StoneSoup: Community Sharing of Social Media Streams

Gaurav Paruthi
University of Michigan
gparuthi@umich.edu

Chuan-Che Huang
University of Michigan
chuanche@umich.edu

Mark W Newman
University of Michigan
mwnewman@umich.edu

ABSTRACT

Community displays facilitate interaction and increase awareness in collocated communities. In this work, we designed, developed and deployed a community driven proactive display system that detects the nearby users and allow its users to personalize content for the display. We have used findings from existing literature to create a community display system that is well-suited for an existing student community. Moreover, we expand the design space for proactive community displays to reduce the burden of updating content, provide greater realtime control, and allow greater exploration of community content and community members.

We deployed the proactive display system in a student lounge of an academic building for three weeks. The deployment of community display system was unsuccessful because of multiple reasons like barriers in usage and insufficient community engagement. Nevertheless, we learnt some important points from the study that we take into consideration for a future study.

Author Keywords

Proactive displays, interactive public displays, collaborative configuration

ACM Classification Keywords

H.5.3 Group and Organization Interfaces: Collaborative computing

General Terms

Design; Human Factors

INTRODUCTION

Public displays are ubiquitous in our daily lives. From restaurants to airports, these displays serve many different social and temporal contexts. An interesting subset of such displays is the community display, which serves the needs of a collocated community. Research on community displays has focused on interpersonal awareness, social relationships, and media sharing around such displays, and have provided many design guidelines and highlighted the importance of the community [6, 11]. Proactive displays utilize sensors to detect the presence of people and respond to them in contextually appropriate ways [18]. This encourages participation from the community, ameliorate privacy concerns, and mesh with existing norms of the community [18, 5]. In this paper, we base our work on the established guidelines set by previous literature and make significant improvements in particular aspects of proactive displays. These aspects include reducing the burden of updating, providing greater real-time control and allowing greater exploration of community content and community members.

We have designed a proactive community display that can sense who is nearby and show online content streams, filtered and selected by the community. Our socio-technical ecosystem comprises of student community, the physical space of a student lounge, interactive mobile devices, presence and gesture sensors and a situated computer display, the goal of which is to increase social interaction amongst its users.

Figure 1: StoneSoup users engaging in social interaction in front of the display.

In this paper, we explain our motivations, goals and designs for building a community driven proactive display system called StoneSoup. It allows community member to create, appropriate and share content on the proactive display with the goal of creating collaboration and awareness within the community. We deployed StoneSoup in a student lounge of an academic building for a duration of three weeks. The students who register with StoneSoup are detected by the display using Bluetooth or are allowed to check-in manually. Users

configure the source of the content using their mobile phones. The content selected by the users are then retrieved by the display and their content is presented on the display as per the user's preference. Figure 1 shows StoneSoup deployed in the student lounge. StoneSoup consists of a large LCD display physically linked to a computer with sensors, that is connected to a server. The server stores user configured content and monitors the activity in the system. StoneSoup also comprises of mobile interfaces that we consider are essential to the success of a stream sharing community display.

Our work is motivated by the findings in the previous literature that entail that proactive displays suffer from problem of static content which becomes a burden on the users to update [16]. We believe that online streams offer a way for users to choose sources of fresh content for proactive displays. Online streams can be said to be a source of media that constantly delivers content to an end user for presentation purpose. We believe that streams generate fresh content that increases social interaction within a community, as well as reduce the burden on the users in managing a community display.

The increased interaction amongst the students is likely to increase the informal engagement around the student lounge. This would in turn increase collaboration amongst students over academic and non-academic activities. Moreover, the content shown through the online streams, shared by the students make the students more aware about the new topics of interest that their community is sharing. This interpersonal awareness is beneficial to a community sharing a space like that of an academic environment [16, 11].

We deployed StoneSoup in a student lounge. We report some quantitative evidence collected over the three weeks of the deployment. Although the deployment was unsuccessful in reaching our target audience, we learnt more about our target community and how a novel proactive display application is used and perceived by it. We share our findings supported by informal interviews with our participants.

In the next section, we highlight the findings from previous works around public and proactive displays. We have designed and developed StoneSoup based on the known principles of what works and what doesn't. Moreover, we carve out the space for the study we undertake with a system like StoneSoup and explain our design rationale.

RELATED WORK

In this work, we explore the use of a community driven proactive display system that extends existing literature on such displays. In order to understand the goals of our work, it is important to understand what community driven proactive displays are. In this section we first introduce public displays and describe the specific subset of public displays that our designs concern itself to. Next, we study the factors on which a proactive display deployment depend upon, the scenarios where such displays have been found to be useful or not useful. We then highlight the gap in the literature which we seek to fill.

Public displays are information displays in a public or semi-public space. These displays are commonly used for purposes like advertising, notification, news displays, in many different physical spaces like airports, restaurants, stadium, museums, offices etc. Public displays can be made interactive by adding interaction mechanisms like touch screen or mobile interfaces. Although there is a corpus of research on public displays, in this paper we only look at a subset of it relevant to understanding our motivation, goals and designs.

Community displays form a subset of public displays that serve a particular community. There is a considerable amount of research done in studying community displays and their use in different socio-technical environments. Community displays have been known to increase interpersonal awareness among colleagues [11, 16], as digital outlets for bulletin board like public posts [5] and to increase workspace interaction [12]. Proactive displays are another subset of public displays that serve a broad range of design goals spanning a range of environments like communities and art installations. These displays have the added capability of detecting nearby people and the context using sensors.

Research has highlighted the factors on which an interactive display deployment depend upon. Huang et al. found five factors that increase adoption: deployment for specific tasks critical to the user, tool flexibility and generality, visibility and exposure to others interactions, low barriers to use, and a dedicated core set of users [12].

Moreover, Churchill et al. extend their findings and showed that the following factors were key in their successful deployment of a bulletin board style community display: participative design encouraging a sense of ownership, fit with existing practices, flexible content sharing, continuity of service, simplicity of form and function (clear identity), neutral digital spaces, communities of curiosity, supportive organization and a network of displays [6].The overlap between the findings of Churchill et al. and Huang et al. suggest the importance of these guidelines in public display deployments. In this work, we have tried to incorporate these findings and also find our target space to be conducive for a community display deployment.

Moreover, Churchill et al. found that including 'social' content like the picture of contributors can serve as a proxy for contributors. This is important in triggering social conversation around content. Our work is inspired by such findings about the role social content and social media plays in building better communities [6, 16]

Proactive display research has highlighted the benefits of increasing interpersonal awareness in workplaces and informal communities. Task based approaches have been used to increase awareness by presenting content targeted to certain use cases like different physical spaces of an academic conference [18]. On the other hand, approaches that promote more reflective awareness and minimize distraction have also been successfully used in certain scenarios [20, 21]. We take a more reflective approach where content is opportunistically presented to users when they are not focused on any other

task.

Enabling the community to contribute content is an important design principle for increasing interaction in a community [6, 11, 16, 17, 19, 18]. Community generated content also requires the community display system to manage the content along the dimensions of privacy, nomination and presentation [7]. Congleton et al. presented a proactive display framework that incorporates more nuanced privacy mechanisms but also notice a difficulty faced by the users in using them [7]. Previous studies have supported the nomination of content by using simple forms to present content from online media sources [16] and user generated media [6].

Prior work has studied different kinds of content being shared on community displays. The proactive display system C3C was developed to be used as a window into the online social media world, where users could share their Flickr streams [16]. The Plasma Poster allowed users to post bulletin board style posters on a network of interactive displays [6]. The AutoSpeakerID and Ticket2Talk proactive display systems showed content relevant to people attending a conference [18]. These systems were primarily developed to enhance the targeted communities.

One drawback found in sharing content on systems like C3C and Plasma Poster was the effort required from the user to maintain freshness in the content being shared [16]. For systems like AutoSpeakerID and Ticket2Talk that dynamically change content based on the context, its easy to imagine that the content would go 'stale' quickly in long term engagements [18]. To solve this problem, our designs of StoneSoup embed the ideas of content freshness offered by online streams within them.

Online Streams

Online streams such as news feeds (e.g. CNN.com, gizmodo.com), blogging and microblogging sites (e.g. Twitter, tumblr), photo streams (e.g. Flickr, Instagram) and social media streams (e.g. Facebook, Google+) are constantly updated by their editors. The editors update the content on the streams directly or they choose a filtered subset of a large multi user stream (e.g. Twitter hash tags).

Streams are generally consumed personally by pointing the browser directly to a rendered output of the streams. For example, to view feeds from a news site, users directly browse the news provider's website. Another way in which streams are consumed is by using feed aggregators. Popular feed aggregators like Feedly and Google Reader (now defunct) can aggregate multiple user defined feeds and output them in a feed browser. The feed readers make it easier for the user to discover fresh content from multiple streams within a single user interface. [3, 2]

A more advanced version of feed aggregation is the use of feed manipulators like Yahoo! pipes, which allow users to filter feeds and manipulate feeds to create mashups. This advanced configurability of feeds allow users to tailor the feed aggregation to their particular needs. However, this flexibility comes at the cost of complicating the interface through which a user configures. High configurability entails a greater number of settings, which are then needed to be learnt by the users to fully understand the flexibility offered by the system. There are multiple ways to make this learning curve smoother for users. One important way highlighted in the literature is the community driven help. The role of experts in such communities has been crucial in debugging configuration related problems [14]. The use of shared configuration artifacts like log files and configuration settings has also been shown as crucial in supporting users in their individualized use of systems [13]. Crowd-sourced help for websites has been shown to be useful in scenarios where users require help in understanding UI elements on the website [4]. Going one step ahead of help, Dong et al. leveraged communities to fix UI related problems on websites [8]. Although the importance of the social nature of configuration has been highlighted, it's not known how mechanisms for social configuration be designed in order to benefit flexible collaborative systems.

STONESOUP TARGET COMMUNITY

StoneSoup is targeted at semi-public community spaces that serve the goal of increasing social interaction and hence building a stronger community. We designed and developed StoneSoup to meet the needs of a similar community existing within our academic building. Our users are students currently enrolled in the Masters program at the School of Information. The target space is the student lounge that is dedicated to the needs of our target students. The lounge is x sq ft. and consists of a kitchen area, a social area, and a quiet study area.

Although there are more than three hundred students, the lounge is frequented by not more than fifty students during the peak time of the day. It is most frequently visited during the lunch time or the time between classes. Students engage in several activities in the lounge that include food consumption, socialise and engage in group discussions, group and individual study. Although, these competing and sometimes conflicting uses of the lounge can be challenging for a socio-technical system to satisfy, we find the physical space ideal for StoneSoup deployment. Students engage in discussions related to current happenings, events around them, new places of interest and weather. All these topics can be supported by creating relevant streams. Moreover, StoneSoup would be ideal in promoting news related to student activities like job fairs, programming events, book clubs and design competitions.

DESIGN GOALS

To understand our design goals for StoneSoup we describe the following hypothetical user scenario- a student Bob enters the student lounge after a class. StoneSoup is deployed on the path towards the couch where Bob is headed. When Bob walks past the StoneSoup display, he decides to stop by for a few seconds and glance at what his nearby community members are sharing on StoneSoup. He looks at the content shared on the display and finds himself interested in a post about an upcoming design competition announced on the school's Twitter channel. He is interested in knowing more about the competition and immediately notices that the content shared by his friend John, is sitting nearby. He goes

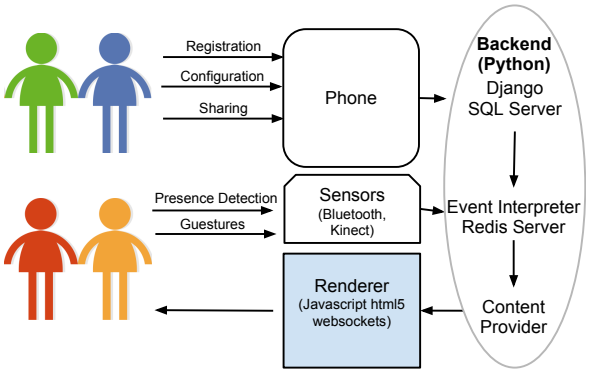


Figure 2: The content creation interface shows recommendations based on recently created widgets.

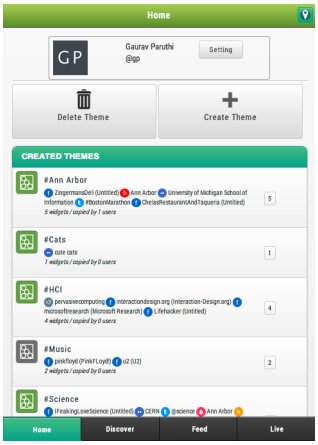
and sits near John to have a conversation regarding the design competition. Later, as John leaves, the content on the display goes away too as StoneSoup automatically checks-out John. Bob notices the absence of the design competition post and feels it would be useful for other students to know. Bob utilizes the discovery feature of StoneSoup to find John's set of content. He copies the configuration representing the design competition to his set and immediately notices his content being shown on StoneSoup. Content shared by him is then read by other students around him, and leads to further discussion that leads to collaboration among them in forming teams for the upcoming competition.

This is just one of the possible scenarios that StoneSoup targets. In the following section we identify the underlying assumptions in making the above hypothetical situation as smooth as possible. Our designs for StoneSoup target the scenarios of content creation, sharing, and appropriation. Furthermore, we have devised social translucency mechanisms [9] via which users discover content and notice the visibility of actions. Next, we describe how the system architecture supports this and then discuss the design and implementation in more detail.

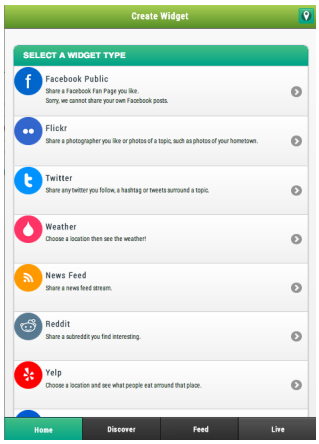
SYSTEM DESCRIPTION

In this work, we leverage existing research to design and build a community driven proactive display system called StoneSoup. The system is primarily designed for community building, and it does this by increasing collaboration and awareness in a shared space. Towards this goal, StoneSoup collectively aggregates the content selected by the users in the space, based on each individual's preference. The design of StoneSoup heavily draws upon the findings and design principles articulated in existing research. We discuss these briefly in the following paragraphs. In addition we present novel designs for answering our social configuration related research goals.

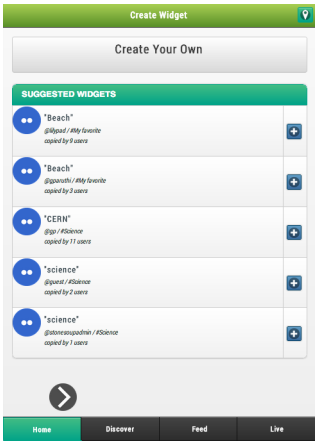
The StoneSoup system consists of a large display attached with bluetooth sensors, a mobile interface for registration and managing content, and a backend server supporting these (See Figure 2). The display and mobile interfaces are developed using HTML5 and Javascript. The backend supporting



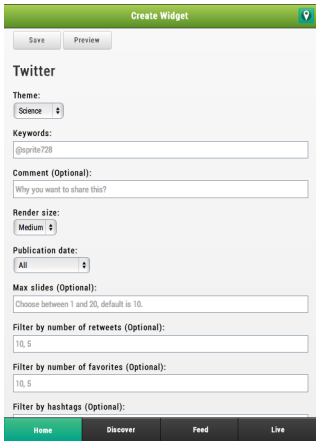
(a) The home screen displays the themes configured by a user. Allows user to change their personal profile and create new themes.



(b) Shows the widget types available for the renderer.



(c) The content creation interface shows recommendations based on recently created widgets.



(d) The widget configuration interface allows the user to change widget parameters.

Figure 3: StoneSoup mobile application interface. Workflow moves from left to right. The top right button on the app is used for check-in and check-out.

these interfaces is implemented as a set of python application servers and uses a sqllite server for storing user profiles and configurations, and a Redis database server that leverages publish-subscribe messaging pattern for sending and monitoring realtime events around the display. The real-time updates that include a user checking in or checking out, or a user changing his/her configuration, are communicated to the StoneSoup display via a websockets server and client.

Registration and Checking-In

A user registers on the system using the signup page on the mobile web application. User enters their username, password, and first and last name. Android phone users have the option to download the StoneSoup app from the Android app store. Currently only the android users are allowed to automatically check-in via bluetooth. The app fetches the user information and turns on the bluetooth of the phone automat-

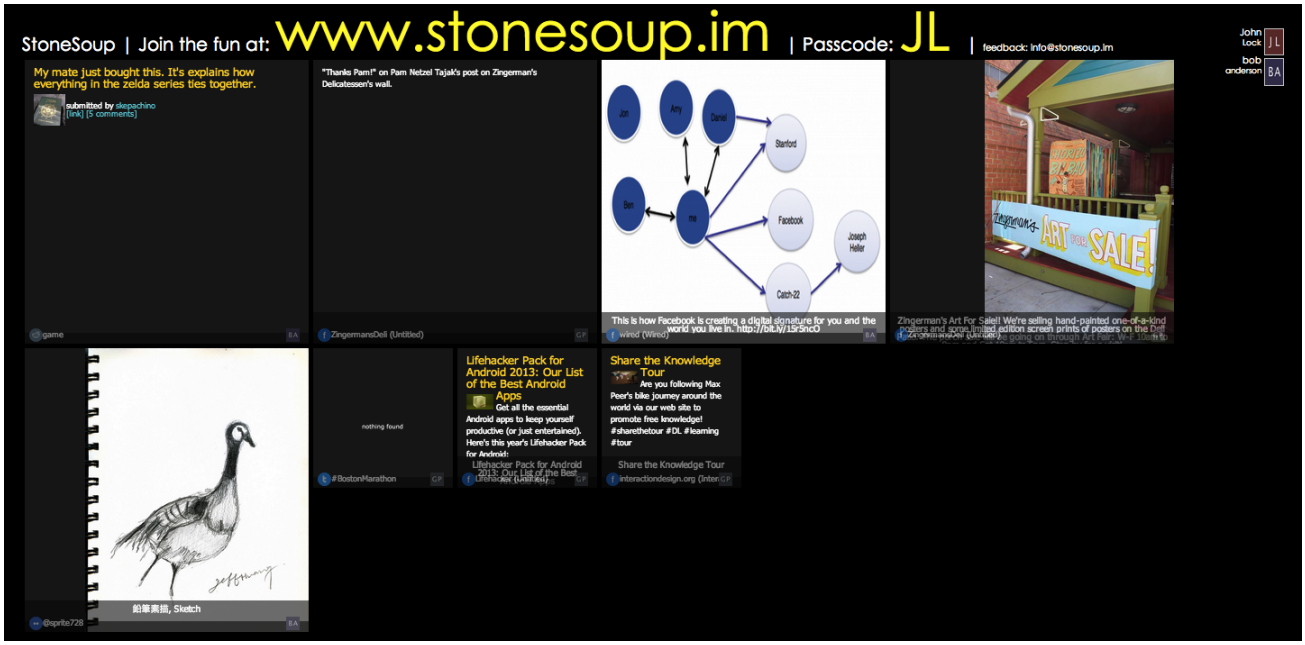


Figure 4: The StoneSoup Display provides a tiles based view for the widgets. The checked-in users are shown on the top-right of the display. Widget allows interactivity via the mobile application and Kinect sensors.

ically. Next, the user is taken to the home screen for managing her content (Figure 3a). Once the user checks-in either automatically using bluetooth or manually by pressing the top right button on the home screen, StoneSoup fetches the content and configuration of checked-in nearby users from the database, prepares the output, and then displays it on the screen.

Configuration

The content and configuration of a user is categorized by themes, which are sets of widgets (Figure 3a). The rationale of a theme is that users would want to create different sets of widgets based on a topic of interest or during different times of the day. For instance, a user Bob wants to create sets of widget for his two interests- a set of science and technology related widgets as a theme named "Technology" and a set of widgets that are related to the upcoming competitions he plan to participate in called "Upcoming". The "Science" theme might contain widgets that show content from a technology news blog, and a widget showing the latest posts by a well known research lab on twitter. On the other hand, the "Upcoming" theme might contain widgets showing the latest updates related to upcoming design competition.

The widget can further be configured on a range of parameters that depend upon the widget type. Parameters define the source, filters and the visual properties of a widget. For instance, the twitter widget can be specified by its source-hashtag or username, filters- how recent the tweet should be (within one day, last 10 days, etc) and its visual properties-size, number of tweets (See Figure 3d).

A user's current configuration consist of a set of configurations, where each configuration represents a widget to be displayed on the screen. A user can set the visibility of each theme and each widget individually by pressing the icon next to each theme or widget. A disabled widget or a theme is assigned a grayed out version of its enabled version (Figure 3a).

Renderer

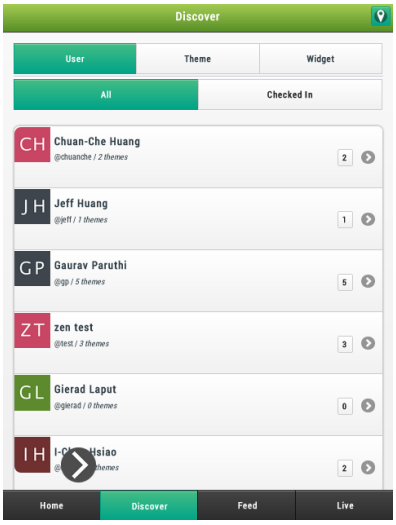
The StoneSoup renderer is shown in Figure 4. The renderer shows the currently checked-in users on the top right and their configured content in the center. This content is shown in a tile based layout. On the top, the renderer shows a link for new users to join StoneSoup and a passcode for non-android users to check-in manually. The passcode is entered on the web interface that is detailed below. The background information for each widget is shown as an overlay information box placed at the bottom of widget. This information includes the type of widget (Flickr, Facebook, etc), the checked-in users who have the widget in their configuration, and the keywords that represent that widget (twitter hashtag, facebook fanpage, etc). Each widget can have multiple slides that rotate every fifteen seconds. Every five minutes the content of the widget is refreshed from the stream source.

The renderer screen when filled by widgets, creates additional renderer screens. The renderer cycles through these screen every minute. Furthermore, a user can interact with the renderer via hand gestures when standing in front of the display to perform renderer related actions.

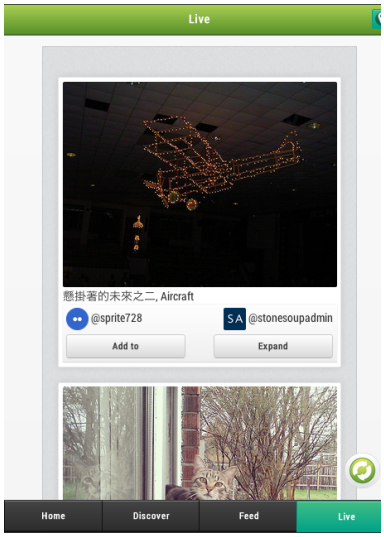
StoneSoup Interaction

Content Creation and Appropriation

StoneSoup mobile interface allows users to nominate and configure kinds of content that currently include: Flickr, Facebook, Twitter, Redit, RSS feeds, Weather and Yelp. This



(a) The advanced discovery interface allows user to browse users, widgets and themes.



(b) Live interface showing the widgets shown by the StoneSoup display on the users mobile.

(c) The feed interface.

Figure 5: StoneSoup discovery and feed interfaces.

list will grow as we discover other content types that are useful to the StoneSoup community. We utilize the APIs provided by these online services to fetch the requested content. A user can create a widget in the following steps:

1. The user selects the type of widget to create from a list of choices (See Figure 3c).
2. The user is shown a list of suggested widgets based on recently created widgets of the same type (See Figure 3c) Additionally the user also has the option to create her own widget from scratch.
3. In either case, the user is shown the widget configuration page where each type of widget can be configured in the source of the content (facebook, twitter, etc) that should be displayed, the filters applied on the stream, and the visual parameters along which it can be configured (See Figure 3d).

An important aspect for making a usable proactive display is to ease users' burden in creating content. Our designs for appropriation support this. It is also essential that these mechanisms be designed for users with no programming experience [15].

The mobile interface provides a simple form based interface for creating and configuring widgets. The content creation interface (See Figure 3c) consists of a simple work flow that allows users to select the widget type and then configure the widget on its parameters (See Figure 3d). All widget types share some parameter types like those related to the layout of the widgets (size, help text). Other widget parameters differ based on the kind of content- a news feed has parameters such as "date of publication", social media feeds have additional parameters that allow filtering based on the popularity of content (likes, shares).

Gesture UI

While designing the interaction mechanisms, we came across scenarios that might require a more direct interaction with the StoneSoup Renderer. More specifically, our designs should allow users near the display to move to different screens of the renderer (Figure 4), freeze the renderer in case there is something of interest and allow users to view more information about the widgets on the screen (Figure 6). We believe that interacting with StoneSoup via gestures would be an intuitive way of completing these actions.

Towards accomplishing this, StoneSoup uses a motion sensing input device to detect gestures. It currently detects four gestures- left swipe, right swipe, left hello and right hello. These are mapped to actions- show previous screen, show next screen, show the widget information view (described later), and freeze current screen respectively.

Social Configuration

In the case of public displays the results of a configuration are visible on the screen to nearby users, and hence has a physical presence. This situated and visible nature of the configuration enables users to observe the results of a configuration change.

Configuration changes represent a move towards a presumably preferable configuration. To support communication and collaboration between users to reach preferable configurations, we have devised social translucency mechanisms for discovering configurations, making visible the configurations and actions, and allowing users to browse the history of configuration.

Discovery of configuration

Users discover content shared on proactive displays in two ways. First, they can directly observe content shared on the

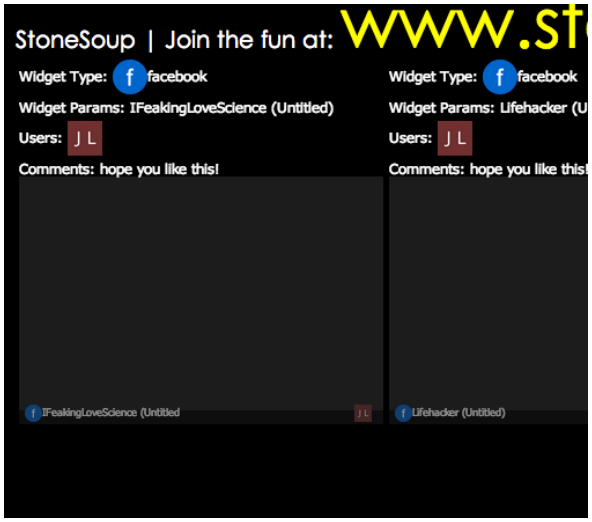


Figure 6: Information view of the renderer is triggered by gestures. It shows detailed information on each widget including the contributor, comments, and widget information.

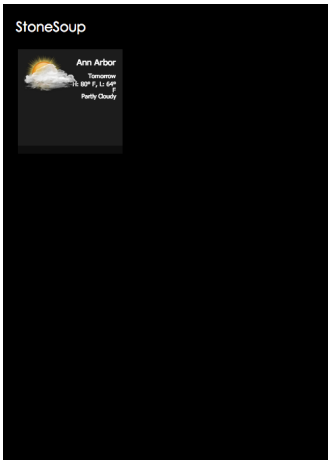


Figure 7: The Preview mode allows users to preview their widgets on their mobile UI. This gives an idea of how the widget might look like.

display and then choose to appropriate. For this our design embeds the identification of content and its attribution along with the content displayed on the display. Attributing the shared content to its creator is a good way of felicitating these interactions and hence increase social information flow [6]. Moreover, users can use a simple hand gesture of "Left Hello" to trigger an information view of the StoneSoup renderer (Figure 6). This view gives a more detailed information about each widget that includes author information, widget type, widget keywords and contributor comments.

Secondly, they can discover content on the mobile web interface for StoneSoup. This interface provides two mechanisms for different discovery scenarios. Firstly, when a user presses the "Live" button, the web interface (Figure 5b) displays the widgets that are currently present on the display. This allows the user to immediately find the widget she sees on the display, and then perform a corresponding action of either browsing the content on her personal device or copying the widget to her set of widgets. Secondly, the advanced discovery interface of the web application allows a user to discover widgets shared by other users based on different parameters of users, widgets and themes (See Figure 5a.) A user can further filter the results to show only the checked-in users and the widgets or themes that are contributed by checked-in users. Furthermore, the user can press on a given result (user, widget or theme) to further navigate and perform an action to copy a widget to her set.

Visibility of configuration

To allow users to recognize the configuration of a widget just by looking at it, it would be useful to make the output of the configuration descriptive of the actual configuration. In StoneSoup, we do this to a certain extent. Each widget shows the keyword associated with the widget. The visual parameters of the widget are self evident. The content specific parameters like "date published" are excluded from the widget information to respect the limited real estate and keep the renderer clutter free. However, this information can be accessed by triggering the information view (Figure 6).

Appropriating content for community displays can be seen as "information staging" [6], where the display is a public stage on which each user is represented by the content they share. Our design for the appropriation of content supports this aspect and tries to prevent unintended representations. As the StoneSoup display maps to the front stage of Goffman's analogy [10], the web interface provides a "backstage" feature that allows users to test widgets on their local system before sharing it with the community. Users can create their set of widgets and then press the "Preview" button (Figure 3d). In this mode, a local renderer opens up inside the user's web browser (Figure 7). Users can look at the results of their configuration to anticipate the actual widget on the display.

Visibility of Actions

A user can interact with a configuration in many ways. Configuration related user actions are relevant in understanding how a given state of StoneSoup display was reached. The news feed interface provides a time sorted list of events related to which a user checked-in or checked-out and which widget was enabled or disabled (See Figure 5c.) These actions give an understanding of how the widgets on the display changed. This would be useful in a scenario where a user notices new widgets on StoneSoup and wants to understand why they were displayed. A user can press on any history element to know the complete state of StoneSoup at that time, which consists of checked-in users and the themes and widgets enabled by them.

History of Configuration

An important aspect of social configuration is to keep track of the history of configuration, so that any user can access a configuration at any given point in time. It would be useful for users to know how a configuration changed thereby changing its resulting presentation and content to a presumably more

Figure 8: The student lounge and the placement of StoneSoup

preferable one. This more preferable configuration can be considered as a stable configuration which users would be motivated to copy to their set, so as to make it easily available.

The feeds interface also provides the users with the history of configuration performed near the display (See Figure 5c.) These actions include addition or deletion of a widget from a user's theme and a change in the widget's configuration, i.e. when one of its parameters was changed.

We incorporate these designs in a deployment of the StoneSoup system. In the following sections, we describe the deployment, the findings from the deployment and discuss our key findings in way that is relevant to the larger HCI community

DEPLOYMENT

To explore the usefulness of the system and learn how it can be improved, we deployed StoneSoup in the Master's student lounge at the School of Information. The system was deployed for a duration of around two weeks at the end of the academic year.

The lounge is a medium sized space (x sq. ft.) that has many tables and seating for the students (See Figure 8). The lounge is frequented by students between their classes and group meetings. It reaches its maximum capacity around lunch time. The students in the lounge participate in activities that include self study, group study, informal interaction and eating meals. Although, the lounge is quite popular with the students, its been known from anecdotal evidence that not all students visit the lounge.

Before the deployment, we conducted tutorial sessions to introduce StoneSoup to select participants. These sessions were useful in two ways- first, it helped us to introduce the system to interested users before the actual deployment, hence giving us an opportunity to test the basic usability of the system. Secondly, the tutorial sessions were useful in reaching out to the early adopters in the student community [21]. This helped in the creation of the initial set of widgets for the actual deployment, hence getting around the cold start problem. We conducted five sessions that were about an hour long and each user was incentivized for $10 per session.

The actual deployment ran for fifteen days and participants who checked-in at least twice in a day were incentivized for a $1 for that day. This incentive was created only to attract users to register and check-in with the system. We were interested in how the system would be used if people were motivated to use it, rather than in whether people were intrinsically motivated. Given the short duration of the deployment and other barriers to the deployment, we felt it was necessary to introduce such a measure.

FINDINGS

In this section we discuss the findings from the deployment and how they relate to our idea of social configuration. Although StoneSoup did not get much usage during the deployment, we have some good insights to share.

There were 34 users who registered with our system out of which 19 users registered during deployment and 15 users registered in the tutorial sessions. In total there were 645 actions performed which included presence actions - check in, checkout; widget and theme actions - copy, create, delete, disable, enable, update. 312 actions were performed outside the tutorial session during the deployment, out of which 109 were performed by the users who participated in the tutorial session.

The low usage during the deployment doesn't allow us to make strong claims about our designs. However, it gave us some insights into building better systems like StoneSoup. In the following subsections we highlight what we have learned from the deployment and also the reasons for the low usage.

Social Configuration

Creation and Appropriation

We observe that the users found the widget recommendations to be useful. When a user is trying to add a widget, she sees a recent list of widgets created by others (See Figure 3c.) More than half the users who created widgets used the recommendations, indicating the usefulness of such a feature.

Unsuccessful Deployment

We conducted informal interviews with some of our users from our tutorials and from our deployment. Based on their comments and our observation, we think the reasons for the low usage can be categorized as follows:

Bad Timing

We deployed StoneSoup at the end of the academic semester. During this time, the masters students are busy with exams, deadlines and job search. Moreover, we observed less than average number of students in the lounge.

Barriers

We noticed many barriers in using StoneSoup. The barriers were observed when a user performed common actions like- sign up, check-in and creating widgets. Some of these barriers could be fixed by future re-design of the mobile app.

Others barriers that change the workflow like check-in, can be improve by adding better sensors to detect nearby users, thereby leveraging the physical nature of StoneSoup. From our interviews, we find many users stopped using StoneSoup after encountering a barrier for the first time.

Community Management

We realized that for a community system like StoneSoup to be successful it takes more than just placing a system in the middle of the community. There was a clear need for a closer engagement with our target users, to present StoneSoup, generate interest, and help them get started. Moreover, we noticed that users had a lot of queries regarding StoneSoup. Due to time limitations, we were not able to answer users' queries and take their suggestions into account. Echoing the finding by Churchill et al., we think that it is necessary to identify representation of the community and include them in the design and deployment process [6]. Community representation also help in knowing the norms governing the communities.

Spatial Characteristics of the Deployment Location

Knowing the way things are arranged in the lounge is vital success of a system like StoneSoup. The limited space in the lounge posed an important challenge which we were not able to overcome. Churchill et al. suggest that a community display system is ideally situated in a space which is already used for social interaction like near the water cooler [6].

One of the problems we faced during deployment was in deciding the location of the display. Given the limited amount of space in the lounge, we were left with no other choice but to reconfigure the seating arrangement of the space. Just before the deployment, we changed the seating configuration of the lounge such that the display becomes centrally located with seats facing it (Figure 8). During the course of the study, we had to revert back to the original configuration because of some complains regarding the new configuration and other concerns.

In our interviews, participants pointed out that the positions of displays went against some existing norms of use. In the original configuration, the seating arrangement blocked the usual path to one of the central tables in the lounge, hindering a student's typical task of sharing food with others. In the second position, a participant mentioned that our final position of the display was unsuitable for social interaction as people standing would be in the way of others walking by (See Figure 1).

In addition to these, we encountered a number of standard usability problems such as text being too small, buttons that were confusingly labeled, confusing task flows, etc. We recommend that all these factors be taken into account before deploying a community display system like StoneSoup.

DISCUSSION

From our findings we think that successful deployment of public displays must take much more effort and thought into the socio-technical environment around the display. Although, we find Churchill et al's nine factors that go into making a successful deployment [6] useful, we realized that there is a lot of depth and nuance to each of those factors that needs to be articulated in more detail. Moreover, we couldn't help but accept the undeniability of some factors.

Participative design encouraging a sense of ownership

We incorporated community engagement in our design process by including some students in our pilot testing. However, we were not able to conduct a pilot with a representative sample of our target population because of the conflicting goal of creating hype around the public display and maintain the novelty of StoneSoup.

Fit with existing practices

We found that a novel service like StoneSoup can sometimes go against the existing practices of a space. We believe that some use cases are novel and useful enough to change people's behavior and their practices. For example, our users currently use Facebook or mailing list to share content and have discussions with the community. This however poses a barrier for many students who are less active in online media sharing. StoneSoup allow its users to share content streams in a more offline sense, where checked-in users are physically located in the same space. This promotes more offline interaction, preferred by many people. We hope that StoneSoup would create new norms regarding the sharing of content for the master's students.

Flexible content sharing

With StoneSoup we have tried to create a flexible content sharing public display platform. The many configuration options for the many content types give gives users enough flexibility to choose from a range of content available. However we find that there is a trade-off in offering too much flexibility- the system is perceived as too complex if we add more complex parameters that serve a rare scenario.

Simplicity of form and function (clear identity)

During our pilot testing with an existing version of StoneSoup, we noticed certain identity issues with the system that had to be solved. Users did not quite understand whether the goal of sharing was to share content that they personally find interesting or that they think others might find interesting. Moreover the novelty of a system like StoneSoup implies that users were just not sure on what to do.

CONCLUSION

Proactive displays are increasingly becoming pervasive. In this work we presented a community driven proactive display system called StoneSoup that aggregates content from its users. By leveraging existing literature, we designed and developed StoneSoup to allow its users to easily create and share content for increased interaction and awareness within the community. However, our deployment didn't turn out to be that successful for multiple reasons like bad timing, barriers in usage, and lack of community engagement. Nevertheless, we believe that our designs related to social configuration and insights from the deployment are an important contribution towards collaborative systems and be considered in future deployments of community displays in general.

NOTE

The title of the paper- StoneSoup is taken from the folk tale inspiring contribution by the community for the greater good [1].

ACKNOWLEDGMENTS

REFERENCES

1. Stone soup. Wikipedia, the free encyclopedia, Dec. 2012. Page Version ID: 527797409.
2. Feedly. Wikipedia, the free encyclopedia, June 2013. Page Version ID: 561621065.
3. Google reader. Wikipedia, the free encyclopedia, June 2013. Page Version ID: 560364087.
4. Chilana, P. K., Ko, A. J., and Wobbrock, J. O. LemonAid: selection-based crowdsourced contextual help for web applications. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* (2012), 1549–1558.
5. Churchill, E. F., Nelson, L., Denoue, L., and Girgensohn, A. The plasma poster network: Posting multimedia content in public places. In *In Proceedings of the IFIP International Conference on Human-Computer Interaction (INTERACT 2003*, Press (2003), 599606.
6. 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 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '04, ACM (New York, NY, USA, 2004), 716.
7. Congleton, B., Ackerman, M. S., and Newman, M. W. The ProD framework for proactive displays. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (2008), 221230.
8. Dong, T., Ackerman, M. S., Newman, M. W., and Paruthi, G. SocialOverlays: collectively making websites more usable. In *Proceedings of the 2013 INTERACT* (2013).
9. Erickson, T., Smith, D. N., Kellogg, W. A., Laff, M., Richards, J. T., and Bradner, E. Socially translucent systems: social proxies, persistent conversation, and the design of "babble";. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '99, ACM (New York, NY, USA, 1999), 7279.
10. Goffman, E. Presentation of Self in Everyday Life. *American Journal of Sociology 55* (1949), 67.
11. 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, ACM (New York, NY, USA, 2001), 514521.
12. Huang, E. M., Russell, D. M., and Sue, A. E. IM here: public instant messaging on large, shared displays for workgroup interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, ACM (New York, NY, USA, 2004), 279286.
13. Huh, J., Newman, M. W., and Ackerman, M. S. Supporting collaborative help for individualized use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, ACM (New York, NY, USA, 2011), 31413150.
14. Jones, M. C., and Churchill, E. F. Conversations in developer communities: a preliminary analysis of the yahoo! pipes community. In *Proceedings of the fourth international conference on Communities and technologies* (2009).
15. MacLean, A., Carter, K., Lvstrand, L., and Moran, T. User-tailorable systems: pressing the issues with buttons. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, ACM (New York, NY, USA, 1990), 175182.
16. McCarthy, J. F., Congleton, B., and Harper, F. M. The context, content & community collage: sharing personal digital media in the physical workplace. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, CSCW '08, ACM (New York, NY, USA, 2008), 97106.
17. McCarthy, J. F., Farnham, S. D., Patel, Y., Ahuja, S., Norman, D., Hazlewood, W. R., and Lind, J. Supporting community in third places with situated social software. In *Proceedings of the fourth international conference on Communities and technologies*, C&T '09, ACM (New York, NY, USA, 2009), 225234.
18. 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 ACM conference on Computer supported cooperative work*, CSCW '04, ACM (New York, NY, USA, 2004), 3948.
19. Nacenta, M. A., Jakobsen, M. R., Dautriche, R., Hinrichs, U., Drk, M., Haber, J., and Carpendale, S. The LunchTable: a multi-user, multi-display system for information sharing in casual group interactions. In *Proceedings of the 2012 International Symposium on Pervasive Displays*, PerDis '12, ACM (New York, NY, USA, 2012), 18:118:6.
20. Ryan, W., Hazlewood, W. R., and Makice, K. Twitterspace: a co-developed display using twitter to enhance community awareness. In *Proceedings of the Tenth Anniversary Conference on Participatory Design 2008*, PDC '08, Indiana University (Indianapolis, IN, USA, 2008), 230233.
21. Zhao, Q. A., and Stasko, J. T. What's happening?: promoting community awareness through opportunistic, peripheral interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '02, ACM (New York, NY, USA, 2002), 6974.