InfoFlo: A Novel Communication Infrastructure for Personal Digital Assistants

Noah J. Ternullo, Nader Mehravari, Robert J. Szczerba Lockheed-Martin Federal Systems, Owego, NY 13827
Patrick H. Madden State University of New York at Binghamton, Binghamton, NY 13902

Abstract

Personal Digital Assistants (PDAs) are becoming commonplace, and feature continually increasing processing and storage capabilities. Distribution of media through wireless means to these devices is becoming less expensive, but is not yet wide spread. In this paper, we describe a novel communication infrastructure to provide distribution of information to PDAs utilizing inexpensive methods; our approach is both scalable and flexible. Most PDAs support Ir. (infrared) communication, and we focus on this in our paper; our infrastructure is not restricted to this, however, and is adaptable to the emerging wireless communication technologies.

1 Introduction

Personal Digital Assistants (PDAs) are becoming commonplace. PDA users enjoy access to email and other services, using devices that are portable enough to be carried anywhere. Small screens, modest processing capability, and intermittent use allow for long battery life. With the growth of the cellular telephone market, obtaining Internet connectivity with PDAs has become practical and reasonably inexpensive.

While Internet "enabled" PDAs do have access to a wealth of data, but no inherent guidance in discriminating for useful information. Lacking from a basic connected PDA is *context sensitivity*: the information that is of primary interest to a user may be directly tied to the users physical location, or can inferred from other information available within the PDA. By taking advantage of the context of use, we seek to improve the utility of PDAs..

Our novel communication infrastructure, called InfoFlo, delivers data and services with content that specifically caters to the local context, while still allowing access to more global e-commerce services. InfoFlo is scalable, inexpensive to implement, and can operate with both the emerging wireless networking standards, and also with the most basic IrDA-compliant devices.

In this paper, we first discuss previous work, focusing on context-aware systems, mobile computing, and (briefly) on methods for networking PDAs. We next describe our motivation for this work, and detail our infrastructure. We then describe our prototype implementation, and conclude with a discussion of our current work.

2 Previous Work

Work in their area of context sensitive and mobile computing has been ongoing for some time. In this paper, we follow the approach of [1], considering both personal and environmental context. Personal context generally relates to the interests of a user, while environmental context relates to physical location. The application described in [1] is an electronic travel guide, which allows personal context through the specification of user interests, and environmental context through tracking of user location relative to landmarks.

A similar set of objectives is considered in [2], which focuses on assisting shoppers in an outdoor mall. GPS-enabled PDAs are utilized to determine the location of a user, resulting in the identification of nearby stores. User goals may be specified, allowing filtering of information provided by nearby stores.

In [3], an agent based methodology for integrating ubiquitous and wearable computing concepts is presented. Here, mobile platforms obtain contextually relevant data from beacons located in the environment. Interaction with the environment is handled by sending out agents onto the local network to interact with whatever services may be found. This approach both a bi-directional connection to a wireless network (here RF based), and the local processing power of a PC class device. The beacons present a particularly interesting concept. Originally developed in [4] (using infrared), beacons were used to emit information about the location of the emitter, and other contextual notes that could be left at the beacon by a passer by. A significant limitation of the beacons was that they had no way of being automatically updated, or of responding to a change in their environment.

Army pilots have avionics software which maintains a certain level of contextual awareness, often referred to as battlefield awareness. In [5], software uses the aircraft's external sensors to detect, and respond to pressing events in the pilot's immediate context. One example of this could be the automatic deployment of countermeasures in response to being fired at. Another example would be the automatic replanning of the aircraft's route after the detection of an enemy along its original flight plan.

Interactive trip planning systems placed within automobiles may not be too far off in the future. These systems would allow drivers to find out where the nearest gas station is, and the changing condition of the route as they drive. Routing software that is context sensitive could alert the driver to an impending traffic jam in time to circumvent it. If the driver has been on the road a long time, the software might also indicate a good location to stop and rest. Some of these services are already available from major automobile manufactures.

Context sensitive computing with a PDA can greatly enhance the PDAs value to its user. A PDA which "understands" the context of its user has the ability to be able to make useful information out of a wealth of otherwise random data. If a PDA is aware that the user is at a conference center, and the user is late for a session, the PDA could display a map with a path between the users current position and the conference room.

Context, as described in [6], is clearly an important and relevant issue for PDA use. For activities ranging from travel, to shopping, the ability to customize data to physical location and individual interests is useful.

A primary difficulty comes in identifying the location of a user. Global positioning is relatively expensive, and works poorly indoors. Triangulation of position based on distances from RF transmitters is possible, but of fairly low resolution. Without reasonably accurate identification of physical location, environmental context is difficult to determine. Use of GPS or RF triangulation also implies that a global database is available, providing accurate and current information regarding the users environment—this clearly poses issues for scalability of systems.

Each of the described systems perform some form of customization of data based on context. A range of methods to distribute this data have also been explored; primary focus has been on cellular and RF technologies. Bluetooth [7] is a radio frequency (RF) standard, which may eventually provide a stable and inexpensive local area networking medium for PDAs and laptops. Most PDAs utilize IrDA [8], a protocol for communication using infrared emitters and receivers. While IrDA is bi-directional and can achieve bandwidths on the order of megabits/second, its range is extremely limited, typically being less than a meter, and is therefore not suited to broadcast data needing to cover a large area.

A second significant challenge is in delivery of data to PDAs in an inexpensive fashion. While cellular and RF technologies are increasing in popularity, the installed base is still relatively small. Methods which provide wide areas of coverage (high power RF, for example) face the problem of providing content broad enough to satisfy the needs of large numbers of users, while also providing sufficient bandwidth for each user. If a large amount of data is provided, PDAs must filter this data to identify material of interest to the user. Because of the limited processing power of a PDA, and its limited sensing capabilities, offloading the task of determining context to the communication architecture itself is desirable.

It is interesting to note that in [9], a "computer in the Year 2000" was speculated on. While many of the predictions (for example, stylus-driven touch sensitive screens, large storage capability, light weight and small size) are remarkably accurate, there are also aspects that are surprisingly absent from current computers. In particular, the authors of [9] suggest that infrared communication will be utilized to connect personal computers to nearby equipment such as printers. While most PDAs support infrared communication, this is in general infrequently utilized.

3.0 The InfoFlo System Goals

Our objectives in this work were to design an infrastructure which allows greater utility of PDAs by providing context sensitive information in an inexpensive and scalable manner. We call our current implementation "InfoFlo." Primary design goals included the following.

1) InfoFlo should be inexpensive to the end user. If the system is expensive, it is unlikely to enjoy wide use. Time Warner's early attempt at interactive television never made it to commercial rollout because it required every end user to purchase a multi-thousand dollar SGI workstation for their home. [10]

This requirement forces the infrastructure to address the lowest common denominator in the PDA market in terms of processing power and networking capability. As such, we seek to avoid both GPS, and current cellular and RF based communication. Our design is portable, allowing support for these technologies when cost permits.

2) InfoFlo must be supported by a network medium that allows at least wireless, unidirectional, broadcast capability. We seek to provide information to the PDA user, but in many instances, we do not require a response. While bi-directional wireless network capability is more flexible, it is not required for many activities in our current design. Our infrastructure provides for bi-directional communication, but assumes that this can be performed through a direct (wired) connection.

3) InfoFlo must allow for communication of contextually relevant data.

4) InfoFlo must accommodate the migration of PDAs dynamically into and out of the network.

5) InfoFlo must be flexible enough to support a myriad of services (everything from traditional e-commerce services to mapping clients).

6) The infrastructure must easily accommodate technological changes in network mediums. Eventually all PDAs will have bi-directional high bandwidth wireless network access. The form this takes may vary from Blue Tooth to IEEE802.11b. InfoFlo accommodates these changes with minimal modification.

7) The infrastructure should easily scale to handle contexts as focused as a single object, up through an entire facility or cluster of facilities.

4.1 A Scalable Architecture Supporting Context Sensitive Computing on a PDA

Broadly, our approach is to provide location-specific data to PDAs through a low cost broadcast mechanism. PDAs "browse" this data, identifying items that are of interest to the user. The user may make requests for additional information, and relay the requests through direct connection to the infrastructure.

The InfoFlo architecture is broken down into several different functional pieces. These pieces consists of services, emitters, network points of presence, clients, and the data that moves among them.

Services are resources, available through the infrastructure, which can perform actions for a user. These actions can be anything from providing e-mail access, reserving movie tickets, to providing a dynamic map, making a voice phone call, or playing a game. Services can be performed either on a stationary, networked machine (offsite or on) or on the client side as an executable (such as the case with the dynamic map, or game).

Emitters specifically refer to those elements that broadcast data to the client wirelessly. An Emitter may be tightly coupled with a network point of presence to provide two way network communication, or only broadcast with limited or no capacity to receive. Emitters may be networked, responding to remote stimulus for change, or stand alone, providing static information.

A network **point of presence (POP)** is a site where a client may obtain bi-directional network access. A POP may be wireless or wired, and may or may not be coupled with emitters.

Clients refer to the mobile computing platform taking advantage of the services. While services can (and often will) be "clients" of other services, (such as a shopping service using a credit card transaction service) that is not what is meant here. A client simply refers to the software running on the mobile computing device (here a PDA) to interact with the communication infrastructure.

There are two forms of data moving within the InfoFlo infrastructure. The first is broadcast data. This is data that is sent one way from the emitters to the PDA. This data consists of information concerning location, available services applicable to the present context, announcements applicable to the present context, and other contextually specific data. Although this data may be available elsewhere, it is always at least available through the emitters.

The second form of data moving through the infrastructure is bi-directional and consists of the communications between clients and services. This data is at least available at every network point of presence.

All data moving through InfoFlo consists of well formed XML elements, allowing the same level of scalability found in the world wide web [11].

The context a user acts within can often by indexed by time and location. In the InfoFlo infrastructure, both time and location are leveraged to provide the user with context sensitive information.

Emitters are placed at known locations and time is well known (internally tracked by the PDA and the network). By sending the various emitters different sets of information to be broadcast at differing times, InfoFlo can provide a solid contextual reference for participating PDAs. As the emitters are at known locations, determining the location of the user is a simple matter.

Emitters and network POPs are organized into domains very similar to the Domain Name Service of the Internet. These domains are created based on the location context of the emitters and the POPs.

The chief operating mode of InfoFlo's minimum configuration is "sync and carry", a familiar form of operation for much of the PDA community. Service definitions and contextual information are broadcast to the PDA. The PDA extracts a subset that is applicable to the user. The user selects services and information that he finds desirable, moving to a POP to sync up and receive the designated information or service. Once the information is received, or the service has been performed, the user moves about his business, observing the results on the go.

A fully wireless bi-directional implementation of InfoFlo changes this operating mode somewhat in that the user requests for services and info can be more immediately satisfied, and more interactive, but does not fundamentally alter the infrastructure.

4.2 Infrastructure

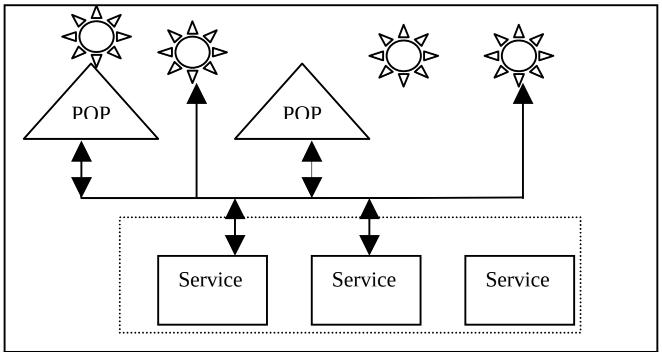

Figure 1: This is the basic InfoFlo Infrastructure architecture diagram. Stars represent emitters, POPs and Services are labeled. The dashed box indicates the service farm. Arrows represent data flow.

A series of emitters are deployed within the target environment to provide the desired resolution of context. Typically, some of these emitters will be coupled with network POPs (point of presence). The vehicle for this coupling in a minimal configuration is a kiosk.

Kiosks are deployed to provide inexpensive bi-directional network connections. A kiosk may be headless, (having no monitor, keyboard, or pointing device), consisting of a simple PC with several network interfaces (such as serial ports using PPP, and an ethernet connection to the LAN). The kiosks are networked through more traditional means forming TCP/IP sub-nets and are connected to the local service farm. The service farm provides the contextually relevant services for the local environment.

In it's minimal configuration, (IE wireless broadcast, and wired bi-directional network participation) InfoFlo works as follows:

A PDA entering the environment either already has the client loaded (download from the web), or may obtain the client through traditional means from any one of the headless kiosks. Once in the environment the client is either activated by the user, or is activated automatically from and internal alarm. The emitters broadcast InfoFlo Broadcast Information (IBI) in a medium the vast majority of PDAs support. This information informs the PDA what context (including location) it is in, and what information and services are directly relevant to that context. The PDA further filters this information to suit its user and presents the user with the results.

These results take the form of textual information about the environment, (perhaps a banner informing the user the gate number of their flight) and iconic representations of the available services. The user selects a desired service in the normal manner.

If the service is known to the client, there may be an associated binary already available for execution on the PDA. This executable may autonomously provide the desired service or may prompt the user for additional information and then synchronize with the network at a kiosk.

If the service is previously unknown to the client, interaction with the service occurs similarly to how a web browser interacts with a server via the common gateway interface (CGI forms). The first time a service's icon is selected on the PDA, the user is displayed a form which was included in the IBI definition of that service. This form serves to obtain preliminary information from the user necessary to completing the desired service function. After filling out the form appropriately, the iconic representation of the service changes to show that it is ready to be synched to InfoFlo. When the user moves to a kiosk and initiates the network connection, the service machine is contacted, and given the form data. The service machine then completes the users request by pushing data back to the PDA. This data may include an appropriately secured binary, a customized iconic representation for the service, an alert message, and any other information requested by the user. The binary takes the place of the dynamically defined form interface next time the user selects the service again.

4.3 The InfoFlo Client

Because of processing limitations, much of the work

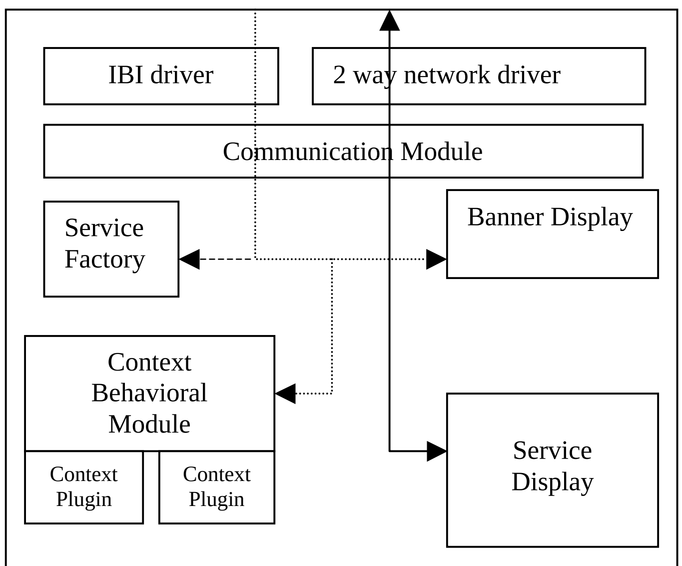

Figure 2: This is the architecture diagram for the InfoFlo client, detailing its internal components. The doted line indicates the flow of IBI data within the architecture. The solid line indicates the flow of bi-directional data.

required for the client to support context sensitive computing is inherently built into the InfoFlo networking infrastructure. A client is still responsible for focusing the available information to provide content specifically useful to the user. This could happen through user interaction, creation of user preferences through observation of the user, and by compiling lists of relevant topics based off of local databases available in the PDA.

To this end, the client must be able to receive contextual information from InfoFlo in the form of IBI, which consists of announcement information (referred to as banner data), service information, and context specific information. Additionally the client must be able to create a connection to the "high bandwidth" bi-directional network when appropriate.

The networking interface must be abstracted from the bulk of the client's functionality insuring that a change in networking medium does not necessitate a change in the client. Different plugins can be used to satisfy different networking requirements. This allows the client to grow with changing technology.

Because of these requirements, the basic InfoFlo Client design consists of a banner display, a service display, a context behavior module, a communication module, and a service factory.

The banner display is responsible for displaying announcement information from the IBI.

The service display is responsible for representing the available services to the user. Typically, an icon is created for each available service.

The context behavioral module is responsible for defining the governing behavior of the client within a particular context. It is this module that "understands" that in an airport the IBI will contain flight and gate information, while at a convention, the IBI may contain show cancellations or updates. This module will interact with the other modules to articulate its context specific behaviors. Behaviors for different contexts (like a mall, a university, or an office) are achieved by "plugging in" code specifically developed for those contexts. These plugins are made available through InfoFlo in the context they are applicable and may also be available ahead of time through the Internet.

The service factory is responsible for creating the iconic representations of InfoFlo services. It is responsible for associating a service available through InfoFlo with a local executable if available, or for generating a default dynamic representation of the service from the available IBI.

5 The Current Prototype

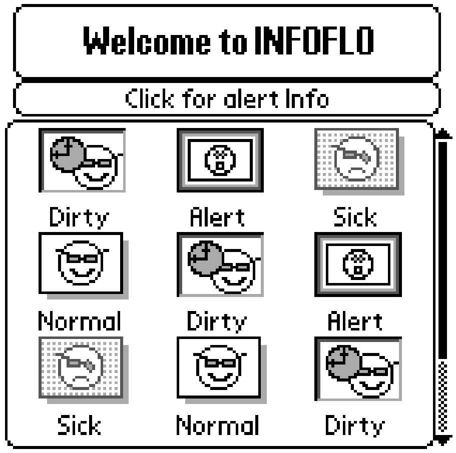

Figure 3: This is a screen capture of the prototype client. The banner is at the top, and the service display area is at the bottom. A button in the middle indicates that an alert is pending. This image details the different states that a service may exist in {Dirty, Alert, Sick, & Normal}.

The current InfoFlo prototype is a proof of concept system showing the viability of the InfoFlo communications infrastructure in its minimal configuration.

Working to minimize the implementation cost of the client, no additional hardware was required on the PDA, a Palm III. A hybrid Diffuse Ir.-IrDA network was set up to allow broadcast of IBI to the PDA client via its Ir. port (as commonly available on most PDAs). When required the PDA could connect to the bi-directional network through serial PPP at a headless kiosk.

The PDA client is written in C++ using Metrowerks CodeWarrior with the appropriate abstraction of the actual communication interface. The client consists of a banner display area, a service display area, and an alert button. The banner display area shows relevant text from the infrastructure as a tickertape. The service display area depicts available services iconically. The alert button appears only when some piece of the client posts an alert, (for example if the context behavioral module determined that the user flight was delayed). When clicked on, further details regarding the alert are revealed to the user.

Flexibility for the client was provided by utilizing a listener design pattern. Internal client components registered listeners for the IBI elements of interested. In this way, plugins to the client, (such as those in the context behavioral module), can use the existing architecture to register for new types of elements.

The mechanism through which PDAs and services dynamically join, discovered and initially interacted with each other was architected with the help of Jini, Sun's impromptu networking architecture.
Interactivity between the C++ client and Jini was done through a proxy object executing on the headless kiosk.

6 Summary and Conclusion

It is not enough to simply connect a PDA to the Internet, even if the Internet offers useful service in and of itself to the PDA. To gain optimal usage of the PDAs wealth of user information it needs to understand its context. Understanding its context allows it to provide useful information to the user out of the piles of data that may be available.

Because PDAs have tight processing and memory constraints the more the work required to determining the user context is offloaded the better. To this end, the InfoFlo Infrastructure allows for the dissemination of contextual sensitive information and services to PDAs.

Currently, we are refining our current system and preparing for a test deployment. Course schedules at SUNY Binghamton are available electronically, and might be suitable for an initial broadcast test. Many students carry PDAs, and we expect that electronic requests and deliver of homework may be an interesting application. We are also considering installation of a prototype system in an airport, to provide flight information to PDA users.

Our prototype browser operates on the Palm platform; a WinCE version of the browser is being developed.

References

[1] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou, "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences," *Proc. ACM CHI 2000*, pp. 17-24

[2] A. E. Fano, "Shopper's Eye: Using Location-based Filtering for a Shopping Agent in the Physical World," *Autonomous Agents '98*, pp. 416—421

[3] Bradley J. Rhodes, Nelson Minar, and Josh Weaver, "Wearable Computing Meets Ubiquitous Computing", *The Proceedings of The Third International Symposium on Wearable Computers*, San Francisco, CA, October 18-19 1999, pp. 141-149

[4] D. Kirsch and T. Starner, "The Locust Swarm: An environmentally-powered, networkless location and messaging system", *The First IEEE International Symposium on Wearable Computers*, Cambridge, MA, 1997

[5] P.N. Stiles, M.O. Hofman, "*Demonstrated Value of Data Fusion and Situation Assessment*", American Helicopter Society Avionics and Crew Systems Technical Specialist's Meeting, Philadelphia, PA, 1997

[6] C.-S. Chang and A. L. P. Chen, "Supporting Conceptual and Neighborhood Queries on the World Wide Web," *IEEE Trans. SMC*, part C, Vol. 28, No. 2, May 1998, pp. 300-308

[7] BlueTooth Special Interest Group, *http://www.bluetooth.com*

[8] Infrared Data Association *http://www.irda.com*

[9] B. W. Mel, S. M. Omohundro, A. D. Robison, S. S. Skiena, K. H. Thearling, L. T. Young, and S. Wolfram, "TABLET: Personal Computer in the Year 2000," *Communications of the ACM*, Vol. 31, No. 6, June 1988, pp. 639-646

[10] Palmer W. Agnew, S. Kellerman, *Distributed Multimedia-Technologies, Applications, and Opportunities in the digital information industry*, Addison-Wesley Publishing Company, 1996

[11] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret, "The World-Wide Web," *Communications of the ACM*, Vol. 37, No. 8, August 1994, pp. 76-82