Orchestrating Multi-Device Presentations with OmniPresent

Maria Husmann, Daniel Huguenin, Matthias Geel, and Moira C. Norrie
Department of Computer Science
ETH Zurich, Switzerland
{husmann,norrie}@inf.ethz.ch, dany.huguenin@gmail.com, matthias.geel@gmail.com

ABSTRACT
Meeting rooms and classrooms are increasingly equipped with more than one screen that could be used for presentations. However, current presentation software provides little support for arranging content across multiple screens beyond mirroring slides. We introduce OmniPresent, a system for orchestrating presentations on multiple devices. OmniPresent offers two modes for authoring these orchestrations. The first mode is suited in particular for legacy, single-device presentations and allows a presenter to orchestrate the presentation on the fly by pinning slides to extra screens. The second mode offers complete freedom in allocating slides to available screens by authoring an orchestration script beforehand. We have evaluated the two modes qualitatively in a user study and report on patterns in content and motivation for distributing presentations across multiple screens.

ACM Classification Keywords
H.5.m Information interfaces and presentation (e.g., HCI): Miscellaneous

Author Keywords
multi-device; presentations; orchestration

INTRODUCTION
Modern presentation environments often offer more than a single screen for presentations. For the audience, making use of only a single screen may not provide the best experience. Using a dual-screen setup has been shown to increase retention of lecture content in an experiment with university students [14]. However, commercial presentation software provides little support for multi-screen presentations beyond simple mirroring of the content. While mirroring may increase the visibility of the presented content for some of the audience, it provides little benefit when the audience already has a good view of an individual screen.

Furthermore, setups in rooms may vary. While some rooms have two screens of the same size, others are equipped with smaller secondary displays or a combination thereof. We have built OmniPresent to experiment with orchestrations in various settings and investigate how presenters would like to use multiple screens in their presentations. OmniPresent contributes a flexible presentation runtime (Figure 1) that supports two orchestration authoring modes. The Pin mode allows a presenter to keep slides visible on extra screens by selecting them from a remote control during the presentation. This approach is well suited to linear presentations that have been built with traditional tools. The Script mode provides more freedom. It allows a presenter to script arbitrary orchestrations and gives them complete control over the assignment of slides to screens. In either mode, OmniPresent adapts the presentation to the available devices, which may not be known to the speaker beforehand or change unexpectedly. While OmniPresent focuses on the presentation runtime, we have also identified the need for a visual authoring tool for multi-screen presentations. We inform the design of such a tool with a qualitative user study of OmniPresent. We found recurring patterns in the orchestrations created by the participants that could be supported by an authoring tool, for example by utilising structural information of a presentation.

BACKGROUND
Researchers have explored diverse approaches to include multiple devices into a presentation. Systems have been built for delivering content to the devices of the audience, for example to allow a remote person to attend a talk or follow a class [11, 21]. While some systems allow the audience to take control over the slides and browse them at their own speed [11], others the flow is controlled by the presenter [21]. Some systems are built not only for remote audiences but also for scenarios where presenter and audience are in the same room [2]. The inclusion of audience devices has been studied in classrooms, in particular [5, 17, 19, 22]. These systems use the audience...
devices not only to push lecture content, but also as a means of interacting with the class, for example through quizzes and polls. ASQ [21] allows the audience to input questions, which are forwarded to the speaker who can decide to answer them on the spot or at the end of the presentation.

While the presenter has little control over the audience devices, some rooms offer more than a single screen that could be used for a presentation. A variety of systems have been built specifically for rooms that are equipped with multiple screens [3, 4, 13, 15, 23]. ModSlideShow [4] links displays into peer-to-peer connected groups. Within a group, slides flow from one screen to the next, for example, when three screens form a group, three successive slides will be shown. The system is aware of the spatial arrangement of the screens and uses gestures to control the flow of the presentation. In contrast, other systems provide tools to specify beforehand how a presentation should be orchestrated, for example by dragging and dropping slides onto a timetable. The screens are either represented as rows [23] or columns [3] with the other dimension used for the steps in the presentation. In each step, content can be assigned to one or multiple screens. Once the timetable has been filled, PreAuthor [23] provides several interesting preview options including a rendered 3D model. As a trade off, this approach requires significant setup steps in the room. Moreover, in these existing systems [3, 23], orchestrations are authored for a specific set of screens and it is unclear if they could handle a change in equipment at runtime, for example due to a technical defect. Furthermore, the exact setup of a room may not always be known in advance or the same talk could be given repeatedly in different settings. Our explicit goal was to cater to these scenarios with a flexible solution that allows a presenter to reuse the same presentation rather than having to update or create a new one for each case. One option is to follow a hybrid approach, like MultiPresenter [13, 15], which allows a presenter to assign slides to screens in a visual authoring tool and to create clippings dynamically during a presentation.

While the above systems build on the concept of linear slideshows, new ways of structuring presentations and including multimedia have been explored. Systems with non-linear navigation [6, 18, 20] allow new ways of linking content and access to the same slide at multiple points in time without having to duplicate it. These concepts have mainly been explored in the setting of single-screen presentations, however, conceptual links between slides provide interesting opportunities for multi-device systems. For example, linked content could be displayed simultaneously.

Instead of using slides, some systems use an infinite, zoomable canvas [8] where a presentation is navigated by moving or zooming a window on the canvas. The approach has also been proposed for multi-display settings due to its suitability for on-the-fly improvisations [12]. Any screen can be updated by adjusting its viewport on the canvas. However, we suspect that this flexibility comes at the price of increased cognitive load on the presenter who has to navigate through the presentation on all the screens in a situation where they might be under stress already. Alternatively, navigation paths could be scripted beforehand for each screen. Intellichalk [1] also combines multiple adjacent screens into an infinite canvas but is conceptually closer to a traditional blackboard. The instructor uses a tablet to write content that will be pushed onto the screens. The focus is on handwritten content, but digital content such as images can also be added. The oldest content is pushed off the screens when space runs out. Recently, SlideSpace [7] successfully combined the slide- and canvas-based approaches by automatically generating a canvas from a hierarchical outline and layout rules. With OmniPresent, we follow a slide-based approach, however, our study reveals that there could be benefits in exploiting references between slides and structural information while authoring orchestrations.

OMNIPRESENT

OmniPresent is a flexible runtime for multi-device presentations. It operates on the concept that screens are assigned roles. The primary role is intended for screens that show the main content. The roles auxiliary_1, auxiliary_2, ..., auxiliary_N are for additional devices that could show supplementary content. Multiple devices can be assigned the same role, but there should be at least one device with the primary role to give a meaningful presentation. Before the presentation, there is a simple setup step where the presenter assigns the roles to the available screens. Screens that are part of the system display a QR code which can be scanned with a mobile device for role assignment. In this paper, we use the term primary screen to denote a screen that has been assigned the primary role and auxiliary screen for screens with an auxiliary role.

OmniPresent comes with a remote control user interface that can be loaded onto a mobile phone (Figure 2), tablet or laptop. With the remote, the speaker can use eyes-free interaction to trigger the next step in the presentation or go back via swipe gestures on a touch area (similar to [16]). Multiple remotes can be used simultaneously, for example in collaborative presentations.
OmniPresent implements two main modes for creating multi-device orchestrations. The **Pin** mode provides a simple way of orchestrating such presentations on the fly. In contrast, the **Script** modes allows arbitrary orchestrations to be crafted before presenting.

### The Pin Mode

The **Pin** mode allows a presenter to pin slides to auxiliary screens while going through their presentation. It was created with basic single-device presentations in mind, as this mode does not require any preparation other than the conversion of the presentation into the OmniPresent format. The presentation is shown on the primary device(s) like a regular presentation. Using a button on the remote control, the speaker can pin the current slide to an auxiliary screen where they remain until they are unpinned (Figure 3). When all screens are occupied, the presenter needs to make space by unpinning a slide. We also experimented with a second strategy where the system overrides the oldest pinned slide automatically. Due to the conceptual division of the devices into primary and auxiliary screens and the assignment of the slides to the auxiliary screens at runtime, this mode requires no knowledge of the room setup beforehand and can be used with any number of screens. When setting up the presentation, the screens only need to be assigned the primary and auxiliary roles.

### The Script Mode

The **Script** mode allows the presenter to author powerful orchestrations. Every transition of every screen can be controlled or pre-defined. Building blocks can be used to reduce the amount of scripting for patterns that we expect to be common. We distinguish two levels of granularity: A **view** corresponds to a slide in a traditional system. Views can optionally contain **incremental steps**, for example list items that appear one after the other. While the presenter could define every orchestration step by specifying screens for every view and every incremental step, we provide the following shortcuts: A **range** of views and their incrementals can be shown sequentially on the addressed screen(s) by specifying the start and end of the range. Two or more views with matching incremental steps can be shown in parallel on different screens using **synchronous build-up**.

Screens can be addressed by their roles and we support basic set operations. For example, a view could be shown on all screens except the **primary** ones by using the expression `all - primary`. The keyword **remaining** addresses all roles that have not been used in a given step. At runtime, the system evaluates the expressions and will display the content on all devices matching the result. If the set is empty, the content will not be shown. This mechanism adds flexibility to the authoring process. Orchestrations can be written without knowing the exact number of screens available by directing the main content to the primary screen and using the auxiliaries for supplementary content that is nice to have rather than crucial to the presentation. Rather than having to update the presentation itself, changes in equipment can be handled by assigning a new screen to an existing role during the presentation setup. If there are fewer screens in a room than roles, the speaker can assign the important roles to the available screens and ensure that the main content is still visible.

### ARCHITECTURE AND IMPLEMENTATION

In this section, we discuss interesting aspects of the architecture and implementation of OmniPresent. First, we look into the OmniPresent runtime and then we explain the format of OmniPresent presentations and their orchestrations.

#### OmniPresent Runtime

OmniPresent has been implemented with web technologies. The runtime consists of three main parts: a central server coordinating the presentation, a screen client, and a remote control client. The server is written in TypeScript, compiled to JavaScript, and runs in Node.js\(^1\), using MongoDB\(^2\) for persistence. It controls the orchestration of a presentation based on the mode that is used (Pin or Script), the configured screens, and the user interaction. When the user pins a slide on their remote, that command is sent to the server, which in turn updates the orchestration and informs all affected devices. Both the screen client and the remote control clients are web-based and run in a browser. This architecture eliminates complex setup steps as no software needs to be installed on screens, since OmniPresent can be loaded on any device with a browser. The remote control client is delivered with a web manifest that allows the application to be installed onto the home screen (on Android devices) and launched without browser controls for an app-like experience. The clients communicate with the

---

1. https://nodejs.org/
2. https://www.mongodb.com/
server over HTTP to access presentation assets such as images. Information on the current orchestration step is pushed from the server using WebSockets.

Presentation and Orchestration Format
Presentations for OmniPresent must be written in its specific XML format. An OmniPresent file consists of two main parts: a set of views and an orchestration. The view content is written in HTML and can be styled with CSS. Incremental steps must be marked with a CSS class. If a presentation uses the Pin mode, the orchestration will be coordinated by the system and no further instructions need to be authored. In the Script mode, the speaker can author each step or use an abstraction, such as range or synchronous build-up. In each orchestration step, the view and optionally an incremental step need to be referenced along with a device role or set of roles that are to show (or hide) the view (Listing 1).

OmniPresent comes with a tool for converting conventional presentations, for example ones written in PowerPoint, into its XML format. Slides need to be exported as numbered images (a feature provided by PowerPoint). The conversion tool then creates a view for each image and arranges them into the right order. The speaker could then give the presentation using the Pin mode or script an orchestration. While this tool helps non-technical users to some extent, we see a clear need for a more visual authoring environment. The XML file format is targeted to be read by machines, and, while technical users can directly author it, it is not suitable for a wider audience. In the next section, we describe our study with presenters that could inform the design of such an authoring tool.

EVALUATION
We conducted a qualitative user study with 15 participants to gather feedback on OmniPresent and to learn more about how presenters would like to use multiple screens in their presentations to inform the design of a visual authoring tool. We recruited participants from our Computer Science department. They were aged between 23 and 39 and four of them were female. All participants had at least some experience of giving presentations. The study consisted of two parts. First, the participants used the Pin authoring mode, while in the second part they used the Script mode for authoring custom orchestrations. After each part, we asked participants to fill in a questionnaire containing a mix of 5-point Likert-type scale and open questions about OmniPresent. We also asked participants for their opinion on using multiple screens for presentations. The study took around 60 minutes in total, with the first part lasting 20 minutes on average. The study was carried out in our lab which was equipped with two projectors and a 32-inch TV (Figure 4). Because the projectors were identical, we labelled them Projector 1 and Projector 2 for reference.

Pin Task
The Pin authoring mode can be used with existing linear presentations. We asked participants to send us a presentation that they had given before and used our tool to convert it into an OmniPresent presentation. We introduced the participants to OmniPresent and explained how they could use the remote control (a Samsung Galaxy S4 provided by us) to pin and unpin slides. Next, participants had time to familiarise themselves with the system. As the main task, we asked participants to step through the first 30 slides of their presentation or to the end if it was shorter than that. Participants did not have to give the actual presentation, however, we asked them to pin and unpin slides where they considered it useful. We noted all instances of pinning and unpinning and asked participants to explain the motivation for their actions. In the study, the system did not automatically override pinned slides when no free screens were available and participants had to manually unpin slides. We opted for this approach to give full control to the participants. The automatic approach would overwrite the oldest pinned slide, but the participant might want to keep that one. During this task, they were sitting in the front row of the room so they had the view of an audience member, rather than standing in the front as a presenter would.

Pin Task Results
On average, participants pinned one in four slides. We consider this a high number and it might be partly due to the novelty effect and the fact that we encouraged participants to use this feature. One participant pinned only a single slide and two participants pinned only two slides. In all three cases, the slides were pinned early in the presentation and remained pinned until the end. Even though these three participants did not use the pinning feature much, they all agreed that the use of multiple screen enriched their presentation. Although none of these presentations had been created with multiple screens in mind, all participants found suitable slides to pin, often many. These observations underline the suitability of the Pin mode, in particular for existing slide decks. Some participants...
mentioned that unpinning slides is tedious and that they would like the system to override slides automatically.

We analysed our notes to find patterns of what and why participants pinned slides and found the following ten purposes. We list them in descending order of occurrence with the number of participants who used them given in brackets.

- **Continuation (7)**: The presentations contained successive slides that represented a direct continuation of each other. The participants pinned the first slide(s) and advanced so that all of the content was visible simultaneously.

- **Reference (7)**: The presentations contained slides with a description or a summary of essential concepts, for example terminology and definitions, theorems, algorithms or hypotheses. The participants pinned such slides to allow the audience to refer to the concepts while presenting the following slides.

- **Overview (7)**: Many participants used slides that gave an overview of a particular situation and then described different aspects of it. For example, one participant pinned a slide showing a paper sketch of an application and then described different aspects of the application.

- **Chapters (7)**: Six participants had included title slides for individual chapters or topics of their presentation and had them pinned while presenting the respective part. These slides would often only contain a title, but there were exceptions. For example, one participant pinned a slide with an open question that they tried to answer in the following slides. Another participant pinned a problem statement and then went on to describe different solutions.

- **Index (5)**: A similar approach for adding structure to the presentation was to pin an outline or index slide in the very beginning and to keep it pinned until the end of the presentation.

- **Title (5)**: Five participants pinned the title slide. Two stated they did so to keep their name visible. One of them was a teaching assistant and considered this useful for students who had questions so they could address her by name.

- **Comparison (5)**: Some people pinned slides in order to compare and contrast them with others. For example, one participant had graphs on two separate slides and pinned one of them to highlight an interesting correspondence when showing the other slide later.

- **Take home messages (3)**: A few people used short summary slides that contained the most important points about a particular topic throughout the presentation, and pinned them to allow their content to sink in.

- **Visibility (2)**: Two people pinned a slide to simultaneously show it on multiple screens in order to improve visibility. One of them pinned a slide with a description of a task that they wanted the audience to solve.

- **Promotion (1)**: One person pinned a slide that contained an invitation to an event related to their presentation. They said that this would allow the audience to copy the date and time of the event. Additionally, people who joined the presentation later would also be able to see the invitation.

### Script Task

In the second task, we explored the *Script* authoring mode. We wanted to investigate how users would orchestrate a multi-device presentation given complete freedom. We provided participants with a set of 28 printed slides about fruit explaining where it grows, why plants bear fruit, and nutritional values. The topic did not require any particular knowledge, but was still complex enough to give a meaningful presentation. The slides contained lists, quotes, charts, and photos. The set of slides included a title slide and an outline slide.

We gave participants time to familiarise themselves with the content of the presentation and then asked them to arrange the slides into a multi-screen presentation. Instead of having our participants write out the orchestration script themselves, we had them lay out the slides on a whiteboard that was divided into columns for each screen (Figure 5). In parallel, we scripted the corresponding orchestration for OmniPresent. We opted against having the participants script their own orchestrations, as we consider this a task that an end-user would not do manually, but rather would use a (visual) authoring tool which we did not have at this point.

Participants did not have to use all slides and were free to present them in any order they deemed suitable. They could re-use a slide multiple times (we had extra copies of each slide on request) and use a pen on the whiteboard, for example to draw transitions from one screen to another. After they had created the presentation, we deployed it with OmniPresent and asked them to step through the presentation to check if it met their expectations.
Script Task Results

On average, participants used 25 out of the 28 slides that we offered them. They made extensive use of the screens. In 12 (out of 15) presentations, all three screens were used in at least half of all steps. On average, participants used at least two (out of three) screens in 90 percent of the steps. Projector 1 was used most to show content and the TV used least. 12 participants placed content on Projector 1 in each orchestration step. The TV was generally used to display auxiliary material. On average, 44 percent of orchestration steps updated only a single screen, 34 percent updated two screens, and 22 percent updated all three screens. When a single screen was updated, that screen was the TV in only 11 percent of cases. When going through the presentation, some of the participants who used such transitions did not notice the update on the TV and thought the system was not reacting. This observation indicates that the projectors were used as primary devices that receive most attention, whereas the TV was used to show auxiliary material.

Again, we analysed the orchestration for patterns and found similar results to the Pin task. As before, the numbers in the brackets denote the number of participants who used the pattern.

- **Reference** (15): All participants at some point showed referenced content together with the slide that made the reference. For example, fruit categories were shown alongside photos of a representative fruit and a definition of the term *phytochemicals* was shown together with a slide that used the term.

- **Comparison** (11): Eleven people showed a slide of the sugar content in table form together with the bar chart visualising the same content.

- **Index** (8): Three participants showed an index slide outlining the presentation during the whole presentation. The remaining five showed it for at least half the presentation. One participant removed the index slide to make room for different content but restored it later.

- **Visibility** (6): Six participants showed the same slide on two or three screens. Only two participants reasoned that this was to improve visibility (for scanning the QR code and reading a quote). Rather, showing a slide on multiple screens was a stylistic choice and mainly used to start and conclude the presentation or increase the impact of a funny slide.

DISCUSSION

Questionnaire data showed that presenters appreciate the opportunity to enrich their presentations with multiple displays. However, our study setup was rather artificial as our participants were not in the situation of having to give a real presentation in front of an audience. During a presentation, many speakers are nervous and a presentation system should avoid adding to their cognitive load. It is unclear, how the *Pin* functionality would be used in a more realistic setting, where a presenter would have to remember during the talk which slides they wanted to pin. However, our study indicates that pinning slides is a simple way of transforming legacy presentations into multi-screen presentations. The functionality could be included in an authoring environment, allowing presenters to choose which slides to pin ahead of time. One participant noted that they would even like to use pinning for scripted orchestrations so they could change the orchestration ad-hoc, for example when answering questions after the presentation.

Multi-screen presentations come with an inherent complexity compared to single-device presentations. As the speaker steps through the presentation, they need some awareness of what is going to be displayed on which screen to create an ideal narrative. As the number of displays with different content increases, we expect the cognitive load to increase as well. In our second task, some participants expressed surprise at some steps and missed some updates completely. To some extent, this can be mitigated by rehearsing a presentation, an opportunity which our participants did not have. Furthermore, a presentation system could provide a preview in a presenter view, similar to current systems displaying the next slide or animation step, highlighting the displays which will update in the next step.

Our study has focused on the presenters. What remains to be studied is how multi-screen presentations are perceived by the audience. Maintaining awareness has been found a challenge in multi-screen groupware systems [9]. The audience could miss content if multiple screens update simultaneously or if their attention is focused on a different screen. Some participants missed an update in our second task when they were stepping through the presentation, even though they had orchestrated it themselves. In multi-device presentations, the speaker would have to carefully guide the audience through the talk to avoid such confusions. Furthermore, audience members typically carry mobile devices that could also be integrated into a presentation [2]. OmniPresent provides support for displaying content on these devices in the Script mode using an *audience* role, however, we did not evaluate this feature in the study in order not to increase its complexity and duration.

CONCLUSION

OmniPresent offers a flexible runtime for multi-device presentations. Our study showed that presenters see value in using multiple screens, when available, to enrich their talks. We found common patterns in the content and motivations observed in the multi-screen presentations of our study. These patterns could inform the design of a visual authoring tool that OmniPresent lacks at this stage. The *Chapter*, *Index*, and *Title* patterns are all connected to the structure of a presentation. An editor with awareness of the hierarchy and structure (such as [7]) could support the presenter in the orchestration. Similarly, the *Reference*, *Comparison*, and *Continuation* rely on a logical connection between multiple slides. Again, an editor could facilitate the creation of these connections and their orchestration. Furthermore, it could allow the presentations to be previewed on different combinations of emulated screens (similar to [10]), so that the speaker could ensure a meaningful presentation even if the equipment is not as expected, for example if a screen is missing.
REFERENCES


8. Lance Good and Benjamin B. Bederson. 2002. Zoomable User Interfaces As a Medium for Slide Show Presentations. Information Visualization 1, 1 (March 2002), 35–49. DOI: http://dx.doi.org/10.1057/palgrave/ivs/9500664


