RJ10357 (A0509-003) September 13, 2005
Computer Science

IBM Research Report

Dialing for Displays: Session Initiation Protocol for Opportunistic Augmentation

John J. Barton, Stina Nylander, Fopefolu Folowosele, Beverly Harrison
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Dialing for Displays: Session Initiation Protocol for Opportunistic Augmentation

John J. Barton Stina Nylander Fopefolu Folowosele
Beverly Harrison
IBM Almaden Research Center
650 Harry Road, San Jose CA, USA

Abstract

Opportunistic augmentation denotes connecting a personal mobile device to another device to gain a transient advantage for the user. For example, a mobile phone user might borrow a large display and keyboard from a desktop personal computer. This uniquely ubiquitous computing activity requires effective device and service discovery as well as appropriate media usable across two or more devices. In this paper we show how Session Initiation Protocol (SIP), the call signaling protocol for Voice over IP, effectively separates discovery from media-rendering selection in opportunistic augmentation. This separation improves system flexibility while allowing users or system administrators to choose the most appropriate discovery technologies for the environment. We also describe two phone-centric discovery mechanisms and demonstrate the practicality of the system by implementation and use in a test environment.

1. Introduction

Small handheld computing and communications devices now offer comparable processor speed and storage capacity to low-end personal computing equipment. However no handheld device has a large display and text input to such devices remains a significant research challenge [1].

Pierce and Mahaney propose "opportunistic annexing", a transient use of input/output peripherals like monitors and keyboards, as a way of overcoming these limitations [2]. The use of "docking stations" for notebook computers gives an existing example of the use-model for mobile but not handheld devices. We have varied the name slightly, using "augmentation" rather than "annexing" to suggest that the mobile device needs access to I/O resources, not total control over them. Specifically, we imagine sharing a display across multiple mobile devices. Either way, the nomadic handheld user gains functionality while retaining mobility by leveraging ubiquity of larger, less mobile I/O devices in their environment.

In this paper we explore a phone-call like approach to opportunistic augmentation, using the Session Initiation Protocol (SIP) [3]. After a brief background on SIP, we describe phone-centric use models for opportunistic augmentation, then show that a SIP-based approach is architecturally flexible. We develop two phone-centric address discovery mechanisms to simplify the augmentation and describe their implementation. We show that this approach is practical and relate it to previous work.

2. Session Initiation Protocol (SIP)

SIP, widely known as the IETF [3] call-initiation protocol for Voice over IP, is not limited to voice call sessions. Most Internet Protocol based device interactions would fit within the "session" model of SIP. The protocol uses ASCII messages resembling HTTP messages to create communication sessions: no audio or telephone switching mechanisms appear in SIP. The software that supports SIP is called a "user agent". Typically each participant in a session runs a user agent with both client (sender) and server (receiver) components. The messages contain identifiers that allow the user agent to handle multiple sessions, information about the session setup progress, and descriptions of the media to be used in a session. SIP readily adapts to initiate non-audio other sessions and it has been used in numerous experimental multi-modal and multi-device settings.

In SIP-based opportunistic augmentation (SIP/OA for short), one device is phone: a personal, mobile device with limited human interaction capabilities. The other device has better affordances for humans. For example, the phone could run a web distributed authoring and versioning (WEBDAV) server to allow access to files in the device through a web browser on a display device. Another example would have the phone run a "virtual desktop" and the display device would act as a remote display for that desktop environment. Other media-exchange protocols may appear in the future and they can be added to a SIP-based system without altering the SIP layer.

SIP's Internet Protocol basis, its adaptability, and its close association with voice calls, the dominant use model for handheld devices, all recommend it for use in opportunistic augmentation. A custom protocol for augmentation could be devised, but a complete solution would overlap most of SIP's existing functionality.

In this paper we concentrate on how the user selects the device and the media-exchange protocol matching the capabilities of the device and the needs of the user. Our design addresses both the user-experience of opportunistic augmentation and the augmentation protocol, while largely remaining orthogonal to issues of how the user interface is arranged after the augmentation. We call the device interaction protocols "media-rendering" even when media is only moved between devices rather than rendered for users to interact with. Indeed, our approach says nothing about what happens to media after augmentation. The final user interaction depends upon the media renderer on each device corresponding to the media selected by the user.

3. Assumptions

Our research builds on certain assumptions about future computing. Technology trends point to phones with large digital capacity (e.g. today's 40GB music devices), good Internet Protocol network bandwidth (e.g. more than 50Mbs in current 802.11 networks), and adequate processing units (at least for communications-oriented media rendering). Simply combining these trends with the vast market for phones and the growth of VoIP solutions leads us to expect a phone with most capabilities rivaling a laptop computer but with a small screen and a cramped, uncomfortable keyboard or input with multiple taps on a phone keypad.

Since SIP is based on Internet protocols, this paper also assumes IP connectivity is widely available. To date most handheld devices have been cellular network phones. Even among PDA devices IP connectivity remains a novelty. However, technology and social events are rapidly increasing the number of handheld devices with IP networking technology and the number of places where wireless IP connectivity works well. In addition, IP technology has been demonstrated over wireless technology originally designed for other purposes, like cellular telephone (via for example General Packet Radio Service (GPRS)) and Bluetooth [4].

Finally, we assume that users can physically access augmenting devices and that the environment of these devices enables users to unambiguously determine which devices might be available. That is, we exclude from our concern issues like finding the nearest useful device in our vicinity but out of our sight[5].

4. Use Model

The essential logic behind SIP as a basis for opportunistic augmentation derives from the mobile user's experience. Given a small, portable but personal device with limited human interface capability and an available device with a larger screen and keyboard, we need a simple interaction model for connecting the two devices that people would learn quickly. A phone call clearly matches the capability; we need only teach users that calls could be placed to computers and that non-audio sessions could result. That is, the user is placing a call between two devices that they control or placing a call to another user in the same physical location for other than audio communications.

The basic user experience for opportunistic augmentation with SIP will be that of a "multimedia phone call to myself". The user "dials" the "phone number" of the display and, when the display "answers", a session of user-interaction can proceed. All the double quotes here signal the use of conventional terminology, accessible to users familiar with cellphones, for unconventional purposes. Specifically, there need be no dials, numbers, or audio involved in the augmentation session. The dialing procedure (discovery and binding) can be fulfilled with, for example, RFID sensing as we describe in the section "Click-a-display". The "answer" starts a session within which a data-oriented computing activity can proceed. Even without overt voice call features, the SIP-based opportunistic-augmentation user-experience is modeled on a phone-call in that there is a caller and callee, a call or session, an exchange and finally disconnect.

The call creates a temporary user-initiated digital communications channel, a SIP session. Through this session the user extends the human interface capabilities of the phone. The phone-centric use model provides user control and folds the connection setup into a protocol that provides for mobility, some aspects of security, and support for networking issues like NAT.

If supported widely, opportunistic augmentation could be used by individuals at home, at work, and traveling. To have dependable publicly accessible kiosks while traveling requires both a significant deployment – so users expect to find kiosks for augmentation – and acceptable digital rights management solutions – to reassure users that kiosks are private. Successful augmentation solutions for home, work, and between home and work have less demanding requirements and yet would motivate pervasive deployment. Therefore we will give some idea of the use models for opportunistic augmentation in a business setting to motivate our technology.

For communications-oriented mobile workers like nurses, sales, and people in distributed workforces, handheld devices allow high mobility. Opportunistic augmentation adds usability. Consider a graphics design team manager named Alice. On the subway to work she uses her phone to review her business email messages and calendar for the day. One email from her German colleague has images of their new design. As Alice arrives at their client's offices she borrows their visitor's cubical office. She connects her phone to the PC in this office to review the design. Soon she is off for a series of meetings to present this new design to the client. On the way she stops to use her phone to print the images sent this morning from Germany. During one of the meetings she presents slides on the projector from the data in her phone. Just before lunch she steps into the visitor office and augments her phone with its desktop computer to type a list of feedback ideas for the next design revision and emails it to her teammates.

This user scenario illustrates that Alice can work in a handheld-centered world with high mobility by leveraging computing and communications resources in her workplace. Her files are stored in her phone: she need not rely on access to remote storage. Alice has the flexibility to communicate and compute without booting up a laptop and finding a flat spot to set it down.

5. SIP Opportunistic Augmentation Architecture

The device-connection events that Alice requires to sustain her mobile activities appear quite different from phone calls. Nevertheless we show that SIP, the VoIP call setup protocol, when combined with discovery protocols, can meet a key part of Alice's needs. SIP initiates "sessions" not phone calls. Typically SIP INVITE messages are requests for audio sessions from one person on one phone to another person on another phone. In our case, the request initiator and the recipient are the same person using two different devices and the sessions may involve media other than audio.

In addition to a session setup protocol, the SIP architecture supports indirect addressing through a registration service called a SIP proxy. The indirect addresses are SIP URIs: for most uses the SIP session initiation begins with SIP INVITE messages that require the SIP URL for the recipient.

The SIP approach to opportunistic augmentation provides a concrete and practical solution to some of the challenges raised by Pierce and Mahaney's proposal [2]. Rather than imagine that the mobile device and the augmenting device are to be part of one overall application, we treat these devices as part of multi-device, multimedia session modeled after multi-party, multimedia work designed for the Internet. This divides the sets of problems into discovery, connection, and media rendering. Complex applications can be built with Internet multimedia protocols, but simple ones like browsing files in a mobile phone can be used without additional research. Our architecture addresses only the connection and leaves all questions of interface and user actions after the connection up to the media rendering programs.

Opportunistic augmentation with SIP builds on the SIP architecture and protocol by 1) adding discovery mechanisms suitable to device augmentation that transfer SIP URLs and by 2) adding session description payloads to the INVITE messages suitable for device-device interaction negotiation. With these relatively small additions we create an architecture that supports multiple, widely varied discovery solutions and supports any IP host-to-host media connection. Most important, we can do this without coupling the discovery mechanism to the particular media rendering programs. The architecture is illustrated in Fig. 1. The next two subsections describe these two additions.

5.1. Phone-centric Discovery Mechanisms

The first addition to SIP is address discovery mechanisms suitable for augmentation. We use the term "discovery" more broadly than it is typically used in networking, more in the way it has been used in discussions of ubiquitous computing. SIP address discovery for audio communications typically relies on spoken URLs or electronic directories. Neither are especially suitable for devices. Since we are working with phones which are wireless and have an existing communications oriented infrastructure, we investigated discovery technologies specific to phone-computer connections. We describe them in the next two sections. For clarity we use "display" to mean a conventional keyboard, mouse, monitor with a simple computer as an example augmenting device. Other types of augmentation are possible with the same technology.

5.1.1 Dial-a-Display

Dial-a-display uses the SIP/OA metaphor literally: the user keys a phone number into their handheld device to provide the URI for SIP INVITE. (See Fig. 2.) The use model is unusual compared to a phone call in two regards: first the call is to a nearby device, not one far away; second, the call is answered by the same human that placed the call, just on a different device. Despite these oddities, the user interface could be straight forward as it is driven by user actions.

In the simplest form the user reads a phone number shown on a display device and keys that number into their phone. Here "phone number" does not need to be a conventional telephone number. SIP requires a Uniform Resource Identifier (URI), typically a user name or number followed by a SIP proxy host name. Typically the SIP proxy acts like a phone number area code: if a user does not specify a proxy in a URI, the user's own proxy is applied. Therefore if the phone and the fixed display are part of the same infrastructure, entering the display's "phone number" alone can trigger augmentation. This would be the case in a business setting for example.

Alternatively, the number could be stored in the handheld device either by the user or indirectly by a service provider. Then the user would pick the number for the display from a phone number list. However, the phone-number-list solutions increase in complexity with the number of potential devices and environments. Therefore we expect users to willfully add commonly used devices while relying on directly keying for other devices. Also note that SIP is not limited to conventional phone numbers: the SIP address could also be the device IP address typed into the phone or a character string keyed or selected by the user. Phone numbers are just the most conventional approach for phones.

5.1.2 Click-a-Display

Click-a-display simplifies the user experience by storing the equivalent of the phone number in an electronic tag. (See Fig. 3). The tag can be on the handheld or on the display. For a tag reader on the handheld, the identification tag would be on the display. A user would apply the handheld to the tag, the handheld would resolve the tag-id to a URL [7] which would then be used as a URI in SIP INVITE. For example, if the link implementation was RFID, an RFID tag would be present on the edge of, say, a display. The phone's RFID reader would be placed up to the tag to read it. For optical tags, existing cameras on phones can be harnessed for this mode of discovery [9].

For a reader on the display, the tag would be attached to the handheld device. If the handheld device had, say, an RFID tag, it could be presented to the reader to trigger augmentation. The identifier would

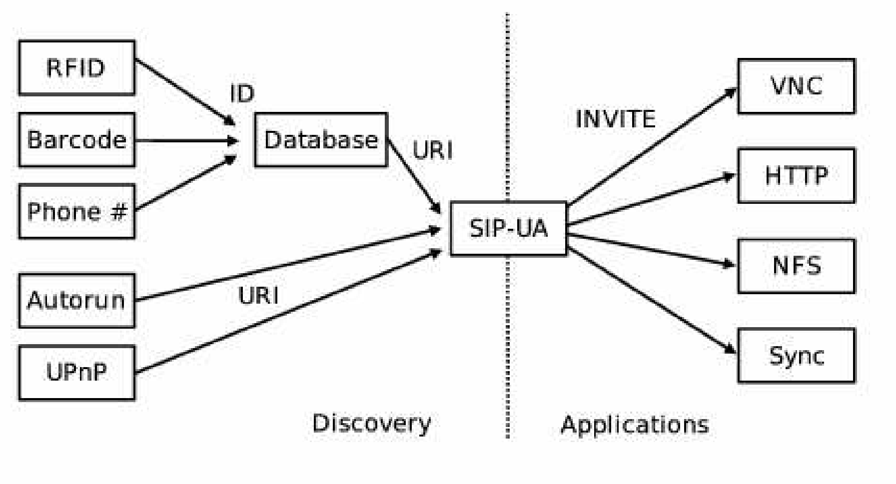

Figure 1. SIP as an architectural separation between discovery and media renderers in opportunistic augmentation. On the left side each box represents a discovery approach: RFID and Barcode mean identification technologies as describe for "click-a-display" "Phone #" refers to "dial a display" in the text. These discovery mechanisms can use a database to convert numbers into SIP URIs. Autorun means using stored files in the mobile device with special names recognized by the augmenting device; UPnP stands in for various network protocol based device and service discovery solutions. In the middle is the SIP User Agent (UA), addressed by discovery, and configured with information on the available media rendering applications. On the right side are media-renderers including VNC ([8]), web browsing where the server could be in the mobile device (HTTP), file editing where the files are served from the mobile device (NFS), and file content synchronization (sync). Discovery leads to a SIP interaction that then negotiates a matching application protocol. The SIP/OA architecture allows flexibility on either side mediated by an Internet standard communications protocol.

be mapped to the user's SIP URI, either by an autonomous identification service [7] or a service combined with SIP service, such as call-forwarding. In the latter case, the display would place a call using the tag identifier in the SIP URI and the SIP proxy would forward the call to the device registered under that identifier. Assuming the user had registered their tag with the SIP proxy, they would receive a phone call to set up the augmentation.

From the user interface perspective, click-a-display is to dial-a-display as mouse clicking is to keyboard command typing. When the number of choices is modest and easy to present visually, clicking will be faster and easier than keying. For click-a-display the RFID reader must be visible and easy to reach, a relatively simple constraint for large displays. Like conventional mouse driven user interfaces, there is some computing and communications overhead to click-a-display because the tag id has to be converted to a phone number.

5.1.3 Multiple Discovery Technologies

Referring back to our scenario, click-a-display might be Alice's preference for editing, printing, and projecting at work. Her company can issue her an RFID tag bound to her phone and instrument the work environment with tag readers [11]. This would add convenience and speed to Alice's opportunistic augmentation at work.

Despite Alice's preference for click-a-display, the existence of alternative discovery technologies could be critical for the overall system success. A newly installed printer may not have RFID technology installed, the RFID reader may fail, the reader could be hard to reach, and so on. In such circumstances Al-

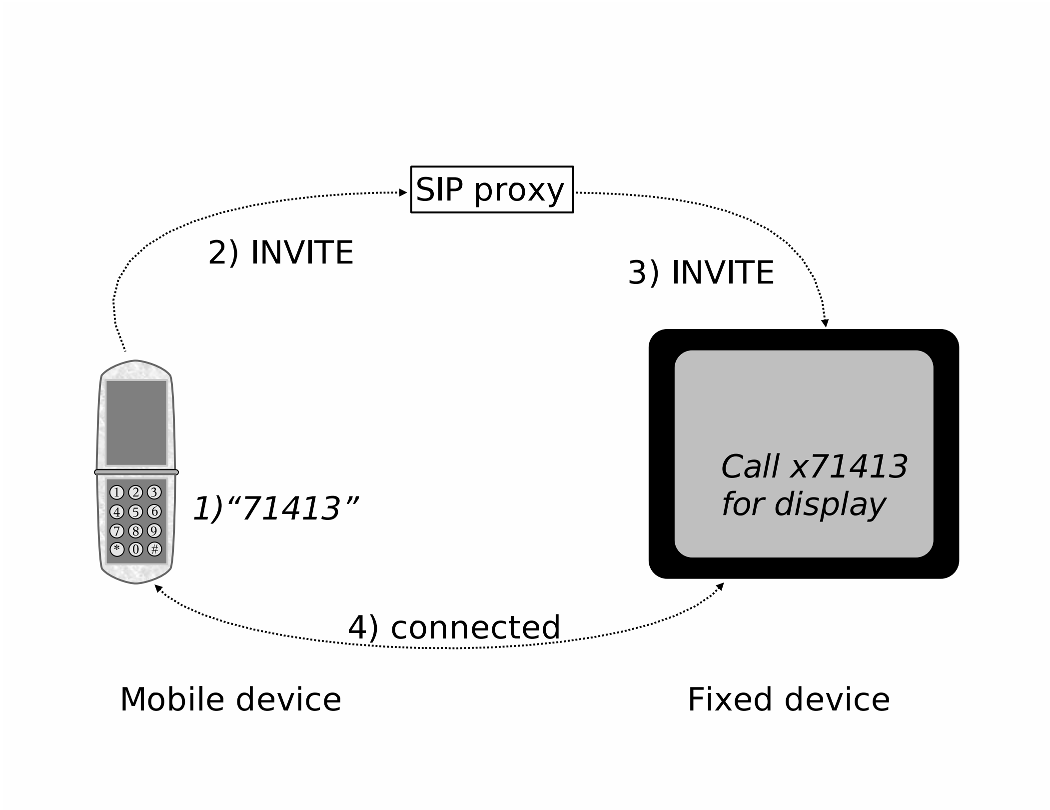

Figure 2. Schematic for Dial-a-display Opportunistic Augmentation with SIP. 1) User reads the phone number of the display and keys it into the mobile device. 2) The mobile device issues INVITE message to the SIP proxy. 3) The INVITE message is forward to the display. 4) The remaining SIP protocol messages establish a session.

ice can fall back to dial-a-display. Indeed she could fall back further to asking some one for a SIP URL over audio or text channels from her phone experience. Moreover, the convenience and speed of click-a-display may be superseded in the future by a new technology. An architecture that separates discovery from application allows flexibility important for pervasive deployment.

5.2. Session Description with Media Type "Application"

The second ingredient in SIP/OA is media description suitable for opportunistic augmentation. The SIP standard [3] relies on the Session Description Protocol, or SDP [12] for specification of potential media sessions. Typically the SIP INVITE message from a caller to a callee will contain in its message body an SDP-formatted message. SDP in turn relies on IANA MIME to specify media types. In Voice over IP (VoIP) uses, the media types in the SDP body are audio and other information in the body details the acceptable attributes for audio communications. RFC 3264 details the communications pattern of SIP messages that allows selection of compatible media and attributes between caller and callee [12].

For opportunistic augmentation we need user agents that support media type "application". Some existing SIP user agents support video media types, but we did not find one supporting media type `application`. Therefore we extended a SIP softphone to add support for media types "application/http" to support web browser/server interaction and "application/x-vnc" to support the VNC remote desktop protocol. The ex-

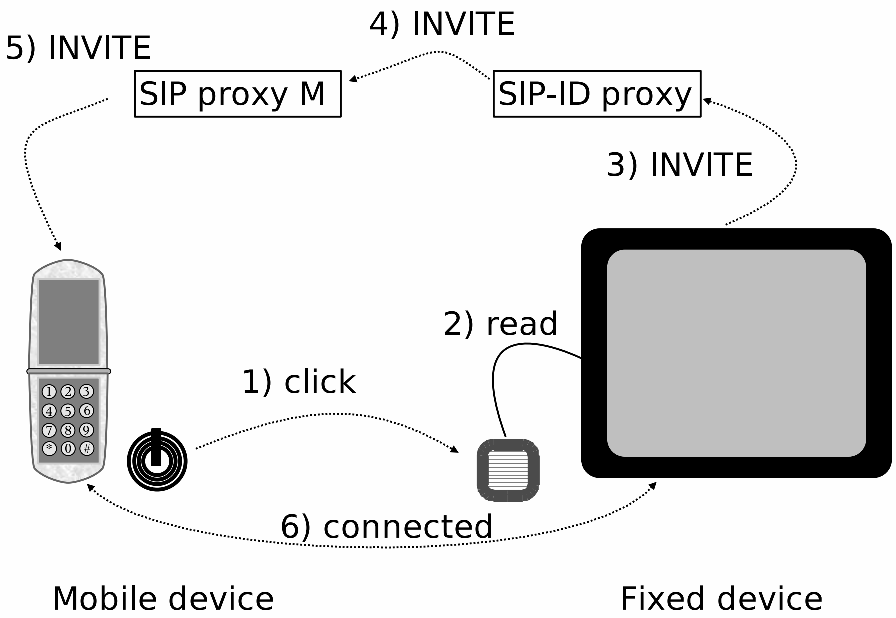

Figure 3. Schematic for Click-a-display Opportunistic Augmentation with SIP. 1) An RFID tag is placed within range of the tag reader for the display. 2) The reader sends the ID value to the SIP UA representing the display. 3) The display's SIP UA issues an INVITE message to a URI created from the ID value and sends it to the SIP-ID proxy. 4) The SIP-ID proxy forwards the INVITE to the registered address for the mobile device at SIP proxy M. 5) The INVITE message arrives at the mobile device. 6) After the human answers, the remaining SIP protocol messages establish a session.

tension required adding new media descriptions to the SDP body sent by the SIP UAC, new "offer/answer" code [13] to compare the session request to the capabilities of the recipient, and new media-handler code for the new media types.

Since SDP evolved for use in Internet telephony, the media descriptions for applications is not worked out in detail. However its basis in IANA MIME types makes extensions orthogonal to the audio and video descriptions. Fig. 4 shows a example SDP document. Our additions in the last two lines offer a web server at port `8080`.

To answer this offer, we apply RFC 2534, iterating over each of the media descriptions (lines starting with `m=`) stanzas and comparing the media types and

```
v=0
o=ThinkLink 0 0 IN IP4 9.1.115.99
s=-
c=IN IP4 9.1.115.99
t=0 0
m=audio 22224 RTP/AVP 0 3 4 5 6 8 15 18
a=recvonly
m=video 22222 RTP/AVP 26 34 31
a=recvonly
m=application 8080 TCP http
a=server
```

Figure 4. An example SDP body with application media formats.

parameters to capabilities of the answering user agent. Any failures to match are marked with port `0` in the answer SDP body. We invented the attributes “server” and “client” and interpret them using the rules for “sendonly” and “recvonly” respectively. The attributes designed for audio and video are not clear when applied to application protocols. Applications tend to be bi-directional so the attribute needs only specify which side will listen and which will talk to start the protocol. Thus “server” and “client” would be clearer.

After the offer/answer phase, we launch a media stream. If this is an application media type, we need to launch or configure a corresponding application. This action will be specific to the user agent operating system and application preferences. Just like web browsers launch external applications to handle media types they do not support directly, our extended SIP softphones launch external applications. We use the same strategy of a locally configured table that maps MIME types to applications.

6. Implementation

To demonstrate separation of discovery and application selection via SIP we implemented a SIP “softphone” to support “application” MIME media types, configured these phones to launch one of two different applications, and connected two different discovery mechanisms to the phones. Running this softphone on a mobile laptop models a futuristic mobile phone; running this softphone on a desktop PC models a display that can augment such a phone. We also worked with a commercially available VoIP phone[15] taking the role of the mobile device, to verify the basic communications elements of our softphone, to understand issues with commercial implementations, and for more realistic user trials. These phones do not support the mime-type “application” so we placed code in our softphone to simulate the reception or transmission of the extended data for the user trials.

6.1. Discovery

In our approach, discovery yields a SIP URI delivered to a SIP user agent. For “dial-a-display” the SIP user agent in the fixed device registered with its SIP proxy under a phone number assigned to the device. The number was displayed on the display user interface (it could also be on a screen saver for the display or written on a sign attached to the display mounting). The SIP proxy address is also needed in general, but we treat this address like conventional phone systems treat area codes: most mobile device users connecting to most fixed devices do not need to specify the proxy because both devices are in the same SIP domain. The user sees the phone number and enters it on the keypad of the phone, completing discovery.

For “click-a-display” an RFID reader was connected to the fixed computer; the reader software feeds any numbers from the reader into the softphone. This SIP user agent prepended the characters “sip:” and appended a SIP proxy address then sent a SIP INVITE to the resulting URI. The SIP proxy is just a conventional proxy [14] with call-forwarding set up to maintain the mapping between RFID tags and mobile devices for the network domain containing both fixed and mobile devices. The RFID tag was placed on the mobile phone. Placing the phone on the fixed-display reader briefly initiates a “phone call” from the display to the phone completing discovery.

6.2. Applications

Discovery ends with the SIP INVITE. For dial-a-display, the body of the INVITE message has `application/sdp` data as shown in Fig. 4. When the fixed display user agent receives the INVITE, it checks the application media types against the media rendering applications configured into the agent. If none match, the call is rejected with 488 Not Acceptable Here code. If one matches, the agent responds with OK and launches the corresponding rendering application. If more than one matches, a dialog box on the fixed display allows the user to choose before continuing with the call set up. This last step differs from the use model for audio where the negotiation between two user agent uses the first matching media format. User testing will be required to determine which approach is most effective for opportunistic augmentation.

For click-a-display, the user-controlled device is receiving the call, but we want this device to control the media offering. We tried sending INVITE from the display with no content. According to the SIP standard this should cause the receiver to respond with a media offer in a SDP body of the ACK message of the protocol. Our particular commercial SIP phone did not respond in this way. Therefore we chose instead to send OPTIONS request from the display. The response to this message should contain all media capabilities of the mobile device. These we display to the user, then send an INVITE from the display with the selected media format as content. Our design principle here is control by the personal device when practical. This is the personal communications device in hand: it should be in control.

7. Practicality

For SIP/OA to be minimally practical, users must 1) be able to connect phones to displays and 2) obtain significant benefit from the augmentation. Using a commercial SIP phone [15], we conducted a 12-person user study to verify that SIP/OA met these criteria. The details of this study are reported separately[16]. The three tests relevant here were for click-a-display, dial-a-display, and–for comparison–manual device connection by looking up the phone's IP address and typing it into a web-browser address bar.

We found the click-a-display (mean 24 ± 1 sec std error of the mean) and dial-a-display (28 ± 2 sec) technology allows people to connect the phone to the display with very little training. These connect times were faster than manual device connection (54 ± 7 sec). Also the variation in connection time as measured by the standard deviation of the times was highest when users typed the phone's IP address in to the web browser (24 sec), lowest for click-a-display (4 sec) and dial-a-display (8 sec). Therefore we claim to meet the first minimum criteria above.

The second criterion, benefit, depends upon the user's needs and skills. For our study we had users enter values in the phone's address book. This task can be done with the phone's keypad through multitap text entry. After augmentation they can use the larger, two-handed keyboard and large screen. We timed users finding the phone-book using the keypad as a a fourth "connection method" for comparison. Access to the phone-book was fast on the keypad, 13 ± 8 sec, but text entry was three times slower than computer keyboard. (These results are consistent with other studies in the field [17, 1] on text-entry speed). Our study showed that users gain from the use of a computer keyboard and screen over the phone's keypad and screen after as little as 15 characters.

8. Related Work

Our work provides a concrete architecture for the concepts of opportunistic annexing explored by Pierce and Mahaney [2]. As we discussed above, we explicitly divide the problem in to discovery and media rendering application so that many issues of two-device user interfaces become conventional problems of media rendering. Unconventional approaches to two device media rendering would be supported as well.

8.1. Phone call metaphor for opportunistic augmentation

We believe the phone-call metaphor will be especially valuable for opportunistic augmentation. The problems facing a nomadic user wishing to use an augmenting device are similar to the problems they face when they want to call someone. They need to obtain the callee's address or "number". They need to establish a session with the callee for the exchange of information. And they need to terminate the session when the exchange is complete. Moreover, users are already familiar with handheld phones for nomadic communications.

Other metaphors for opportunistic augmentation include "syncing" and "web-browsing". The "syncing" metaphor derives from the current use model of Personal Digital Assistants (PDA) when slaved to a master personal computer. The advent of powerful handheld computers with dense storage suggests that the handheld device could one day be the master rather than slave in a synchronization arranegment. Either way, syncing implies (partial) data equivalence as the only application for augmentation. A system that allows more variety in the application used over the connection will suit a wider range of users over a longer period of time.

Want et al. use a web-browsing metaphor in their personal server work [4]. They demonstrate a handheld device with little or no user interface, and thus no ability to "dial". The discovery in their personal server relies on proximity wireless strength with the user interaction driven completely from the augmenting device. Presumably users will imagine their device emitting an invisible aura that is picked up by all nearby displays. These displays may either actively notice every user as they come near – creating a privacy issue – or they may await a user signal to search out nearby handhelds – creating a usability challenge since the user must wait for discovery then correctly select their device from all those within radio range. In addition, disconnection depends upon the highly variable decay of radio strength.

Despite these differences, SIP-based opportunistic augmentation overlaps with both syncing and personal server browsing. Like syncing, our phone-call metaphor is driven by user interaction on the handheld device. Also like syncing, we expect handheld data-replication to be a common activity during an augmentation session, but the replication may not be PC-master to handheld-slave. Like the personal server, our handheld device is a full peer to the augmenting device, not a slave to it. Also like the personal server, we expect users to view data stored in the handheld device on the augmenting display

8.2. Related discovery techniques

Two well-known ways to connect small devices to computers could be used for opportunistic augmentation: docking and network broadcast discovery. Small mobile devices are typically connected to personal computers via docking stations or "cradles". This model arose before wireless communications were widely available and is used by PDAs and portable-memory-devices (such as USB memory devices). The interaction between the personal computer and the mobile device does not resemble opportunistic augmentation: the mobile device acts like a peripheral to the PC rather than other way around.

This docking approach could be used for phone augmentation connections. On the plus side, the deliberate act of docking ensures that the user intends to create a connection. The approach also allows human physical activity for debugging: the physical connector can be examined and retried; removing the device clearly breaks the connection. On the other hand they may not be able to use the handheld device interface comfortably in the docking location and they may tend to forget the device when they leave the location. We expect this connection approach be useful in user-controlled environments where the handheld is stored for recharging, for example in a home application.

Network-based discovery is widely used for small groups of fixed devices and it has been adapted for wireless devices [6]. Here a discovery protocol is run on the network that will later be used for communications between two devices. This approach works well in a laboratory setting or when devices are first deployed in small numbers. As the number of devices and networks increase, we expect the user experience to decline because selection from a long list of potential matches becomes onerous. Rekimoto [24] showed how to use human button-press timings for discovery to avoid this problem. All of these solutions fail if the two devices are on different subnets. Some efforts to overcome the problems of network protocol based discovery have been proposed [18, 19], but they require added network infrastructure.

The dial-a-display discovery method resembles the most common early method for connecting computers: using IP addresses directly in media rendering applications. For example, URLs containing IP addresses can be typed directly into web browsers. The multimedia handling by web browsers is similar to the SIP approach we use. However, the SIP approach fits with the predominant handheld device, the phone; phone numbers naturally work well with phone keypads; SIP already works with NAT and similar multi-network issues; SIP allows either device to initiate the connection.

The click-a-display discovery method looks like SIP INVITE with physical hyperlinks [7, 9]. We could say that the user triggers a physical hyperlink to their handheld device or uses their handheld device to select a physical hyperlink to an augmenting device. Scott et al. [10] use visual tags decoded with cell phone cameras to discover Bluetooth device identifiers, giving user experience similar to click-a-display. They couple application selection with device discovery, but it seems from their work that Bluetooth would support an approach similar to the one implemented here for SIP.

8.3. Security and Privacy

A significant area omitted in this study is security and privacy. For use in businesses and homes, solutions already exist: the SIP phone and augmenting device are simply computers on a network and the connection between them is based on IP protocol with well studied security issues and solutions. Computers supporting augmentation can be secured using convention technologies plus technologies designed to secure VoIP soft phones. For wide applicability in public spaces where trust in the fixed, augmenting computer is harder to establish, opportunistic augmentation requires new security solutions. We note that the problem securing media rendered on a augmenting device that is not controlled by the user of a personal mobile device is a form of "digital rights management", a subject of considerable academic and commercial interest. We anticipate that solutions for digital music and video will apply equally to application media needed for SIP/OA.

8.4. Related Architectural Approaches

Berger et al. [20] integrate a personal device, a watch, with other devices in the user's environment. Their exploration complements ours: they have a non-phone device controlling audio while we concentrate on a phone device used in non-audio tasks. They also emphasis the potential of SIP on personal devices as a technology for user-control of infrastructure.

Raghunath et al. [21] discuss a more advanced personal device that leverages pervasive I/O devices. Their approach aims to give users a more integrated and continuous experience, but requires more elaborate technology.

Olsen et al. [22] (XWeb) and Kindberg et al. [23] (Cooltown) describe ubiquitous computing systems that use HTTP with a flexible discovery approach. XWeb extends web service to add new support and thus fixes the application at XML document manipulation; Cooltown uses simple HTTP which has some media type negotiation similar to SIP. Our work is more specific to a valuable use model: opportunistic augmentation.

Both Bluetooth [10] and IRDA [23] have been adapted to out-of-band discovery and multiple service selection as we have done here with SIP. If SIP were used over an ad-hoc 802.11 network (point-to-point), these three choices would differ primarily in radio frequency and standards organization philosophy. SIP allows infrastructural solutions and its standards organization – IETF – has a strong track record.

9. Conclusion

Our contribution includes 1) an architectural element for ubiquitous computing system that decouples opportunistic augmentation into discovery and session phases, 2) application of an emerging Internet Protocol – SIP – to an important ubiquitous computing paradigm, opportunistic augmentation, 3) demonstrating how existing or proposed discovery techniques would work with SIP in our approach, 4) elaboration and implementation of two light-weight discovery techniques "dial-a-display" [2] and "click-a-display" [11].

10. ACKNOWLEDGEMENTS

Our softphone implementation used the NIST-JAIN SIP stack available from http://www-x.antd.nist.gov/proj/iptel/.

References

[1] P.-O. Kristensson and S. Zhai. SHARK2: A large vocabulary shorthand writing system for pen-based computers. *CHI Letters* 6(2), 2004.

[2] J. S. Pierce and H. E. Mahaney. Opportunistic Annexing for Handheld Devices: Opportunities and Challenges. *Human-Computer Interaction Consortium (HCIC)* 2004, Fraser, CO

[3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol, *IETF Request for Comments: 3261.* June 2002

[4] R. Want, T. Pering, G. Danneels, M. Kumar, M. Sundar, and J. Light. The Personal Server: Changing the Way We Think about Ubiquitous Computing. *UbiComp 2002* Göteborg, Sweden, September 29 - October 1, 2002. G. Borriello, L.E. Holmquist (Eds.).

[5] C. Jiang and P. Steenkiste, A Hybrid Location Model with a Computable Location Identifer for Ubiquitous Computing, *Ubicomp 2002*, LNCS 2498, pp 246-263, 2002.

[6] E. Guttman. Autoconfiguration for IP Networking: Enabling Local Communication. *IEEE Internet Computing* 5(3):81-86, May 2001.

[7] T. Kindberg Implementing physical hyperlinks using ubiquitous identifier resolution. *Proc. of the 11th Intl Conf. World Wide Web*, 2002

[8] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual Network Computing. *IEEE Internet Computing*, 2(1):33–38, Feb. 1998.

[9] I. Haritaoglu. InfoScope : Link from Real World to Digital Information Space. *Ubicomp 2001*: 247-255

[10] D. Scott, R. Sharp, A. Madhavapeddy, and E. Upton. Using Visual Tags to Bypass Bluetooth Device Discovery. *Mobile Comput. and Comm. Rev.*, 9(1):41, Jan. 2005

[11] R. Want, K. P. Fishkin, A. Gujar, B. L. Harrison. Bridging physical and virtual worlds with electronic tags *Proc. SIGCHI conference on Human factors in computing systems* p 370 - 377 1999.

[12] M. Handley and V. Jacobson. SDP: Session Description Protocol *IETF RFC 2327*.

[13] J. Rosenberg H. Schulzrinne. An Offer/Answer Model with Session Description Protocol (SDP). *IETF RFC 3264*

[14] SIP Express Router (SER) http://iptel.org

[15] ZyXEL 2000 Prestige W

[16] J. J. Barton, F. Folowosele, S. Nylander, M. Houston and S. Zhai, "Connection Time for Strange Devices" to be submitted to CHI 2006.

[17] C. L. James and K. M. Reischel. Text Input for mobile devices: comparing model prediction to actual performance. *Proc. of the SIGCHI conference on Human factors in computing systems*, 2001

[18] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, J. Lilley. The design and implementation of an intentional naming system. *Proc. Of ACM SOSP*. ACM, Dec. 1999

[19] S. E. Czerwinski, B. Y. Zhao, T. D. Hodes, A. D. Joseph, and R. H. Katz. An Architecture for a Secure Service Discovery Service. *Fifth Int'l Conf. on Mobile Computing and Networks* 1999

[20] A. Acharya, S. Berger and C. Narayanaswami. Unleashing the power of Wearable Devices in a SIP infrastructure. *Third IEEE Int'l Con. on Pervasive Computing and Communications (PerCom)* 2005

[21] R. Caceres, C. Carter, C. Narayanaswami, and M.T. Raghunath. Soulpad: Reincarnating PCs using portable devices. *MobiSys 2005* p65

[22] D. R. Olsen, Jr., S. T. Nielsen, D. Parslow. Join and capture: a model for nomadic interaction. *UIST 2001.*

[23] T. Kindberg, J. Barton, J. Morgan, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frid, V. Krishnan, H. Morris, J. Schettino, and B. Serra. People, Places, Things: Web Presence for the Real World. *Proc.of the Third IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'00)*, 2000

[24] J. Rekimoto, Y. Ayatsuka and M. Kohno. SyncTap: An Interaction Technique for Mobile Networking.*MOBILE HCI* 2003.