

Small meets Large:

research issues for using personal devices to interact with public displays

Alan Dix

Lancaster University, UK

6th January 2005 (web version March 2008)

Abstract

Public screens are becoming common from advertising displays in airport lounges to the vast city centre screens. Small private devices, notably mobile phones are an obvious compliment and potential interaction device. This discussion paper looks at research issues raised by the potential synergy of these, looking at several levels from human interaction to system infrastructure. The paper is written in the context of eCampus, a large-scale deployment of public screens around the campus of Lancaster University, giving it one of the most extensive and most flexible public screen infrastructures in the world.

About this report

This report was written in January 2005 as an internal discussion paper. However, it was subsequently distributed more widely and has also been the impetus for a number of other projects and papers. In order to be able reference it and in particular the scenarios written in it more formally, I am now making it available on the web. Since it was first written several of the things that were simply ideas have become reality, but others are still 'to do' ... feel free to mine for ideas for your own work!

Note: I have not made extensive edits, so it should be read in this light, an informal discussion document.

1. User experience and interaction issues

1.1 Detailed Interaction

Key issues: knowing what LD is being addressed, primitives for SD/LD interaction, experiments for fine level details

Low level binding of devices to locations/displays must reflect users perception of this. This is rather like the old 'unselected window problem'. In a GUI how do you know where your input is going - in SD-LD how do you know whether and which LD you are addressing.

What are the common repertoire of things we want to do between small devices and large displays? Many will be similar to GUI, but there are more choices of where interactions are focused and where they have effect.

Some potential primitives in the repertoire ...

- (a) use SD to perform large display selections (e.g. glyph, send SMS code)
- (b) use SD to scroll, spin, move large display objects (e.g. sweep, joystick)
- (c) navigate menus on SD to control LD content (e.g. WAP menus)
- (d) use LD selection to control SD content (e.g. select glyph, enter code and get additional private info)
- (e) move content locus – fluid movement of information, interaction between SD and LD (e.g. browse private SD, send large object to LD)
- (f) move interaction focus – which LD! (e.g. type code, camera-glyph, bluetooth)

My feeling is that quite a lot of fine presentation issues (e.g. appropriate font size, Fitts' law properties of sweep) can be addressed by looking at existing design knowledge coupled with things like the viewing angle of displays, tracking speed of devices, etc. On the other hand formal experiments on these kind of things are just what is likely to make strong TOCHI papers! ... MRes projects ??? The most promising of these would be ones which are not obvious extensions of existing knowledge: e.g. phone joystick Fitts' law - boring, sweep Fitts' law ... more interesting, font size at distance - standard, fonts when looking at the display from odd angles – more interesting.

1.2 Multi-user issues

Key issues: 'competing' for screen yet not (necessarily) collaborative group, use of implicit means of contention management??

Standard synchronous groupware with large displays (e.g. colab) faced multiple cursors, contention etc., but were focused on cooperating groups. Main difference here is that screen real estate is a fixed resource, but the potential 'users' are not necessarily cooperating! However, still many lessons may carry over - e.g. contention often 'sorts itself out' if people are made aware of each other's actions.

Much of traditional groupware does not focus on precise physical locations; indeed it is often designed to deliberately reduce the influence of space. The obvious exception to this are electronic meeting rooms (Colab et al.) where there is some sort of plinth or hot spot and certainly the social dynamics are influenced by physical layout (see my book!!).

In addition, ethnographies of control rooms, workplaces etc., emphasise the importance of spatial arrangements – overhearings, peripheral awareness, etc.. Often the strength of the physical arrangement has arisen over time by 'local' adaptation and have not been 'designed' from outside. Typically when there is an explicit design/designer these tend to be disaster stories (could be biased ethnographers of course).

With public displays in architectural space there is perhaps less room for local adaptation, so getting spatial designs right is more challenging!!

There has been quite a bit of work already on collaborative use of large displays. Again this dates back certainly to the Colab work, although one could see the military battle boards (not sure what the right word is those big maps they move model planes about on) as early examples. However, things like the shared displays at Nottingham (forgotten the name, but the ones they have trialed in sixth-form common rooms) and seen others as well (need to look in Keith's book!). The focus is often on collaboration, whereas for public displays we may have a combination of individual interactions, group interactions and more community effects all potentially 'competing' for the same screen real estate and even physical space.

These user interaction issues are about public LD interaction in general. However the presence of SD may give opportunities to help – see scenario 3.5.

1.3 Use of space

Key issues: blocking of LDs, wide and odd viewing angles, use of space to claim control, for auditability, stand back / stand forward styles of interaction

As well as these inter-social aspects, space is central to situated displays in many other ways.

The physical placement of displays, the context of place, locations of other objects, etc., all affect what can you see from where and what you can't ... people (buses) blocking views, blocking devices etc. This all means that the locations from which a display is visible may be dynamically changing and have different typical patterns at different times etc. Even when a screen is visible it may be seen at an oblique angle.

The sheer size of screens is an issue too. With desktop or even command and control displays, the devices are usually arranged to give near front-on views. In contrast, a single or series of large displays (e.g. in the underpass), may mean that the overall viewing angle is far greater than a standard desktop (e.g. my current display is 25cm across and about 50cm from my eyes, a viewing angle of about 0.5 radians ~ 30°). This means that (i) some parts of the screens will be at distorted angles (ii) more of the screen is in peripheral vision.

Note that both the potential for oblique views of screens and the sheer size mean that the part of the screen on which the user (Is that the right word? viewer, onlooker?) is focused may be very distorted. Note that text is particularly difficult to read at an angle.

Jennifer Sheridan has noted too that the nature of interaction may be different if one is close to a large screen or far back (stand back/forward). For example, close to the screen one may use a camera as a physical pointer, but of course only within the range one can reach. Further back one may use camera to snap glyphs, or joystick or sweep.

Space can be an interaction opportunity, using the physicality of devices and their placement to 'solve' or bypass issues that seem problematic if one views them purely in the virtual/electronic domain.

One example is where there is the potential for contention between users of a display. One could create complex floor control algorithms, priorities, resource sharing schemes. Alternatively, the system can use physical proximity as an arbiter. This means that people

can 'seize' control by getting close to a display (or part of it) and off loads the contention resolution to social interaction.

In Hermes we have also seen how the 'accountability' of interaction in a public space reduces the need for system controls on access etc.

1.4 Temporal Issues

Key issues: broadcast mode of LD in conflict with user control, use SD for control of pace of navigation, information about what is to come and capture/replay

In 'traditional' desktop systems changes happen when the user does things and hence the pace is controlled by the user (or strictly the slower of user and system!). If the user wants time to think the system patiently waits.

In groupware and other sorts of open system this is not the case. Other people or environmental factors can create events that are outside a single user's control. Interestingly, outside the factory control room or aircraft cockpit, the more asynchronous aspects of interfaces tend to be boxed in: IM window, email application etc., within a desktop environment that is still largely user controlled. In remote groupware applications this is possible because each person's display can in some way throttle the impact of everyone else's interactions. The exceptions to this (as noted above)

For public displays the content may be responding to other users, context, broadcast streams etc. Furthermore, there may be many users, some explicitly 'connected', but many simply viewers, so a single user may not be able to take control of the screen.

Small devices can help get round this :-)

- *controlling pace of input* – Navigation may be performed on the small device leading to changes in content without needing to use of the large screen for the display of navigation feedback (see scenario 3.5)
- *monitoring pace of delivery* – The displays may well have some indication of scheduled content, but this is likely to be limited in horizon, both because of limited screen space and also because dynamic schedules will not be planned too far in advance. However, information about when items of interest are coming can be shown on a user's own device (see scenario 3.5).
- *escaping pace of output* – The displays are constantly changing, but they can be snapped by the phone, not in the sense of taking a real picture, but simply by noting the time and display. This can then be replayed and examined at leisure both at the time using the SD or later on a web portal or stand alone application (see scenario 3.6).

1.5 Level of integration

Key issues: levels of integration from simply accessing the same content to behaving like a single interface, ? most interesting issues at higher levels of integration

The SD and LD may be integrated at various levels:

- shared services and information – The SDs may access services and information that is also being broadcast on public displays. In the case of interactive SD services this may affect the LD content, but in a way that is not intimately connected to location. For example, if Jenny vote for a SU position on line this may change the overall leader board shown in the underpass no matter where Jenny is at the time.
- incidental links – Because the SDs and LDs display similar content there may be associations in the users mind. For example, Jenny may decide to vote because she sees

the leader board. The connection of location may be important, but the system/infrastructure does not know about it, only Jenny.

- shared reference – Where the public display has IDs, SpotCodes or other glyphs, IR links, RFID ... which can be picked up by the phone and used to link then or later to related information. In this case the LD is not affected by the SD, but there is some explicit system link (possibly mediated by hand in the case of typing in a code) between the LD and SD.
- SD as input device – Using the SD in some way as an input/control device for the LD. This includes using the sweep as virtual mouse, dragging glyphs on the screen, using the keypad or joystick for data entry, pointing device. In this case there is a very clear connection, and the phone is acting as a dynamically connectable input device. Note that a few of the SpotCode examples have elements of this.
- uniform interaction environment – Finally, there may be a close level of cooperation so that the interface on the SD and that on the LD seam more like one. For example, some video controllers have small LCDs on the controller that you can use to set record times etc., without disturbing other watchers. The controller, VCR and television function as a single interaction environment. Similarly in a public environment the SD may be used to do local navigation for content that will appear on LD (see scenario 3.5). Also some content may be viewed on the SD and only when large screen space becomes necessary it may be streamed out to LD. (This is like the various forms of sweeping windows between portable devices.). In scenario 3.4 we imagine a 'video' game that involves interfaces on SD and LD working seamlessly together.

2. Technical issues

By this I mean under the bonnet things like architecture, toolkits, protocols. I keep find myself using the term 'technical' in this way and hate it, but it seems to be what people expect.

2.1 Device Issues

Key issues: novel methods and algorithms (e.g. sweep, use of camera or screen for location), abstractions over capabilities of SD and LD, self description both

Small devices can function as input: the sweep mode, glyph recognition and general use of the camera offer novel capabilities for small devices or alternative ways for achieving standard controls. Phones, PDAs etc. are also interaction devices in their own right as well as simply input devices for other devices. This means there are possibilities to trade interaction between the surfaces. Binding is also an issue with various technical solutions (GPS/location, bluetooth, camera+glyph).

For traditional GUI, in fact further back to PHIGS etc., there have been various abstractions of input devices (e.g. Card et al. degrees of freedom, Buxton 3 state model), and for standard 'desktop' interface pared down to generic pointer and text entry. SDs offer different capabilities (e.g. camera) and may not be uniform.

One issue therefore is to catalogue the kinds of concrete potential of different kinds of device and to build suitable abstractions on these. This process clearly interacts strongly with interaction repertoire below, architecture and toolkit.

Of course we cannot run the install a driver disk on the public display every time someone comes into the underpass with a new device ;-). So some form of self-description needed to tie devices into the capability ontology and allow some form of plug-and-play.

Different public displays may also have different capabilities, so there is likely to also be some form of content negotiation required there. For example, I am in a pub and want to show my photos using a remote photo album service. There has to be some process whereby it can find out the left panel of the display in the corner can respond to customer services and can show JPEGs at 480x360 resolution. Furthermore, somewhere between the service, my device and the display I need to know that this is possible. [[user service discovery: actually knowing what you can do when and how, seems a really big user interaction issue]]

One of the examples Jennifer Sheridan has described is the use of flashing phone screens that are picked up by cameras (rather like fairylights). The use of this form of spatial location depends on both the kind of device (a phone capable of connecting closely enough to be able to display light patterns on demand) and on the 'display' hardware (web cams to do tracking). So, the abstract capabilities available to a service or application are based on combined device capabilities and interactions.

2.2 User Interface Architecture issues

Key issues: Seeheim for situated displays!!

Standard UI architectures like MVC and PAC deal well with multiple visualisations of the same data - public displays? Certainly they all implicitly assume that the locus of interaction is the locus of feedback. Also they tend to have implicit models of display surfaces, assume single bindings of input device and displays etc.

Multi-user architectures tend to focus on shared information space (really focused on remote-synchronous interaction). These not so different from the string content focus that has been emerging. However, again there is an implicit assumption of fixed bindings, certainly between input devices, screens and people, even if the set of people may change.

There is a lot of work on rendering for different display devices, and although not 'solved', this is well populated. Less I think on dynamically changing this within an interaction (e.g. moving content between SD and LD), so this would be more fruitful place to spend effort.

2.3 Toolkit issues

Key issues: interaction primitives, dealing with varying device capabilities

GUI toolkits do not simply stop at mouse and pixel, or even menu and window, but package up interaction paradigms.

Similarly what are the equivalents for SD-LD interaction (are they radically different or 'more of the same')

There are multiple levels of this from low-level APIs and 'widgets', through higher level event-action type scripting to 'canned' templated interactions.

This stage interacts with the system-level scheduling/infrastructure issues, the device capabilities and self description, and also the user-level interaction repertoire.

2.4 Authentication, Security and Privacy

Key issues: establishing levels of trust between user, service and platform in heterogeneous open infrastructure, use of phone/PDA as 'trusted device' by user.

The use of public displays to access remote services has some similarity with web access, but with differences. In the web the PC or other display device is 'trusted' by the user. The service needs to authenticate the user, but otherwise treats the user and display as one. The user is also seen as 'owning' the PC hence anything requested by the user can legitimately be displayed on the PC.

In fact the latter is not universally true and company or family net-nanny services are because there are separations between ownership, control and use.

However, for public screens the tacit understandings of control and trust collapse entirely: the platform is potentially an 'alien' environment for the user and service and vice versa the display has 'alien' applications using it and 'alien' small devices connecting to it.

The user-service authentication is reasonable standard, and can use the public display communications infrastructure (e.g. WAP through bluetooth) using standard untrusted communications methods.

Perhaps more interesting is the mutual need for trust between the user+SD+service on one side and the public display on the other:

- Do the service and user trust the display?

Will the display keep information about the user gained from personal devices? Will the display keep information displayed on it?

Perhaps like the web it is effectively the user's job to establish trust in the devices, but in, say, an open urban environment, how do we know that the screen in the second hand HiFi shop is really to be trusted?

Also, how do we know which display we are connected to anyway?

- Does the display trust the service and user?

Is the service producing salacious, illegal, libellous material which is being seen by others?

For semi-private areas, such as a PC in the library, this is more like the web, but for, say, a display in Alexander Square we would probably only want to allow access to trusted services.

Even when the different platforms trust each other (to some extent), how do they communicate the situations and allowable content. For example, a bank might not want to allow a customer's credit card statement to be displayed on the big Manchester screen, so needs to know something about the publicness of the display. On the other side, if the display does not want certain sorts of content displayed it needs to be able communicate this back to the service – perhaps PICS ratings for content?

Establishing policies, protocols, and frameworks for this seems both a necessary aim for eCampus and interesting research. Furthermore, the, albeit unintentional, slightly heterogeneous nature of the hardware/software platforms we are developing seems ideal for this.

The place of the phone or other SD seems interesting in that it is a trusted device of the user with significant computational power. For example, standard public or private key encryption can be used with user PIN/password entered directly through the SD thus avoiding the use of public keypads that could be tamperable-with or monitored.

The SD can also be used as a 'holder of context', rather like a web browser holds cookies (spatial cookies!). For example, in eCampus navigation scenarios we have discussed using bluetooth addresses as a way of 'tracking' people around. This means we can centrally store information such as where they are intending to go, where they are now, etc. Alternatively the infrastructure could deliberately hide this from services and instead leave cookies with the phone "this person wants to go to X", giving more control and privacy to the user. Furthermore, this could allow certain cross-service use if cookies were not service specific (as web ones tend to be), but also had more service-type oriented

N.B. Perhaps for open networks we really want the equivalent of WiFi MAC or Bluetooth addresses that are assigned randomly on a per connection basis thus avoiding unintended tracking, rather like dynamic IP and firewalls do for web? (see Bluefish!)

3. Scenarios

Trying to think of examples that are do-able in a 3 year time span, but demonstrate interesting interaction issues.

3.1 Display configuration using phone

Demonstrates: use of glyphs for attachment to display, selection and movement

Display has a number of standard framed layouts showing different kinds of content at different times. The graphic designer or content manager has spent some time at a workstation arranging these, but needs to see how things work in situ.

She uses her phone connects into the local infrastructure and she uses a local application on her phone to enter configuration service. A small glyph appears on the top corner of each display nearby and she selects the one she wants to configure by pointing her camera at the glyph and snapping. The glyphs now disappear from the displays and she

She uses her phone application to chose the layout she wants to test and it appears on the screen with representative content in each panel. When she chooses 'select panel' on her phone, each panel on the display shows a glyph and she selects one panel using point and snap. She can then use her phone to chose different types of content and check that they look right when displayed in context.

She does the same to several layouts until she gets to one that seemed OK on her office screen, but just doesn't work out here in situ. She chooses 'change layout' on her phone application and each panel displays glyphs at their boundaries. She 'grabs' a panel boundary using snap-and-hold when the relevant glyph is in the hot spot of her phone. As she drags her phone down the display it drags the grabbed glyph.

N.B.

- (i) glyphs only need to be unique within local area, so can be smaller than the standard Cambridge ones.
- (ii) the glyph drag uses the glyphs location within the camera field rather than sweep

3.2 Cursor wars at the Graffiti Wall.

Demonstrates: use of sweep or joystick to deliver 'abstract' interaction

It is late at night in the underpass and a group of friends are waiting for the bus. Ged has a miniature phone a camera and Bluetooth. Ted's PDA has a WiFi/GPRS PCMIA card and a really nifty thumb joystick, but no camera, . One of the screens is showing the Graffiti Wall. Ged connects with Bluetooth to the local infrastructure and a spray can appears on the wall with his nickname beside it. He uses his phone like a mouse with camera 'sweep' and when he presses a button sprays paint. He begins to draw a face. Ted's PDA doesn't have bluetooth, but he connects instead using GPRS. His own spray can appears and he uses his joystick to move it around and draw a beard on the face. "Hey" says Ged "get off my face" ...

3.3 Night Sky ... under cover

Demonstrates: use of sweep to naturally control virtual navigation

One evening the local junior astronomical society book one of the underpass 'screens' for a meeting. When they arrive they start off outside looking at the stars. The sky is partly

covered in clouds, but they can make out Orion, the head of Taurus and the diminutive Seven Sisters barely visible.

As it starts to drizzle and the children start to shuffle their feet, the leader takes them into the underpass. She connects her phone and a picture of the night sky faces them. As the leader sweeps her phone in front of her the stars wheel by and scale at the bottom and sides shows the compass direction and azimuth. She takes them below the horizon to the southern sky to see the Southern Cross, then back to the north and the constellations they have seen that night.

She makes a selection on the phone and the stars are overlaid with drawings of the fantastical creatures behind the names of constellations: Orion with his bow, dog, bull, serpent, ... They talk for a while about the legends and myths of the stars.

Then back to the wonders of the real cosmos. The pictures disappear and she moves the focus to the Seven Sisters. The application on her phone is configured so that sweeping the phone whilst holding the joystick towards her pans and tilts the view and holding it away from her makes the sweep control zoom. With an expansive sweep of her hand and phone towards the screen the images flood nearer, stars fly by and the Seven Sisters fill the wall in front of them. As it gets closer they see one start divide into a double and one of those grow and become a whole galaxy spiral. [[I think the astronomy is right!]]

3.4 Starfighters

Demonstrates: use of SDs to give local interfaces combined with large screen

In the bar the local infrastructure notices that several of the bluetooth phones belong to regular Starfighter players. In the ticker tape it announces a game will start and some of the potential players have registered to receive alerts and their phones beep to tell them. They start the Starfighter application on their phone and when sufficient have said they are ready the game gets scheduled.

When the movie trailer for the Film Society ends the prelude for the game begins describing the background to the game and showing the pilots getting into their starfighters ready to attack the empire battle star . More people connect until there are three empire fighters and five rebel fighters. the system creates several more robot fighters on each side to even the odds and the game begins.

Each phone's display shows an individual heads-up display. On the large screen the right hand side shows 'birds-eye' view of the battle star from several directions and along the bottom of the screen are a series of miniatures of the heads up on each player's phone. The players can look up to the birds eye view to give them global idea of the battle and some of their friends who are non-combatants give strategic advice.

As well as the birds-eye views and heads-up miniatures, a large area of the screen is given up to a 'movie' of the game. An automatic algorithm finds 'hot spots' in the game and chooses camera angles, sometimes from outside looking at the fighters, sometimes over the shoulder from one of the pilots. "Hey that's me" says one, then "Oh s**t" as he is shot during his lapse of concentration. This movie is in console style realism generated at the local display controller compared to the low-res schematics on the individual phones.

3.5 Choosing the channel

Demonstrates: use of SD to give local navigation for shared large screen

It is Saturday afternoon and Jenny is waiting for a friend in Alexander Square. In a shady corner is one of the public screens. It is showing latest news headlines.

"I wonder how Golgate Rangers are getting on in their away match at Carnforth", she thinks.

Jenny has a miniature phone with no camera, but it does have WAP and GPRS. She connects the eCampus WAP site and selects the sports channel. Great! two nil up! She selects 'show hi-lites' and the WAP application asks her to select a screen. Screens at common locations she visits are listed for quick selection, but the current screen is not listed and she types in the screen ID that is shown in the top left hand corner of every screen. As it is a new screen the application comes back with a confirmation page giving the name "Alexander Square C" which corresponds to the name at the top of the screen. She confirms it.

While she was interacting a (silent) Film Society trailer had started.

After a few moments the phone display refreshes and says "scheduled for 2 minutes". Thirty seconds later it refreshes again "1 minute 30 seconds", "1 minute", then into a 30 second countdown, and as the phone counter steps to zero the trailer ends and the sports channel starts. One side of the screen are shown the latest scores and on the other a series of short clips of goals and hi-lites, including of course, the two Golgate goals!

Note. Because it is a large public screen the scheduler attempts to satisfy Jenny's request within broader content. This helps make those viewers who are simply watching have a smoother flowing experience. Also by doing this it makes it less obvious what content a particular viewer has chosen. In general the information THAT a person is interested in particular information is as important and potentially private as the information itself!

3.6 Snap it 4 later

Demonstrates: use of SD to escape from fixed temporal flow of LD

Charles is eating in Pizza Republic. On the eCampus display there is a short trailer for the Film Society's movies that night, and also a ticker tape news feed. As he eats his pizza he glances up at it and wonders about going to the film that evening. Suddenly he notices something about Golgate Rangers in the ticker tape, but has missed the beginning of the item. he takes out his phone and 'snaps' the display

(N.B. The 'snap' only needs to identify the display, this could use the camera and glyph recognition, bluetooth proximity, or simply, entering the display ID by hand.)

On his phone display is shown a frozen miniature of the screen just as if he had photographed it. He can then use his joystick keys or small numbers overlaid on each panel to select a panel. Once he has the panel he uses the joystick keys like a video remote control to scan backwards and forwards through the channel until he has found the item he wants. "Oh know" he shouts "not signing up Beckham, he's rubbish". The girl at the next table gives him an odd look and Charles thinks it is time to move on.

When he gets back to his room, he chats to Jenny and Ted using IM. "There looked like a good film at the Film Soc" he types. "What is it?" asks Jenny.. Oops! Charles has forgotten, but he quickly opens up the myCampus portal and looks at his snaps. The displays he has snapped are tiled across the web page with times and locations under each. The one from Pizza republic also has a few words underneath from the Golgate news item as that was the point he had scanned to at the time. Happily he was not connected using audio as his language when reminded was not nice.

He selects the snap and gets a larger view of the eCampus display at the moment he snapped it. He uses on-screen VCR style controls to choose the right place and sees the title "Total Recall" and types back into IM. He also notices that in the myCampus text info. window there is extra stuff about the film: what time it is showing, who acts in it, web link to the official film site, reviews etc. So he selects "send to friend" which sends emails (or IM

messages) to Jenny and Ted with a clickable eCampus URI which indicates the display, time and channel so that they can see for themselves.

Some web refs.

SpotCodes

<http://www.highenergymagic.com/spotcode/>

BlueFish

<http://www.nobodaddy.org/portfolio/bluefish.htm>

IR tags on posters

<http://news.bbc.co.uk/2/hi/technology/4081289.stm>