SmartContacts: A Large Scale Social Context Service Discovery System

Yong Liu and Kay Connelly
Indiana University
{yonliu, connelly}@cs.indiana.edu

Abstract

The proliferation of cell phones has led to an ever increasing number of inappropriate interruptions. SmartContacts provides a solution to this problem by empowering the caller to make better, more informed decisions about the appropriateness of making a call by providing him/her information about the receiver's context. SmartContacts also demonstrates the usefulness and feasibility of a large scale service discovery mechanism for a ubiquitous computing environment. It supports automatic service discovery over the Internet in real time and on demand. It adopts a purely decentralized service organization structure, which provides better scalability and fault tolerance than most of the existing service discovery systems.

1. Introduction

SmartContacts is a social context service discovery system that addresses the need to reduce inappropriate incoming calls and the corresponding interruptions to cell phone users. It adopts a large scale service discovery mechanism targeting a ubiquitous computing environment. Drawing on information retrieval technology and the established Internet infrastructure, SmartContacts provides a platform for cell phone users to publish their own context and retrieve other users' context, if allowed by the information owners. It has global automatic context discovery and utilization capability.

There are many scenarios in which SmartContacts is useful. Similar to existing proposed solutions, a person can check the context of someone already in their address book. Unlike existing system, however, they can also retrieve the context of someone they have never called before, as long as they know the cell phone number. For example, imagine a college student finds a missed call on her cell phone after a class. The caller's number is unknown to her. She would like to call back but is wondering if it is an appropriate time to make the call. The challenge in this case is that a user wants to call a person she has never contacted before, and she does not know how to find the receiver's context.

A few characteristics of SmartContacts distinguish it from other service discovery systems. First, it enables mobile devices to automatically discover and consume services to retrieve desired social contexts of a particular cell phone without human intervention. Second, it supports Internet-scale service discovery for mobile devices. Third, it works in a purely decentralized manner, providing better scalability and fault tolerance than most of the existing systems.

Below we describe SmartContacts in detail. Section 2 gives an overview of related work in cell phone interruption issues and service discovery mechanisms. We discuss our approach in Section 3. The architecture of SmartContacts is described in Section 4. Finally, our current implementation is discussed in Section 5.

2. Related work

The number of cell phone users has doubled since 2000 [14]. However, because of the static nature of their notification mechanisms, cell phones often create interruptions to receivers such as ringing in a meeting or movie theatre. Previous research shows that such interruptions may lead to an increased level of receivers' stress and errors [1, 2] and cause annoyance to people in the surrounding environment [3, 4]. Various approaches have been proposed to solve the problem of inappropriate incoming calls [5-7]. Among them is a category known as caller-based solutions [8, 9], which provides the caller with the context of the receiver. Examples of context include a person's location, the activity in which they are engaged, or the number of surrounding people.

The caller-based approach to minimizing inappropriate cell phone interruptions assumes that communication is a cooperative act, and leverages the caller's judgment in determining whether it is an appropriate time to call. This type of solution requires the receiver's social context either to be sensed automatically, or manually set by the receiver, and then delivered to the caller. Current proposed solutions do not address the infrastructure required for the caller's

mobile device to automatically obtain the receiver's context information without human intervention and assume some type of manual bootstrapping. Our approach focuses on this automatic discovery using the existing Web infrastructure.

Service discovery protocols in use today are either local area network protocols (Sun Microsystem's Jini [10] and Microsoft's UPnP - Universal Plug and Play [11]), or target the wide area network but lack scalability (i.e., UDDI [12], standardized by OASIS.), hence do not meet the needs of service discovery in ubiquitous computing.

Based on Java technology, Jini uses service registries to register service proxy objects and provide lookup services. During the service discovery process, a client downloads the service proxy and invokes it to access the service. In Jini, services are identified by the Java class hierarchy.

Different from Jini, UPnP doesn't have a central service registry, but services multicast their announcements to control points that are listening to these messages. Control points can also multicast discovery messages and search for services within the system. XML is used to describe the services in detail.

UDDI is used to build discovery services on the Internet. It provides consistent service publishing and discovery interfaces for users. Services are described in XML and stored in a centralized service registry.

Jini can cover both small networks and larger enterprise networks whereas UPnP is appropriate for home or small office networks. The main drawback to these two systems is that they do not cover a wide area network that spans the whole Internet. Although UDDI is originally designed for Internet-scale service discovery, its centralized architecture greatly limits its scalability in practice.

3. Our approach

Since cell phones are constantly with their owners, callers cannot easily judge if it is an appropriate time for a cell phone owner to receive a call. SmartContacts provides a solution by enabling a caller to retrieve context information provided by the receiver's phone. Examples of such information that can be obtained by state-of-the-art cell phones is the phone's current notification mechanism (e.g. silent and normal), and for smart phones, the description in a calendar entry. It has been shown that calendar entries are a good cue as to whether a person is available for a call [15]. Future cell phones may include sensors that can detect contextual information like the presence of other people [16], or access local-area context services to gather more detailed context information such as the activity in a room. Providing such context information to the caller leverages human judgment in determining the appropriateness of making a call given the current social environment of the receiver [13].

The key technical problem that the caller-based solutions need to solve is the dynamic retrieval of the receiver's context. The caller-based solution should provide ways for receivers to publish their context and for caller's mobile device (in most cases, the cell phone) to identify where to find such information and automatically retrieve it. This information may change frequently, since the receiver's social context is dynamic. We believe instead of publishing context information directly to web pages, a better way is to publish a service handle, through which the current context can be retrieved on demand. There are two reasons. First, a web page does not provide access control capability. When a web service is used, authentication and authorization mechanisms can be added to the service to achieve better security. Second, a web page based approach is not fit for large volume context publishing. For example, a user's context information may include her location, current activity, and number of people surrounding her, etc. It is not convenient to put all of this information on a web page. A better approach is to provide a web service, using some parameters to represent the type of context the service consumer would like to retrieve.

However, most of the existing service discovery protocols do not meet the unique needs of the caller-based solutions. Protocols using a central registry are not a good choice since for a caller, the number of potential receivers could be up to tens of millions. Protocols designed for local area networks do not work since receivers may be distributed over a country, or even the world. Furthermore, we believe the solution should be able to scale beyond phones to any device a person has in a ubiquitous computing environment, where it is not possible for a single authority to maintain the context information published by all of the devices. Hence, we did not desire a hierarchical service registry structure that could not scale with to all devices. In this paper we propose SmartContacts, a context publishing and retrieving system using a new service discovery mechanism that utilizes the existing Web infrastructure.

SmartContacts works in a way similar to retrieving a piece of information from billions of web pages on the Internet. SmartContacts enables cell phone owners to publish their context in the form of a web service and a service description document on a web server of their choice. The description document is then crawled and indexed by major Internet search engines like
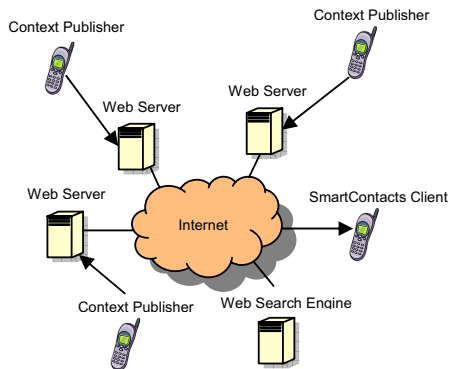
Figure 1. Overview of SmartContacts

Google. When a caller would like to know the context of a specific receiver, his SmartContacts client finds the address of the corresponding web service via Internet search engines and retrieves the context through the service handle. The entire discovery process is automatic. All the caller needs to know is the phone number of the receiver.

The service discovery mechanism adopted in SmartContacts offers the following advantages:

- *High scalability* – services are organized in a decentralized way
- *Wide area service discovery* – services are discovered Internet wide by the existing Internet infrastructure
- *Automatic service discovery and consumption* – services are discovered and consumed according to their descriptions, e.g., a cell phone number
- *High fault-tolerance* – there is no single point of failure in the discovery system maintenance

4. SmartContacts architecture

SmartContacts utilizes the existing Web infrastructure to achieve an automatic service discovery of context Web servers, which can then be queried in real-time to obtain context information. In this section, we provide a description of the system components and describe the context publishing and context retrieving processes.

4.1. System overview

As shown in Figure 1, SmartContacts consists of four major components:

- *Context Publisher* is a program on the receiver's cell phone which automatically publishes the current context.
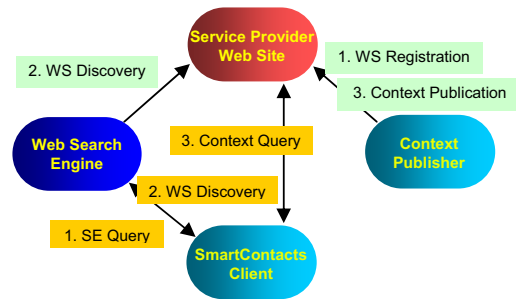

Figure 2. SmartContacts workflow

- *SmartContacts Client* is a program on the caller's cell phone that retrieves a receiver's current context. It uses the receiver's cell phone number to retrieve the information on demand in real time without human intervention.
- *Web Server* hosts context Web services for one or more cell phone users. It can be any third party Web server and does not need to be dedicated for SmartContacts.
- *Web Search Engine* is an existing search engine which is used to locate the service description document of a specific phone, from which a service handle may be attained.

4.2. Service/context publishing

SmartContacts service publishing consists of the following steps, as shown in Figure 2.

- *Registration:* Users publish their context querying Web service WSDL file and a description HTML document containing the address of the WSDL file to a third party Web server. The user must provide a Web server address, but a default web server could be provided by a cell phone service provider to further automate the process.
- *Discovery:* The description HTML document is crawled and indexed by major Internet search engines.[1]
- *Publication:* Context is obtained by the Context Publisher. This could be as simple as determining the current notification mechanism (e.g. vibrate or ring). The Context Publisher runs as a background application, periodically checking the context information and updating the third party Web server via the corresponding Web service.

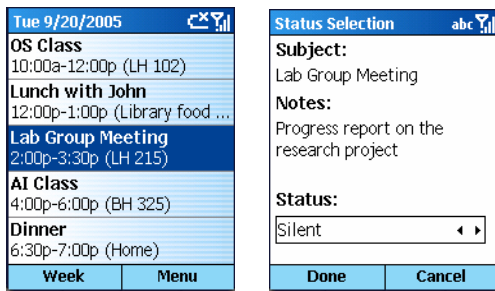[1] Google currently takes 1-2 weeks to discover a new page.

Figure 3. Phone status setup interface

Figure 4. SmartContacts client interface

4.3. Service querying

SmartContacts service querying consists of the following steps, as shown in Figure 2.

- *Search Engine Query*: When a caller wants to check a receiver's context for the first time, her SmartContacts client issues a query to an arbitrary Internet search engine (e.g., Google) via some interface provided by the search engine (e.g., Google Web service API). The keyword used in the query contains some information which uniquely identifies the receiver, such as his cell phone number.

- *Context Web Service Discovery:* Once the corresponding HTML document is obtained, the SmartContacts client parses the HTML to get the address of the Web service WSDL file and download it.

- *Context Query:* By analyzing the WSDL file, the SmartContacts client knows where and how to invoke a remote Web service method to retrieve the context information.

In subsequent use, the SmartContacts client will use the existing Web service address and method to obtain context. If the context is stale or the web service no longer exists, the SmartContacts client will perform the entire process, starting with the search engine query, to obtain a more recent Web service handle, if it exists.

5. Implementation and future work

To demonstrate the feasibility of our design, we have developed a SmartContacts proof-of-concept prototype on the Microsoft Smartphone platform. We use Google as the third party Internet search engine and the Google Web service API to obtain other SmartContacts users' context Web service handles. Curren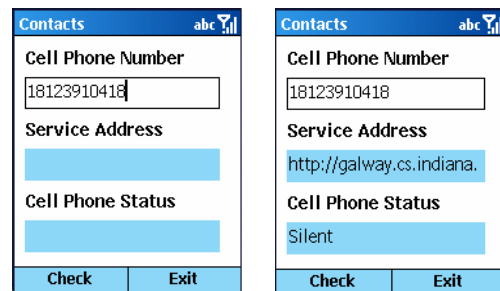tly SmartContacts users need to apply for their own Google API license key before starting to issue their queries via the SmartContacts client. This procedure can be incorporated into the SmartContacts installation process.

The context our prototype publishes is simply the cell phone status. To demonstrate how SmartContacts can be used with other approaches, we developed a cell phone status configuration toolkit on the Smartphone platform similar to that proposed by Khalil and Connelly [15]. Figure 3 shows how users can browse their calendar items and assign a status to each item using this tool. The phone status options are the same as the profiles widely used by major cell phone vendors, including *Normal*, *Silent*, *Meeting*, and *Outdoor*. The tool automatically sets the cell phone's active ring style according to the pre-assigned phone status for the current time period.

The Context Publisher periodically updates the phone status to the remote Web server.

Figure 4 shows our prototype's interface for the SmartContacts client. Once the user types in a cell phone number, the SmartContacts Client automatically retrieves the context Web service address (either from a local cache or the Web search engine) and the current status of the remote cell phone. It is then up to the user to decide to make a call or not.

The entire service query and context retrieval process can be accomplished in a few seconds, most of which is for the one-time Google search to find the Web service. This demonstrates that the current SmartContacts service discovery system can work in real time, which meets the requirement of the caller-based solutions to inappropriate incoming calls.

Currently, the SmartContacts Client interface is a separate application. We intend to integrate it with the current address book, so that the context can be displayed in the address book, and not using a separate interface.

Security and privacy protection mechanism will be the focus of the future development of SmartContacts. In our current implementation, the context Web service

IEEE
COMPUTER
SOCIETY

is found by searching on the phone number. In the future, a more complicated keyword can be used, such as a string with both the receiver's phone number and name encoded in it. This can avoid context information from being exposed to strangers who search cell phone numbers randomly via a search engine. Further protection could be provided with the receiver providing some type of capability to callers, but this would require an offline capability exchange.

6. Conclusion

SmartContacts is a large scale service discovery system which supports cell phone users in publishing their own and retrieving others' context information over the Internet. Using such information, a caller can be better informed weather it is an appropriate time to make a call. Drawing on third party Web servers and Internet search engines, SmartContacts enables callers to access the receivers' context in real time. These features provide necessary supports to the caller-based solutions to inappropriate incoming calls proposed in previous research.

SmartContacts also demonstrates the feasibility of a large scale service discovery mechanism and its usefulness for ubiquitous computing. In the ubiquitous computing era, people may carry multiple electronic devices in their daily lives, and each device may need to publish information to the surrounding environment and retrieve information from the environment via various types of services. Considering the size of the set of all possible services existing in a ubiquitous computing environment, such services are very likely to be organized in a purely decentralized way, just like today's World Wide Web. SmartContacts foresees this trend and leverages modern information retrieval techniques to implement a large scale service discovery mechanism. Its purely decentralized structure achieves better scalability and fault tolerance than most of the existing service discovery systems.

7. Acknowledgements

8. References

[1] Eyrolle, H. and J. Cellier, "The effects of interruptions in work activity: Field and laboratory results", Applied Ergonomics, 2000. 31: p. 537-543.

[2] "Study: All cell phones distract drivers", August 16, 2001, CNN.

[3] Wei, R. and Leung, L., "Blurring public and private behaviors in public space: policy challenges in the use and improper use of the cell phone". Telematics and Informatics, 16, 1999, 11– 26.

[4] Monk, A., Carroll, J., Parker, S., and Blythe, M., "Why are mobile phones annoying?", Behaviour and Information Technology, January–February 2004, vol. 23, no. 1, 33–41

[5] Pering, C., "Taming of the ring: context specific social mediation for communication devices", In Extended Proceedings of CHI 2002, NY: ACM Press, 712-713.

[6] Nelson, L., Bly, S., and Sokoler, T., "Interactive Quiet Calls: Talking Silently on Mobile Phones", In Proceedings of CHI 2001, NY: ACM Press, 171-181.

[7] Horvitz, E., Jacobs, A., and Hovel, D., "Attention-Sensitive Alerting", In Proceedings of UAI 1999, Morgan Kaufman, 305-313.

[8] Tang, J., Yankelovich, N., Begole, J., Van Kleek, M., Li, F., and Bhalodia, J., "ConNexus to awarenex: extending awareness to mobile users", In Proceedings of CHI 2001.

[9] Schmidt, A., Takaluoma, A., and Mantyjarvi, J., "Context-Aware Telephony Over WAP", Personal and Ubiquitous Computing, 4(4), 2000, 225-229.

[10] Sun Microsystems. "Jini architectural overview", Technical report, 1999.

[11] UPnP Forum. "UPnP device architecture 1.0", Technical report, Dec. 2003.

[12] UDDI.org. "UDDI technical white paper", White paper, Sept. 2000.

[13] Avrahami, D., Gergle, D., Hudson, S., and Kiesler, S., "Improving the Accuracy of Cell Phone Interruptions: A Study on the Effect of Contextual Information on the Behavior of Callers". To appear in Behaviour and Information Technology.

[14] REUTERS, "Mobile phone users double since 2000", Dec, 2004.

[15] Khalil, A. and Connelly, K., "Improving Cell Phone Awareness by Using Calendar Information", In the Proceedings of Interact 2005, Rome, Italy, 2005.

[16] Siewiorek, D., Smailagic, A., FuruKawa, J., Moraveji, N., Reiger, K., and Shaffer, J., "SenSay: A Context-Aware Mobile Phone", In Proceedings of IEEE International Symposium on Wearable Computers (ISWC), New York, NY, 2003.

IEEE COMPUTER SOCIETY