel.com/>. (2018).
- [47] Shahram Izadi, Harry Brignull, Tom Rodden, Yvonne Rogers, and Mia Underwood. 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)*. ACM, New York, NY, USA, 159–168. <https://doi.org/10.1145/964696.964714>
- [48] Giulio Jacucci, Ann Morrison, Gabriela T. Richard, Jari Kleimola, Peter Peltonen, Lorenza Parisi, and Toni Laitinen. 2010. Worlds of Information: Designing for Engagement at a Public Multi-touch Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2267–2276. <https://doi.org/10.1145/1753326.1753669>
- [49] Nassim Jafarinaiimi, Jodi Forlizzi, Amy Hurst, and John Zimmerman. 2005. Breakaway: An Ambient Display Designed to Change Human Behavior. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1945–1948. <https://doi.org/10.1145/1056808.1057063>

- [50] JCDecaux Group. 2014. JCDecaux UK | No. 1 Outdoor Advertising Company. <http://www.jcdecaux.co.uk/> [Last accessed: May 2014]. (2014).
- [51] E. Douglas Jensen, C. Douglass Locke, and Hideyuki Tokuda. 1985. Time-Driven Scheduling Model for Real-Time Operating Systems. *IEEE*, 112–122.
- [52] Rui José, Nuno Otero, and Jorge C. S. Cardoso. 2014. Dimensions of Situatedness for Digital Public Displays. *Adv. in Hum.-Comp. Int.* 2014 (2014). <https://doi.org/10.1155/2014/474652>
- [53] Rui José, Nuno Otero, Shahram Izadi, and Richard Harper. 2008. Instant Places: Using Bluetooth for Situated Interaction in Public Displays. *Pervasive Computing, IEEE* 7 (2008), 52–57. Issue 4.
- [54] Keith Kelsen. 2012. *Unleashing the power of digital signage: content strategies for the 5th screen*. Routledge.
- [55] Christian Kray, Areti Galani, and Michael Rohs. 2008. Facilitating Opportunistic Interaction with Ambient Displays. In *Workshop on Designing and Evaluating Mobile Phone-Based Interaction with Public Displays at CHI 2008*. ACM, New York, NY, USA.
- [56] Thomas Kubitz, Sascha Thullner, and Albrecht Schmidt. 2015. VEII: A Toolkit for Editing Multimedia Content of Interactive Installations On-site. In *Proceedings of the 4th International Symposium on Pervasive Displays (PerDis '15)*. ACM, New York, NY, USA, 249–250. <https://doi.org/10.1145/2757710.2776806>
- [57] Hannu Kukka, Heidi Oja, Vassilis Kostakos, Jorge Gonçalves, and Timo Ojala. 2013. What makes you click: exploring visual signals to entice interaction on public displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1699–1708.
- [58] Peter Lasinger and Christine Bauer. 2013. Situationalization, the new road to adaptive digital-out-of-home advertising. In *Proceedings of IADIS International Conference e-Society*. 162–169.
- [59] Joseph F. McCarthy, Tony J. Costa, and Edy S. Liongosari. 2001. UniCast, OutCast & GroupCast: Three Steps Toward Ubiquitous, Peripheral Displays. In *Proceedings of the 3rd International Conference on Ubiquitous Computing (Ubicomp '01)*. Springer-Verlag, London, UK, 332–345.
- [60] Nemanja Memarovic, Ivan Elhart, and Marc Langheinrich. 2011. FunSquare: First Experiences with Autopoiesic Content. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia (MUM '11)*. ACM, New York, NY, USA, 175–184. <https://doi.org/10.1145/2107596.2107619>
- [61] Mateusz Mikusz. 2019. *Next generation analytics for open pervasive display networks*. Ph.D. Dissertation. Lancaster University. <https://doi.org/10.17635/lancaster/thesis/572>
- [62] Mateusz Mikusz, Sarah Clinch, and Nigel Davies. 2015. Are You Feeling Lucky?: Lottery-based Scheduling for Public Displays. In *Proceedings of the 4th International Symposium on Pervasive Displays (PerDis '15)*. ACM, New York, NY, USA, 123–129. <https://doi.org/10.1145/2757710.2757721>
- [63] Mateusz Mikusz, Sarah Clinch, and Nigel Davies. 2017. Design Considerations for Multi-Stakeholder Display Analytics. In *PerDis '17 Proceedings of the 6th ACM International Symposium on Pervasive Displays*. ACM, New York, NY, USA. <https://doi.org/10.1145/3078810.3078830>
- [64] Mateusz Mikusz, Sarah Clinch, Rachel Jones, Mike Harding, Chris Winstanley, and Nigel Davies. 2015. Repurposing Web Analytics to Support the IoT. *Computer* 48, 9 (Sep. 2015), 42–49. <https://doi.org/10.1109/MC.2015.260>
- [65] Jörg Müller, Florian Alt, Daniel Michelis, and Albrecht Schmidt. 2010. Requirements and design space for interactive public displays. In *Proceedings of the 18th ACM international conference on Multimedia*. ACM, New York, NY, USA, 1285–1294.
- [66] Jörg Müller, Juliane Exeler, Markus Buzeck, and Antonio Krüger. 2009. ReflectiveSigns: Digital Signs That Adapt to Audience Attention. In *Proceedings of the 7th International Conference on Pervasive Computing (Pervasive '09)*. Springer-Verlag, Berlin, Heidelberg, 17–24.
- [67] Jörg Müller and Antonio Krüger. 2007. How Much to Bid in Digital Signage Advertising Auctions. In *Adjunct proceedings of Pervasive*. Springer-Verlag, Berlin, Heidelberg.
- [68] Jörg Müller and Antonio Krüger. 2007. User Profiling for Generating Bids in Digital Signage Advertising Auctions. In *Adjunct Proceedings of User Modeling*. Springer-Verlag, Berlin, Heidelberg.
- [69] Jörg Müller and Antonio Krüger. 2009. MobiDiC: Context Adaptive Digital Signage with Coupons. In *Proceedings of the European Conference on Ambient Intelligence (AmI '09)*. Springer-Verlag, Berlin, Heidelberg, 24–33.
- [70] Jörg Müller, Antonio Krüger, and Tsvi Kuflik. 2007. Maximizing the Utility of Situated Public Displays. In *Adjunct Proceedings of User Modeling*. Springer-Verlag, Berlin, Heidelberg, 395–399.
- [71] Jörg Müller, Robert Walter, Gilles Bailly, Michael Nischt, and Florian Alt. 2012. Looking Glass: A Field Study on Noticing Interactivity of a Shop Window. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 297–306. <https://doi.org/10.1145/2207676.2207718>
- [72] Jörg Müller, Dennis Wilmsmann, Juliane Exeler, Markus Buzeck, Albrecht Schmidt, Tim Jay, and Antonio Krüger. 2009. Display Blindness: The Effect of Expectations on Attention towards Digital Signage. In *Pervasive Computing*, Hideyuki Tokuda, Michael Beigl, Adrian Friday, A. J. Bernheim Brush, and Yoshito Tobe (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–8.
- [73] Kenton O'Hara, Mark Perry, Elizabeth Churchill, and Daniel Russell (Eds.). 2003. *Public and Situated Displays*. Springer Netherlands. <https://doi.org/10.1007/978-94-017-2813-3>

- [74] Timo Ojala, Hannu Kukka, Tomas Lindén, Tommi Heikkinen, Marko Jurmu, Simo Hosio, and Fabio Kruger. 2010. UBI-Hotspot 1.0: Large-Scale Long-Term Deployment of Interactive Public Displays in a City Center. In *Proceedings of the 2010 Fifth International Conference on Internet and Web Applications and Services (ICIW '10)*. IEEE Computer Society, Washington, DC, USA, 285–294.
- [75] Terry Payne, Ester David, Nicholas R. Jennings, and Matthew Sharifi. 2006. Auction Mechanisms for Efficient Advertisement Selection on Public Displays. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006)*. IOS Press, Amsterdam, The Netherlands, 285–289.
- [76] Peter Peltonen, Antti Salovaara, Giulio Jacucci, Tommi Ilmonen, Carmelo Ardito, Petri Saarikko, and Vikram Batra. 2007. Extending Large-scale Event Participation with User-created Mobile Media on a Public Display. In *Proceedings of the 6th International Conference on Mobile and Ubiquitous Multimedia (MUM '07)*. ACM, New York, NY, USA, 131–138. <https://doi.org/10.1145/1329469.1329487> capturing.
- [77] Michael L. Pinedo. 2008. *Scheduling: Theory, Algorithms, and Systems*. Springer, New York, NY, USA.
- [78] Fernando Ribeiro and Rui José. 2009. Timeliness for dynamic source selection in situated public displays. In *5th International Conference on Web Information Systems and Technologies*.
- [79] Fernando Ribeiro and Rui José. 2013. Smart content selection for public displays in ambient intelligence environments. *International journal of ambient computing and intelligence* 5, 2 (2013), 1–20.
- [80] Scala. 2014. Scala – Digital Signage Software. <http://www.scala.com/> [Last accessed: April 2014]. (2014).
- [81] ScreenCloud Signage. 2019. ScreenCloud Signage. <https://signage.screen.cloud> [Last accessed: August 2019]. (2019).
- [82] Sony Electronics Inc. 2005–2014. Ziris™ Digital Signage Software. <http://pro.sony.com/bbcs/ssr/cat-digital signage/resource.solutions.bbcscoms-assets-cat-digsignagedev-solutions-Ziris.shtml> [Last accessed: April 2014]. (2005–2014).
- [83] Oliver Storz, Adrian Friday, and Nigel Davies. 2006. Supporting Content Scheduling on Situated Public Displays. *Computers & Graphics* 30, 5 (2006), 681–691.
- [84] Ströer Digital Media GmbH. 2013. Infoscreen. <http://www.stroerdigital.de/en/infoscreen/> [Last accessed: January 2013]. (2013).
- [85] Constantin Taivan, Rui José, and Ivan Elhart. 2012. Selection and Control of Applications in Pervasive Displays. In *6th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2012)*.
- [86] Yukinobu Taniguchi. 2018. [Invited Paper] Content Scheduling and Adaptation for Networked and Context-Aware Digital Signage: A Literature Survey. *ITE Transactions on Media Technology and Applications* 6, 1 (2018), 18–29. <https://doi.org/10.3169/mta.6.18>
- [87] Lars Tiede, John Markus Bjørndalen, and Otto J. Anshus. 2013. Cloud Displays for Mobile Users in a Display Cloud. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications (HotMobile '13)*. ACM, New York, NY, USA, Article 12, 6 pages. <https://doi.org/10.1145/2444776.2444793>
- [88] Ville H Tuulos, Jürgen Scheible, and Heli Nyholm. 2007. Combining web, mobile phones and public displays in large-scale: Manhattan story mashup. In *International Conference on Pervasive Computing*. Springer, 37–54.
- [89] Ville Valkama and Timo Ojala. 2011. Stakeholder Value Propositions on Open Community Testbed of Interactive Public Displays. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems (MEDES '11)*. ACM, New York, NY, USA, 107–113.
- [90] Mettina Veenstra, Niels Wouters, Marije Kanis, Stephan Brandenburg, Kevin te Raa, Bart Wigger, and Andrew Vande Moere. 2015. Should public displays be interactive? Evaluating the impact of interactivity on audience engagement. In *Proceedings of the 4th International Symposium on Pervasive Displays*. ACM, New York, NY, USA, 15–21.
- [91] Mark Weiser. 1991. The Computer for the 21st Century. *Scientific American* (September 1991).
- [92] Xiao Emila Yang and Martin Tomitsch. 2014. Designing interactions with pervasive displays for location-based storytelling. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: the Future of Design*. ACM, New York, NY, USA, 133–136.