

# Smart Wireless Access Points for Pervasive Computing

Roel Ocampo<sup>1,2</sup> and Hermann de Meer<sup>2,3</sup>

<sup>1</sup> Department of Electrical and Electronics Engineering, University of the Philippines,  
Diliman, Quezon City, 1101 Philippines

<sup>2</sup> Department of Electronic and Electrical Engineering, University College London,  
Torrington Place, London WC1E 7JE

<sup>3</sup> Faculty of Mathematics and Computer Science, University of Passau,  
94032 Passau, Germany  
{r.ocampo, h.demeer}@ee.ucl.ac.uk

**Abstract.** Research in ubiquitous computing has traditionally focused on sensing and making use of user-related context. However, a pervasive computing environment is also a mobile computing environment, where QoS adaptation is often a major concern. We present an integrated framework applicable to both user- and network-oriented adaptation, that uses a common base for sensing context. Smart wireless access points called M-WASPs provide a wireless network infrastructure, and perform sensing, adaptation and other cognitive functions in a distributed fashion, within a pervasive computing environment.

## 1 Introduction

More and more computing power is being integrated into everyday appliances, which in turn are increasingly being networked. We share the pervasive computing vision described by Mark Weiser in 1991 [1], a future environment where computing power and computing services will be found everywhere, but paradoxically, will increasingly be transparent to the user. Although these services are hidden, it does not mean that they are neither useful nor accessible; in contrast, much of their utility will stem from the fact that user need not explicitly issue commands to these hidden computers in order to use available services.

While there have been significant advances in the area of context-aware computing in research environments, we still have to see it making significant inroads in our daily lives. A major prerequisite is the physical infrastructure needed to sense users, their activities, and their environment. Some forms of context might be relatively easy to extract, especially in indoor environments that are already automated to a certain extent. Lighting levels and temperature, for instance, may be obtained from home or building automation systems, or in future home gateways that serve to monitor and control indoor environments. Obtaining other forms of user context, such as the location and orientation of people, may be more challenging. This often involves the deployment of dedicated and often specialized infrastructures of sensors.

A complementary and practical approach might be the integration of sensors as add-on functions on commonly-seen and already-useful devices, services, appliances and infrastructures. Research efforts such as Smart-Its ([www.smart-its.org](http://www.smart-its.org)) have focused on providing small everyday devices with some sensing, communication and computational abilities. Schmidt and Van Laerhoven call these smart appliances: devices that are aware of their environment [2]. The core sensing and processing hardware of such devices are often governed by considerations on their mobility, size, weight, and power consumption, and have quite limited computational power. In addition, most of these devices associate with the rest of the computing environment in a wireless and often ad-hoc fashion.

We believe that these small, highly-mobile smart devices will indeed play a critical role in the pervasive computing landscape. However, *smarter* devices that are less mobile and that possess less restrictions on their computing and sensing capabilities will also play an equally important role in providing a ubiquitous communications and computing infrastructure. Aside from sensing the environment, these devices form part of a relatively fixed network infrastructure, participate in distributed computation, and crucially, provide wireless access to their smaller counterparts.

## 2 Context-Awareness and Adaptation

Much of the research in pervasive and ubiquitous computing has focused on context-aware applications: applications that provide relevant information and services to a user, using information about that user's situation, such as her location, identity, and the state of people, groups and nearby objects [3]. In response to certain situations or changes in context, such an application may automatically alter its execution, trigger the execution of another service, or prompt the user to act in a certain way. We call this type of response *user-oriented adaptation*.

A pervasive computing environment, particularly one that handles diverse forms of media-rich information, also shares many of the issues inherent in the area traditionally known as mobile computing. A user interacting with and moving within such an environment may generate large variations in network traffic due to the diversity of applications being executed, and experience high variability in connectivity characteristics provided by the different access technologies that are available. Since we are interested in smart appliances that provide access to the network, we are equally interested in *network-oriented adaptation*, or the ability of a system to adapt to variability in network conditions such as traffic levels, available bandwidth, congestion levels, and connectivity characteristics [4]. This form of adaptation has been closely studied in systems such as Transend [5], Odyssey [6], Conductor [7], Coda [8], Transformer Tunnels [9], and CM [10].

A valid question would be whether network-oriented context and adaptation are relevant in a pervasive computing environment. The relevance of context always depends on the user's objectives and applications, i.e., her task [3]. In

applications where the quality of the information being transmitted, or the perception of its quality by the user (i.e., fidelity [6]), its timeliness, or the user experience may be affected by variations in the performance of the underlying network, such context is indeed relevant. A fairly sophisticated level of context-awareness in a system would be needed for it to be able to judge whether such context is relevant or otherwise, and to manage both network-oriented and user-oriented adaptation in parallel. To emphasize the level of “smartness” needed to support such context-awareness and adaptation, we call services that enable them *cognitive services* within a pervasive computing system.

### 3 A Framework for Cognitive Services

Serving as our guide in the design of software that enables cognitive services is a framework that defines the various component functions. A framework can be used as a guide to understanding existing systems, and a useful tool in building new ones. Decoupling the various components of a framework and implementing them separately promotes the design of reusable components with well-defined interfaces, and abstracts the underlying implementation details. Services may be composed from ‘mix-and-match’ combinations of components that satisfy certain criteria for functionality, performance, and resource utilization.

Our goal is to have an integrated framework for both user- and network-oriented adaptation that builds upon a common base for context sensing. We found several useful frameworks for the analysis and design of context-aware and adaptive systems in the literature. One such conceptual framework for network and client adaptation is presented by Badrinath et al. in [11], summarizing the results of several network-adaptive systems. In their framework, adaptation mechanisms are implemented by adaptation agencies, or AAs. An AA consists of an event manager (EM) component that monitors the environment, a resource management and monitor (RM) component that handles resources, and application specific adapters (ASA) that perform adaptation on a data stream.

For context-aware applications, Pascoe identifies four generic capabilities that are needed, namely: contextual sensing, adaptation, resource discovery, and augmentation [12]. In this framework, contextual sensing refers to the detection of environmental states and their subsequent presentation to the user. Applications then adapt their behavior to this contextual knowledge. Contextual augmentation extends these capabilities further by adding information, either through the digital data augmenting reality, or reality augmenting digital data. Contextual resource discovery makes context information and information resources available to interested entities.

Several other models have evolved from work on location systems. WhereMoPS [13] provides a layered system model for indoor geolocation systems that includes data collection, location computation, location normalization, and location provisioning components. In contrast with Pascoe’s approach, WhereMoPS decouples location sensing into separate data collection and computation components, permitting the use of different positioning algorithms. Normalization then

transforms the computed location into a standardized representation, and the information is provided to applications in the provisioning step. The Location Stack [14] model, similarly focused on location context, consists of a seven-layer stack that includes sensors, measurements, fusion, arrangements, contextual fusion, activities and intentions. Although both the WhereMoPS and Location Stack models were originally proposed to handle location context, the essential components may be applied to other types of context sensing.

Dey, Salber and Abowd offer a conceptual framework that includes context widgets, interpreters, aggregators, services and discoverers [15]. Context widgets abstract underlying sensors and acquire context information. This information is further abstracted by interpreters into higher-level information. Aggregators gather information relevant to an entity. Services then execute behaviors using context. Finally, discoverers maintain information on which of these components are available for use by applications.

Schmidt proposes a “perception architecture for context-aware systems,” consisting of sensors, cues, contexts, and the applications that use them [16]. Physical and logical sensors provide information about the world, to be abstracted or processed into symbolic or sub-symbolic values called cues. The context layer then abstracts cues into situations and decides whether a situation satisfies the definition of a particular context. The context is then passed on to applications.

Our framework for cognitive services thus builds upon the functions, characteristics, components and models previously described. It serves as a common framework that integrates network-centered adaptation in mobile computing, and application-centered adaptation in context-aware computing over a unified sensing base. It accounts for the different processes and stages that bridge sensing and adaptation, such as perception, awareness, reasoning, judgment, and augmentation. We impose no strict precedence between these components, and they may be executed in a variety of ways: sequentially, iteratively, or in a recursive fashion:

**Sensing.** Sensing refers to the collection of measurable or quantifiable physical data or the observation of an event. Such data or events may either be directly measured by hardware or software sensors, or may be higher-level data and events previously detected or generated by other cognitive components and services. Examples of low-level sensed data or events at various levels might be the strength of an 802.11 RF signal, a mobile device associating with an access point, beacon signals impinging on a receiver, the execution of an application, or statistics on the utilization of a network link.

**Interpretation.** The sensed data are transformed into useful form by interpretation, which may involve the application of a numerical process or an algorithm, by comparing it with a model, or by the application of a logical process such as reasoning. Values obtained may also be checked if they are within expected or acceptable range, and a confidence parameter may be applied to a measurement or estimate. This component may also detect the occurrence of an event based

on the values of data obtained. As with most other frameworks, the output of this stage may be numeric or symbolic.

**Augmentation.** Sensed data and events may be aggregated with, or examined in relation to, other pieces of information or knowledge. These may be recently-sensed or interpreted data or events originating from other sensors (sensor fusion), or historical data from the same sensor. Historical data may be useful in establishing a trend or in improving the accuracy of estimates from new sensor data through statistical means. Information previously generated may be retrieved from databases, maps, or models. For example, the sensed location of a user may be compared to a map of the location of objects and spaces in order to generate new information on the proximity of users with respect to other entities (“users  $X$  and  $Y$  are near each other”), or containment within spaces (“user  $Y$  is within the space in front of workstation  $W$ ”). An important aspect of the augmentation stage is the ability to detect and form relationships and connections between sets of data, events, and prior knowledge.

**Adaptation.** Adaptation, which we broadly define as goal-oriented action in response to changes in context, may benefit different entities of the system. Some examples we have previously given are user-oriented adaptation and network-oriented adaptation. Although these are by no means the only forms of adaptation that are necessary or present in a pervasive computing system, these are the forms in which we are primarily interested.

The adaptation component in our framework includes the specific actions in response to the detection of certain contexts, or changes in these contexts, as well as the policies and strategies necessary to execute the adaptation itself. Some adaptation strategies include launching new services or applications, modifying application or network behavior, suggesting a course of action to a user, or reserving resources [17]. There may be cases where adaptation may apply even to the components of the framework: sensing, interpreting and augmentation processes may also have to adapt, under certain situations. An interesting form of adaptation that leads us one step closer to truly cognitive services would be *learning*, the ability of a system to modify its view of the world through newly-acquired knowledge.

## 4 A Platform for Distributed Cognitive Services

We share the view in [17] that motion is part of everyday life and thus a pervasive computing environment should provide mobile and ubiquitous connectivity. Towards this end, we earlier suggested the need for *smarter* appliances that possess not only sensing and distributed computing capabilities, but can also provide connectivity to other devices as well.

Our generalized architecture and physical platform that models and simulates the capabilities of computing appliances in future pervasive environments,

providing computing, connectivity and sensing functions, are called *M-WASPs*, or *Multi-modal Wireless Access and Sensing Platforms*. M-WASPs possess the following basic functions and characteristics:

1. *Access points*. M-WASPs are wireless access points, providing ubiquitous connectivity in a pervasive computing environment. M-WASPs may extend the wireless network by associating in an ad-hoc manner, or by provide wireless access to a wired infrastructure. We also enhance the usefulness of networked appliances and devices by enlisting them to provide wireless access, through embedded M-WASPs.
2. *Multi-modal*. To reflect and support the diversity of connectivity options within pervasive computing environments, M-WASPs are multi-modal, employing a wide variety of connectivity and sensing technologies such as IEEE 802.11, Bluetooth, infrared, and acoustic sensing capabilities. We also explore a wide range of user input devices and classify them as sensors in our framework.
3. *Sensing platforms*. A key requirement for obtaining context is the use of physical and logical sensors for the acquisition of data or detection of events. Sensing may either be a dedicated function for a device or a software program, or it may be an add-on or overlay function. For example, in RADAR [18], 802.11 wireless access points were used beyond their traditional function of providing connectivity, to sense user locations. We intend to exploit this fully using other interfaces and technologies.
4. *Programmable*. A key feature of the M-WASP architecture is programmability in the active networking context [19],[20]. An active networking approach allows us to flexibly deploy cognitive service components within the network on an on-demand basis. We push the active networking paradigm further by supporting not only mobile code, but mobile data as well, within our wireless access points.

Active networks have long been recognized as a promising infrastructural solution for adaptation [7],[11]. The Mobeware middleware toolkit [21], for example, is based on an open programmable networking architecture and runs on mobile devices, wireless access points and mobile-capable switches and routers. This effort has produced a useful set of objects and APIs to provide QoS support for adaptive mobile networking.

In application-level active networks (ALAN), active networking principles are applied at the application layer rather than the lower layers of the network. FunnelWeb [22] is an ALAN implementation that supports the dynamic deployment and execution of proxylets in active nodes in the network. These proxylets implement network-oriented adaptive functions such as transcoding, compression, and caching. We have previously used the ALAN approach for the segmented adaptation of traffic aggregates [23] and to dynamically manage and optimize peer-to-peer traffic on bandwidth-limited links [24].

In real deployments, M-WASPs may represent a wide range of devices: infrastructure elements such as routers and dedicated wireless access points, mobile

user appliances such as PDAs and mobile phones, or computing elements embedded in a diverse range of appliances such as office equipment, networked environmental control systems, pieces of furniture, and household appliances. At the extreme end, M-WASPs may be dedicated computing devices such as PCs. M-WASPs may vary in terms of actual capabilities and configurations such as processing power, memory and storage, interfaces, connectivity and sensors, according to their intended application. However, the heterogeneity of the underlying hardware is abstracted by an architecture whose primary function of interest from our viewpoint will be the provision of computing, connectivity and sensing functions – distributed cognitive functions – in addition to any general or specialized embedded computing function the particular device might serve.

## 5 Sensing and Representing Location Context in M-WASPs: Examples

M-WASPs are *smart wireless access points* in the sense that in addition to providing wireless access, they are also actively involved in providing distributed cognitive services, such as adaptation functions at the user and network levels. Sensing various forms of user and system context is also another main function. While the notion of “context” that may be relevant to a user may involve a wide range of things, the location of users and surrounding objects plays a very important role and consequently has been the subject of much interest for many researchers.

In this section we discuss some of the schemes we use in M-WASPs to sense and represent fine-grained location information in an experimental setting.

### 5.1 A Method for Sensing Location

For our experimental use, we needed a location sensing scheme that would be compatible with commercially-available off-the-shelf devices such as a PDA. To roughly estimate the position of users within a relatively large area, such as within a 3–4 meter radius, techniques such RADAR [18] and its variants may be used with a PDA outfitted with an IEEE 802.11 interface. For fine-grained positioning, such as within 10–20 cm., while a number of systems such as the Active Bat Location System [25] and Cricket [26] have been discussed in the literature, most of these systems usually use ultrasonic transducers and a channel for transmitter-receiver synchronization such as an RF or infrared channel. We preferred a position sensing scheme that would use available interfaces and require only a bare minimum of hardware interfacing, if any at all.

Aside from our preference in using off-the-shelf components and built-in interfaces, we likewise made an assumption that the located object would operate asynchronously with respect to the positioning system. A location estimation technique known as hyperbolic multilateration does not require the tracked object to be synchronized with the positioning system. If beacons, or pairs of beacons can be closely time-synchronized with each other, a receiver could detect the

arrivals of each beacon’s signal and measure the relative delays between them. If signals from beacons  $i$  and  $j$  arrive at a receiver at  $t_i$  and  $t_j$ , respectively, referenced to the receiver’s clock, then the time-difference-of-arrival (TDOA) is simply  $t_i - t_j$ . For a receiver located at coordinates  $(x, y, z)$ , and any two beacons  $i$  and  $j$  located at  $(x_i, y_i, z_i)$  and  $(x_j, y_j, z_j)$  respectively, the equation describing the range difference  $r_k$  corresponding to the TDOA  $t_i - t_j$  for this pair is given by:

$$\begin{aligned} r_k &= c(t_i - t_j) \\ &= \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \end{aligned} \quad (1)$$

where  $c$  is the propagation speed of the beacon signal used. Equation (1) describes a hyperboloid, and with at least four beacon signals, three independent TDOA values may be obtained, producing three independent equations. The solution to these three simultaneous equations yields the  $(x, y, z)$  position estimate of the receiver. Graphically, this corresponds to the intersection between the hyperboloids generated using (1) for any set of three TDOA values.

We applied this technique using acoustic spread-spectrum signals, with standard PC speakers as beacons and a receiving microphone as the located object. The use of spread-spectrum facilitates the detection of the arrivals of beacon signals at the receiver due to their excellent correlation properties, and provides some measure of resilience against noise and environmental scattering [27]. Four PC speakers, each with a single 3.5-inch driver, were positioned in a room. Three of these were mounted on the ceiling, approximately 2.2m above the floor on average, while one was mounted on a wall, approximately 80.5 cm above the floor level. Acoustic beacon signals, consisting of 127-bit Gold codes with a chip rate of 10 kchips/s and BPSK-modulated with a 10 kHz sine wave, were simultaneously transmitted through the speakers. A microphone (simulating a PDA in our test scenario) recorded the received signal every 10 cm on a 130 cm x 110 cm grid. The recorded signal was then successively correlated with each of the transmitted Gold codes. A correlation peak indicated the instant that a beacon signal arrived at the microphone. The speed of sound is approximated to the first order using the formula

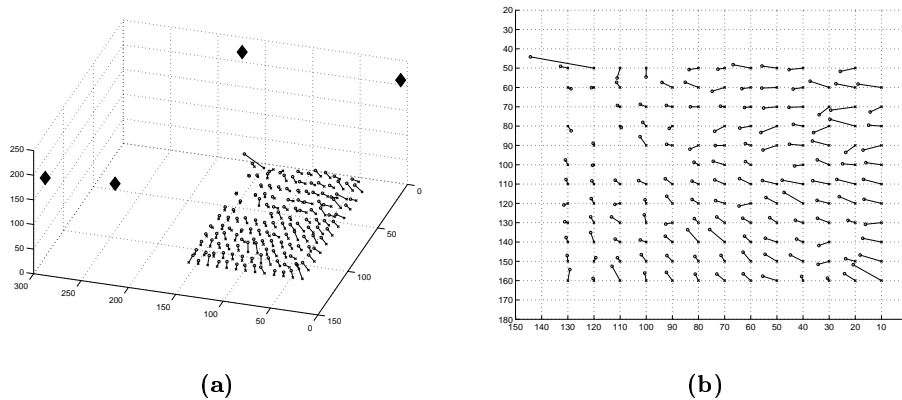
$$c \approx 331.5 + 0.610t_{\text{air}} \quad (2)$$

where  $t_{\text{air}}$  is the air temperature in degrees Celsius and  $c$  is in meters/second. In our experiments, the temperature was recorded from a digital thermometer. In an actual implementation, the ambient temperature may be supplied by an online sensor. Alternatively, a fixed approximate value may be used in environments where the temperature is regulated or typically does not vary to a large degree. Since the equations for each TDOA pair represented by (1) are nonlinear, we linearized them using the first two terms of their Taylor series, and used least squares to solve the resulting equations.

The results of one of the trials of our acoustic position sensing scheme is shown in Fig. 1. The positions marked with “x” indicate the actual microphone



positions, while positions marked with “o” indicate the position estimates computed through hyperbolic multilateration. Lines interconnect pairs of actual and computed positions. The gaps in the grid where there are no “x” marks represent points where the least-squares algorithm did not converge within the maximum number of iterations, or the resulting computed position was outside the coordinate system. For the data shown in the figure, the computed position deviated from the actual position by 7.0 cm on average, and 80% of all computed positions deviated by less than 9.4 cm from their actual positions. In sensing the location of people and objects, we are normally more interested in their  $(x, y)$  position rather than their elevation above the floor. For the data represented in Fig. 1, the deviation from the actual positions along the  $x - y$  plane was around 4.6 cm on average, and less than 7.5 cm for 90% of all computed positions.



**Fig. 1.** (a) 3D plot showing measurement points, computed positions and beacon locations. (b) Top view. (Axes are marked in centimeters.)

In an actual implementation the position may be computed on the PDA itself by performing the hyperbolic multilateration computations on the sensed beacon signals and a preloaded database of beacon codes and the corresponding  $(x, y, z)$  coordinates of the speakers transmitting them. Alternatively, the position information of the speakers may be contained within the beacon signals, although this would require more processing on the part of the PDA and the use of longer code sequences. Either way, the PDA’s location would not be known to the network, so this may be considered a mode of operation that preserves the privacy of the user. A non-privacy or tracking mode would have the PDA transmit the recorded beacon signals back to the M-WASP, using either an 802.11 or Bluetooth interface, for the location computations. This mode sacrifices privacy in favor of hardware simplicity, as it shifts the computational burden from the tracked object to the network. The system we have described can be rapidly deployed and used, as no additional hardware construction is

needed. Our technique does not require any synchronization between the beacons and the located object, and allows very simple commercial devices with little or no computational power, such as an analog wireless microphone, to be tracked. Even an ordinary audio recorder, for example, may continuously record acoustic beacons as it moves within an area, and its traversed path may later be post-processed and reconstructed. However, in a real-life scenario, rather than a laboratory deployment, the use of audio beacons in the audible range might be annoying to users. In such cases, it would be necessary to shift the working frequencies to the ultrasonic range, and although this would require some simple hardware modifications, the basic principles would remain the same.

## 5.2 Representing Spaces

For location context to be useful to applications, information on the location of users, nearby objects and spaces of interest must be represented in a form that can easily be stored, transmitted and processed. Having a simple and efficient scheme for representing location and spaces, and other forms of context in general, is desirable for the following reasons:

*Support for simple but smart devices.* A scheme that is simple enough such that context may be exchanged, stored and processed by simple devices with limited memory and processing capacity promotes a distributed model of context-awareness and cognition. Such a model not only supports users who ask “What useful objects or services are nearby?” but also allows simple devices to pose the similar question “Who are the nearby potential users of my service?”

*Robustness.* If context can be distributed throughout the system and processed in a distributed manner, then there is less chance of catastrophic failure, in contrast with a system that relies on a centralized server to process and store location information.

*Privacy.* Privacy can be enhanced if the user can select the entities that will be made aware of her context. In some cases, it may be sufficient to share context locally to nearby devices on a need-to-know basis. This is possible if these nearby smart devices can process context information locally within the constraints of their limited computing and storage resources.

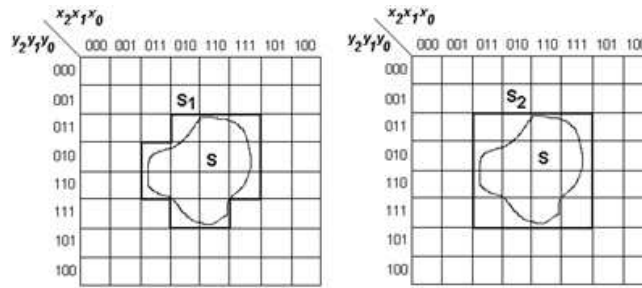
*Network-friendliness.* A scheme that uses a simple representation minimizes the bandwidth consumed by the exchange of location context, a crucial consideration for wireless networks.

We now discuss a scheme for representing locations and spaces that satisfies our design requirements for simplicity and efficiency.

Large spaces such as rooms, are partitioned into small cubes, and each unit cube is uniquely identified by a set of Gray-coded coordinates, similar to Karnaugh maps [28]. A more precise term used in Boolean algebra for these unit cubes are 0-cubes. A 0-cube may be contained within one or more spaces of

interest, such as “the space in front of workstation  $W$ ,” and such a space of interest is completely defined by the set of cubes that completely enclose it. This is similar to the definition of space containment in [25]. Spaces of interest may thus be represented in terms of Boolean functions that describe the logical sums of 0-cubes that completely enclose them. With spaces represented by Boolean functions, various logic operations may then be applied to determine relationships between spaces and the location of objects and users, such as intersection and containment. Operations such as combining spaces may be done through a logical union of Boolean expressions. Proximity may be determined by testing for inclusion within larger enclosing spaces.

Most of the spaces in which we are interested, e.g. “the space in front of workstation  $W$ ” consist of collections of contiguous groups of unit cubes. We exploit this property to obtain compact space representations by applying logic minimization techniques. This approach is similar to the representation and compression of images in [29], however, our goal is not to compress an image of the entire area including the details of all objects contained within. Rather, our goal is to obtain simple and compact representations only of the spaces and locations of interest. A two-dimensional simplified example of this space representation scheme is shown in Fig. 2. A space of interest,  $S$ , is shown on this figure. We also indicate two areas,  $S_1$  and  $S_2$ , that completely enclose  $S$ . The Boolean



**Fig. 2.** A space of interest  $S$  and enclosing spaces  $S_1$  and  $S_2$  are drawn on a Karnaugh map

expressions for  $S_1$  and  $S_2$  in minimal sum-of-products form are

$$f_{S_1} = x_1y_1y'_0 + x_1x'_0y_1 + x_2x_1y'_2y_1 \quad (3)$$

$$f_{S_2} = x_1y_1 \quad (4)$$

It can be seen from this example that the geometry of the enclosing area used to approximate the original space of interest affects the extent to which the switching expression can be minimized. Generally, in building the maps that represent our spaces and the objects contained, we can select geometries that will lead to more compact representations. One advantage of this space representation

scheme is that it implicitly encodes both position and size information. However, there are some potential weaknesses. First, as previously mentioned, space geometries other than those that may be conveniently enclosed in rectangular cubes with dimensions that are integer powers of 2 may tend to have more complex expressions. Second, it tends to view the world in “black and white,” that is, if the location of an object is represented by Boolean expression  $f_{S^*}$ , then the representation implies that the probability of actually finding the object outside  $S^*$  is zero. Third, within  $S^*$ , the probability of finding the object is uniformly distributed. These might not be the case if the area being represented by  $f_{S^*}$  is generated by a sensor that inherently can only provide an estimate of location.

One solution to the first problem above, aside from careful geometry selection, is to explore other shapes that lead to simple expressions in other minimal two-level Boolean representations such as exclusive-OR sum of product (ESOP) forms. Additionally, if the space information is to be shared, the simpler product terms in the minimized expression (corresponding to larger and more regular sub-areas) can be transmitted first for rapid, “short-circuit” logic evaluation. In some cases this may eliminate the need to transmit the other product terms, and thus while the stored representation remains complex, it does not generate too much traffic. Finally, to represent probability distributions that are non-uniform, a space may be built from composite overlapping or non-overlapping sub-areas and a probability assigned to each component.

## 6 Related Work

Girod and Estrin use acoustic spread-spectrum techniques in a ranging system that uses frequencies in the audible range [27]. Although their system only produces range (distance) information rather than location, it may be extended through multilateration in order to estimate location. This has in fact served one of the technical bases for the acoustic location sensing component we have developed for M-WASPs. In addition, their philosophy of using COTS hardware has served as a guide for us in designing a system that can be rapidly deployed and used in conjunction with commercial devices such as PDAs.

A privacy-oriented location system based on ultrasonic DS/CDMA spread-spectrum and pseudorange has recently been presented by Hazas and Ward [30]. Although the physical sensing base is identical to ours, the difference in approach (hyperbolic multilateration vs. pseudorange), i.e., the interpretation component, has some implications in the overall design of the location system, particularly in the need for synchronization within the system. A pseudorange system typically requires beacons to be tightly synchronized with each other, and there is likewise some benefit in synchronizing the located object with the beacons as well, as this minimizes the magnitude of the clock bias that needs to be estimated. In a hyperbolic multilateration system similar to ours, tight synchronization is required only among pairs of beacons to provide accurate TDOAs, and to a lesser extent, across different pairs. While synchronization between the beacons and the located object may minimize the number of acoustic

data samples that need to be processed, the algorithm itself does not require it. At any rate, our ability to use a different interpretation component, i.e., post-processing algorithm, over the same physical sensing base as may be dictated by the sensor deployment scheme or other system concerns, precisely illustrates the usefulness of the modular framework we have presented.

Broadly, while our research shares a number of common objectives with similar research efforts in context-awareness within the pervasive and ubiquitous computing research community, we extend these further by paying equal attention to the problems caused by the mobility of users and devices, as well as their information flows, within these environments. Our work is more related to efforts such as Carnegie Mellon's Aura [31], which seeks to create distraction-free environments that adapt to users' context and needs. Aura applies the concepts of proactivity, or the ability to anticipate requests from a higher layer, and self-tuning, or autonomous adaptation by layers. Unlike many similar efforts in the ubiquitous computing domain, Aura correctly provides attention to the network-oriented adaptation that is required to support user mobility. We seek to achieve similar broad objectives through our application of a well-defined framework for cognitive services, delivered within the pervasive computing environment in a distributed, scalable and robust fashion, through our programmable, smart wireless access points called M-WASPs.

MIT's Project Oxygen (<http://oxygen.lcs.mit.edu>) aims to provide human-centered computation freely available everywhere. Environmental devices, called E21s, provide sensing, computational and communication functionality for intelligent spaces. These devices are interconnected through flexible, decentralized, adaptive networks called N21s. Our M-WASPs seem to share characteristics similar to E21s, and our active networking approach allows the deployment of services necessary to manage networks similar to N21s within the same platform, using an integrated framework for cognitive services.

## 7 Conclusions

We have introduced M-WASPs, which are smart wireless access points for pervasive computing. M-WASPs provide a wireless infrastructure linking users with smart devices and other networks. They also provide sensing and cognitive functions, enabling user- and network-oriented adaptation. We have also presented a framework for distributed cognitive services that includes components for sensing, interpretation, augmentation and adaptation. This framework serves as our guide in designing the active code that can be dynamically deployed and executed on M-WASPs. Having efficient and simple means of sensing, representing, exchanging and processing context, such as space information, promotes a distributed, modular approach to cognitive services. As an example, we presented a scheme we use in M-WASPs to sense and represent location and spaces. These simple techniques for sensing and representing location and spaces may be of interest for experimenters in context-aware computing and in related domains where positioning may be needed, such as in robotics and virtual reality.

## Acknowledgment

Roel Ocampo's work is supported by a Doctoral Studies Fellowship from the University of the Philippines.

## References

1. M. Weiser. The Computer for the 21st Century. *Scientific American*, September 1991.
2. A. Schmidt and K. Van Laerhoven. How to Build Smart Appliances? *IEEE Personal Communications*, 8(4), August 2001.
3. A. K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing Journal*, 5(1), 2001.
4. R. Katz. Adaptation and Mobility in Wireless Information Systems. *IEEE Personal Communications*, 1(1), 1994.
5. A. Fox, S. D. Gribble, Y. Chawathe and E. A. Brewer. Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives. *Proc. 16th Intl. Symposium on Operating Systems Principles (SOSP-16)*, France, October 1997.
6. B. Noble, M. Satyanarayanan, D. Narayanan, J. Tilton, J. Flinn and K. Walker. Agile Application-Aware Adaptation for Mobility. *Symposium on Operating System Principles*, November 1997.
7. M. Yarvis, P. Reiher and G. Popek. Conductor: A Framework for Distributed Adaptation. *Proc. 7th Workshop on Hot Topics in Operating Systems*, March 1999.
8. L. B. Mummert. *Exploiting Weak Connectivity in a Distributed File System*. PhD thesis, Carnegie Mellon University, School of Computer Science, 1993.
9. P. Sudame and B. R. Badrinath. Transformer Tunnels: A Framework for Providing Route-Specific Adaptations. *USENIX Annual Technical Conference*, June 1998.
10. D. Andersen, D. Bansal, D. Curtis, S. Seshan, and H. Balakrishnan. System Support for Bandwidth Management and Content Adaptation in Internet Applications. *Proc. 4th Symposium on Operating Systems Design and Implementation*, San Diego, CA, October 2000.
11. B. Badrinath, A. Fox, L. Kleinrock, G. Popek, P. Reiher, and M. Satyanarayanan. A Conceptual Framework for Network and Client Adaptation. *IEEE Mobile Networks and Applications*, 5(4), December 2000.
12. J. Pascoe. Adding Generic Contextual Capabilities to Wearable Computers. *Proc. 2nd International Symposium on Wearable Computers*, October 1998
13. M. Wallbaum. WhereMoPS: An Indoor Geolocation System. *The 13th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications*, Lisbon, Portugal, September 5-18, 2002.
14. J. Hightower, B. Brumitt, and G. Borriello. The Location Stack: A Layered Model for Location in Ubiquitous Computing. *Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002)*, Callicoon, NY, June 2002
15. A. K. Dey, D. Salber, and G. D. Abowd. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction (HCI) Journal*, 16 (2-4), 2001
16. A. Schmidt. *Ubiquitous Computing - Computing in Context*. PhD dissertation, Computing Department, Lancaster University, November 2002.
17. M. Satyanarayanan. Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, 8(4), August 2001.

18. P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location And Tracking System. *Proc. IEEE Infocom 2000*, Tel-Aviv, Israel, vol. 2, March 2000
19. D. L. Tennenhouse and D. J. Wetherall. Towards an Active Network Architecture. *Computer Communication Review*, 26(2), April 1996.
20. A. T. Campbell, H. G. De Meer, M. E. Kounavis, K. Miki, J. B. Vicente and D. Villela. A Survey of Programmable Networks. *ACM SIGCOMM Computer Communication Review*, 24(2), April 1999.
21. A.T. Campbell, M.E. Kounavis and R. R.-F. Liao. Programmable Mobile Networks. *Computer Networks*, 31(7-8), April 1999.
22. M. Fry and A. Ghosh. Application Level Active Networking. *Computer Networks*, 31(7), April 1999.
23. H. De Meer and P. O'Hanlon. Segmented Adaptation of Traffic Aggregates. *Proc. Quality of Service - IWQoS 2001, 9th International Workshop*, Karlsruhe, Germany, June 2001.
24. H. De Meer, K. Tutschku, and P. Tran-Gia. Dynamic Operation in Peer-to-Peer Overlay Networks. *Praxis der Informationsverarbeitung und Kommunikation (PIK Magazine), Special Issue on Peer-to-Peer Systems*, June 2003.
25. A. Harter, A. Hopper, P. Steggle, A. Ward, and P. Webster. The Anatomy of a Context-Aware Application. *Proc. 5th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MOBICOM)*, Seattle, WA., August 1999.
26. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. *Proc. 6th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MOBICOM)*, August 2000.
27. L. Girod and D. Estrin. Robust Range Estimation Using Acoustic and Multimodal Sensing. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, Maui, Hawaii, October 2001.
28. M. Karnaugh. The Map Method for Synthesis of Combinational Logic Circuits. *Trans. AIEE. pt I*, 72(9):593-599, November 1953.
29. A. K. Chaudhary, J. Augustine, and J. Jacob. Lossless Compression of Images Using Logic Minimization. *Proc. Intl. Conf. on Image Processing*, Vol. 1, September 1996.
30. M. Hazas and A. Ward. A High-Performance Privacy-Oriented Location System. *Proc. First IEEE International Conference on Pervasive Computing and Communications (PERCOM '03)*, Dallas-Fort Worth, USA, March 2003.
31. D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste. Project Aura: Toward Distraction-Free Pervasive Computing. *IEEE Pervasive Computing*, April-June 2002.