

The ProD Framework for Proactive Displays

Ben Congleton¹, Mark S. Ackerman^{1,2}, and Mark W. Newman^{1,2}

¹School of Information

²Department of Electrical Engineering and Computer Science

University of Michigan

1075 Beal Avenue

Ann Arbor, MI 48109, USA

{bcx,ackerm,mwnewman}@umich.edu

ABSTRACT

A proactive display is an application that selects content to display based on the set of users who have been detected nearby. For example, the Ticket2Talk [17] proactive display application presented content for users so that other people would know something about them.

It is our view that promising patterns for proactive display applications have been discovered, and now we face the need for frameworks to support the range of applications that are possible in this design space.

In this paper, we present the Proactive Display (ProD) Framework, which allows for the easy construction of proactive display applications. It allows a range of proactive display applications, including ones already in the literature. ProD also enlarges the design space of proactive display systems by allowing a variety of new applications that incorporate different views of social life and community.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: proactive displays, public displays, awareness, audience-awareness, software framework, ubiquitous computing, pervasive computing, collaborative systems, CSCW.

INTRODUCTION

Public electronic displays are cropping up everywhere. While typical use is currently dominated by entertainment, advertising, and information dissemination, there remains a compelling promise that such displays can be used to facilitate social awareness and interaction. Indeed this has emerged as a robust research theme within the ubiquitous computing and computer-supported cooperative work communities over the past decade.

An important class of public display is a proactive display—an application that selects content to display based on the set of users who have been detected nearby. Proactive display applications can be traced to McCarthy et al.'s

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST '08, October 19–22, 2008, Monterey, California, USA.

Copyright 2008 ACM 978-1-59593-975-3/08/10...\$5.00.

formative work on displays that adapt their content based on the presence of one or more users [16, 17]. They have been found to be useful in practice [19], but until now the design space of proactive displays has been under-explored because each proactive display application had to be constructed by hand.

It is our view that the previously reported examples of proactive display applications are sufficient to paint a picture of the most promising patterns for the design and construction of proactive displays. Now we face the need for a software framework to support the range of applications that are possible in this space.

In this paper, we present the Proactive Display (ProD) Framework, which satisfies just such a need. It allows a range of proactive display applications, including ones already reported in the literature. As well, ProD allows a variety of new applications that incorporate different views of social life and community, and thus it also enlarges the design space of proactive display systems.

This paper begins with an introduction to proactive display applications, their design space, and the relevant literatures. It then explains the ProD framework in detail, and follows with a presentation of interesting applications that have been built atop ProD. The paper finishes with some discussion of ProD's strengths and limitations and a brief conclusion.

The Proactive Display Design Space

As mentioned, a proactive display selects content based on the set of users who have been detected nearby. The choices made in performing this mapping have a significant impact on the social interactions fostered by the display. To illustrate the design space of possible proactive displays applications, we begin with a discussion of several previously implemented systems.

AutoSpeakerID [17] represents perhaps the simplest possible proactive display application. It detects a single user (the most recently detected user) and displays a single content item associated with that user (a tuple consisting of name, affiliation, and photo). The primary purpose of such a system is to introduce the “present” user to all other viewers under the assumption that many of the viewers will not already know him or her. The original AutoSpeakerID system was used to identify audience members asking questions in a conference setting.

Ticket2Talk [17] is a slight variation on AutoSpeakerID where “present” users are entered into a queue to take turns presenting content they would like to talk about on a display. At any given moment Ticket2Talk displays the name, affiliation, and one pre-selected content item representing the current user’s interests, as well as a visual representation of the list of users waiting their turn. Ticket2Talk allows only a single, static content item to be associated with each user. Similarly to AutoSpeakerID, Ticket2Talk is designed to introduce people who have not had much prior interaction with each other, with the additional goal of providing them with a topic to start an initial conversation.

From these two simple examples, one begins to see that a range of choices can be made regarding “presence” (who is considered *present* for the purposes of selecting content to display) and regarding content selection. If, for example, users are given a wider range of choice in selecting their own preferred content, the range of possible social interactions is widened as well. Supporting such varied interactions is important when considering the range of situations into which public display applications may be deployed. Previous examples range from conferences with hundreds of loosely connected attendees (e.g., [17]) to tightly-knit workgroups of a dozen or less (e.g., [8]). In the face of such variety, a number of common patterns and mechanisms emerge that point towards a need for abstracting common functionality into a set of reusable components.

ProD, then, facilitates building systems within this design space, allowing the construction of a range audience-aware public display applications, including those with different social goals.

RELATED WORK

There is a large literature on public displays. This work has included interaction techniques and frameworks [3, 24, 27], social awareness [7, 8, 19], collaboration [22], and the sharing of multimedia content [5, 9].

The literature on proactive displays is considerably smaller. “Proactive displays,” as a term, was introduced McCarthy, et al. in [17]. Proactive displays differ from systems like BlueBoard [22] and Dynamo [21], which provide methods for users to manually log into the system, but are not aware of users who have not explicitly signed in to the display. In addition, systems like BlueBoard and Dynamo are focused around supporting explicit interaction with the display’s content, whereas proactive displays emphasize implicit interaction along with the display of information.

The existing proactive display literature primarily focuses on the discussion of specific proactive applications, with an emphasis on the evaluation of their use. There has been very little attention to system design.

Proactive applications include Ticket2Talk, Neighborhood Window, and AutoSpeakerID. We have already discussed Ticket2Talk and AutoSpeakerID above. They and Neighborhood Window [19] were three proactive display systems deployed at UbiComp 2003 (a medium-sized academic conference with approximately 500 attendees) using RFID

tags as an identification mechanism. As mentioned, Ticket2Talk allowed conference attendees to specify an image to talk about when they were near a proactive display. In AutoSpeakerID a RFID reader was attached to the microphone in an auditorium and used to display information about audience members asking questions. Neighborhood Window was arguably the most information-rich of these three systems, presenting connections between the interests of nearby users as a network visualization.

All of these applications centered on individuals showing something about themselves. Some work has also been done on rendering information based on the group of users located nearby. MusicFX [15] was an early proactive system designed to aggregate the music listening preferences of multiple individuals in order to determine the music to be played in a corporate gym. MusicFX used a simple collaborative filtering approach to select the music to play, but had no knowledge of the underlying social relations between users of the system, or even interests on more dimensions than music. Groupcast [16] was another display system designed to recommend content (e.g., web sites) to several present users, but its effectiveness was limited by underspecified user profiles. The webPendle application [26] allows users to define a set of keywords describing their interests. When one or more users are detected near a display, a set of web pages based on the users’ keywords is displayed, thus endeavoring to show a set of content that will be of interest to the group.

Our work draws on several additional research streams in addition to that of proactive displays. Because of our interest in collaboration and the social use of these systems, we necessarily also draw on group recommendations and collaborative governance.

Briefly, a proactive display application must make decisions about what content to place on the physical display. This can be a form of group recommendation. A discussion of recommendations to groups can be found in [20] and [10]. Masthoff [13] discusses strategies for combining individual user models to form group models for recommending television programming.

Governance, or how groups (or larger collectivities) are structured and governed, plays a subtle role in these systems. It operates in several different ways. At a large scale, as Kling [11] observed, systems can incorporate the political stance, or value orientation, of their designers. The most prevalent case in computer applications tends to be one of libertarianism. Indeed, many existing proactive display applications follow such a governance philosophy, since they allow each user to put whatever he or she wants as content onto the screen. Kling pointed out that one could imagine other polities, however. One could imagine including other governance policies, for example, ones that promoted social cohesion, social identity (as opposed to individual identity), or even social control. At a lower level, these governance orientations play out in the forms of matchmaking that are used [25] and the views of social networks that are incorporated [18]. ProD provides for

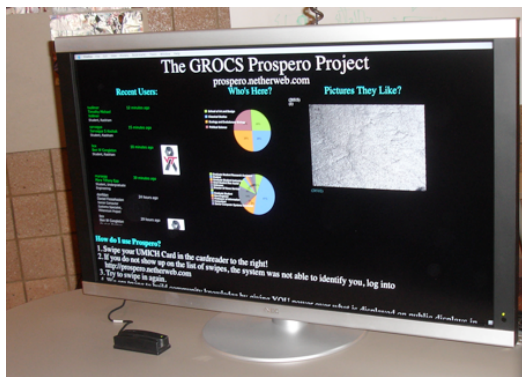


Figure 1: The University of Michigan Prospero system allowed developers to create “widgets” that would adapt their contents based on information and preferences provided by the users that were detected nearby.

differing governance orientations, and allows a restricted range of these to be incorporated into proactive display applications.

PROJECT HISTORY

The ProD framework builds upon lessons learned during the development of two prior display frameworks—the Michigan Prospero system, developed and deployed at the University of Michigan (shown in Figure 1.), and the Context, Content, and Community Collage (C4) developed and deployed by the first author with Joseph F. McCarthy and others at Nokia Research Palo Alto [14] (shown in Figure 2).

The Michigan Prospero System

The Michigan Prospero system (Prospero) was an experimental framework for rapidly exploring a variety of proactive displays. Prospero was designed to facilitate third-party development of presence-based widgets: small applications that react to nearby users. Users of the Prospero system specified personal profile information and preferences, e.g., keywords of content from Flickr. Four developers used this platform to create 17 simple widgets, the most popular of which being a widget that projected Flickr images based on user preferences. It was during the creation and refinement of Prospero that we identified the basic outline of a proactive display framework: presence, processing, and presentation. Users and developers responded positively during our initial three-week deployment, but Prospero was discontinued in favor of C4 below.

The Nokia C4 System

C4 was a focused proactive display application that supported the display of user-specified content feeds, rather than a system supporting a wide variety of display applications. C4 followed a simple presence, processing, and presentation model, with additional abstractions for user-specified content feeds. These abstractions provided the tools necessary to build content feed plug-ins to support a wide variety of data sources including Yahoo Pipes, Last.fm, Flickr, PhD Comics, and Google’s Picasa. C4 built collages of user selected content based on the pres-



Figure 2: The Nokia Research Context, Content, and Community Collage (C4) allowed users to specify content streams (e.g., RSS feeds or Flickr photo streams), that would be tapped for content items to display whenever they passed nearby. Groups of nearby users would each add content to the screen, with shifting priorities over time.

ence of nearby users. For example, if two registered users, John and Beth, were near the display, C4 might show some Flickr photos from John’s recent trip to Florence and Beth’s favorite comics. C4 was deployed at Nokia Research Palo Alto for about 10 months. Like Prospero, C4 provided an interface to allow third-party developers to create new proactive display modules. Although in C4 the user experience was made more cohesive by restricting each module to providing a stream of content items to a standard presentation engine.

The longer C4 deployment made the importance of designing a framework to support the needs of both social and technical systems more apparent. Early in the deployment of C4 we recognized the need for users to be able to quickly filter their content feeds, since “public” on Flickr did not translate directly to public for sharing in the office. In addition, because C4 was deployed in a corporate research lab situated within the norms of a larger organization, content considered personally acceptable for display was not always appropriate by corporate standards. To address these issues we implemented functionality for individuals to flag inappropriate content on the displays and additional methods for filtering personal content before it reached the displays. These enhancements were largely reactionary and not incorporated as part of a larger C4 framework. ProD explores the need for a more socially aware proactive display framework.

THE PROD FRAMEWORK

Building on our experiences with Prospero and C4, we designed ProD to meet the following goals:

- Provide a simple, easy to use programming model to encourage the development of diverse proactive display applications.
- Abstract low-level details into reusable services to allow developers to focus on the display’s functionality.

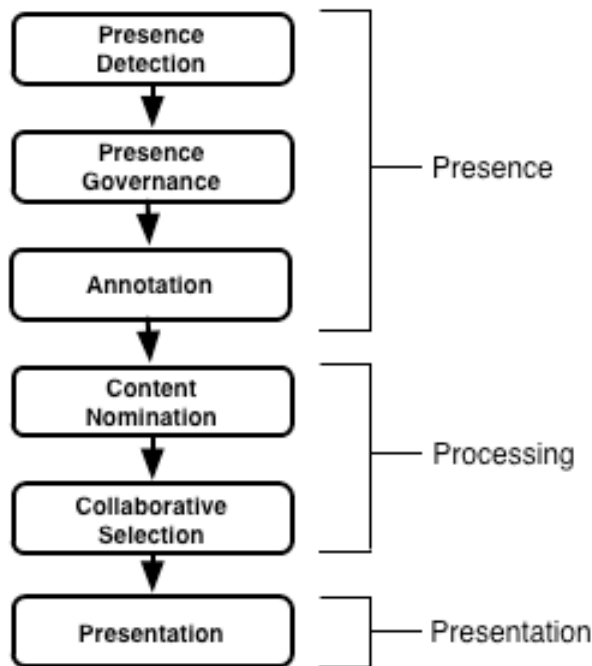


Figure 3: The ProD pipeline consists of 6 stages, each managing a different aspect of the detecting, selecting, and annotating of users considered “present”; processing user preferences and application rules for selecting content; and rendering the selected content to the display. Taken together, these stages cover what we consider to be the three major functional components of proactive displays: presence, processing, and presentation.

- Facilitate the fostering of different social arrangements in public display applications.

As we shall see, ProD’s features align with each of these goals. In the remainder of this section, we describe the pipeline programming model presented by ProD. Along the way we note stages of the pipeline where complex details (e.g., presence detection) are handled by pluggable modules whose default configurations provide useful services to developers but can also be easily replaced by alternative implementations. Following the description of ProD, we will discuss a set of applications that demonstrate the utility of our framework.

The programming model presented to developers is that of a pipeline that takes as input a set of users (detected via a proximity-based sensing such as RFID or Bluetooth) and produces as output a display comprised of various content items. Each of the intervening stages provide the developer with opportunities to focus on the policies that will affect the mapping of users to content, which will in turn affect the social interactions afforded by the application.

The ProD pipeline (see Figure 3) consists of six stages that are sequentially triggered at a regular interval (in our experience 2 seconds works reasonably well), each time generating a new set of content items to be displayed. The content set is based on inferences about the currently sensed set

of nearby users as well as processing rules that have been defined by the application writer.

The six stages are:

1. User presence is detected producing a list of present users.
2. Present users are filtered and prioritized by local policy enacted in a governance module.
3. The list of users who are present is annotated with social relationships, profile information, and display history.
4. Each user nominates content items according to their own preferences as well as application-driven rules in order to determine appropriate content candidates for display.
5. A collaborative selection mechanism then filters and prioritizes the content candidate pool.
6. The presentation engine renders the resulting content to the proactive display.

Presence: Determining Who Counts

The goal of the pipeline stages in charge of “presence” is to determine which users should be taken into consideration when making decisions about content to display, and to retrieve or generate additional information about those users that will assist in later stages’ rule processing.

Presence Detection

Detection of nearby users is the first step in any proactive display application. ProD supports the detection of users but is not tied to a particular sensing mechanism. It is designed to take input from a variety of sensor platforms; we have implemented support for Bluetooth and magnetic card readers, though other proximity-based mechanisms such as RFID or 802.11b could be implemented with relative ease. The result of the presence detection stage is a list of identified users, potentially including metrics regarding their relative proximity to the display.

Presence Governance

Given a list of detected users, the next ProD stage determine which users should be considered in further steps of the ProD pipeline. After presence detection, a presence governance engine allows normative inferences about who is considered to be present and how present users are prioritized. This is best explained with a few simple examples.

Consider a proactive display that presents news feeds, and suppose it is to be located in a common area along with a web-cam. Are users who are watching the web-cam considered present in the room? In this example, determining the presence of web-cam viewers is a technical issue, but deciding whether they count as being ‘present’ is an open issue, to be determined by the social norms and rules of the place where the display is situated. In McCarthy’s Auto-SpeakerID system, where proactive displays presented information about audience members asking questions at a podium in an auditorium hall, many users were near the

display, but only one was present for the sake of the Auto-SpeakerID application.

Present users can also be prioritized in many ways. For example, users who have just arrived at the display could be prioritized over those who have already had their turn, giving the newcomers priority over incumbents. One might prioritize incumbents for some other application. In some situations, simple, inflexible prioritization rules such as “when company executives are near the display, they get top priority” provide more nuance than that found in current proactive display deployment.

Priority is used throughout the ProD framework as a method of ascribing importance to users, rules, and content items. User priority is used primarily in the presentation stage of the ProD framework. These priorities help the presentation engine make decisions for situations with limited display resources. Priority is carried through the pipeline, however, and can be used in any stage.

Governance rules about user presence and prioritization can be re-used across multiple proactive applications. This can provide a simple mechanism for local social norms and policy to be integrated into proactive display applications.

User Annotation

In the annotation stage, ProD determines a set of annotations for users, both individually and collectively. These annotations add information about users to the system state, which can be used by the nomination and selection mechanisms to make more nuanced judgments concerning content to be selected for presentation. In the current implementation, ProD annotates each present user with the following types of information: social proximity determined through an analysis of the social relations (currently, a simple social network analysis), group membership (currently a list or organizational chart), a matrix of common interests between each present user (currently their tag clouds), and individual interaction histories including previously displayed content and time intervals spent near the display.

Application authors can extend the existing annotation mechanisms to introduce more elaborate techniques for determining social proximity, new methods of determining user affiliations, and more nuanced comparisons of user profile information. For example, our current implementation calculates a social proximity metric based on closed triads in a social network [28], but an application author might find the shortest path a better measure. Each user can also be annotated with multiple measures for each type of annotation (i.e. two users may be compared using multiple similarity measures).

The resulting annotations provide application writers with a rich set of information to use when creating the rules that direct the content nomination and selection.

Processing: Selecting Content for Display

At this point, ProD has an annotated set of users who have been determined to be present as well as a prioritization scheme. The next step in the ProD pipeline is the processing component (see Figure 4). The processing component

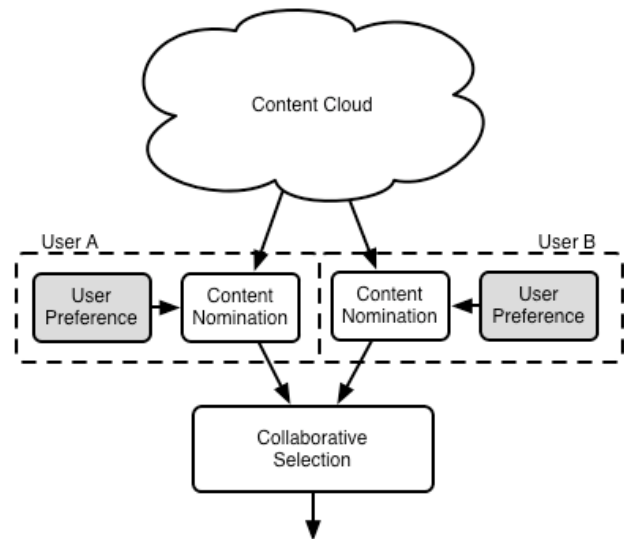


Figure 4: The processing stages in ProD begin by having all of the users who were deemed “present” in the early stages of processing nominate content items based on their personal preferences. The collaborative selection stage applies application-specific logic to select a final pool of content based on the annotated information about the present users, including such information as their social proximity, history, and relative priority.

of ProD takes this prioritized list of users as input, and produces a prioritized list of candidate display content. Processing to select content follows a two-step process: the nomination of content by each user who is present, and a collaborative selection step that filters and orders the candidate content items.

Content Nomination

The content nomination step of ProD processing takes a prioritized list of users as input and evaluates rules to produce candidate content pools for each user. These candidate content pools will be further filtered by the collaborative content selection mechanism (below).

Each present user begins the content nomination stage of the ProD pipeline with an empty candidate content pool, a collection of user-specified content feeds, a set of user preferences, and any annotations that might have been added in the previous stage.

A nomination rule uses system state (present users, relationships among users, and user-defined preferences if any) to select among content feeds and obtain candidate content items. Rules, then, are the primary method of mapping present users to content for later display. Each rule can perform add, intersect, tag, union, order, and filter operations upon a target set of content. These operations provide rule creators (i.e., application writers) with the tools necessary to combine content from multiple content feeds in a simple yet powerful way.

For example, if a given user had defined a content feed about his party the previous evening, he might want to

show it to only his friends. If the present users were not all his friends, he might want to show his other content. He would specify this in his user preferences, demarking two groups and their preferred feeds. The rules, then, would be:

```
present(friends_only) :
  candidates += pref(friends_only)
default :
  candidates +=
    allContent() - pref(friends_only)
```

These rules are evaluated by examining the user preferences. The first rule specifies providing content for a group called `friends_only` to that only group. The second rule specifies showing everything else as a default.

Rules can also be used to prioritize content. Two such orders include temporal ordering:

```
default:
  candidates = order_by_date(candidates)
```

and selecting content randomly:

```
default :
  candidates = order_random(subset(all,0,100))
```

Application writers can also specify general rules that are not user-specific. For example, an application developer wishing to deliver personalized notes to the display might write a rule to put the note on the display: "If user X is near the display, and user X has a pending note, show that note on the display."

At the end of the content nomination step, a prioritized set of candidate content items will have been produced. An application writer, however, may wish to further process his content pool in order to produce social effects. This is a goal of the next stage in ProD, collaborative selection.

Collaborative Selection

After content nomination, the collaborative selection stage provides a way for application developers to consolidate the individual user content pools. In contrast to the presence governance stage described above, which selects *who* is present for the purposes of nominating content, the collaborative selection phase makes the final decisions about *what* content to display. Application developers can again use system state to make decisions about how to combine the content of the individual candidate content pools for each present user. For example, a simple guideline such as "take the union of the individual candidate content sets" would provide for previously reported proactive displays. A more advanced content selection technique might involve selecting interests users have in common or prioritizing content for a specific high-status user.

The collaborative selection stage, then, filters and prioritizes potential content based on rules determined by the norms or preferences of the place where the proactive display is situated. The collaborative selection stage provides a convenient location for inappropriate content to be filtered, and for candidate content to be reorganized to fit a higher goal. For example, a community manager wanting

to use a proactive display to emphasize the work-life balance of an organization could prioritize pictures with tags related to families and vacations. Whereas a community manager who wanted to use the display as more of an information resource could prioritize content that was related to the company's core business. Perhaps most interestingly, one could construct applications that might help build social cohesion and better group dynamics.

The final output of the collaborative selection stage is an ordered list of content passed to ProD's presentation stage.

Presentation: Rendering Content to the Display

After content has been selected, ProD's presentation engine lays out the content on the display. ProD's presentation stage draws from previous HCI research separating presentation from content (e.g., [12]). A presentation engine determines relative layout following either heuristics or rules, based on such factors as display size, screen resolution, overlay policy, timing policies, and so on. Design of the screen layout may be based on aesthetic considerations such as an item's preferred display size. Finally, the presentation engine must also include prioritization decisions from the selection rules that preceded it.

Currently, we have implemented a number of simple presentation engines. Each presentation engine can create quite dissimilar looking applications. These presentation engines include: Maps, an engine that renders geo-coded metadata for present users on a map; Collage, a layout engine similar to that found in C4 that randomly places serial pieces of content on a display to create a dynamic collage, and Ticket, an engine that displays one piece of content at a time similar to the Ticket2Talk and AutoSpeakerID applications. The ProD framework makes it simple for application developers to construct new presentations that render selected content and present users in a variety of forms.

Implementation details

The ProD framework is implemented in Ruby using the Rails programming framework; the database layer is provided by MySQL. It contains approximately 12,000 lines of program code. The current implementation uses our Simple Sensor Architecture for Pervasive Prototyping (SSAPP) toolkit to detect user presence using nearby Bluetooth devices and swiped magnetic cards. SSAPP is written in Python and C and consists of 36,000 lines of program code. The presentation engine uses Ajax and the Yahoo UI toolkit. We plan on releasing the source code to both ProD and SSAPP after further hardening. Our prototype applications have been deployed on 42" plasma displays.

APPLICATIONS

One of the most valuable assessments of a software framework is the variety of applications that can be created using it. In this section we discuss a number of different applications that we have built using ProD. These applications demonstrate the ability of ProD to produce widely differing proactive display applications, while also demonstrating the utility of the governance and collaborative selection stages.

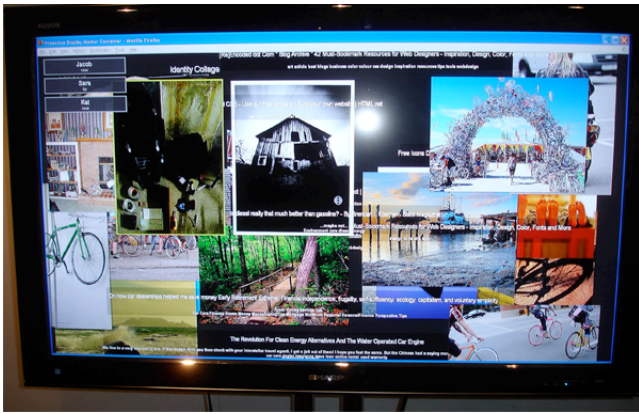


Figure 5: The Identity Collage content from delicious and Flickr based on tags set by the three users present at the display.

First, ProD can cover any of the previous proactive display applications described in the literature. MusicFX would require minor modifications, as it uses audio streams instead of video.

We have also implemented a number of new applications. The *Identity Collage* (see Figure 5) is a re-implementation of the C4 system described above. We will use this as the basis for discussing other new applications, so we discuss it in detail. Identity Collage allows users to select and filter a set of web-based data sources (e.g., RSS feeds, Flickr photo streams, Yahoo! Pipes results) from which content will be shown whenever they are near the display. The primary social goal of Identity Collage is to support the presentation of self by community members. A secondary effect is to foster some level of connection among community members by allowing individuals' interests and personal information (e.g., personal photos) to be projected into group situations. In terms of ProD's facilities, Identity Collage implements a simple policy for presence (i.e., any registered user who is detected near the display is given a subset of the collage's available display regions), with priority given to new arrivals. In addition, recent departures are given a gradually decaying slice of the display so that an audience that approaches the display shortly after a community member has moved may still view that member's selected content.

Other new applications show how ProD enlarges the proactive display design space by providing capabilities at each of its stages. The Remembrance Wall shows the power of the governance stage. The Cohesion Collage shows the advantages of the collaborative selection stage, as well as providing for annotations on present users. Finally, the Nearby application shows the utility of having a separate presentation stage.

The *Remembrance Wall* (see Figure 6) provides a very different social world [23] from the Identity Collage. The Remembrance Wall is designed to remind its audience about community members who have not been seen for a while. It employs a radically different presence policy by retaining a history of all community members ever seen. It

inverts the Identity Collage's priority ordering for present members by giving the highest priority to the community members who have not been seen recently, thereby highlighting content that will call those members to mind. Thus, rather than focusing on identity projection and fostering communication among currently present community members, the Remembrance Wall places an emphasis on the long-term and historical identity of the community as well as maintaining presence for community members who spend large amounts of time away from the shared community spaces. This application can be also be seen as the opposite of Huang and Mynatt's Active Portrait [8], which provides awareness about lab members who have been recently seen in the lab, while fading those who have not been seen from view.

Alternatively, by providing socially-aware rules at the collaborative selection stage, it is possible to create other new types of applications. These differ from one another in their assumptions about the nature of social cohesion and group dynamics.

One such application is the *Cohesion Collage* (see Figure 7). The Cohesion Collage extends the Identity Collage's secondary goal of enhancing social cohesion by selecting content for display that matches the intersection of interests among all members currently in front of the display. Thus, where Identity Collage allows users to explicitly project their interests (as well as personal information) to whoever happens to see it without regard to who the audience might be at any given time, the Cohesion Collage allows users to suggest data sources and interests (implemented in the form of tags and query terms) that match their preferences. The specific content items selected for display are then generated by the overlap in group members' interests. While the current implementation of the Cohesion Collage uses a simplistic technique for finding the intersection of interests—simply selecting the tags and data sources that occur most commonly among the set of collocated community members—we envision that greater sophistication in the identification of both shared interests and opportunities for fostering cohesion could lead to improved experiences for



Figure 6: The Remembrance Wall displays content related to community members who have not been seen near the display recently.

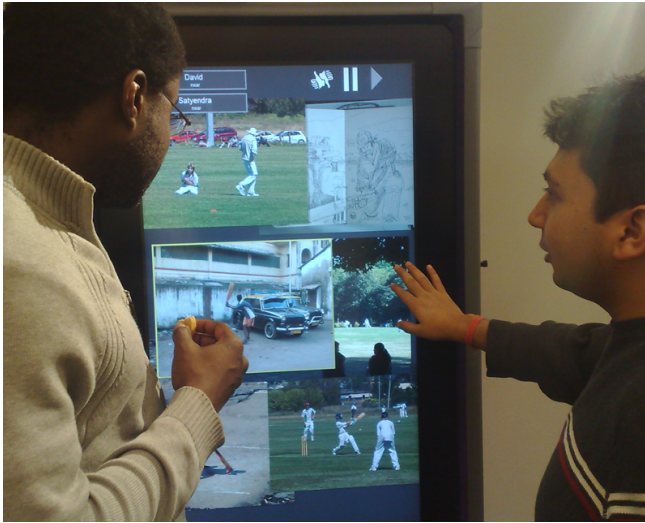


Figure 7: Two students see their common interest in cricket expressed on the Cohesion Collage.

community members. For example, by taking advantage of information about the community social network, intersections among collocated community members with the weakest ties (or the strongest, if so desired) could be weighted more heavily. Also, a small or nonexistent intersection of explicit interests among a set of community members could be resolved by searching the social network for community members with connections to and shared interests with each of the collocated members and displaying information relevant to the connections with him or her.

The Cohesion Collage can be seen as an extension of MacDonald, et al.'s Neighborhood Window [19], which displays a graph visualization of how a set of viewers are related to each other in terms of interests. However, the Cohesion Collage shows interests as content items instead of Neighborhood Window's textual labels, thus perhaps increasing the opportunity for discovery and exploration of shared interests. It extends Neighborhood Window by also adding social cohesion as the goal, rather than just the exposition of shared interests.

Finally, separating out presentation in different presentation engines can give a very different visual feel to applications. *Nearby* (see Figure 8) is an application built on top of the Maps presentation engine. *Nearby* provides additional context for in-person interactions near the display by rendering each present user's recently geocoded activity on a shared display. Copresent users share where they have been, and what they've been up to. Presently the content displayed by *Nearby* is limited to geocoded micro-blog content (i.e. GeorSS), but could be easily extended to other forms of geocoded content (i.e. Flickr).

As these examples show, ProD supports the construction of a range of proactive display applications. In each case, these applications can be designed, keeping in mind the social interactions desired by a community or group.

DISCUSSION

While we have expounded the ability for ProD to support proactive display applications, it is important to clarify the contribution provided by ProD and some of the limitations of the ProD framework.

ProD's primary contribution is a *software framework*, and thus does not address all specific aspects of the implementation of proactive display applications. For example, user interfaces for creating or managing profiles, setting preferences, and selecting content are not specifically addressed. In addition, we do not propose (or, say, provide a library of) any particular governance rules to dictate or influence particular social arrangements. We view the decision to constrain our focus as one of ProD's key strengths. By making such considerations orthogonal to ProD, we strive to provide a framework in which each of these challenges can be addressed somewhat independently from the proactive display pipeline.

While we believe ProD will be useful in creating proactive display applications, it has two important limitations. First, it is clearly restricted in the range of social worlds [23] it can create – it cannot handle all types of social arrangements. One might wish for unrestricted flexibility, for example, in adding and removing people from ad-hoc groupings. While people handle social arrangements like this seamlessly and with great nuance, systems cannot [1, 6].

ProD has no better (but no worse) methods of dealing with the problem of social nuance and flexibility. In order to create reasonable applications, we have necessarily restricted the scope of the kinds of social arrangements one can invoke. With ProD, one can create an application that fosters relatively simple notions of group and of group cohesion. We have done this because we felt that more nuanced notions are likely to run quickly into problems. Computational mechanisms, such as rules or social network



Figure 8: The *Nearby* proactive display renders recent geocoded RSS (e.g., Twitter-based) feeds using the map presentation engine.

metrics, are limited. This is not a fault of ProD. We have argued elsewhere that computer science overall lacks effective mechanisms for dealing with social activity [1], and in order to create effective and usable systems, we must restrict what we try to do.

ProD also has limited views of privacy. Privacy is a critically important problem for proactive displays, as with any presentation of identity [2]. ProD currently allows users, in their user preference settings, to specify an access control list. This ACL, however, has only limited settings, since a range of research has found that users have difficulty setting complex preferences. This provides some privacy capability, but users attempting to allow very private content up onto the display are likely to make mistakes. In short, nuanced privacy decisions are difficult, if not impossible, to do with ProD (and in all other systems).

Again, we find ourselves limited by the state of the art, here in privacy policy mechanisms. ProD has been designed to make privacy, as much as possible, into an orthogonal problem. ProD can incorporate more nuanced or more advanced privacy mechanisms, as they become available. For example, privacy policy engines that use task-level data [4] have shown some promise.

Despite these limitations, we believe that the social arrangements and privacy preferences that can be expressed by ProD are likely to be useful, as our applications described above indicate.

CONCLUSION

Proactive displays have emerged as an important class of public display systems. By adapting the content shown on the display based on the users who are present, these displays hold the promise of delivering customized information to users as well as fostering a variety of social interactions. In this paper, we have presented the Proactive Display (ProD) Framework and demonstrated how it can be used to create a variety of applications that, in turn, facilitate the construction of different social worlds. By providing reusable modules for basic functionality and a simple programming model, application developers are free to focus on the collaborative and functional aspects of proactive display applications, rather than low-level details and architectural considerations. While we have begun the process of exploring the proactive display design space with our initial set of applications – including the Identity Collage, the Remembrance Wall, the Cohesion Collage, and Nearby – we believe that ProD affords the opportunity for further design exploration.

The development of the ProD framework has introduced a number of interesting avenues for future exploration. One is the potential to evaluate the effects of different ProD public displays. By observing the types of social interactions and user reactions that are engendered by different approaches to presence governance, content nomination, collaborative selection, and presentation, we can begin to understand the potential of proactive displays to impact social interactions among people who share a physical space.

Furthermore, proactive displays, partially because of their relative simplicity, turn out to be very useful testbeds for looking at a number of important HCI problems - collaborative rules and preference setting being two. For example, we see the potential to use ProD as a platform to explore users' needs and capabilities with respect to managing privacy preferences in pervasive environments. Tools and user interfaces to support end-user privacy configuration remain an open question, and proactive displays are a very visible, but admittedly limited view of how personal information becomes "public" in ubicomp systems. Their very visibility may provide users with a more clear view of the impact of their privacy settings on system behavior, allowing us to more rapidly develop an understanding of user needs and best practices around this thorny topic.

In light of these future directions, we see ProD as a significant step in the emerging understanding of the proactive display design space. On the one hand, ProD synthesizes previous work and provides an encapsulation of the most promising patterns we have observed in that work. On the other hand, ProD opens up new possibilities and new avenues for investigating key questions regarding users' experience of ubiquitous and collaborative computing.

ACKNOWLEDGEMENTS

This work has been funded, in part, by the National Science Foundation (IRI-9702904), Nokia Research, and Intel Research. The original Prospero system was funded by the University of Michigan GROCS program, and the first author has been funded by the University of Michigan/NSF STIET program.

We would like to thank David Hutchful, Perry Wong, Hung Truong, Mouly Kumaraswamy for their help with the design and development of applications for ProD. Tom Finholt was of great help in setting up this project. We would also like to thank Max Harper, Katherine Connors, and Josh Palay for their insights and help with this work. Finally, we especially acknowledge Joe McCarthy for his encouragement and support.

REFERENCES

1. Ackerman, M.S. The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. *Human-Computer Interaction* 15, 2-3 (2000), 179-204.
2. Ackerman, M.S. and Mainwaring, S.D. Privacy Issues and Human-Computer Interaction. In Cranor, L. and Garfinkel, S., (eds.): *Security and Usability*. O'Reilly Press, Cambridge, MA, 2005.
3. Ballagas, R., Rohs, M. and Sheridan, J.G. Sweep and point and shoot: phonecam-based interactions for large public displays. In *Extended Abstracts of the SIGCHI conference on human factors in computing systems (CHI '05)* (Portland, OR, USA), 2005, pp. 1200-1203.
4. Borders, K., Zhao, X. and Prakash, A. CPOL: High-Performance Policy Evaluation. In *Proceedings of the 12th ACM conference on Computer and communications security (CCS '05)* (Alexandria, VA, USA), 2005, pp. 147-157.

5. Churchill, E.F., Nelson, L., Denoue, L., Helfman, J. and Murphy, P. Sharing multimedia content with interactive public displays: a case study. In *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '04)* (Cambridge, MA, USA), 2004, pp. 7-16.
6. Dourish, P. *Where the Action Is: The Foundations of Embodied Interaction* MIT Press, Cambridge, MA, 2004.
7. Greenberg, S. and Rounding, M. The notification collage: posting information to public and personal displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '01)* (Seattle, WA, USA), 2001, pp. 514-521.
8. Huang, E.M. and Mynatt, E.D. Semi-public displays for small, co-located groups. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '03)* (Ft. Lauderdale, FL, USA), 2003, pp. 49-56.
9. Izadi, S., Brignull, H., Rodden, T., Rogers, Y. and Underwood, M. Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *Proceedings of the 16th annual ACM symposium on User interface software and technology (UIST '03)*, pp. 159-168.
10. Jameson, A. More than the sum of its members: challenges for group recommender systems. In *Proceedings of the working conference on Advanced visual interfaces (AVI 2004)* (Gallipoli, Italy), 2004, pp. 48 - 54.
11. Kling, R. Value Conflicts in the Design and Organization of EFT Systems. In Dunlop, C. and Kling, R., (eds.): *Computerization and Controversy*. Academic Press, Boston, 1991, 421-435.
12. Mackinlay, J. Applying a theory of graphical presentation to the graphic design of user interfaces. In *Proceedings of the 2nd annual ACM symposium on User interface software and technology (UIST 1988)* (Alberta, Canada), 1988, pp. 179-189.
13. Masthoff, J. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. *User Modeling and User-Adapted Interaction* 14, 1 (2004), 37-85.
14. McCarthy, J.F., Harper, F.M. and Congleton, B. The Context, Content & Community Collage: Sharing Personal Digital Media in the Physical Workplace. To Appear in *Proceedings of the 2008 conference on Computer supported cooperative work (CSCW '08)* (November 8-12, San Diego, CA), 2008.
15. McCarthy, J.E. and Anagnost, T.D. MUSICFX: an arbiter of group preferences for computer supported collaborative workouts. In *Proceedings of the 1998 conference on Computer supported cooperative work (CSCW '98)* (Seattle, WA, USA), 1998, pp. 363-372.
16. McCarthy, J.F., Costa, T.J. and Liongosari, E.S. UniCast, OutCast & GroupCast: Three Steps Toward Ubiquitous, Peripheral Displays. In *Proceedings of the 3rd international conference on Ubiquitous Computing (UbiComp 2001)* (Atlanta, GA, USA) Springer-Verlag, 2001, pp. 332-345.
17. McCarthy, J.F., McDonald, D.W., Soroczak, S., Nguyen, D.H. and Rashid, A.M. Augmenting the social space of an academic conference. In *Proceedings of the 2004 conference on Computer supported cooperative work (CSCS '04)* (Chicago, IL, USA), 2004, pp. 39-48.
18. McDonald, D.W. Recommending collaboration with social networks: a comparative evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '03)*, 2003, pp. 593-600
19. McDonald, D.W., McCarthy, J.F., Soroczak, S., Nguyen, D.H. and Rashid, A.M.. Proactive displays: Supporting awareness in fluid social environments. *ACM Trans. Comput.-Hum. Interact.* 14, 4 (2008), 1-31.
20. O'Connor, M., Cosley, D., Konstan, J.A. and Riedl, J. PolyLens: a recommender system for groups of users. In *Proceedings of the 7th European conference on Computer supported cooperative work (ECSCW '01)* (September 16-20 Bonn, Germany) Kluwer Academic Publishers, 2001, pp. 199-218.
21. Rogers, Y. and Rodden, T. Configuring spaces and surfaces to support collaborative interactions. In (O' Hara, K., Perry, M., Churchill, E. and Russell, D., eds.) *Public and Situated Displays: Social and Interactional Aspects of Shared Display Technologies*. Kluwer Academic Publishers, 2004, 45-79.
22. Russell, D.M., Trimble, J.P. and Dieberger, A. The Use Patterns of Large, Interactive Display Surfaces: Case Studies of Media Design and Use for BlueBoard and MERBoard. In *Proceedings of the 37th annual Hawaii international conference on system sciences (HICSS '04) - Track 4 - Volume 4* (January 5-8, Big Island, Hawaii) IEEE Computer Society, 2004.
23. Strauss, A.L. *Continual Permutations of Action*. Aldine de Gruyter, New York, 1993.
24. Streit, N.A., Ro cker, C., Prante, T., Stenzel, R. and Alphen, D.v. Situated Interaction with Ambient Information: Facilitating Awareness and Communication in Ubiquitous Work Environments. In *Proceedings of the 10th International Conference on Human-Computer Interaction (HCI International 2003)* (Crete, Greece), 2003, pp. 133-137.
25. Terveen, L. and McDonald, D.W. Social matching: A framework and research agenda. *ACM Trans. Comput.-Hum. Interact.* 12, 3 (2005), 401-434.
26. Villar, N., Schmidt, A., Korteum, G., and Gellerson, H-W. Interacting with proactive public displays. *Computers and Graphics* 27, 6 (2003), 849-857.
27. Vogel, D. and Balakrishnan, R. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proceedings of the 17th annual ACM symposium on User interface software and technology (UIST '04)* (Santa Fe, NM, USA), 2004, pp. 137-146.
28. Wasserman, S. and Faust, K. *Social Network Analysis: Methods and Applications*. Cambridge University Press, New York, 1994.