Abstract: Web accessible display devices from smartphones to large public displays are ubiquitously available. Therefore in many cases, multiple display devices exist within an area. However, the current applications or services are generally designed for a single-user-single-display interaction on a single computer system, unable to fully take advantage of such rich service potentials. In this paper, Pervasive Display Markup Language (PDML), an XML-compliant language is introduced along with Pervasive Display Management System (PDMS) for both distributing, and compounding heterogeneous web content and services across multiple display devices. PDML abstracts a web content or service as a UI component. It describes both the UI component and the displays to define the semantic relations among them for UI separation and distribution. The flexible management mechanism of such applications and services are introduced with the privacy assurance of outdoor display systems.

Key words: pervasive display management, web service management, multiple display interaction, semantic description language, collaboration

1. Introduction

A single-user-single-display interaction has been the most common use case of the computer system. However due to the increasingly affordable and popular deployment of display devices and systems, the digital media environment has evolved into a multiple display environment (MDE) where numerous displays co-exist within an area. In such environment, issues on single-user-multiple-display, or even multiple-user-multiple-display interaction has emerged [10, 14, 17, 18]. Various research issues arose as multiple factors needs to be considered in such environment. Most of the research questions lie on: when, what, where, and how the information needs to be displayed to give the best user experience, in both interaction and information display perspective. The term distributed display environment (DDE) has come up among the researchers in the past few years as the displays are deployed in more distributed manner. People are now accessing multiple displays at home, at work and on the streets either using their personal display devices or public large displays. However research in DDEs investigates purpose-oriented systems considering co-located displays. We use the term pervasive display environment (PDE) where both private and public display systems are networked for multi-purpose. PDE includes any type of network accessible displays to support an open-ended set of applications. Unlike the DDE, it does not necessarily use multiple displays simultaneously. However both co-located and disperse displays are considered in the PDE. Heterogeneous display systems with display sizing from hand-held smart phones to wall-size large display infrastructures may perform as a single coordinated system [11].

Based on such environment described above, we confront a new issue of managing information and services across multiple displays, both indoors and outdoors. How can a single content or application be adaptively rendered to multiple displays for more efficient information delivery? What contents or functions in an application should be selectively distributed according to the characteristics of the pervasive displays? How to assure privacy in terms of service delivery in such privacy fragile environment?

In this paper, we propose system architecture and PDML, an XML-compliant language for describing web documents to be pervasive display environments in which display contents is selectively shown according to the display property and meta-data in the contents. Displays physically located at various places are modeled and described with PDML in order to distribute selective contents on to the display.
Motivating examples are given to describe the benefits of the PDML based information distribution and requirements of semantically distributing information in PDEs are deducted from the scenario. Finally prototype system implementation is introduced, describing a scalable method in PDML architecture for practical deployment. It provides a privacy assured, environment adaptive information distribution for multi-purpose task on both indoors and out in PDEs. Further functions and PDML system capability will be described in the next few chapters.

2. Related Work

2.1 Distributed Display Environment

The earliest research regarding multiple display systems occurred from the 1990’s by Funke et al.[3] They developed an intelligent system of automated window management for dual-monitor system. The attempt was to show or hide the windows by analyzing the window usage time and position in an autonomous way. However the evaluation showed that the automated system often incorrectly showed or hid the window in focus. This research revealed the difficulties of application design in multiple-display system compared to the single-display system.

Perry and O’Hara showed how people use their desk and walls to arrange objects for information display and deducted implications for designing of interfaces for the DDEs [15]. Grudin showed empirical results of the users placing UI tools and widgets on one monitor while placing information on another [5]. Hutching et al. showed that interaction components as MS-Windows TaskBar tend to be less used by multiple-monitor users [7, 8].

The research trend above reveals that the main issue is placement or location of on-screen objects and information. In another words, as the display environment evolves into multiple display environment providing more space of information representation, the opportunities for improvement and innovation in interaction with the information regarding the multiple screens have emerged.

2.2 Context Modeling

In the higher level of research relating multiple displays, there are practical implementation of systems and domain specific context modeling in the area. In order to access variety of mobile devices of heterogeneous capabilities (regarding data input, computing capacity, display format etc) a uniform way of representing device, user and the environment context is important. Research in modeling of context information for pervasive computing domain varies according to the representation extent and their formats.

Christopoulou et al. developed GAS Ontology to describe the semantics of the basic concepts of a ubiquitous computing environment and defined their inter-relations [2]. A service discovery mechanism was proposed. However, such approach may meet its limitation in describing each one of the elements of the PDE. Predefined set of devices and their properties may be limited and changes need to be made by the user each time dynamic changes occur in the environment.

Albert et al. introduced Comprehensive Structured Context Profiles (CSCP) based on RDF to overcome the shortcomings of the Composite Capability/Preference Profiles language (CC/PP) regarding structuring [1]. He suggested some requirements for representation of the context information. He noted the importance of decomposability in managing context information

The display hardware properties (e.g. size, resolution, interface etc) can be described explicitly in most of the languages. However previous approaches may have problems supporting the dynamics of implicit properties (e.g. display usage, relative location etc). Among the various languages, an XML based language was sought to meet the needs for wrapping web applications and services at UI component level as a metadata, along with its applied system for UI level content redirection.

Considering the increasing number of displays with various property and usage (mobile vs static; touch UI vs non-touch UI etc) as well as the pervasive computing technologies are suggesting a new paradigm of information service in the PDE.

3. Motivating Example

The baseline assumption of the following examples is that the information and applications are composed of UI (User Interface) components in order to be provided to the end-user. Generally, what we call “information” on the display is a piece or pieces of text, pictures or video wrapped in a UI. For example, a huge storage of information we access every day is the Internet, and we are able to access its information through a web browser. The pieces of information are encapsulated in the webpage frames to represent information. Through
Figure 1. (a) banking on public display scenario; (b) display mediated multi-display information sharing scenario

the webpage, bi-directional information exchange occurs in an interactive way, which we call a web application. Such application also consists of UI, composed of groups of function modules in relation with one another to function as a whole. Therefore we abstracted the information, contents and applications on pervasive displays as groups of UI components.

One of the example scenarios is accessing bank account from the web for money transfer using a public display. Since the display’s context is public (possibly located outdoors, on the streets or a crowded place), the user might expose private information, such as his name, bank account number or the balance to the public. This may cause a serious issue in terms of privacy. The user should either halt the transaction process or block the screen in order to secure his private information.

In this case, the frames, or the UI component consisting private information on the public display can be removed and rendered on his private display - the smartphone as shown on Figure 1(a). This way, the user can secure his private information from being exposed, as well as additional activities such as password input can be done from his private device.

Another example scenario is sharing information among peers by building a virtual information bridge mediated by shared display. In order to share digital information visually, a display device is required. It can either directly shared through one’s personal display or on public display or projector display by connecting the personal device. However in order to visually share multiple user’s content at once, possibly for a group meeting, the digital information must be gathered to a single digital display output device. This process is usually performed by sending an email, connecting to the target device and transferring the file, or even using a memory device to manually copy and paste. Once the information is gathered, they are displayed or performed one by one. This seriously lowers the natural workflow of information sharing. Moreover, the contents gathered onto the central computer can only be manipulated one by one due to the limited input interface. Therefore by holding their own display device and the content, and only sharing the visual display on the shared display with the access permission of peers’ contents, multiple information can be generated as well as edited in real time as shown on Figure 1(b).

The approach described in the above examples can benefit in the following abstracted domains.

3.1 Collaborative Work

Collaborative work indicates a task that produces a single output by simultaneous contribution from the participants. In the context of simultaneous information sharing and editing, multiple users can either visually provide information for peers as well as merge displayed content by using their own display device, possibly mobile devices as laptops or smartphones. This also means that they can make use of their familiar device. This could enhance work efficiency by making use of multiple devices at once with no men empty handed.

3.2 Reducing Cognitive Overload

A complex arrangement of multiple UI components might hinder users in terms of cognitive overload. In a
single display environment, user can make an arrangement of UI components within a display. Frequently used menus are maximized and less or unused menus are minimized to the user's intent. However, small sized display may not be able to hold the entire frequently used interface at once. In another hand, large sized display may provide too much space between each interface for an efficient task. Dual monitor environment, or multiple display system specially designed for large data management in control rooms and trading floors provide information control interfaces within a single reach for task efficient manipulation, deals with problems above. However, it is not scalable, does not allow more displays to join dynamically and depends on hardware connectivity. Our approach enables any display to join in via a network and bring a part of UI components from the main display in action.

3.3 Content Filtering

According to the type of information in UI components, there exist cases either hindering the communication or even attacking privacy, accessing in the public display. The example of accessing banking account from the web using the public display shows its needs. Also, heavily packed information can be blinded when accessing from a small display device according to its predefined descriptions showing the most important content first, least important last.

By changing some parameters, the users can easily adjust the content frames. For example, display with the “age under eighteen” or “public” settings may blind the UI component tagged harmful display material. Instead, the filtered content UI component can be distributed to other displays meeting the requirements if available.

Such application domains are well examined to extract the requirements for designing the descriptive language as well as the management system for distributing content across multiple displays.

4. Requirements

Approaches to enable UI components to be distributed to multiple displays were described. Requirements were deducted as follows by comparing existing technologies.

4.1 Representing the Links between UI Components and Displays

HTML is a markup language that represents how contents are rendered in a Web browser and more importantly it represents hyperlinks. In terms of description of contents, a HTML document represents contents with hyperlinks that point to another HTML document stored in a computer in the Internet. It allows us to navigate hypertext in the Internet by simply viewing a HTML document and following hyperlinks using a Web browser. Likewise, we can think of our approach in that UI components are displayed in the network of displays. To model it, we need a way of representing the UI components with "links" to display. The link can be viewed as a pair of UI component and display, indicating that the UI component needs to be shown in the display. A XML-compliant language, PDML is introduced in specific in the following sections.

4.2 Managing the Application–Display Relational System

Through consideration of the display-display relation management as well as the application or service distribution onto the dynamic displays are far more complex than a single display system requesting a service. Since the display context and setting may change dynamically, the available displays for service need to be tracked in real-time, but efficiently. Also, managing the sessions for “service-linked-displays” is an issue, because the service may request different number of displays anytime. Therefore the displays linked for a certain service should register in and out flexibly.

In chapter 5, the management system is described with multi-session control for multiple displays. Also session DB schema is shown for managing the dynamic context change occurring of the displays.

In summary, to satisfy the requirements above, we propose PDML for describing UI components and multiple displays. We also propose PDMS, a management system for centralized control over the multiple displays for dynamic information distribution service.

5. System Architecture

5.1 PDML

PDML is a declarative language describing what and where the UI component should be represented onto multiple displays. It is designed according to the requirements deducted from above sections. PDML
document could simply be a web document page consisting text and image frames, or a complex web runtime application. In our current prototype system, PDML schema consists of following descriptions: target display property, onMove, authority, and position for selectively distributing PDML UI components onto the set of displays connected to the management system.

Target display again declares two types of descriptions—relative and absolute property. Relative property refers to the target display candidates and compares them to the current display status. For example, the <TargetDisplay> the UI component directs to is the “nearest” as shown in the figure 2. The PDML management system searches for the nearest display registered in its management database and translates the “nearest” to the absolute property—the IP address of the nearest display. These translations of relative attributes may change according to the display environments and their semantic context. In the other hand, the absolute property does not need to undergo any translations and directly searches for the matching display. The intersecting results of relative and absolute properties derived for selecting the target displays. There are unlimited extensible set of attributes that can be added for the <TargetDisplay> with a simple XML-based query language.

The <onMove> declares whether or not to leave the UI component once it is distributed to another display. It either has “copy” for duplicating the same UI component, or “move” for cutting out the UI component from the original display and pasting it to the other.

The <Authority> declares the control access permission on each display devices. It basically has mdWrite, mdWrite_pdfWrite, meaning the write access permission for this UI component on both Main Display and Peripheral Display. mdWrite_pdfRead, mdRead_pdfWrite, and mdRead_pdfRead provide access permission to each displays accordingly.

The <Position> declares the absolute or relative position of the UI component within the target display device. PDML management system includes the UI component arranger which collects multiple UI components and repackages them into the multiple set of PDML documents before UI component distribution. More specifics of the PDML management system is described in the next section.

![Figure 2. An example of PDML schema](image)

5.2 PDMS (Pervasive Display Management System)

PDMS consists of a Session Manager, Session DB, Context Manager, and Content Distribution Manager. PDMS interacts with a cloud of web applications and services based on PDML and distribute them onto the Main Display and multiple peripheral displays accordingly.

First, each one of the displays, including the Main Display (user handling display) and peripheral display to serve as a distributed service representation, are registered on to the PDMS. Session Manager collects each display’s property including display’s physical address (IP address), Location (GPS data), display size, resolution, mobility (static vs dynamic), privacy type (private vs public) in the Session DB (see Figure 4).
Once the Main display requests a PDML document, which is a web application or service, the Session Manager searches for the matching display lists for each one of the UI component description. Session Manager refers to the Context Manager for the relative display property and opens a session for the application, including and excluding the target displays into the session.

The Content Distribution Manager then receives Target PDML documents, also referring to the Context Manager to transcode display’s absolute address, and repackages the PDML documents according to its requirements. Finally, the PDML documents are selectively distributed according to the display context and property.

For each PDML document service, single session is generated until the application termination. Within the service runtime, the participating display may be included or excluded according to the PDML requirements and the changes of display context.

| Session ID | Display List | Target PDML Doc | ...
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Display ID</td>
<td>Location</td>
<td>Resolution</td>
</tr>
<tr>
<td>Mobility</td>
<td>PrivacyType</td>
<td></td>
</tr>
</tbody>
</table>
|            | ...

![Figure 4. Example of Session DB Schema – the display property may change in real-time by the Context Manager](image)

### 6. Conclusion

In this paper we proposed a novel method for selectively distributing contents and applications to the available displays in the PDEs. Issues in harnessing multiple pervasive displays for information and application access were examined. Requirements of information and system design for the PDE was deducted. The display information, contents, and applications are abstracted into the UI components. In order to make such UI components available across the PDEs, an XML-compliant language PDML and its management system PDMS was proposed. We showed that our approach can be supportive in dealing with issues in the PDEs as follows.

First, once the UI components and target environment is represented using PDML, it can be adapted into an actual display environment dynamically. Second, the PDML based display property and context description is scalable for extension to any type of information display service. Third, the selective distribution mechanism of public and private information assures privacy safe ubiquitous computing.

The future work is to conduct user experiments to measure and evaluate a domain specific PDML application in terms of cognitive load, and task efficiency. Based on the results we expect to extend the languages in PDML describing the relations among the displays for more application customized, context customized and user customized PDEs. Finally after the verification, we expect PDML to be standardized at W3C to boost pervasive display related information service industries.

### 7. Acknowledgements

This work is supported by Korea Communications Commission under grant no. 11912-03001.

### 8. References


