Pervasive Web Access via Public Communication Walls

Alois Ferscha and Simon Vogl

Department of Computer Science
Johannes Kepler University of Linz
{ferscha,vogl}@soft.uni-linz.ac.at

Abstract. Multi-user communication and interaction via public displays together the pervasive and seamless access to the WWW in public areas via mobile phones or handheld devices is enabled via the WebWall system. A software framework for the operation of WebWalls has been developed, strictly separating WebWall access technologies (like HTTP, email, SMS, WAP, EMS, MMS or even simple paging protocols found on mobile phones) from the physical display technologies used and the presentation logic involved. The architecture integrates ubiquitous wireless networks (GSM, IEEE802.11b), allowing a vast community of mobile users to access the WWW via public communication displays in an ad-hoc mode. A centralized backend infrastructure hosting content posted by users in a display independent format has been developed together with rendering engines exploiting the particular features of the respective physical output devices installed in public areas like airports, trainstations, public buildings, lecture halls, fun and leisure centres and even car navigation systems. A variety of different modular service classes has been developed to support the posting or pulling of WWW media elements ranging from simple sticky notes, opinion polls, auctions, image and video galleries to mobile phone controlled web browsing.

1. Introduction

The growing availability of wireless communication technologies in the wide, local and personal area, together with the pervasive use of mobile and embedded computing devices gives strong raise for WWW services adapted to context, particularly to the person, time and location of their use. The seamless provision of services to anyone (personalized services) at any place (location based services) and at any time will presumably fertilize – besides the "desktop-WWW" – a qualitative growth of the Web towards an "embedded WWW", enabled by wirelessly networked autonomous special purpose computing devices (i.e. Internet appliances). Applications and WWW services will have to be greatly based on the notion of context and knowledge, will have to cope with highly dynamic environments and changing resources, and will need to evolve towards a more implicit and proactive interaction with users [3,16], and will have to be accessible in a more ad-hoc fashion – not only in privacy (from the desktop), but also in the public (via shared display artifacts). We therefore explore the fact that visual displays have played an important role in individual WWW usage, but very little research has been conducted to explore the potential of large, shared visual displays for group and community communication

F. Mattern and M. Naghshineh (Eds.): Pervasive 2002, LNCS 2414, pp. 84-97, 2002.

and interaction. While the use of visual displays and desktop projections is getting quite popular in group work settings (shared whiteboards, smartboards, etc.), their use in public spaces to allow for a ubiquitous WWW access for a broad, loosely related, non-determined and unstructured audience is only rudimentarily understood today.

With this work we address the potentials of ad-hoc communication in public spaces using a wall metaphor. We have developed a software framework, the WebWall framework, providing a seamless WWW access over visual displays in public spaces via a manifold of access technologies including HTTP and email, but most importantly SMS and WAP. The WebWall framework is presented as a means to enrich public places with digital communication and interaction means for people to access their personal ‘multimedia memories’, to share information (e.g. notes, videos, pictures) with others or to interact with others (e.g. opinion polls, auctions, games) – all over the WWW and possibly all over mobile devices like mobile phones or Internet appliances.

2. The WebWall Framework Architecture

The design principle of the WebWall framework appears to be independence with respect to potential access technologies, display technologies, and configurability and dynamicity of interactive services. The software architecture hence isolates a request handling component on the input side, a data management component in a backend system, and a presentation component on the output side. As for the input side, the integration of Internet- and mobile networking technologies demands for flexible and standardized access to a WebWall system which is granted via the representation of access requests in a standardized format, irrespective of the media (SMS, WAP, email, and HTTP). The strict separation of request handling and display rendering provides extensibility by means of being able to integrate new technologies as they evolve. As for the physical presentation of WebWalls, various display technologies exist today (projectors, plasma screens, CRT, etc.), and further technologies will evolve (laser projection). The WebWall system therefore is designed to provide flexible support for the full range of exisiting and upcoming display technologies. A presentation module is responsible for arranging the service instances on a physical screen according to service type and priority. Services are provided as instances of services classes with dedicated functionality. A so called “rendering engine” for each service class is responsible for translating the requested WebWall data into a displayable form, for example a HTML page of a given size.

Users interact with the objects (i.e. service instances) on a WebWall by passing messages and/or commands through one of the access modules. The current implementation of a WebWall provides GSM, IEEE802.11b WLAN as well as standard LAN access to receive requests, which are then passed on to the service access module that is responsible for translating the text into requests to specific service instances or classes. Personal preferences, login information as well as pre-defined objects are managed by the backend system.

Users may create service instances not only by direct interaction with a WebWall, but also by accessing the backend system via a web-interface. This way, many service classes – like picture *galleries* or personal *videos* – can be costumized and saved for display on a WebWall at any later time. Besides the user related data, the backend system hosts the code for the service classes and the renderer classes that are downloaded to Java-based clients whenever they start up. It also handles configuration sets for individual clients that define the services that should run, as well as the display areas where individual instances may appear on a visual display. This central storage of configuration sets and class code enables application providers to implement new service classes and distribute them to a defined set of clients.

WebWalls support a range of service classes that differ in presentation as well as in functionality (see Figure 1). The most basic service is the one for posting notes (service class *Note*) to a WebWall that can be viewed by everyone in the spatial proximity of the (public) display. Replys to a note may be sent to a WebWall, which, depending on the reply mode, either display on the WebWall or are routed invisibly to the author of the referred note. After a defined lifetime, notes are removed from the WebWall. While notes may be posted instantly when viewing a WebWall, there are other service classes that are better defined first over the Web-client: *Video* and picture *galleries* (service class *Gallery*) can be used to display multimedia content by composing URLs of the media to display and save them under a userdefined names. *Polls* may be used to solicit the public opinion on local issues that may arise in the geographical vicinity of a WebWall, for example. Polls display an up-to-date view of the current collective opinion, thus providing an effective means for instant democracy. To allow for ad-hoc buy&sell applications and commercial advertising the framework provides two service classes: *Auction* and *Banner*. Banners work analogously to their WWW counterparts, but could be used to send vouchers to the interested reader upon request. Auctions lets users bid for an item on sale, with the highest bid being on display on the WebWall.

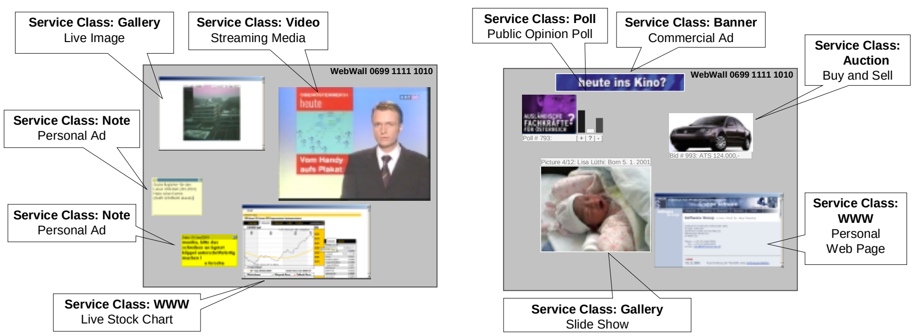

Fig. 1. WebWall and service classes

The software architecture of the WebWall framework can be separated into four major entities (Figure 2, grey rectangles) and two interfaces (Figure 2, red rectangles). The *Request Generator* (RG) module on the access side, and the *Backend System* (BS), the *Community Management System* (CMS) and the *Show Module* (SM) on the processing and display side.

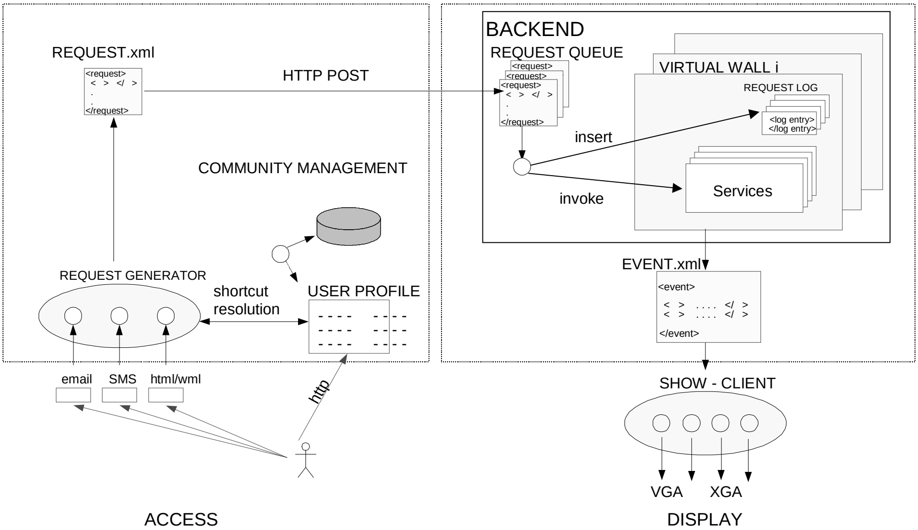

Fig. 2. WebWall software architecture overview

The *Request Generator* module accepts incoming requests from different media types (currently SMS, E-mail and HTTP) and prepares those requests for further processing. The RG is responsible for transforming each request into its XML representation (request.xml) which is the input for the BS. The RG logs all incoming requests, compares them with profiles managed by the CMS and adds data from the CMS where necessary (profile, resolves shortcuts, etc). A *request.xml* file is the interface between the RG and the BS. It is created by the RG and holds information of a specific request of a specific service class. request.xml is being passed from the RG to the BS via a HTTP-Post-command. In the BS the data of this file will be transformed into a JDOM-Object for further processing [15]. The *Backend-System* processes the request.xml files generated by the RG. It is the most complex module of the whole WebWall-system. It manages so called *Virtual WebWalls* (VWWs) and permanently stores the state of each VWW. It is responsible for creating, scheduling (including a waitlist which also can be displayed on the WebWall if needed) and positioning all service-class-instances, generating HTML-content for each visual representation of an instance and creating the file event.xml which contains display-information for the SM placed on a specific WebWall client. The BS can be configured via a WWW-Interface. The *event.xml* file is being created by the BS each time a SC is due to be displayed on a WebWall. It contains data which is needed by the SM to successfully display a service class. Like request.xml this file is sent to the SM via a HTTP-Post-command, which means that the BS contacts the SM every time a event.xml is generated and transmits that file. The *Show Module* component is responsible for displaying the results of BS processing. It receives standardized event.xml documents which are passed over from the BS via HTTP/POST. As this passing happens in a “push” oriented manner, the SM has to provide rudimentary HTTP server functionality, in order to listen to incoming requests from BS. As the content of the event.xml is classified by MIME types, the SM has to provide functionality in order to interpret the corresponding data formats. The current prototype is based on the “text/html” MIME type as a visualization format for service instances [13]. The SM provides the functionality to interpret and render the html encoded content passed over with the event.xml, creates browser (IE) instances containing the HTML-content rendered by the BS, and displays the WebWall service within an IE-instance (neither contain menu-bars, task-bars nor status-bars, only the common browser-window is being displayed). The *Community Management System* is implemented as a WWW-interface for users and administrators of the system for customisation and administration. Customisable features include the definition of user profiles, shortcuts (representation of longer strings or even whole services for easier handling via SMS), styles (colour and text properties), picture-upload and instant or scheduled posting.

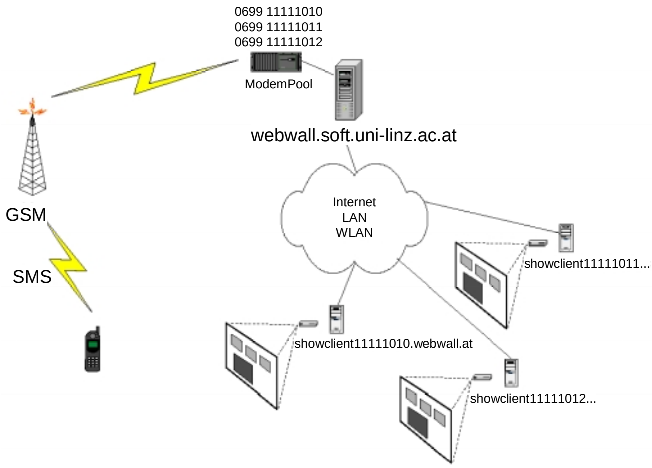

Fig. 3. Virtual WebWalls in a distributed multiple display setup

The WebWall system can either be distributed on physically different servers/clients, or be set up on one single machine to satisfy different applications of this service (for example a system managing many WebWalls used in Underground stations). All modules communicate with each other over HTTP gateways and pass data in XML format (request.xml and event.xml). As an example for a distributed WebWall setup, Figure 3 shows a configuration of a geographically dislocated WebWall system. It illustrates the possibilty to operate multiple WebWalls (Show clients), which are distributed over the Internet (LAN, wireless LAN, etc.). All clients are managed by one server with the WebWall-Backend component installed. The example depicts the centralized request reception via SMS with a modem pool attached to the WebWall server. Three individual *Virtual Web Walls* are maintained and their content is displayed at possibly three different physical displays.

3. Visual Components and Styles

A variety of different visual components have been created for the individual service classes, some of which are displayed below. For the video service class (Figure 4, upper left) the streaming video is displayed in the main frame. The service instance id is placed in the upper right header and can be used to stop, replay or removie the video. The gallery class (Figure 4, upper right) overlays image by image out of a collection of objects in `img` MIME type from the CMS. The auction class (Figure 4, lower left) displays an image and description text of an entity upon which an auction is set up in the public. New bids are posted by referring to the instance id, and once registered by the WebWall overwrite the current bid tag. An opionion poll like e.g. the evaluation of presentation by the audience accepts votes for one of the displayed alternatives (Figure 4, lower right), counts the votes, computes percentiles and displays the information in real time.

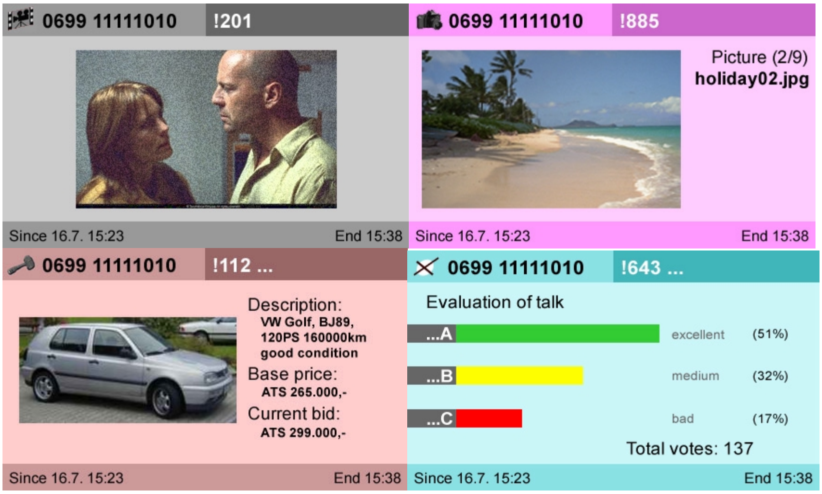

Fig. 4. Service class visuals: video, gallery, auction, poll

4. Scenario: Posting a Note to a WebWall via SMS

To illustrate the operation of the WebWall system, let us consider the creation of an instance of the service class "Note" on a WebWall identified by the MSISDN "+436991001010". Assume a person wants to post the text "Hello WebWall!" in the style "blue" with a GSM mobile phone via sending an SMS message (Figure 5). It is assumed that at the moment the WebWall does not contain any information elements (i.e. no request has already been posted to WebWall), that the requesting user is identified by the MSISDN 436645229250, and that the request to post a note "Hello WebWall!" to the WebWall has been issued at the time 10:29. In the first step the RG translates the incoming SMS information into a request document that looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<request id =”1” timestamp=”9897668787” >
 <user authType=”sms” id=”436645229250”/>
 <service command=”new” class=”note” screen_id=”436991001010”>
       <parameter name=”text” value=”Hello WebWall!”/>
    <parameter name=”style” value=”blue”/>
    <parameter name=”lifetime” value=”30000”/>
 </service>
</request>
```

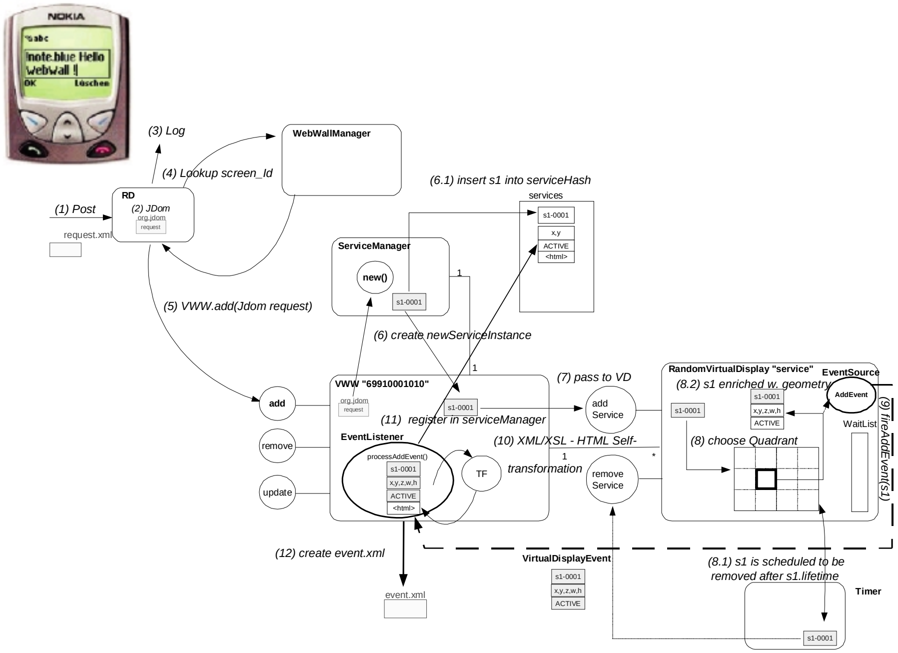

Fig. 5. SMS to WebWall 436991001010

To accomplish the request, the BS executes the following steps:

1. As a first step, after the SMS message has been received, parsed and transformed into a request file, request.xml is submitted via HTTP/POST from the RG to the WebWall BS, where the RequestDispatcher servlet receives (1) the incoming request as a *java.io.InputStream*. The RequestDispatcher creates an *org.jdom.Document* representation of the request based on the incoming string (2), and logs the request (3) appropriately.
2. With the information stored in the JDom Object (i.e. the *screen_id* attribute and the command attribute) the RequestDispatcher first checks whether the specified screen_id matches a VirtualWebWall instance within the Web-Wall Manager (4). If it exists, it delegates further processing to the specified VirtualWebWall instance. In this case it is the VirtualWebWall which is identified by the key “+436991001010”. As the command attribute of the request specifies that a “new” instance of a service class note should be created, the appropriate public

add(org.jdom.Document) method of this VirtualWebWall instance is invoked (5) with the org.jdom.request object as parameter.

3. The VWW receives the *org.jdom.Document(request*) and asks its ServiceManager to create a new instance of a service class. According to the class attribute of the service Element, the ServiceManager calls its internal newNoteInstance method to create an instance of a Note (6). Assumed that this is the first request to the WebWall Backend, the initial value of RNum is “0001” (6.1). This number is the identifier of the just created Note instance, and is used as key for the storing in the ServiceManagers Hashtable (services) for later reference. The service instance is inserted into this Hashtable (6.2).
4. The Note instance is returned to the VWW(7.1). The VWW passes the Note to its default VirtualDisplay(VD) (7.2) where the layout/geometry allocation is done: As the VD is empty, a random Quadrant is chosen (8.1) and the Note is being enriched with the appropriate geometrical data (8.2) and the status field of the Note is set to “ACTIVE” describing that the ser-vice is actually being displayed on the WebWall. Also time relevant fields of the Note (ActivationDate, ExpirationDate) are set accordingly. The Note instance is scheduled within the VDs internal Timer according to its lifetime attribute. In this case a lifetime of 5 minutes (300000 milliseconds) is as-sumed as the default lifetime of the service class Note. After expiration of this lifetime, an automatical call of removeService will be fulfilled.
5. The VD invokes its fireAddEvent method, in order to indicate that a VirtualDisplayAddEvent has occured (9). The listening (VirtualDisplayListener) VirtualWebWall is notified that a service instance has been placed on the VirtualDisplay as the VD invokes the processVDAddEvent of the VWWs VirtualDisplayListener (10).
6. Based on the event that a change in the VD has occured, the VirtualDisplayListener invokes the process of rendering a HTML representation of the Note instance. First a XML representation of the service instance is generated (11.1), which in turn is the basis for the XSL transformation into the HTML representation (11.2). Subsequently, the invokation of the to-EventXml method of the service leads into the generation and pushing of the event.xml to the associated waiting Show module, which in turn uses the event.xml to display the Note instance on the physical projection display. In this scenario the corresponding event.xml looks like (HTML code omitted):

```
<?xml version="1.0" encoding="UTF-8"?>
<event requestId="1"
   id="1000456158960"
   serviceId="436991001010-0001"
   class="note"
   command="new">
    <geometry x="30" y="480" z="1" w="200" h="150"/>
    <content type="text/html"><![CDATA[]]></content>
</event>
```

The positioning within the WebWall projection display depends on the associated geometry data. In this case the Note is positioned at the coordinates (x=30,y=480) with a dimension of (width=200,height=150), and z-order=1 within the display of the WebWall.

5. Related Work

The use of the wall metaphor as a means to enable “shared artifact interaction” among humans is not new. Indeed is wall computing research closely related with artifact research enabling collaboration and awareness [10,11]. At least for awareness solutions for workgroups quite a few wall metaphor based approaches have been proposed in the literature. We briefly discuss the main efforts and relate the WebWall approach to them.

The *Notification Collage* (NC) [8,11,12,23], developed at University of Calgary, Canada, is a groupware system that provides ways to post media elements onto a real-time collaborative surface (display). It is inspired by the metaphor of a bulletin board containing a collage of randomly positioned and possibly overlapping visual elements like sticky notes, video (used for providing virtual “presence” through Webcams), slideshows, or activity indicators to provide a means of indicating the amount of activity at a persons site through collected (proximity sensoring) movement data in the appropriate users room. The emphasis of the Notification Collage lies in the consideration of support of interpersonal awareness and interaction within small communities of colleagues. It supports both co-located and distributed members of a group in the means of providing functionality to allow all group members to post, notify other users about, and use shared resources within the context of a Notification Collage. The Notification Collage is based on conventional PC and Internet technologies and incorporates MS Windows as operating system in order to provide broad support for various multimedia formats. Display technology incorporates both standard monitors for workstation oriented dual screen setups, and large projection displays for public applications.

DATA Wall is a project at MIT Media Lab [18], which aims to overcome limitations given by conventional display technologies. The DATA Wall has a resolution of 2048x3840 pixels and provides a seamless, gapless fullmotion ultra high resolution projection display. It can be used in either front or rear projection, direct or folded optics mode. The main scope of the DATA Wall is to provide a projection facility with variable and configurable displays. It therefore can be used as a physical design component for various projection applications.

LIDS (Large Interactive Display Surfaces) is a project of the University of Waikato; New Zealand [1]. The LIDS project explores applications of large display surfaces for teaching and distance learning, meeting support, and personal information management. For teaching, and particularly distance learning, the focus lies on the ability to efficiently record, retrieve and disseminate lectures, seminars and tutorials, with almost no additional effort on behalf of the teacher. Similar technology is applied to the support of meetings and tutorials conducted over multiple sites, and this technology will be adapted to support both informal and formal face-to-face discussions and meetings. The system consists of a rear projected glass screen with standard data projector and a Mimio whiteboard digitizer.

The University of Minnesota in collaboration with Silicon Graphics Inc., Ciprico Inc. and IBM Storage Products Division have developed *PowerWall*, a high performance, high resolution visualization system [21]. The purpose of the PowerWall is to visualize and display high-resolution data from large scientific simulations. In addition to this high resolution, the PowerWall provides a large 1.8 x 2.4 m display area to facilitate collaboration of small groups of researchers using the same data set. All the collaborators can see the display clearly and without obstruction, and the rear-projection technology makes it possible to walk up to the display and point to features of interest, just as one would do while discussing work at a blackboard. The PowerWall is a single 1.8m x 2.4 screen driven by a 2x2 matrix of video projectors. A successor project called *InfinityWall* has been portrayed in [5], providing a 2048x1536 pixel stereoscopic display for large audiences.

The *HoloWall* [19,20] is a wall-sized computer display that allows users to interact without special pointing devices but by gesture recognition. A rear-projection setup is used in combination with an infrared light source and an IR-camera that films the back side of the display. Since the rear-projection panel is semi-transparent, the user's hand or any other objec in front of the screen reflects IR light and thus becomes visible to the camera, if it is close enough (somewhere between 0 and 30 cm). Image processing algorithms are used to track the shapes and cause interaction with the displayed contents.

i-LAND [10,26] integrates several so-called roomware components into a combination of real, physical as well as virtual, digital work environments for creative teams. By roomware, computer-augmented objects in rooms are considered, like furniture, doors, walls, and others. The current realization of i-LAND covers an interactive electronic wall (*DynaWall*), an interactive table (InteracTable), computer-enhanced chairs (CommChair), and the Passage mechanism. The objective of the DynaWall is to represent a computer-based device that serves the need for being able to interact with virtual content in an intuitive way, relying on standard gestures known from the interaction with physical objects in the real/paper world. It is possible that information objects can be taken at one position and put somewhere else on the display or thrown from side to the opposite side. These features are realized by an advanced interaction mechanism based on the penguin concept [9]. The current realization uses three rear projection electronic whiteboards (SMART Boards [25]) with a total display size of 4.5m width and 1.1m height and a resolution of 3072x768 pixels.

The *Stanford Interactive Workspaces* (SIW) project (also called i-Room) [7] is exploring the integration of high-resolution wall-mounted and tabletop displays (Interactive Murals [14], Interactive Tables), as well as personal mobile computing devices such as laptops and PDAs connected through a wireless LAN. Specialized input and output devices such as LCD-tablets, laser pointer trackers, microphone arrays and pan-and-tilt cameras are also present in the environment. The Interactive Mural is a large, high-resolution, tiled display, constructed using 8 projectors connected either to a SGI dual-pipe IR or a cluster of 8 myrinet-connected PCs with NVIDIA graphics cards. A scalable graphics library has been designed and implented that provides a single virtual display abstraction to the programmer, regardless of physical display properties (multiple overlapping projectors, multiple independent graphics accelerators and multiple processors). The framework supports multiple people, and pointing devices in interactive spaces including dynamic configuration and deals with failure, removal, addition and reconfiguration. Another research focus lies in interaction styles and assoicated toolkits that are appropriate for large displays, multiple devices, and multiple users.

Using the Stanford Interactive Mural, Davis and Chen [6] present new input methods for people collaborating one this shared display area. They use laser-pointers as input devices and are able to discriminate between several simlutanouos input gestures to enable a natural interaction. A similar interaction technique is used at the Fraunhofer institute [27].

The *Scalable Display Wall* (SDW) at Princeton University is another exponent of large-screen tiled display architectures, delivering 6000x3000 pixels via 24 aligned projectors, driven by an array of desktop computers. Their research efforts focus on frameworks for clustered displays, especially performance and scalability issues [4] for high performance data visualization.

The Table 1 summarizes wall computing projects and compares them with the WebWall approach. Most of the research projects presented deal with problems in display technology or CSCW-related issues, especially with closely coupled multiple users (workteams, interest groups) and appropriate interaction metaphors [2].

Table 1. Comparison of research work in wall computing

	NC	DATA Wall	LIDS	POWER Wall	HOLO Wall	DYNA Wall	SIW	SDW	Web Wall
Application area	CSCW	visualiza-tion	CSCW / CSCL	scientific visualiza-tion	ubiquitous computing	CSCW	CSCW	scientific visualiza-tion	public communica-tion, CSCW
Display	Conventio-nal monitor, rear projection device	proprietary projection device	rear projection screen, whiteboard digitizer	rear projection (2x2 matrix) video projectors	rear projection (IR cut filter)	rear projection whiteboards (SMART board)	rear projection board, tiled video projectors	tiled video projectors	independent of projection system, video projectors light emitting displays
Access	direct	passive	direct	direct	direct	direct	direct	direct	direct
Interface	Windows GUI	none	pen like interaction devices	touch screen	IR LED emitter, IR filtering camera with image processing techniques	touch screen, gesture techniques	LCD tablets, laser pointer trackers, microphone arrays, pan and tilt cameras	trackers	HTTP WAP SMS EMS MMS
Internet linkage	yes	no	no	no	no	no	no	no	yes any MIMEtype
Scalability	high	single system	medium	single system	single system	single system	medium	single system	high
Extendi-bility	high	none	none	none	none	none	medium	none	high
Status	under developmt.	idle	prototype	idle	research	commercial	under developmt.	under developmt.	fully functional

The Notification Collage is more related to the WebWall system than other wall computing projects, as it supports different media types that can be shared in a distributed setup. In contrast to WebWalls, it is closely coupled to a specific operating system for its providing its services, and lacks the possibility to access it via wireless phones (GSM) or other mobile devices – keyboard access is necessary to interact with the system. Furthermore, contents looks different on every screen, as users may arrange items at will, whereas a WebWall may be exported to different clients resulting in the same view on the data. The DataWall focuses on questions of abstracting logical from physical displays to construct larger interactive areas, and does not take into account any networking aspects, but could be used as an ouput medium for WebWalls. The POWER Wall as well as the InfinityWall deals only with high-performance data visualization problems and local multi-user interaction, networking is not taken into account. Likewise, the Scalable Display Wall focuses on clustered rendering of 3D content and uses networking only to distribute internal data sets. LIDS uses a whiteboard metaphor for user interaction so users need to have physical access to the wall and a pointing device, while WebWalls can be used from anywhere with a mobile phone or the Internet. Similarily, DYNA Wall allows access to networked data but needs direct interaction with the physical device, with binds the user to a specific location. SIW supports different pointing, input and output devices, but makes also heavy use of the room metaphor – only taking into consideration objects that are in a room. HOLOWall is a singular system that explores an alternative input technology and does not deal with networking, different service classes or other multi-user considerations.

There exist several notification services that transport information from the Internet or other data sources to mobile phones [28], like the various info services of GSM network operators. Another example of Internet/phone integration is iValet [17]. It informs users of incoming emails and lets the user react to individual mails. These examples are a strict one to one type of communication, there is no ability to share information with others or even publically.

Current research efforts can therefore be summarized as concentrating on three major areas: *Display technology* research covers advanced uses of projection systems - often in combination with cameras for system feedback - to provide seamless output of multiple beamers on arbitrary surfaces, even deliberately integrating physical objects into the digital realm [22]. Several architectures for the configuration, calibration and transparent access of a multi-display Wall have been proposed [4,14]. Projects focusing on *groupware* issues deal with the interaction of a known group of users on a shared display, using a variety of input devices. These efforts deal with the cooperative manipulation of artifacts on a shared display, dealing with privacy [24]. The size of the displays creates new problems for *human computer interaction,* as normal keyboard and mouse input becomes impractical (if not impossible). Therefore, new input devices have been proposed like laser-based pointing devices [6,27].

The WebWall project, in contrast, makes use of a variety of dislocated displays to enable the ad-hoc communication and interaction of people with on another as with Internet-originated artifacts.It makes use of large displays as one possible output technology, but does not limit itself to this presentation medium. Instead it can be adapted to a wide variety of interfaces.

6. Conlusions

In this paper we have presented a framework architecture enabling multi-modal access to multimedia information sources over wireless as well as fixed networks for the purpose of communication and interaction in the public – employing the wall metaphor. By separating data from access and display technology, WebWalls provide an open, flexible, extensible architecture that offers instant access to Internet information sources on an ad-hoc basis. The access to information and the direct interaction among possibly anonymous users in public spaces is novel and unique. Furthermore, the WebWall framework extends GSM network services from simple synchronous voice-streams to an interactive WWW services with advanced multimedia and broadcast capabilities.

Fig. 7. WebWall location scenarios

Possible locations for the use of WebWalls as considered by GSM network operators are envisioned in Figure 7: in public waiting areas for general public communication and access to multimedia information; for large assemblies as public opinion polls or for democratic interaction with digital systems (such as light shows), mobile WebWalls for location-based services enhanced by its notification capabilities, or as a source of up to date event information.

Acknowledgements. This work was supported by Connect Austria under grant WebWall. The valuable design and implementation contributions of Gerold Kathan, Maxl Miesbauer, Josef Blüml, and Daniel Simma are acknowledged.

References

1. M. Apperley, M. Masoodian: "Supporting Collaboration and Engagement using a Whiteboard-like Display." Techn. Report, Department of Computer Science, University of Waikato, Hamilton, New Zealand, 2000. http://www.edgelab.sfu.ca/CSCW/cscw_masoodian.pdf
2. W. Buxton, J. R. Cooperstock, S. S. Fels, K. C. Smith: "Reactive Environments: Throwing Away Your Keyboard and Mouse." Techn. Report, Input Research Group, Computer Science, University of Toronto, 1998.
3. M. Coen: "The Future of Human-Computer Interaction or How I learned to stop worrying and love my Intelligent Room." Techn. Report, MIT Artifcial Intelligence Lab., 1999.
4. H. Chen, G. Wallace, A. Gupta, K. Li, et.al., "Experiences with Scalability of Display Walls", To appear in Immersive Projection Technology Symposium (IPT), March 2002.

5. M. Czernuszenko, D. Pape, D. Sandin, T. DeFanti, et.al., “The ImmersaDesk and Infinity Wall Projection-Based Virtual Reality Displays”, Computer Graphics, Vol 31, No 2, 05/01/97-05/01/97, pp. 46-49, 1997.
6. J. Davis, X. Chen, “LumiPoint: Multi-User Laser-Based Interaction on Large Tiled Displays”, to appear in Displays, Mar 2002.
7. A. Fox, P. Hanrahan, B. Johanson, T. Winograd: “Integrating Information Appliances into an Interactive Workspace.” Techn. Report, Stanford University, 2000.
8. S. Greenberg, M. Boyle and J. LaBerge: PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal**.** *Personal Technologies*, Vol.3, No.1, March 1999.
9. J. Geissler: „Shue, throw or take it! Working efficiently with an interactive wall.” Techn. Report, German National Research Center for Information Technology, Darmstadt, Germany, 1999.
10. J. Geissler, T.Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P.Rexroth, P. Seitz, R. Steinmetz, N. A. Streitz: „i-LAND: An interactive Landscape for Creativity and Innovation.“ Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 120 – 127. ACM Press, Pittsburgh, Pennsylvania, USA, Mai 1999.
11. S. Greenberg, M. Rounding: “Using the Notification Collage for Casual Interaction.” Techn. Report, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, 2000.
12. S. Greenberg, M. Rounding: “The Notification Collage: Posting Information to Public and Personal Displays.” Techn. Report, Department of Computer Science, University of Calgary, 2001.
13. M. Grand: “MIME Overview”. 1993. http://www.mindspring.com/~mgrand/mime.html
14. G. Humphreys, P. Hanrahan, "A Distributed Graphics System for Large Tiled Displays", Proceedings of IEEE Visualization 99, San Fransisco, CA, October 24-29, 1999.
15. S. L. Jones: “An Introduction to JDOM.” XML Journal, 7. Sep. 2001.
16. U. Leonhardt: “Ubiquitous Location-Awareness.” Techn. Report, Imperial College UK, 1998.
17. S. Macskassy, A. Dayanik, H. Hirsh, "Information Valets for Intelligent Information Access", to appear in AAAI Spring Symposia Series on Adaptive User Interfaces, 2000.
18. R. E. McGrath, M. D. Mickunas: “An Object-Oriented Framework for Smart Spaces.”
19. M. Nobuyuki, J. Rekimoto: “HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall.” Proceedings of UIST'97, Sony Computer Science Laboratory Inc., 1997.
20. M. Nobuyuki, J. Rekimoto: “Perceptual Surfaces: Towards a Human and Object Sensi-tive Interactive Display.” Tech. Report, Sony Computer Science Laboratory Inc., 1997.
21 POWER WALL: http://www.lcse.umn.edu/research/powerwall/powerwall.html
22. R. Raskar, G. Welch, M. Cutts et.al., "The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays", ACM SIGGRAPH 1998, Orlando FL, 1998.
23. J. Rekimoto,”A multiple device approach for supporting whiteboard-based interactions”. CHI '98 Conference Proceedings, p. 344 -351, 1998.
24. Garth B. D. Shoemaker and Kori Inkpen, "Single display privacyware: augmenting public displays with private information", CHI 2001, pp 522-529, 2001.
25. SMART Technologies, http://www.smarttech.com/products/smartboard/
26. P. Tandler, “Software Infrastructure for Ubiquitous Computing Environments: Supporting Synchronous Collaboration with Heterogeneous Devices”, In: Proceedings of UbiComp 2001: Ubiquitous Computing. Heidelberg: Springer LNCS 2201, pp. 96-115, 2001.
27. M. Wissen, ” Implementation of a Laser-based Interaction Technique for Projection Screens”, ERCIM News No.46, July 2001.
28. T. Woo, T. La Porta, „*Pigeon: A Wireless Two-Way Messaging System* “, IEEE Journal on Selected Areas in Communications, Vol.15, No. 8, 1997.