In Proc. of IEEE Services Computing Conference (SCC'07)
Salt Lake City, Utah. pgs. 146-153. July 9-13, 2007.

Ubiquitous Services

Claudio Pinhanez
IBM Research, T.J. Watson
pinhanez@us.ibm.com

Abstract

This paper proposes the use of service-based architectures as a way to bring the benefits of ubiquitous computing solutions to real world problems. We compare this services approach to the traditional view of computer applications as user tools, noticing that service providers can bring significant economies of scale not only in monetary terms but also in computational simplifications based on multiple user data aggregation and large scale data information processing and gathering. We argue for such ubiquitous services *by analyzing the evolution of the World Wide Web in the last two decades and the success of service providers such as* Google, eBay*, and* Mapquest *in changing the way people live; and by showing how a services approach can make feasible many solutions to real problems such as urban congestion, preventive health care, and global trade.*

1. Introduction

"[Technologies] weave themselves into the fabric of everyday life until they are indistinguishable from it." was the fundamental claim of Mark Weiser's seminal paper of 1991 [13], which eventually led to the establishment of the *ubiquitous computing (ubicomp*) research field in computer science. For the last 15 years, researchers around the world have created an incredible amount of highly innovative and creative visions of how computers can disappear [2], prototyping some, piloting them sometimes, and even making some few available commercially. Of those visions, however, very few have or are likely to materialize in the everyday world in the near future.

A powerful analogy is often cited in support of the ubicomp efforts, in which computers today are compared to cars in the beginning of the 20^{th} century. In those days, the driver needed the knowledge of a mechanic: going for a ride often involved a great deal of technical skill to fix the invariably occurring mechanical problems. Nowadays, most of the technology of an automobile is hidden from its users to the point that they can concentrate solely in the act of driving, if so.

There is, however, an often overlooked mechanism by which technologies also disappear: they often become services provided by people or companies. From the point of view of an airline passenger, flying is as ubiquitous as driving a car — if not more — because the flying technology is offered as a service. There is no need to know about flying techniques, protocols, regulations, or even the technology behind the flying machine itself. Airline customers just enter the plane, sit, try to relax, and wait for the arrival.

In many other cases, a user has also the option to choose between using herself a set of tools or procuring a service to accomplish a goal. To go from Boston to New York, you can drive your car (the tool option) or take the air shuttle (the service option). Similarly, enterprises and organizations often have to decide to offer a new product as a tool or as a service. For instance, users can now buy music files (seen here as tools to produce audio) in *iTunes* music stores or subscribe to Internet radio services like *Rhapsody*.

This paper argues that, in many scenarios, a better paradigm for ubiquitous computing applications is based on the notion of a *service* — *ubiquitous services* — instead of the traditional view of a *tool*. However, in this paper we will not demonstrate the adequacy of this approach by examining an implementation of the concept in detail. Instead, we support our idea with three complementary arguments.

We first argue that many of the changes in computer and network usage in the last 15 years were due to the emergence of innovative services instead of powerful or smart personal tools. In our view, service providers such as *Yahoo*, *Google*, *Travelocity*, *eBay*, *Skype*, as well as *IBM*, *EDS*, and *InfoSys* in business services, were the *de facto* agents of computing change in the last two decades, with the majority of the killer applications enabled because a services instead of a tool approach was taken.

A services approach to application design, development, evaluation, and deployment is based on architectures where the user's client application/interface utilizes a service provider instead of trying to produce the information or services itself. Common advantages of this approach are economies of scale and aggregation gains due to access to data from multiple users. Such factors, as we discuss in the second part of our argument, can simplify considerably the solution of some traditional problems in the field of ubiquitous computing such as privacy management, technology deployment, and, of course, user support.

As the third and final part of our argument, we show how a services approach can make ubicomp a technology to solve real, important problems, moving away from a tendency of the ubicomp researchers to focus on solvable but unimportant annoyances of everyday life. We discuss some of these real-world problems at the end of the paper and how a services approach may simplify the design and deployment of practical solutions.

We notice here that even when ubicomp applications have explicitly used a service-centered approach as, for example, in the cases of [5, 6, 8], there were hardly any effort to understand what the use of a service paradigm actually implies, and how simpler it can render the whole system.

At this point is, however, important to be clear: this is a vision paper that proposes the use of a new structural foundation for the design and deployment of ubicomp applications based on the services paradigm. It is not possible to prove that this is the right approach for the field, even if we could list hundreds of successful cases where this paradigm had been applied. The goal of this paper is to suggest that there is possibly a better alternative to the current tool paradigm and that the service paradigm is able to, theoretically and according to our view, overcome some of today's ubicomp challenges and limitations.

2. Forecasting Tools, Getting Services

Mark Weiser's seminal paper [13] is part of an extraordinary collection of essays about the upcoming era of integration of communications, computers, and networks published in a special issue of *Scientific American* in 1991. The issue includes articles from technology visionaries such as Michael Dertouzos, Vinton Cerf, Nicholas Negroponte, Alan Kay, Mitchell Kapor, and then US Senator Al Gore. Among other things, the articles predict in 1991 the appearance of large scale broadband networks, the non-centralized structure of today's World Wide Web, and the emergence of India as a software outsourcing powerhouse, as well as problems such as junk mail, cyber-crime, and identity theft and trade. Of course, as in any collection of futuristic ideas, many predictions in the special issue never materialized, including a deep change in how we educate children and increases in democracy and peace across the world.

More interesting for the sake of our argument, however, are some of the essential components of the current computer network environment that were never suggested by this extraordinary group of futurists. In particular, they all fail to recognize that the massive interconnection of users would open the space for large service providers that mediate and simplify the relationship between users and the vast amount of data in cyberspace. Intermediaries that negotiate, process, and filter public and private data, collect and analyze people information, and provide simplified information access, as services, such as *Google*, *Yahoo*, *eBay*, *Travelocity*, *Amazon*, *Skype*, etc.

The common thinking 15 years ago seemed to be that the access to the myriad of computers in the network and the browsing and filtering of their data would be performed by personal tools or software agents that would scout and explore the web for information relevant to their users. A good exemplar of this view is the concept of *knowledge robot*, or *knowbot*, proposed by Robert Kahn and Vinton Cerf [7], *"…programs designed by their users to travel through a network, inspecting and understanding similar kinds of information…"* as described by Dertouzos [4, pg.35]. In Cerf's view, knowbots would be unleashed to fulfill specific user requests for information, moving *"…from machine to machine, possibly cloning themselves […] dispatched to do our binding in a global landscape of networked computing and information resources."* [3, pg.44].

The problem with this and similar visions based on powerful personal tools is that they do not scale up. In the current world of distributed information, this "knowbot" approach to information search would require each of us to run (and possibly store) the equivalent of *Google*'s operations of crawling the web and query matching. What the authors of the *Scientific American* issue could not see is that, as networks and their users grow, there are tremendous economies of scale when millions and millions of queries are handled by a single system that crawls and indexes all the information available and provides information finding as a service. For a fee or, as it turned out, in exchange for a chancy click on a paid link.

Scalability is one of the fundamental arguments for our view of applying a services paradigm to ubiquitous computing: satisfying a large group of users is (per capita) significantly simpler than pleasing one user, as demonstrated initially by *AltaVista*, and then by *Yahoo* and *Google*, among others. Not only economies of scale are possible but also the last decade have seen the appearance of several *"cast of thousands"* data handling methods such as collaborative filtering [10] and social matching [12].

Similarly, compare the simplicity of using a service provider of geographical position information such as *Mapquest* to most location devices sold in the market. While in the services case all the data access, search, and updating is handled by the service provider, often location devices require the user to manually find/request and install new maps in the system. Even when using in a limited area, it is very common that the information in the device becomes outdated as time goes by. *Mapquest* users, instead, are oblivious to data maintenance, which is cared for by expert people in the services provider organization. The most profound technologies are those that disappear.

Being fair to the authors of the *Scientific American* issue, no one in the early 1990s seemed to have grasped the real meaning of connecting in quasi-real time thousands, millions, and billions of people. But there is more here than failure to understand how gargantuan networks of people behave: there is a reluctance to consider service providers as being more effective than a personal tool. Possibly this is fruit of the traditional computer engineering training which tends to see introducing another agents in a problem as always increasing the complexity, not possibly diminishing it.

It is illustrative that this difficulty of including the possibility of large scale service providers also framed the failure of the HCI research community in leading the design of user interfaces for information search and filtering in the web. Throughout the 1990s, two information filtering approaches battled within the HCI community: intelligent software agents, often associated with the work of Pattie Maes [9], vs. user-controlled visually-based direct manipulation, championed by Ben Shneiderman [11]. A famous debate in CHI'97 publicly clashed the two researchers and views, without a clear winner. But we all know now the real winners of that battle: *Altavista*, later followed by *Yahoo* and *Google*, which took an approach of filtering based on large-scale crawling operations, massive index databases, and deep analysis of the structure of the network and of trillions of user queries — only economically feasible in the framework of services!

3. Examples of Ubiquitous Services

Services are the standard approach for maintenance and support of most of the machinery we use today. People routinely take their cars to repair shops and hire contractors to fix or renovate their homes. We also routinely rely on services provided by experts — physicians, lawyers, financial advisors — to navigate the information mazes of medicine, law, and the market.

In the computer and network arena, although hardware and software installation is becoming increasingly performed by users themselves, we still rely on service providers for maintenance and support. In a different level, enterprises aggressively embraced the idea of outsourcing their IT needs to service providers during the last decade, to companies such as *IBM*, *EDS*, or *InfoSys* in India.

But we are not talking about IT outsourcing here. What we are advocating is a paradigm shift in the way we architect software solutions and, in particular, user applications. We are arguing that there are significant simplifications, in many areas of ubiquitous computing, by considering application architectures where the bulk of the work is provided by a services system instead of being performed by a tool in the user's system. To better understand the difference, let us examine three traditional problems in ubiquitous computing and see how a services approach can simplify some of their inherent difficulties.

PRIVACY MANAGEMENT. A customer enters a store populated with cameras, microphones, RFID readers, and other sensing devices, and wants her identity privacy protected. In a typical ubicomp scenario, her cell phone automatically starts to negotiate with all sensor devices appropriate levels of privacy handling, such as not allowing customer identification unless necessary (for instance, for a purchase), or video recordings in the dressing room, and warn the user if there are violations. The basic problem is how to deal with the variability of hardware and the complexity of privacy policies in the ever-changing environments of everyday life. The traditional tool-based approach requires some form of structured dialogue between the user's phone and each sensing system, using a standard supposedly accepted and adhered to by all manufacturers. A services approach to privacy management would be based on the existence of privacy management providers. As the customer enters the store, her cell phone picks up the store ID and sends it to the privacy management provider. This triggers a high-level dialogue between the privacy provider and the store surveillance system, using standard, pre-negotiated privacy "packages" that both enterprises (possibly with human input) have previously agreed. These contracts may be negotiated one store at a time by the privacy provider, which can leverage the cost of this work through its multiple users. For the store, negotiating privacy in bulk is also a winning option. Moreover, a privacy provider may include as part of its service a guarantee that it periodically checks whether the sensing system of particular locations are actually following their promises, by human inspection or large operations of data mining, even possibly by negotiating direct access to the store databases (which would hardly be given to customers). Eventually, standards may automate part of the process, but in the meantime a practical solution is technically and economically feasible because a services approach, due to its scale, is able to handle the complexity of the real world. A similar but limited version of this approach has been proposed in [8].

DETERMINING LOCAL CONTEXT. A traveler is driving after a long day of work and decides to have a quick dinner and maybe go to a bar to listen a local band, if there is anything interesting playing that night. This scenario, often referred to in mobile and context awareness research, is normally tackled through some kind of search in the web, either by direct GPS coordinates (using *Google Local*, for instance) or by entering the name of the street/city in a search engine. Let us forget, for the moment, that these approaches are, in fact, already using service providers to collect and filter information. The problem here is how to select the restaurant and the possible night outing, since spatial information collected through search engines is often contradictory and many times outdated. A key advantage that a service provider of traveler information has is the ability to gather the opinions and complaints of all its users, compare their impressions, and possibly apply profile-based, collaborative filtering techniques [10] to consolidate the results. Computer science has struggled for many years to model individual human beings but statistics has repeatedly been shown to be a powerful tool to model multitudes of people, especially in non-critical scenarios such as this. Moreover, a services provider for local context can also utilize direct human intervention to improve the quality of its data, either by having employees responding to customer complaints such as in *Craigslist*, by direct user reports as used by *Amazon*, or by carefully analyzing and filtering the result of user surveys as in *Zagat*.

MONITORING TEMPORAL HUMAN BEHAVIOR. A 70-year old widow lives alone in a house equipped to monitor her activity and alert her daughter, who lives across the city, of potential health problems. Typically she goes every Sunday morning to church but this Sunday she decided to stay home because of the shocking news about the tsunami that is happening in the other side of the world. How an automatic system analyzing her motion habits can avoid triggering a false alarm in a situation like this? A tool approach, based on direct analysis of her motion habits through time is clearly unable to detect the impact of cognitive-level events such as the tsunami occurrence and its effect on people. How much AI and common-sense would be needed to have an event-detector like this? However, if the information about the widow is being collected by a service provider and compared across the board with other people, a pattern of staying at home watching TV is likely to be detected for that day, which could be used to increase the alarm threshold. As before, the sheer scale of a service provider able to process temporal data from a large number of users can simplify the human behavior monitoring tremendously. It could be argued here, as in other cases, that a peer-to-peer information sharing approach could also be used. However, peer-to-peer systems are really only possible when access and processing of global data is not essential and when no sensitive data is involved. Who would like to share its temporal behavior data when a possible peer is a thief, insurance agents, or the CIA? Service providers have all kinds of reasons to keep their customer data private and secure, including the trust of their customers and privacy laws.

4. Pros/Cons of Ubiquitous Services

Let us summarize some of the main advantages of the services approach illustrated by the cases and scenarios discussed so far:

- **Avoids duplication of large scale efforts:** as in the case of web search, a service approach may have a significant advantage when solving an information task that requires a large amount of effort in the case its results can be shared by a large pool of users.
- **Enables development of libraries of special cases:** to handle complex interactions between agents, such as in the privacy management case, gathering a case-by-case library detailing all the special cases is feasible under the services paradigm but hardly an option for tool-focused applications.
- **Allows the use of expert, expensive knowledge:** as illustrated in the privacy management system, when the problem can be solved by a small collection of agreements or interventions of human experts that can be shared by large number of users, a services approach may be the simplest option.
- **Provides opportunity and space for human support to users:** a service system can circumvent the limitations of automated technology and have a

fail-safe network of support agents to help users.

- **Simplifies maintenance and user support by allowing human agents:** service-based systems can significantly decrease the maintenance requirements of tools. Service providers can share the cost of human supervision and support through their many users, instead of having to establish fully automated, self-healing tools.
- **Permits the aggregation of information from multiple users:** relatively simple statistical methods can extract information from a collection of users that would require very sophisticated reasoning, sensing, and modeling to otherwise generate. For example, it allows the use of machine learning techniques to model temporal behaviors as illustrate in the elderly example. Of course, there are limitations on how much can be detected due to the typical heterogeneity of human data.
- **Creates a "trusted" partner to aggregate sensitive information:** often it is simpler to negotiate with the service provider an agreement on how to manage one's sensitive information than to do that with each member of a peer network. Also, sensitive informative does not have to be given to other users, often not even in aggregated form.
- **Expands range of applicable business models:** traditionally tools are sold, often in "shrink-wrap"-like packages. In comparison, as we learn from our discussion about the web development in the 1990s, services sometimes are even "free", i.e., supported by advertisement or paid by other actors interested in the transactions. In comparison, virtually all initiatives to sell software with advertisement seemed to have failed.
- **Fosters innovation by allowing competition:** in general it is less costly to switch to a new service provider than to acquire a new software tool as we witnessed in the rapid succession of Internet search providers in the 1990s.

Notice that many of the advantages listed above are mostly enabled by the economies of scale of a services approach. But creating a large pool of users is not easy both for research and the market place. Often it is hard to get a service-based system to take off, since it may need a large number of users and/or interactions to start exhibiting a satisfactory level of performance. This is a well-known problem in collaborative filtering and social matching (see [12]). Accordingly, we choose what we call the chicken-and-egg problem of services to head our list of <u>disadvantages</u> of the services approach:

- **Requires often fairly large systems to exhibit adequate performance:** how to attract and keep users when the initial performance is often inferior to what a large scale system can deliver? One approach to deal with this issue is to use a more labor-intensive system in the early stages of deployment, monitoring performance and manually intervening to compensate for poor performance.
- **Makes prototype and research difficult:** another consequence of the fairly common dependences on large number of users are difficulties in prototyping and testing the system.
- **Requires extra attention in privacy handling and building trust:** introducing an intermediary between the users and the information they want requires a third party that has to be trusted by the users. Building trust often demands much better handling of sensitive information than traditional tools.
- **Allows less user control:** a downside of economies of scale is that they often require the services customer to be profiled and to be served within a set of parameters. Often, tools suffer from the same problem, but in the latter case it is mostly an issue of interface design, while in the former customization and personalization are often constrained by economic and profitability considerations.
- **Brings uncertainty to the provision, quality, and cost of the service:** while a tool tends to work as far as the user retains control of it, the provision of services is often outside the control of the user. Service providers can stop providing a service, decrease its quality, refuse service, be overwhelmed by the demand, or decide to charge more. Also, service providers may adopt lock-in policies that tend to weaken the users' bargaining position.
- **Requires methodology that is not well defined:** in spite of the extensive use of service-centered application architectures since early 1990s, there is hardly a body of literature describing how to build service-based applications.

The services approach should not be confused with the notion of client-server. Although service architectures usually employ a client-server communications paradigm, the fundamental difference here is the assumption of a large number of users. In fact, this approach seems to be mostly advisable when the application domain can take advantage from the aggregation of information from multiple users and/or when there are economies of scale in information search or processing.

5. Solving Problems, Not Annoyances

How much better your life would be if you would always know when there is fresh brewed coffee in your office? Or glancing to a window and getting the weather report overlaid on the glass? Or having your computer booted, warmed up, and automatically logging you in the moment you sit at your desk in the morning? How much would the realization of all these scenarios contribute to the quality of your life?

Scenarios such as the above — from Weiser's description of life in *"… a world full of invisible gadgets."* [13, pg.74] — are abundant in the ubicomp research. Unfortunately the ubicomp research community has a long history of worrying about small annoyances of everyday life (such as the ones listed above) which, even if perfectly, transparently and seamlessly, solved would be very unlikely to have a significant impact on people's perception of their quality of life. Perhaps the fascination with such scenarios is their usual presence in science-fiction literature and films. But in Sci-Fi, these trivial indexes of future can be regarded as literary and visual artifacts that assure the readers and viewers that the narrative is happening in a different point in time. They work so well to characterize a future time precisely because they are not an essential part of life, but rather intriguing details that owe their presence to a world that has changed radically. They are a beacon of the future, but solving annoyances often does not lead to substantial improvements in quality of life.

However, solving annoyances is often fairly easier than trying to solve some of the really important issues of contemporary life. Now that research in ubicomp is getting close to two decades, we feel that it is time for it to mature and find a good reason why the rest of the world should care about it. How can an invisible, transparently controlled, embedded network of computers can actually improve the life of ordinary human beings?

In the following pages we discuss our vision of a handful of ubiquitous computing systems that people would probably care about. These are systems that we believe the service-based architecture is the right and sometimes the only feasible approach. But first, let us be clear that the service-centered development is not a panacea for all scenarios of ubiquitous computing. What we argue here is that there are solutions for some important problems in which the service-centered approach seems to be much more likely to achieve actual results than a tool approach. In other words, we believe there are real, important problems that can be feasibly solved — technologically and economically — by service systems that aggregate information gathered by ubiquitous sensors from a large number of users, and delivered by embedded devices.

DECREASING URBAN CONGESTION. There has been much discussion about embedding chips in roads to detect traffic flow and feed back this information to drivers in real-time so they can take alternative routes and increase the capacity of the road system [1]. Although there are concerns about this strategy, these proposals tend to put governments in charge of the effort, since it involves a lot of work in the physical infrastructure of streets and highways. Instead, imagine a service provider that fits cars with GPS sensors which send positional data, via the SMS infra-structure or similar, to a service provider. By statistically matching the position of vehicles through time to the road grid, it is possible to create a fairly accurate model of where congestion points are. Data about particular vehicles need not to be kept beyond some minutes after arriving to the service provider, removing some possible privacy concerns. In the driver's dashboard, a navigation aid similar to the ones used today would query the service provider, every minute or so, about the best route between the current position and the destination, using the SMS network or other cell phone-based communication infrastructure. Starting this service is possible even with a limited number of initial subscribers if their spatial distribution is carefully controlled and the system is also informed by traffic data collected by human operators. It is also possible to obtain critical mass by initially offering the service free for taxis and buses which can provide a much higher volume of information as they stay longer on the road. This kind of service could be offered for a fee or even provided by a smart, forward-thinking local transit authority as a way to decrease future investments in road infrastructure. As noted in [1], transportation policy is moving beyond building new roads and mass transit systems to encompass informational means to manage and control traffic flow.

FIGHTING FLU EPIDEMICS. One of the best ways to fight a flu epidemic is to decrease the possibilities of contagion. Consider a system that could warn individuals of areas where there is a high level of people suffering from flu, or organizations such as schools that an epidemic is happening in their spaces. People could avoid such areas altogether or prophylactic measures such as washing their hands could be advised or strictly enforced. A ubiquitous computing approach, suggested by our colleague Maria Ebling, could use cell phones to process the sound of their owners and detect likelihood of flu symptoms such as coughing, nasal congestion, and hoarse throat. This information, together with the approximate position of the phones, could be sent to an epidemic monitor and control provider that would aggregate the data and create maps of flu incidence. This aggregated information could be sent back to the users, either as a paid service, or laced with advertisements for cold remedies, or simply funded by the government, health care providers, and/or large scale enterprises as a way to decrease the high costs of flu. To bootstrap the system, the service could be offered for free to major players in the contagious network such as teachers of young children.

SIMPLIFYING GLOBAL SHIPPING. The bulk of global trade is performed by containers transported by ships. As noted in a recent study on global innovation opportunities, [1], *"[w]ith a mishmash of different standards, byzantine customs policies, inefficient manual processes and aging infrastructures, many of the world's ports cannot accommodate the massive influx of traffic brought by increased global trade."* [1, pg.34]. There is possibly a huge opportunity for service providers that track the global position of containers with GPS, match with the RFID tags of their contents, and electronically perform customs functions such as tariff collection and cargo inspection. A good analogy of the envisioned system is in fact a TCP/IP network, where containers are regarded as information packets and the service providers are routers. The "routing" and "protocol verification" functions, traditionally performed in containers at the nation borders by customs officials, can be possibly more efficiently performed by trusted service providers that use embedded sensors and computers to track and monitor the containers and their contents, possibly publishing the information in real-time to all involved parties. Many experts believe the appearance of such agents as essential to keep up with the rise in global trade [1]. Here, the service-centered approach is a requisite of the multinational characteristic of the system, which traditionally distrusts government officials and already relies on intermediary traders and service providers for efficiency. These service providers are likely to specialize at first in containers between two nations or even between two ports, maybe only for specific products, developing electronically-monitored procedures that satisfy trade laws from both nations, collect the appropriate tax. Eventually, the volume of transactions pays off the initial human and resources investment to develop and negotiate the container trade procedures.

IMPROVING DISABLED AND ELDERLY CARE. As described before, aggregating temporal tracking information of groups of people may considerably simplify the understanding of their behavior. We see the opportunity for service providers to offer services for families that need to remotely monitor disabled or elderly people. Movement sensors, either wearable or installed in their homes, could gather information about patterns of motion and issue warnings to the people being monitored and their overseers. Even with state-of-art behavior and action recognition technology, it is very hard to account for contextual impacts on people's activity. As discussed before, a possible way to handle this complexity and avoid a system too prone to false alarms or too insensitive is to comparatively monitor a large group of individuals. Given enough data, it is even possible to use the tracking data itself to cluster individuals with similar habits. There is a clear market for service providers to install and maintain the sensors, collect the data, aggregate the tracking information, and communicate alerts to its customers. The main challenge here is getting enough subjects and tracking information to start providing the service reliably. Approaches such as targeting specific segments of the population that are likely to exhibit similar behaviors may help.

EMPOWERING CITIZENS BY FACILITATING CONSCIOUS SHOPPING. In a world where increasingly large enterprises have a disproportional share of power, one of the most effective ways for political action is having people shaping their consumption habits according to their social and political beliefs. The last decades saw the emergence of such *conscious shopping* movements such as the one used by organizations trying to stop animal testing and child labor, the "buy American" campaign, or the "fair trade coffee" initiative. However, in the globalized world is increasingly difficult to determine which products a particular cause or idea recommends or boycotts. For example, for a supporter of organizations against unjust labor practices such as child labor to recognize which products are actually being targeted is often difficult. In a not-so-distant future where many products will have an incorporated RFID tag and phones will have RFID readers, a political action group can offer a service where scanned products are checked against the organization's database and the consumer is alerted about approved or objectionable behavior associated with a certain product or manufacturer. A similar service could be offered today by simply sending a picture of the barcode taken by a web- and camera-enabled cell phone. Consumers could subscribe to conscious shopping services as easily as registering for a RSS feed and even be able to set up degrees of supporting/boycotting beyond the dichotomy of the yes/never approaches used today. Also, the services could consider not only individual products but all the items of a shopping cart which could be, for instance, balanced by the total carbon footprint of the products in it.

6. Final Discussion

Using a service approach to design and deploy systems is not a new idea, even in the context of ubiquitous computing (for example, [5, 6, 8]). What we try to accomplish in this paper is to bring to the foreground the assumptions behind using a services approach for an application and in particular, for an ubicomp application. By doing so, we are able to discuss when the approach is advisable and what kind of new, real-world applications seem to be feasible.

Simply put, computer systems tend to be more transparent and ubiquitous when they are managed by other people. The service approach is not at odds with the original vision of ubiquitous computing by Weiser [13] or his contemporaries, but it is remarkable how much service approaches were and are still not acknowledged as an alternative, distinct option for an application architecture. In particular, that leads to a failure in understanding the potential economies of scale of services approaches, not only in monetary terms but also in terms of improved data gathering, processing, organizing, and management.

It is also clear from our analysis that a services approach to ubiquitous computing is not a slum-dunk option. First, there are disadvantages such as often requiring large amounts of data to exhibit performance; making prototype and testing difficult; needing privacy and trust handling and management; and the lack of a set of known, documented, and tested design, evaluation, and deployment methodologies.

However, we do see the services approach as a way out of the current status of ubiquitous computing of both partially solving unimportant problems and annoyances, and also hand-waving solutions for important problems such as elderly care and privacy management. We believe the five service-centered solutions we outlined for problems in urban congestion, preventive health care, global trade, disabled and elderly care, and political participation demonstrate the potential of this approach to deliver ubicomp-based solutions to the real-world in a reasonable timeframe.

To make computers disappear it is not necessary to make them small, embedded, automatic, or autonomous. The key is to focus on the user experience, to, as proposed by Weiser, *"… make using a computer as refreshing as taking a walk in the woods."* [13, pg. 75]. Computers also "disappear" when they are installed, controlled, and managed by another people, system, or network-based service which trades complexity with economies of scale. Ubiquitous computing, as well as many other types of computer applications, does not need to be restricted to fully automated computing systems: complex but well-designed service systems involving machines and human beings, supporting a large number of users, are often the best option to solve real world problems.

References

[1] *"Global Innovation Outlook 2.0,"* IBM 2006.

[2] G. D. Abowd and E. Mynatt, "Charting Past, Present, and Future Research in Ubiquitous Computing," *ACM Transactions on Computer-Human Interaction*, vol. 7 (1), pp. 29-58, 2007.

[3] V. G. Cerf, "Networks," *Scientific American*, vol. 265 (3), pp. 42-51, 1991.

[4] M. L. Dertouzos, "Communications, Computers and Networks," *Scientific American*, vol. 265 (3), pp. 30-37, 1991.

[5] E. Eagle and A. Pentland, "Social Serendipity: Mobilizing Social Software," *IEEE Pervasive Computing, Special Issue: The Smart Phone*, vol. 4 (2), pp. 28-34, 2005.

[6] I. Haritaoglu, "InfoScope: Link from Real World to Digital Information Space," in *Proc. of Ubiquitous Computing 2001 (Ubicomp'01)*, Atlanta, Georgia, 2001.

[7] R. E. Kahn and V. G. Cerf, *The Digital Library Project, volume I: The World of Knowbots*, Reston, Virginia, 1988.

[8] E. Kiciman, "Using Internet Services to Manage Massive Evolving Information for Ubiquitous Computing Systems," in *Proc. of the UbiComp'02 Doctoral Consortium*, Goterborg, Sweden, 2002.

[9] P. Maes, "Agents that Reduce Work and Information Overload," *Communications of the ACM*, vol. 37 (7), pp. 31-40, 1995.

[10] U. Shardanand and P. Maes, "Social Information Filtering: Algorithms for Automating Word of Mouth," in *Proc. of CHI'95*, Denver, Colorado, 1995.

[11] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 3rd edition, Addison-Wesley, 1998.

[12] L. Terveen and D. W. McDonald, "Social Matching: A Framework and Research Agenda," *ACM Transactions on Computer-Human Interaction*, vol. 12 (3), pp. 401-434, 2005.

[13] M. Weiser, "The Computer for the Twenty-First Century," *Scientific American*, vol. 265 (3), pp. 66-75, 1991.