Content Exchange Appliances

Dejan Milojicic, John Ankcorn, Rich Gossweiler, Jim Rowson,
Larry Rudolph, Sonia Garg, Franklin Reynolds, Rajnish Kumar,
Himanshu Raj
Mobile and Media Systems Laboratory
HP Laboratories Palo Alto
HPL-2003-139
July 3rd , 2003*

E-mail: [dejan, ankcorn, rcg, jrowson] @hpl.hp.com, [rudolph, garg] @lcs.mit.edu, franklin.reynolds@nokia.com [rajnish,rhim] @cc.gatech.edu,

displays, appliances, personal appliances, content exchange

WWW and current Information Technology have made it easy to display a wide variety of content on desktops and personal devices. Unfortunately, little progress has been made for access to the content in public areas. Some technologies, such as Internet Kiosks and narrowcast, enable content access (primarily viewing), but not exchange. There is a growing need to access, adapt, and exchange desired information with public displays for improved user experience.

In this paper we propose a new approach to content exchange using special purpose devices, which we call Context Exchange Appliances (CEAs). A CEA is a display appliance capable of presenting content and supporting a simple *"put/get"* interface for content exchange. Based on what is displayed on the CEA, a user can download/upload content to his or her personal device. CEAs have the potential to improve user experiences, maintain privacy and minimize information overload while offering intuitive yet secure user interfaces for content exchange.

* Internal Accession Date Only
Approved for External Publication

Content Exchange Appliances

Dejan Milojicic, John Ankcorn, Rich Gossweiler, and Jim Rowson
[dejan, ankcorn, rcg, jrowson]@hpl.hp.com
HP Labs

Larry Rudolph and Sonia Garg
[rudolph, garg]@lcs.mit.edu
MIT

Franklin Reynolds
franklin.reynolds@nokia.com
Nokia Research

Rajnish Kumar and Himanshu Raj
[rajnish,rhim]@cc.gatech.edu
GaTech

Abstract

WWW and current Information Technology have made it easy to display a wide variety of content on desktops and personal devices. Unfortunately, little progress has been made for access to the content in public areas. Some technologies, such as Internet Kiosks and narrowcast, enable content access (primarily viewing), but not exchange. There is a growing need to access, adapt, and exchange desired information with public displays for improved user experience.

In this paper we propose a new approach to content exchange using special purpose devices, which we call Context Exchange Appliances (CEAs). A CEA is a display appliance capable of presenting content and supporting a simple *"put/get"* interface for content exchange. Based on what is displayed on the CEA, a user can download/upload content to his or her personal device. CEAs have the potential to improve user experiences, maintain privacy and minimize information overload while offering intuitive yet secure user interfaces for content exchange.

1 Introduction

Although graphical displays are ubiquitous in modern life and are an integral part of our interactions with information processing devices, there is so much more that we can do with them, especially in the public arena. Public digital displays are used for advertisement (e.g., in stores, along highways, and in restaurants), entertainment (e.g., at sports events, cinemas, and toy stores), for emergency information (e.g., traffic alerts on high-ways), etc. Most public displays are either used only for viewing content or they are used for very limited interaction, typically through touch screens. We envision a future where people can have a greater variety of interactions with signage, notice boards, and even futuristic display devices such as animatronic figures or holograms.

We realize that this future poses many research challenges. Since the public displays can be used in many ways, we need a generic system that can allow the interaction between displays and users' personal devices. What should be the interface between users' device and the display, how can the content be more personalized, what are the related privacy issues and how to solve them – our system needs to face these challenges.

In this paper we propose a new architecture for content exchange based on Content Exchange Appliances (CEAs). CEAs require minimal installation and management efforts. Once setup, these devices display information provided by the owner. CEAs enable users to download or upload content using personal devices (see Figure 1). Interaction is primarily done by value, not by name, i.e. a picture, video clip, or a piece of displayed news is available for exchange with the personal appliance. CEAs are deployed as public displays and in the rest of the paper we will use these two terms interchangeably.

The content on CEAs is adapted in selection (filtering only content of interest to user), form (audio, video, headline, short/long), and presentation (browser, fonts, language, etc.). The more information users reveal about their interests, the more content filtering and adaptation that can take place. There is an analogy in real life: if you frequently go to the small grocery store and get acquaint-

Figure 1. CEA Concept. *A user interacts with a public display using a personal device, such as a phone. She can navigate the content, download it, or view additional content not visible on the display.*

ed with the owner, you will receive discounts. If you have a supermarket membership card, you are revealing information about yourself in exchange for discounts. In a similar way, if you reveal more information about yourself and your intentions to the CEA, you receive more personalized information in return (a discount on the amount of effort you invest to get what you want).

The rest of the paper is organized as follows. Section 2 overviews CEA. Section 3 presents a design and Section 4 a prototype implementation of CEA. Section 5 discusses CEA applications. In Section 6 we evaluate lessons learned and in Section 7 we discuss remaining challenges. Section 8 compares CEA to related work and Section 9 summarizes the paper.

2 CEA Overview

At the very minimum, a public display is a device capable of presenting some graphical information that anyone can see. We also make some additional assumptions. There is at least one owner of the display that has ultimate control over what is seen by the general public. There may be multiple copies of the display at geographically different locations, all of which present some variation of the same public information. The novelty comes with the assumption that there is a set of "consumers" who can consume or capture the public content as well as non-visible content through the use of personal devices, such as a PDA or a cellphone. Finally, both owners and consumers can have access to different resources and content based on their privileges. The public content is the main means of navigation or naming of the non-visible content.

2.1 Motivating Scenario

Jamie is flying out for an interview at a company in the Bay Area. As Jamie walks by the large public displays at the airport, she sees something of interest. Instead of trying to remember or jot down a URL and later re-enter it, she simply uses her phone to "record" the information.

While recording the information, the phone also personalizes the experience. It allows the user to navigate the publicly visible content without revealing her interests or disturbing the public display. Jamie was always interested most in the picture that has just disappeared from the screen. By navigating back on her phone, she can view the images that appeared earlier on the public screen, she can preview on the personal device what will appear on the public display next, or see the details that were not appropriate or would not fit on a public display.

Jamie can also decide what, where, and how much content to download v. bookmark, aided by her phone's awareness of its resources and capabilities. She casually bookmarks the URLs and picks up the thumbnails of new movies into her phone, however phone redirects all of the movie trailers into the tablet PC in her backpack. As she stands longer in front of the CEA, all the content she watched gets picked up too.

Jamie is especially happy about all of her options with the content. She can show it to her colleague on the train, while disconnected from the Internet; she can upload it on her private CEA at home (a desktop computer or a projector), or if she forgets about it, the content will get stored in her "memories shoebox".

While she is waiting at the lobby for her host to arrive, she notices some new product brochures and technical reports. She downloads a few of them to her phone. She was also interested in the new benefits announcement, but unfortunately this content was accessible only to full time employees, a few of whom casually picked it up while badging in for work.

Jamie is pleased with the social life at the company. The employees gather around a billboard in a social area to see the clips and images one of the employees took during his vacation in Brazil. The same content propagated across other company sites, except for the ad "cat looking for home", which was featured only at the local site. Jamie pondered how this scenario applies to stores, apartment complexes, and real-estate ads.

Jamie's phone also customizes the presentation, for example, by alerting her of the content that is available through the public display. On the way back to the airport, Jamie is so happy when her flight number gets highlighted on a huge screen as she is rushing towards the gate. Her trip was a success.

2.2 Classes of Use

Uses for public displays come in the following styles:

- Published by a few, used by many: e.g. movie times or seminar information.
- Symmetric *put-get* by large populations: an equivalent to bulletin boards in supermarkets or on the Web.
- Symmetric *put-get* by a small, closely related group, such as a project team, a family, or buddies.

End-uses of CEAs include:

- Casual gathering of information: e.g. interesting headline or seminar, house listing, and ads posted by individuals rather than organizations.
- Simple communication: e.g. leave a note for a friend or post a review of a nearby restaurant.
- Access to shared information: e.g family calendar, "where are the kids?" or "has my husband left the office to go pick up the kids?".
- Emergency information: in disaster, take over all screens to provide useful information.

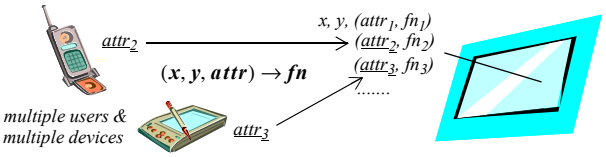

Figure 2. CEA Model. *Every pixel on CEA is associated with a number of functions, enabling customization to preferences of present us-*

Some of these end-uses are context dependent: the information presented is local and it makes sense to have it associated with a specific display. Some of it is equivalent to an online bulletin board. Some of the uses are community oriented: a family calendar display in the office, home, or car, so that one can access it in the common places without the need to carry anything.

2.3 Model

In the past, a display was represented by a collection of *{((x, y), val)}* – a set of points with a value coming from some producer. A user could capture these values: *getVal((x, y))* – this is what a "print screen" command would do. Or if you point a camera at the screen, one could capture the values at the positions. At present, touch screens can be considered to be a display with a collection of *{(x, y), val, function}*. In addition to *getVal*, the user can also invoke the function associated with a location. The display helps the user to identify the function. So, user sees the display and clicks at the right location.

In the future, we envision public displays will go further with a set of functions potentially associated with each pixel. In other words, a display becomes a set of *{(x, y), val, {(attr, function)}}*. A click at a position comes with a set of attributes. Only the function that matches this set of attributes will be invoked (see Figure 2).

Having multiple functions associated with each pixel or *(x, y)* also allows users to upload content onto the display, provided the *attr* associated with *(x, y)* is not already assigned. Setting the *val* of point *(x, y)* on a public display requires some higher privilege. People can associate information or functions with a pixel but they can not affect what others see. The *attr* associated with the mobile user provides a way for managing the metadata and enables more flexible content selection. Allowing upload and download to be performed in a similar manner makes it easier to generate a rich display environment as well.

A very diverse set of users, with many different needs, can thus be supported. The public, visual graphics of the display provides a navigation tool or a language by which the user can specify a function, e.g. location *(x, y)*, but the desired function may vary from user to user.

To make this yet more concrete, consider a possible interaction mode. The user has a PDA or a mobile phone. Upon approaching the display, the user is recognized and given a "cursor". There may be multiple active cursors on the display; one for each person interacting with the display. Using a mobile phone or a PDA the user can move the cursor around the screen. With a PDA, the touchscreen can act like a mouse, on a cell phone, the keys can act like keyboard arrow-keys (2 for up, 6 for right) and something for click. The user can choose an application on the PDA as a way of specifying the type of content to be chosen. For example, choosing the address book application on a mobile phone and then "clicking" on the appropriate location on the display will transfer an address-book entry from the display to the mobile phone. Alternatively, if the user chooses the MPEG-3 player, clicking on the same location on the display would transfer a music file.

2.4 Challenges

Intuitive User Interfaces. There is an increasing overload of information available as well as an increasing overload of people's schedules. Content needs to be filtered for the present user, provided at the location, and in the way user prefers. Adaptation can take place based on the user preference and on the data-type or display properties. In *(x, y, val, {(attr, fn)})*, *fn* may be the transcoder or filter for doing the generic adaptation, and *attr* may be the user preferences provided to the *fn* for user-customized adaptation of the content.

The choice of a pointing metaphor(s) for CEA is another open research question. For example, should "pointer" be visible on the CEA or should users be able to point privately on the displays of their mobile devices, or both? Can a pointer be associated with a user if it is not controlled by the mobile device, e.g., if a CEA had a touch screen, does this imply that only one user at a time should try to use the CEA, and how would that be enforced? How many types of pointing actions are needed, e.g. for select, drag, drop, invoke, or meta operations? How to best support drag-and-drop functionality for multiple present users. For example, if I am interested in saving the recent news alert on my PDA, I could drag the pixels corresponding to the news into the pixel assigned for the presence of my appliance. Finally, futuristic devices, such as animatronic figures or holograms open a slew of user interface research questions.

Privacy. Revealing the *attr* field of each person enables displays to track down user's identity, interests, presence, and habits. It is an open research question how to optimize content adaptation while retaining the maximum privacy. Although we might wish to access some information available on a sign, we may not want to be bothered by email from that site later. In some cases, we may

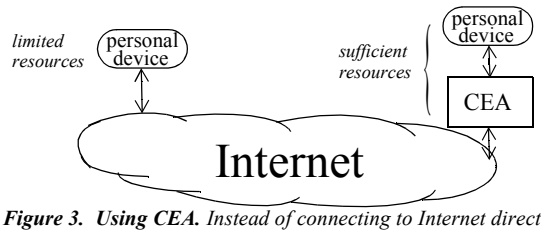

Figure 3. Using CEA. *Instead of connecting to Internet directly via a personal device, CEA acts as an intermediary – a "resource amplifier", providing capabilities users would not get from direct Internet connection: richer interface (size, power, I/O) and personalization (preferences, storage, familiarity).*

not even want to acknowledge that we were viewing content at a particular location. Becoming important places for exchanging information in everyday life, CEAs become focal points for many privacy issues. A person accessing an "employee-only" web site may need to prove that they are employees, but may wish to keep their name anonymous. If they wish to post information to a restricted notice board, they may be required to identify themselves to gain this privilege.

3 CEA Design

A CEA is a standalone appliance connected to the Internet. It can interact with the surrounding users by means of a wireless network and/or touch screen. A CEA runs a standard operating system, an Internet browser, and a run-time environment for content distribution, we call the Agile system [15]. Each CEA can discover the person near by (with the person's permission) by means of a wireless network (e.g., IR, Bluetooth, or 802.11). Upon discovering a user, CEA establishes a secure session with the user's personal device.

CEAs offer a qualitatively different context exchange as compared to users connecting directly to the Internet through personal devices. The user has more resources at their disposal, such as a larger screen, more power, I/O (e.g., touch screen, surround sound audio, and printers), processing, storage, and network bandwidth (see Figure 3). In addition, the display can be used as a shared resource among multiple users. Because CEAs will have varying capabilities, the content presented is adjustable to available resources. The content has different presentation forms, e.g. content on large displays is adjusted when viewed on smaller displays (see Figure 4). A display can only have the headlines of the news, while enabling *get* operation of the complete news items. Similarly, the video clip can be distributed in its entirety on the display, but only a small clip can be available using *"get"* operation. The availability of certain content may be restricted to particular users, whereas the titles (headlines) can be widely distributable.

The content presented on a CEA is organized in layouts (XML documents), with predefined times for repetition as well as expiration. The *attr* is maintained both on the CEA and on the personal appliance. If requested and permitted, the personal *attr* is matched against the CEA *attr*.

3.1 Agile Platform

A CEA depends on Agile for content store, content propagation and consistency, content adaptation, and for administration. The Agile system is aimed at building distributed applications that run on a set of appliances. The run-time consists of a distributed cache along with a web execution environment, forming a peer-to-peer overlay network for Web applications. Agile applications and data move dynamically onto the appropriate computing resources and execute locally.

While developing the CEA, we benefited from the Agile global uniform name space and data archival. We were able to map our content into the name space at will and not worry about where it was archived. The same applies to synchronization models and consistency across multiple machines. Agile also supported the security model and data encryption. One specific advantage is in its Web-based interfaces, enabling us to view the same content either locally on the CEA or remotely (e.g. for testing purposes or simply for viewing remote content).

For the development, Agile offers template and action language. Templates are being executed locally on the devices and enable dynamic content creation by manipulating the XML data (cf. Figure 9). Agile actions are propagated to the Agile servers in response to user input (cf. Figure 10).

3.2 Playing the Content

The content on a CEA is organized as bundles of HTML frames with individual frames contributing images, video, and text. Sequences of individual frames are presented on the screen at different intervals. They are viewed on a CEA in one form factor and on the personal device in the smaller form factor (see Figure 4). Using the personal device, the content of individual frames can be selected, zoomed in, and then played by user. Each frame is an Agile template that can be executed against the CEA's Agile node or against the local Agile if there is one running on the personal device. When the templates execute, they traverse the content organized in the CEA space using XML structures. When user navigates the content, the state is recorded in the Agile space, for example, the speed of the slideshow, direction (backward/forward), and the content selection for upload/download.

3.3 Creating, Organizing, and Importing Content

Content developers typically can use a variety of professional content creation tools and work with a rich set of different media types (e.g. HTML, Flash, video, images).

Figure 4. The Thumbnail Model of Navigating Content. *A thumbnail version of content is displayed on the personal device enabling it to browse and exchange the content independently from the CEA.*

The CEA has the ability to play several different media consecutively, much like a slide show or series of television advertisements. To do this, the raw assets that the content developers have created need to be enhanced with meta-information (e.g. play order, play duration, expiration, etc.). We use small XML files to describe this meta-information. In the future, we plan on using SMIL [17].

For example, the content-developer may want the CEA visual layout to be broken into a 3x3 layout. The upper left corner will show a static logo image, while the upper right corner will rotate through a series of images, changing every 30 seconds. The middle square will show video (see Figures 5 and 6).

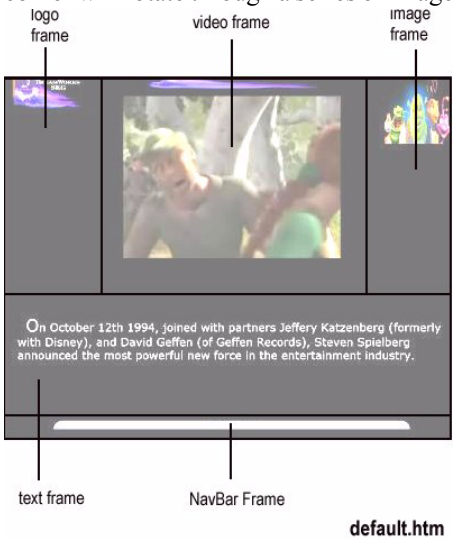

Figure 5. Organization of Content on Cea. *Video, images, and HTML are contained within different frames.*

Since a content-developer may not be familiar with the XML syntax, we have developed small direct-manipulation utility applications to help them collect, organize and publish their media content to the CEA. The tools support drag and drop, copying all of the media into a playlist. Pressing "save" then exports the media (e.g. video or images) as a bundle, along with the associated XML meta-data.

These tools not only make it easier for the developer to get data into the system, but also provide future opportunities to support other important operations (for example, version control, an upgrade migration mechanism, content optimization and encoding Digital Rights Management).

3.4 CEA-User Interaction

A wireless network enables the CEA to locate a user in front of the CEA (see Section 3.5). Once discovered, the CEA establishes a session between user's device and the CEA. At that point, the user can access the content on the CEA subject to access rights. The interaction is performed in an intuitive way. For example, while viewing a picture on a CEA, the user can touch that part of the display on CEA, or the thumbnail version on the personal appliance (see Figure 4), or even use a combination of a camera and a laser pointer. Distinguishing between users in front of many CEAs (and vice versa) is assisted by user's input (e.g., prompting the user to browse among different CEA icons).

Both CEAs and users are represented in the Agile name space. If users do not have a personal appliance, but rely on other identification mechanisms, such as badges, and use touch screens instead, the selected content will be pushed into the user's space in Agile and later accessed (see Figures 7 and 8).

3.5 Discovery

The personal device can use either Bluetooth or WiFi as a medium of interaction. Bluetooth is preferable because

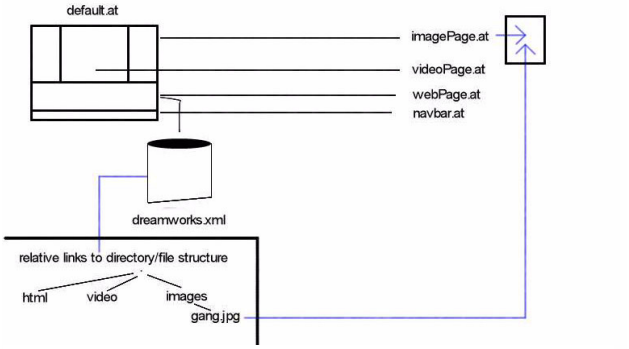

Figure 6. Content Playing. *Content is played by separate players that read content from Agile spaces. Players themselves live in Agile.*

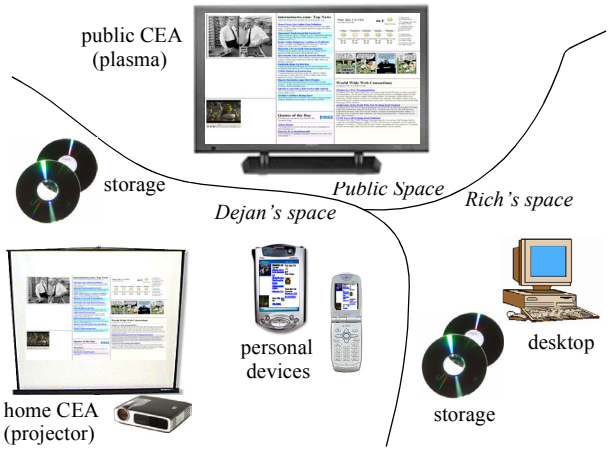

Figure 7. Private and Public Content: User and Device Spaces. *There are public and personal spaces for content. Each public CEA belongs to the public space, which user can access. In addition, each user may have his personal devices, a private CEA, or some designated computers and/or storage where they can copy the public content.*

of its ability to use proximity to connect. The display acts both as an 802.11 base station as well as a blue-tooth master. Making a connection to a public display from a handheld device must satisfy several competing goals. We have prototyped both a Bluetooth and a wifi enabled handheld device. In the case of Bluetooth, the handheld is the master and the display is the slave.

Since the display must act like a slave to multiple masters, the initialization protocol is somewhat complex. Since we wish to support both types of connections, we use an IP connection for both. We also use Secure Socket Layer (SSL) for the communication with a heavyweight public-key encryption system to make the initial connection followed by an inexpensive MAC encryption scheme for the rest of the communication. The display remembers the MAC key to make future reconnection with the handheld device fast. Although Bluetooth does provide functionality similar to SSL, it is important to use just a single secure interface in order to reduce the need to maintain twice as many secure interfaces; maintaining one is hard enough. The handheld can change its Bluetooth name, MAC key, and nonce so that it appears as a new device to the display. Thus, the only price to pay to avoid tracking, is a slower reconnect time. If one does not care about privacy or simply trusts the display, reconnect is fast.

With WiFi, the situation is somewhat different. The display can act like a base station and DHCP server. The WiFi handheld gets an IP from the display and the display then makes a connection to the handheld. The rest is the same as in the Bluetooth case. As one approaches a display, the handheld initiates the connection. There may be other handhelds already connected and there may be other displays nearby. The signal strength indication is

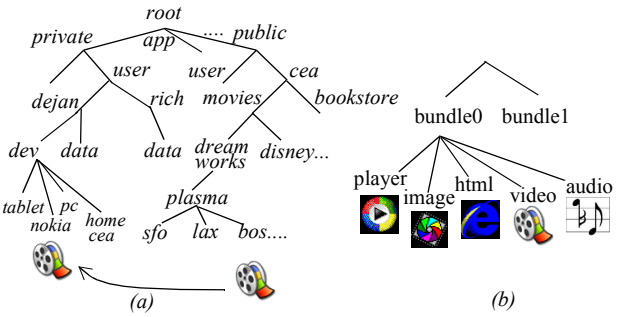

Figure 8. Name Spaces. *Device and user spaces are mapped into the Agile global name space; "get" content copies content from a location into another (a). Content is encapsulated with player, so user picks also the look and feel of playing it. Player consists of a number of template files, content is a collection of images, video, audio and html (b).*

usually sufficient make the display choice obvious, however, the user may be called upon to select the display from a set of Bluetooth slaves. Once an initial connection is made, information is exchanged and the correction is broken.

3.6 Security

In making information available, whether in public kiosks or privately, we often wish to limit the allowed uses and users of that data. In a corporate kiosk, although we may make general information public, detailed telephone directories and organization charts are usually accessible only to employees. Although we want to allow a large group of people to post messages on public bulletin boards, we may want to restrict ability to post or to have some traceability of authors as a means of preventing abuse. Our ability to use kiosks to improve accessibility to a wider range of information will depend on being confident that the information and facilities that we have made available are properly used.

The simplest method to manage authorization for tasks is the presentation of short-lived capabilities at the time of access. These capabilities are minted by the original content owner in conjunction with one of many authentication mechanisms that are already in use today – key cards, VPN network logins, and specialized security devices. Once a user is able to confirm the identity that they wish to assume for the transaction (perhaps their personal name, "anonymous employee" or "person with money"), they can present this confirmed identity to the content access control service, receiving a certificate to be presented to the kiosk to obtain information or services. In this way, we can easily deploy CEAs, leveraging off the existing infrastructure investments.

To support interaction sessions with a single certificate, we use intermediate proxies to perform authorization control without affecting the original format or content of the client and server protocol interactions. By the use of transparent proxies in the CEA infrastructure, we can

```
#set curimageudl=$[$xml/$slide#@image]
#set nextslide=$[$xml/$slide#@next]
#set nexturl="http://$agileOriginalHostPort/agile/View?t=$t&xml=$xml&slide=$nextslide"
<html>
<head>
  <title>Image show</title>
<script language="JavaScript">
function cycleImages() {
   location='$nexturl';
}
</script>
</head>
<body>
<a href="$nexturl" target="_top"> <img name="ceaImage" border="0"
                                src="/agile/Read?udl=$curimageudl"> </a>
<script language="JavaScript">
  setInterval("cycleImages()", 15000, "");
</script>
</body>
</html>
```

Figure 9. Agile Template Example. *Variables curimageudl, nextslide, and nexturl get initiated from the XML database.*

perform certificate presentation out of band with the actual content access, but synchronized to it.

As users access public devices more frequently, there is a danger that information about them could be collected and collated by untrusted third parties. To counteract this trend, we would expect users to have multiple, uncorrelatable identities for use in different transactions. In this way, a user would only have to reveal minimum information about themselves in order to get access to a service. As identity management becomes more complex, we think that users will depend on personal trust assistants to remember and present appropriate identity information at the relevant time. These will act much in the same way automobile key fobs perform today when they authenticate the user as the "owner of the vehicle" but reveal no other information about the user.

4 Implementation

Our prototype CEA implementation is based on plasma displays connected to PCs running either Windows or Linux. For a personal device we experimented with Nokia 3650 phone using Bluetooth and with HP iPAQs using 802.11.

The first prototype implementation was deployed in the printer area at HP Labs in Palo Alto. The prototype consists of a display connected to the Internet. The CEA displays short videos, images, weather forecasts, comics and news. An Agile server runs on each CEA. Agile executes XML templates describing individual pieces of content, such as source, type, layout, and content lifetime. After the cache expires, it is refetched upon new access. The content on the display is refreshed using JavaScript (see Figure 9). For this prototype, we use badge readers for user discovery because all employee are wearing badges, so it is ubiquitous. A badge reader is physically attached to CEA using a serial line. Swiping a badge causes the value of the badge to be read from the serial port and then sent to Agile. Agile interprets it and causes an Agile action to execute (see Figure 10). The

```
<action transaction="commit">
 <define name="badgeno" xml="/pub/cea/hpldejan8080/swipe" xpath="@id"/>
 <define name="src" xml="/admin/device" xpath="@id"/>
 <if>
  <test xml="/cea/users/users" xpath="user[@swipe='$badgeno']">
   <if>
    <test xml="/pub/cea/hpldejan8080/login" xpath="user[@badgeno='$badgeno']">
     <define name="user" xml="/public/cea/hpldejan8080/login"
                                xpath="user[@badgeno='$badgeno']/@name"/>
      <delete xml="/public/cea/hpldejan8080/login" xpath="user[@badgeno='$badgeno']"/>
    </test>
    <else>
     <define name="user" xml="/cea/users/users"
                                xpath="user[@swipe='$badgeno']/@name"/>
     <define name="url" xml="/cea/users/users" xpath="user[@swipe='$badgeno']/@url"/>
    <define name="img" xml="/cea/users/users" xpath="user[@swipe='$badgeno']/@img"/>
    <define name="urls" xml="/cea/users/users" xpath="user[@swipe='$badgeno']/@urls"/>
     <insert xml="/public/cea/hpldejan8080/login" xpath="/logged">
      <user name="$user" url="$url" urls="$urls" img="$img" badgeno="$badgeno"/>
     </insert>
     <create xml="/device/dejanob61008080/popup">
      <anything url="http://hpldejan.hpl.hp.com:8080/agile/View?t=
                                /app/cea/thumbs3.atr&amp;xml=$src&amp;"/>
     </create>
    </else>
   </if>
  </test>
  <else>
   <print label="found no user for badge" value="$badgeno"/>
  </else>
 </if>
</action>
```

Figure 10. Agile Action Example. *This action gets initiated when the badge is swiped. First it searches if the badge number is in the database. If so, the other relevant info is extracted from and entered into the login structure; Creation of the popup.xml will trigger popup window on personal device.*

Agile action attempts to identify the user, to log it into the CEA, and to popup a thumbnail popup window on its the personal device. The window is popped up only with user's permission. We rely on the security implementation by Reddy, et al. that has proven to be easy to deploy and maintain [13].

On iPAQs we use 802.11 and connect to the HP intranet using VPNs, much the same way as we would use laptop or desktops. On iPAQs we use Pocket PC and Pocket IE as the underlying infrastructure along with the Agile port to iPAQ. Because of the different screen size, we are executing templates in a different form factor, however most of the mechanisms for playing the content is the same. In order to support ease of use, as the user swipes the badge, the thumbnail of the CEA content automatically appears on the iPAQ. This presumes that the iPAQ owner has enabled this preference.

For our prototype implementation on Nokia 3650 phone we decided to use Java as the development platform. The intention was to minimize the effort of porting applications on different phone platforms in future. Nokia 3650, supports a J2ME MIDP 1.0 environment. We establish communication between our Java application on the phone and target application on the CEA using Bluetooth serial port. Unfortunately, the Java environment does not yet support Bluetooth stack. We developed a small proxy application for the Symbian environment (using Symbian's C++ SDK) that communicates with our Java application on the phone using local sockets.

5 Applications

We explored three applications for the CEA: Content-to-Go, billboard, and content cascade. Prototyping of the first two is underway and we recently started the last one.

5.1 "Content-to-Go"

In the Content-To-Go Application, CEAs consisting of large screen displays and self-contained computers are scattered about in public places, such as in coffee shops, across several malls or even across the world. These devices can provide public information and timely alerts, but their primary purpose is to deliver advertisements and incentives (e.g. digital coupons). People who see interesting content such as a movie trailer or an advertisement about an interesting item can use their personal device (e.g. cell phone or PDA) to capture the content. In addition to getting the appropriate content based on the device's available resources, the person can also get other, related information. This might include a coupon for the item or a map to the store nearest them. Additionally, the Content-To-Go application can be context-sensitive, for example, showing umbrellas when it is raining out. Since people are using their own device to navigate the content, the same display can be shared by multiple people.

5.2 Billboard

The billboard application reflects on the model that already works in practice. Billboards are a successful way of informal communication at the universities, street corners, in front of the drugstores, and other places where people of similar interest or locality gather. The billboard model typically relies on leaving an information and additional pointer for others people to take away; for example, apartment to sublet; car for sale; and lost pet. At the bottom of the note, there is half-cut piece of paper with the phone number or contact address. In a similar manner, the billboard CEA application allows people to leave their notes on the CEA billboard. The strength of our billboard application is in being able to propagate content to other interested CEAs based on their preferences and on the content topic.

5.3 Content Cascade

In this application, the role played by CEA is primarily of an active preview provider. Whenever a user carrying a mobile device visually browses the contents being displayed by the CEA at the public display, a summary of the content is pushed to the mobile device by the CEA. The summary contains the URL from where the content can be accessed and a small preview of the content itself, e.g. a clip from the movie or a snippet from the text. In case the user spends more time in front of the display based on immediate interest, even the actual content can be pushed to the mobile device (this also depends on the business model and user's willingness to remain passive in this transaction). The scenario naturally supports multiple users since there is no explicit interaction of users with the display. The summary can be seen as the cached snapshot of data browsed at different locations. At a later time, users can go through the previews available in the summary, and based on their liking, decide to obtain the related content (remove the preview). Preview/content existing on mobile devices can be synced with devices with better user interaction capabilities, such as laptops or desktops, in order to have a richer experience of the content. User might store different versions of the content on different devices based on device profiles and accessibility/interest factors.

6 Lessons Learned

What you see is not what you want. When you see something of interest on a CEA and think "I want that." What is it that you mean by "that"? Do you want to copy that exact data at that moment in time (copy-by-value)? Or do you mean that form of information (e.g. I want to get the daily weather report – copy-by-reference)? To complicate matters, even with copy-by-value (I want that movie trailer), do you want those exact bits, which you can carry to another display, but cannot possibly enjoy on your personal device, or do you want the data in a format suitable for your personal device?

Privacy: ease of use and ubiquitous deployment. We can not underestimate the ease of use of the infrastructure. Some of the obvious interfaces, such as badge reader made a big difference in using the CEA. Badges were used both for authentication and authorization of users within the company. Swiping the badge allowed the CEA to recognize who is in front of it in order to adapt the content, as well as to automatically pop up the navigation screen on the personal device eliminating the need to type in the URL. On contrary, using PDA with VPN proved to be cumbersome because of the need to type in the password, followed by the output from the ActivCard. Due to the heavy fragmentation of user "identity" in the future and the difficulty of managing authentication interactions, it appears that a personal identity management device will be required to allow users to interact with digital infrastructures with the flexibility and confidence that they do with interpersonal exchanges today.

Choice of a personal device - PDA vs. phone. At the start of our implementation we had to choose a personal device. We were aware that mobile phones are widely deployed and few people carry PDAs and that there were speculations about PDA functionality converging into mobile phones in the near future. However, at that time there were no phones available with a wireless connec-

tion so we decided to proceed with the PDA as a development and demonstration vehicle. We furthermore selected Pocket PC instead of Linux because more people were using off-the-shelf PDAs. In retrospective, this was a poor choice. Even though Pocket PC is compatible with Windows environment, there are subtle differences that slowed down our development. For example, resource constraints prevented Agile from stable execution requiring continuous adjustments; the IE Object Model (even though powerful enough) is different from the Windows IE model requiring rewrite of the CEA players; The CE version of the VPN software was unstable; and finally the development cycle is slow on PDAs. If we were to start the project now, we would probably start with only phone development for the personal device, even if we would have to wait for the wireless to be supported.

Name spaces and storage model. Name space organization for the content is personal and our efforts to enforce some adopted structure, such as where to put devices (public, personal, private) and where personal content failed even between the two developers. In addition, there was a mismatch in the original design requirements between Agile and CEA with respect to global v. p2p name space model. Agile model is inherently global: there is a root node, which maps all other namespaces underneath it, whereas CEA model is inherently p2p, enabling any device to name any content and pass it to another device irrespectively of devices' and content's location in the name space.

7 Remaining Challenges

CEA-CEA Interaction. The CEAs can be used in three different communication models, which also define the consistency models, and the architecture:

- narrowcast (push of content), strong consistency, client-server architecture,
- billboard/newsgroups (lazy digest push/pull), weak consistency, p2p architecture, and
- broadcast (emergency, real-time push of content), best effort consistency, a combination of a p2p and a client-server architecture.

It is a unique requirement that CEAs should support all three models. For example, in a supermarket, a CEA can present ads, only to switch to the billboard mode for real estate advertisements by an interested user, and suddenly to turn into an emergency display in case of a reported fire. Propagation of the content between CEAs is based on the type of content and the purpose of CEA. Availability and reliability of content exchange is achieved through redundancy (p2p techniques applied to CEAs).

Group Collaboration and Competition. A public display creates interesting social dynamics. Complete strangers may gather around momentarily, based on seeing something of common interest. What sort of interaction model will support this informal collaboration? Can more information be shared based on group interests? Can these momentary collaborations become longer term relationships? Also, these devices, much like vending stalls at a marketplace, may allow people to gather on a regular basis to trade information while meeting over coffee. How can a public CEA support these regular collaborations? In addition to momentary and regular collaboration, these devices have the potential to support competition. Much like eBay and real-world auctions, people can gather to present and to purchase commodities (digital and physical). Given the opportunity to support private information on personal devices and public information reaching a larger audience, can new models for competition and exchange be developed?

Zero-Configuration. We deal with self-organizing behavior in at least three types of network situations:

- IP level networking (v4 or v6 autoconfig) for CEAs that may be attached to the Internet.
- CEA communities need to self-organize as CEAs join and leave communities of the same type.
- Mobile devices need to be able to "show up at" and "depart from" a CEA with a minimum of user distraction.

Using Agile. While Agile was well suited for CEA as a content delivery network, the Agile programming model has two main challenges in supporting CEA:

- It is somewhat weak: conscious choice in first implementation was made that actions support only tree manipulations on XML which is good because it is simple and we can guarantee that certain properties hold however, it is challenging because we can't do everything we want.
- There are inconsistent syntaxes: html, template, action, etc. to get content developers to use this we need to make it easier for non-programmers to use: wysiwyg, visual languages, IDEs, etc.

There is a tension between natural Agile model which is that user's should not need to care about where an asset is stored and the natural CEA model where the user gets an asset from a CEA onto his PDA and then puts that asset onto another CEA. Current Agile storage model has a single archive per asset, though there can be many archives for different purposes. Agile should move to a p2p model where the notion of single namespace is either lose or does not exist and where CEAs cooperate to ensure persistence.

8 Related Work

There are three classes of related work: public displays, Human Computer Interfaces (HCI), Computer Supported Collaborative Work (CSCW), and traditional p2p.

Examples of public displays include Internet kiosks, teller machines, and information appliances. Internet kiosks have not achieved wide deployment despite initial spike in interest. ATMs have been widely deployed for banks, and their use is being extended to express-checkin at the airports and even Starbucks. Finally, there is emerging deployment of information appliances at cinemas [11], elevators, McDonalds, WallMarts, etc. As a public display, CEA is positioned between Internet kiosks and information appliances. It offers more limited interfaces compared to Internet kiosks (which we consider an advantage), but more intuitive and secure. It has superior interfaces compared to ATMs and information appliances, enabling also content exchange. It has reduced cost of ownership compared to information appliances.

Related work in HCI and CSCW includes large collaborative displays and inter-appliance communication. Display appliances projects encompass I-room [6], LiveBoard [1], Web Signs [9], and Blue-Board [16]. Inter-appliance communication projects consist of Personal server [18], Pick-and-Drop [14], InfoStick [8], and work by Greenberg and Boyle [5]. Most of these systems are targeted for collaborative work and not for public displays enabling content exchange. We can derive some of the experience in user interfaces, but privacy, security, and content distribution are typically not addressed.

There are many p2p systems applied to content exchange both in industry and academia [10]. Industrial efforts typically address MP3 swapping and download, such as KaZaA [7], Napster, and Morpheus [12]. Among many others, academic efforts related to CEAs encompass Gnutella [4] and Freenet [3]. CEA differs in that it is targeted for public places rather than desktops, introducing different requirements for user interfaces, security, and content presentation.

Our work is also anecdotally related to the movie "Minority Report", although in contrast to the movie we consider our goal to protect the privacy of the users of CEA.

9 Summary and Future Work

We believe that the displays will become ubiquitous in the near future and we propose how to extend their functionality. Supporting higher levels of functionality in these appliances will open up a new area of targeted context delivery, enhancing people's experiences in public and private areas. Some of the observations we have made so far consist of the following:

Interaction with the Web is not the only thing one can do with a phone/PDA. These personal devices can be used for interaction with displays and to transfer the content.

Physical proximity is a workable way of enforcing security. Requiring users to be present in front of CEAs to access content limits scalable security threats.

Information access need not be a solitary activity. We are particularly interested in the group use of CEAs.

We are proceeding with CEA prototype implementations and deployment at HP Labs, Georgia Tech, and MIT.

ACKNOWLEDGEMENTS

We are indebted to Fred Douglis, Patrick Goddi, Bob Mayo, and Jeff Mogul for reviewing versions of the paper. Their comments significantly improved the content and presentation.

REFERENCES

[1] Elrod, S., et al., "LiveBoard: A large interactive display supporting group meetings, presentations and remote collaboration," *Proc. CHI'92*, pp. 599–607, 1992.

[2] Ferscha, A., et al., "WebWall - An Architecture for Public Display WWW Services," *Proc. WWW2002*, May 2002.

[3] FreeNet. 2001. The FreeNet home page, freenet.sourceforge.net. www.freenetproject.org.

[4] Gnutella. 2001. The Gnutella home page, gnutella.wego.com.

[5] Greenberg, S., Boyle, M., "Moving Between Personal Devices and Public Displays," *Proc. Workshop on Handheld CSCW* Nov. 1998

[6] Johanson, B., et al., "Multibrowsing: Moving Web Content across Multiple Displays," *UBICOMP 2001,* Sept. 2001.

[7] KaZaA, www.kazaa.com.

[8] Khotake, N., et al., "InfoStick*: an Interaction Device for Inter-Appliance Computing", Workshop on Handheld and Ubiquitous Computing (HUC'99)*, 1999.

[9] Lewis, S., "Web Signs, Display Appliance Platform," Appliance Studio *Technical Report*.

[10] Milojicic et al, "Peer-to-Peer Computing", Submitted for publication, Fall 2002.

[11] Marriott, M., "Movie Posters That Talk Back," Article from NYTimes.com. Dec. 12, 2002.

[12] Morpheus, www.morpheus.com.

[13] Reddy, P., et al., "Authentication and Authorization of Mobile Clients in Public Data Networks. HPL-2002-213

[14] Rekimoto, J., "Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments," *Proc UIST'97,* pp 31–39.

[15] Rowson, J, et al., "Agile Computing," HP Labs TR. 2002.

[16] Russel, D., Gossweiler, R., "On the Design of Personal & Communal Large Information Scale Appliances," *Proc UBICOMP* 2001, pp 354–361, Sept. 2001.

[17] SMIL, The Synchronized Multimedia Integration Language Web Page, http://www.w3.org/AudioVideo/

[18] Want, R., et al., "The Personal Server: Changing the Way we Think about Ubiquitous Computing," *Proc UBICOMP* Sept, 2002, pp 194–209.