# What next, Ubicomp?
# Celebrating an intellectual disappearing act

**Gregory D. Abowd**
School of Interactive Computing & GVU Center
Georgia Institute of Technology
abowd@gatech.edu

## ABSTRACT
Weiser's landmark *Scientific American* article inspired many researchers to explore an exciting socio-technical vision of a third generation of computing. At the 21st anniversary of that published vision, I want to assess ubicomp's maturity and explore the identity challenge it faces. Today, ubicomp as a niche research topic no longer makes sense; we must celebrate its "disappearance" as a well-scoped research agenda because it has become a profound agenda across most of computing, and beyond. This should not be surprising; the $2^{nd}$ generation of computing, the personal computer revolution, experienced the same profound disappearance. In celebration of this imminent disappearance, I will highlight the unique contributions of the ubicomp community, express some remaining intellectual challenges, and speculate on how to formulate new visions of computing that might succeed this third generation.

## Author Keywords
Ubicomp as a field

## ACM Classification Keywords
K.2 History of Computing; H.4.0 Information Systems Applications—General; D.2.6 Programming Environments

## General Terms
Design, Experimentation, Human Factors, Languages

## INTRODUCTION
When Mark Weiser wrote his seminal article defining the concept of ubiquitous computing, he did so in a world that had embraced the personal computing revolution. Two decades later, the world has embraced many of the notions of ubicomp, and it is time to reflect on that reality and decide where to go next. The discipline of ubiquitous/pervasive computing has spread so widely throughout the computing universe—the research and practice of computing—that it should disappear as a niche topic in computing. My thesis is that ubiquitous computing, the third generation of computing, is here and no longer requires special attention, as its ideas and challenges spread throughout most of computing thought today. The struggle to maintain an identity for ubicomp is an intellectual distraction, albeit one that serves a social function that is hard to abandon.

There is a difference between the intellectual area of ubiquitous computing and the community of people who identify themselves as ubicomp researchers. The former is what I believe to be broadening to the point of disappearance. The latter still remains, and is not entirely defined by an intellectual agenda. The ubicomp community will likely remain for many years beyond the publication of this simple commentary, but I offer some constructive ways for it to best operate.

In this paper, I will further explain why ubicomp has and should disappear, based on an observation that its ideas already pervade much of computing research and practice. It is increasingly hard to identify what constitutes ubicomp research today, because it is hard to rule anything out as being unrelated to this current generation of computing. I will explore what has made ubicomp research valuable in the past and distinguishable from other research communities. I will also clarify the relationship between application domains and ubicomp research. Through a comparison to the personal computing generation, I will frame a remaining research challenge to further simplify the development of ubicomp applications.

If ubicomp as the third generation has arrived, what characterizes the next generation of computing? This is an interesting challenge to ponder. Visions of computing are difficult to offer up, as they are far more likely to be wrong than right. However, if we consider visions as ideas whose time has come, we can actually revisit ideas from the past and ask whether today's computing climate will provide an opportunity for those ideas to be realized in new and compelling ways.

Before I proceed with the arguments of this paper, I offer a few notes to the reader to best interpret and respond. Though many of my comments are relevant to other disciplines and communities, both related and unrelated to computing, I am not explicitly attempting to make that more general argument. I am not approaching this paper in the broadest historical context that might make sense to


*UbiComp' 12*, Sep 5 – Sep 8, 2012, Pittsburgh, USA.

some readers. I do not feel qualified to make that generalized argument with academic rigor.

I am not a social scientist, and I have arguably tread into that domain by making bold statements about how a community of practice has and should evolve. It will also be evident to the reader that the majority of my examples and references are of technical contributions. I do not apologize for these limitations; I only provide them as an explicit warning to the reader, and a potential call for those with different perspectives to augment or refute my claims.

I am writing this essay as a researcher who has committed the majority of his professional life to a specialized community. The opinions expressed here are based on nearly 20 years of experience helping to create a community and style of research. This paper does not reflect a change in what I think is important to do as a researcher. It is attempting, however, to get us all to think more deeply about how to continue as a community of research practice. While I can see a logical conclusion of this paper is that the ubicomp-themed conferences need to fade into the sunset, an alternative is that our mission and identity be revisited and solidified.

## THE DISAPPEARANCE OF UBIQUITOUS COMPUTING

Of all of the work and writings of Mark Weiser, most of the attention and scrutiny has been placed on his September 1991 *Scientific American* article [45] (also available as [47]). That work was indeed inspiring and should rank as one of the most influential computing papers of the 1990's. Following Weiser's lead, others have come to explain ubiquitous computing as a natural progression of computing, from the first generation of the mainframe to a second generation of the personal computer to the third generation of ubiquitous computing (see, for example, Want's historical account in Krumm's edited book [44]). Whenever I have trouble describing what ubiquitous computing is, I fall back on this definition that it is the third generation of computing. That definition, however, brings with it a harsh reality. As with any generation, its time comes and then it goes, to be followed by the next generation.

But it is too simple to say that ubicomp's time is up simply because time has passed. A more convincing argument is found in the (revised) words of Weiser himself.

*The most profound research topics are those that disappear. They weave themselves into the fabric of everyday research until they are indistinguishable from it.*

Weiser's article in the *Communciations of the ACM* in 1993 inspires my argument that the intellectual agenda of ubiquitous computing has become so profound that it is increasingly indistinguishable from the overall agenda of computing research today [46]. Weiser presented technical and human-centered challenges for hardware design, wireless networking, and interaction design. In the 1990's and early 2000's, others followed suit, demonstrating how the vision of ubicomp and mobile computing was relevant to a variety of research fields, including operating systems [41], software engineering [1], mobile computing [39], privacy [29], and application areas like the home [20] and sustainability [30]. Indeed, the 2002 inaugural issue of *IEEE Pervasive Computing Magazine*, in which Weiser's original article was reprinted, presented a variety of perspectives from leaders in the field that outlined a decade's worth of advances in the wide intellectual swath that Weiser's vision had inspired.

An early 2012 issue of *IEEE Pervasive Computing Magazine* provided another reflection on the state of ubicomp. Twenty years after the original vision was made public, we realize that many of Weiser's predictions have come true, most clearly the proliferation of connected devices of different scales and ownership models. Some predictions were not exactly accurate, such as inch-scale devices being readily disposable and impersonal, the smartphone being a stark counterexample.

Commentaries on Weiser's ubicomp vision have critically assessed the counterproductive directions those influenced by that vision have taken. Rogers argued that the vision presents too "hard" of a challenge of autonomous, invisible technology, as opposed to visible technology that enriches human activity through effective engagement [38]. Bell and Dourish cautioned against research motivated by a utopian "proximate future," whose goal is to shoot for an impossible future in which technological infrastructure functions seamlessly [7]. Indeed, the mundane aspects of technologies that are ubiquitous are a fascinating concept to ponder and leverage in design [42], and the apparent seams that mark the otherwise annoying aspects of an imperfect technological infrastructure can be used to advantage, even for enjoyment [10].

It is not the point of this paper to argue about the correctness of past predictions for ubicomp. The volume of predictions and reflections and the wide variety of intellectual perspectives of the authors of those writings demonstrate that Weiser's vision has succeeded in pervading the thoughts of a large community of researchers. Connected devices, at a variety of sizes and with varying models of ownership, define our world of computing today. Those who do research in computing necessarily couch their ideas in the context of this current generation of computing. It is no longer the specialized domain of Weiserian disciples.

To those who argue that ubicomp is still its own specialized domain in computing, I offer the following thought experiment. Consider any paper published in the proceedings of this version of the *International Conference on Ubiquitous Computing (Ubicomp 2012)*. Ask whether that paper could have rightly appeared in one or more other conferences or journals that do not align by name with ubiquitous or pervasive computing. I think you will find that "our" work is relevant to so many other communities.

As a follow-on experiment, consider a paper published in some other computing conference or journal in the past year, and ask whether that paper would be in or out of scope with Ubicomp. More and more of the work that appears outside of Ubicomp is relevant to what we call ubiquitous computing.

The intellectual agenda of ubiquitous computing is becoming broader and broader; as a result, attempts to maintain a well-scoped, niche research community on the topic is increasingly difficult. I would further argue that it is dangerous, because it threatens to marginalize those who remain insular. We should instead celebrate the profound impact of ubicomp and prepare to move on. Ubiquitous computing is now indistinguishable from computing itself.

*Ubicomp is dead! Long live (ubiquitous) computing!*

## HOW UBICOMP HAS DISTINGUISHED ITSELF

While the previous argument is meant to be a positive celebration of a vision gone viral, I recognize that it may be taken in a negative light. One of the reasons I believe that ubicomp as a discipline is disappearing is it is increasingly hard to identify computing research that is uniquely ubicomp. Conversely, it is hard not to see the relevance of much of today's computing research towards the goals of ubicomp. There are, however, some examples of research activities that came out of the ubicomp community that are positive influences on all of computing research. The examples serve to define what it has meant to "be ubicomp" over the years: putting computing into the real world; leveraging existing infrastructure to create new services that scale to a broad form of use; and appreciating the need for integrative, multidisciplinary research.

### Living laboratories: Hacking in the real world

What excited many of the early researchers in ubicomp was that Weiser's visions were both futuristic as well as attainable. While Bell and Dourish comment upon the dangers of the "proximate future," that is, the vision of a tomorrow that is just around the corner and an improvement on what we have today, it is exactly this proximate feature that was so compelling [7]. By the mid 1990's, commodity hardware for mobile and handheld computing, and emerging standards for streaming multimedia, outdoor location and information exchange was making it affordable to explore applications outside of the laboratory. Experiments involving research deployments for off-the-desktop applications were gaining in popularity, and the best examples of this first appeared in the ubicomp and wearable domains (see, for example, projects like the Remembrance Agent [36], MediaCup [22], GUIDE [13], and Classroom 2000 [2]).

This do-it-yourself (DIY) mentality is prevalent in ubicomp and is fueled by the larger DIY, or "maker culture" that has emerged as the hardware and software tools have matured over the past few years. One of the lasting impacts of ubicomp research is the renewed interest in training computer scientists to have both software and hardware development skills.

### Your noise is my signal: Practical deployment of new services

At the heart of ubicomp is making computational services more available, literally making things more ubiquitous. Recognizing the practical, large-scale deployment challenges to new capabilities coming out of the research community, mobile and ubiquitous computing researchers have shown how to take advantage of existing capabilities and commandeer them for new purposes.

One of the first examples of this is in developing indoor location technologies. Outdoor location became widely available in the mid 1990's through affordable GPS receivers that could connect to, and ultimately were integrated inside of, commercial mobile platforms. GPS was never a good solution for indoor location, and there was no simple alternative. While some compelling applications of indoor location showed up in the late 1990's (e.g., Cyberguide [3]), they were mostly demonstrations without any practical hope for large-scale deployment. Bahl and Padmanabhan's RADAR system was the first example of a promising approach to indoor location that made practical sense [6]. RADAR took advantage of existing communications infrastructure, WiFi access points, and used signal strength measurements and a "fingerprinting" pattern recognition approach to produce location information. A frenzy of war-driving efforts followed to create publicly-accessible datasets of WiFi access points and their locations.

Beyond the important contribution of practical indoor location, the research approach introduced by RADAR revealed the idea that we could use already deployed infrastructure to create services that the infrastructure was not designed to provide. With location, this idea was quickly replicated with cellular telephone infrastructure [32,43], FM radio signals [28,50], and the domestic powerline [33,40]. "Your noise is my signal." is a hallmark of ubicomp research over the last decade that makes valuable computing or sensing services available to a broader audience. Dietz, Yerazunis and Leigh exploited the characteristic of the LED as a photodiode to show how this common and ubiquitous display element could be exploited as an input device [18]. Patel, Reynolds and colleagues demonstrated a variety of examples in which variations of physical properties in the infrastructure of a home (powerlines, plumbing, gas, HVAC) could be sensed in one place and reveal information about human-initiated activities around the home [11,21,34,35]. This is now a common and respected research approach for developing new and practical sensing solutions.

### We need each other to advance: Being truly multidisciplinary

The ubicomp community is one of the most eclectic research communities in computing. The conferences attract results from very different research communities.

Where else do we bring together hardware, sensing, activity recognition, systems, middleware, industrial design and applications researchers with social science and privacy researchers? Some of the most innovative work our community has seen resulted from initiatives that explicitly brought these communities together, such as the European Disappearing Computer Initiative and the UK's EQUATOR project.

Part of the reason ubicomp encourages multidisciplinary research is because it appeals to doing things in the real world. Any applied research in computing requires the integration of a variety of skills and perspectives for proper investigation. But multidisciplinary work does not mean that every disciplinary perspective is rewarded with results that break new ground in that area. A good example of that balance can be seen in research that applies pattern recognition and machine learning. Albinali, Goodwin and Intille demonstrated how on-body sensors can be used in conjunction with mobile phones to train a system to recognize the highly idiosyncratic stereotypical motor behaviors of a child with autism [5]. This is a perfect example of the kind of work that the ubicomp community highly regards, for it is focused on a real problem and exploits results in a variety of technical areas (mobile phone interaction, wearable sensing, and activity recognition). Ubicomp provided a home for this kind of technically driven applications research.

Solving real problems does require multiple disciplinary perspectives, but does not always require breaking new ground in those disciplines. It takes a special kind of researcher to accept the compromises of doing multidisciplinary research, and often those researchers must do double duty to both impress the ubicomp community as well as retain their credentials within their own, more focused research community. Has this requirement for doing double duty as a researcher helped or hindered the ubicomp community?

As the hardware and software tools mature, more applications researchers will be empowered to use the technologies of ubicomp, perhaps even without the aid of collaboration with researchers who specialize in those technologies. This is again a cause for celebration for ubicomp, but it will be signaled by a disappearance of this kind of research at ubicomp conferences. We are already seeing this happen with applications on mobile phones. The ubiquity of mobile phones makes it the perfect platform for large-scale deployment studies in a variety of domains. But services on a mobile phone, such as texting, are so reliable and straightforward to use that the ubicomp community no longer values their use alone as a research contribution, and that is a perfectly valid stance.

These comments on the evolution of our multidisciplinarity in the ubicomp community are reflective of a constant tension faced by applications research in computing, a topic I address next.

## APPLICATIONS RESEARCH AND UBICOMP

*"Applications are of course the whole point of ubiquitous computing."* [46]

This mantra from Weiser is a two-edged sword. While there is little reason to develop a computing technology unless it has some hope of being applied to a problem people care about, we sometimes over-emphasize this point in assessing the contributions of different ubicomp research activities. There is a continuum in the balance between technology development and application.

A technology advance that improves availability or accessibility to a core technical capability (e.g., location) is of value regardless of the application of that capability. This kind of *application agnostic* research features prominently in our research literature. Previous examples already discussed in this paper include turning the LED into an input sensor, and infrastructure mediated sensing. There are also fine examples of this approach in research fields that highlight interaction technologies (e.g., UIST and CHI) over the past two decades, such as DiamondTouch [17], sensing on mobile devices [25], and Skinput [24]. Much of the research in systems, networking and software engineering pride themselves in being broadly applicable across a wide variety of application domains. An effective litmus test for this kind of research, in fact, is the demonstration of relevance across many different application domains. And as the examples above show, the results, while of great interest to the ubicomp intellectual agenda, appear in a variety of forums.

More human-centered researchers often favor an approach that begins with a specific application domain. Even technology-centered researchers sometimes explore how a particular technology, with all of its advantages and imperfections, can best address the challenges in a well-characterized domain. This *applications motivated* research has found a comfortable home in the ubicomp community because we appreciate the balance between understanding the needs of a given problem domain, highlighting the opportunities to make a difference through some technological intervention, and doing initial, exploratory validation to reveal the gap between technology introduction and making a difference in the application domain.

Our community has explored a variety of domains in this way, including education [9], health [12], religion [49], and sustainability [48]. A potential danger of applications motivated research appears when it serves as a façade to mask what really is applications agnostic research. We are all familiar with research that begins with a single motivating application scenario that quickly evolves into a technical exploration that shows very little appreciation for the motivating scenario. Skeptics scoff at this kind of technology-centered work as "a hammer in search of a nail," or similar dismissive remarks. The litmus test for applications motivated research is whether domain experts believe that the infusion of ubicomp will make a meaningful difference, even if that difference has not been established with ultimate rigor.

The final category of ubicomp applications research, *applications driven*, is the Holy Grail, an introduction of technology into a problem domain that makes a research contribution to that domain itself. We are beginning to see this happening in education with work like the Open University's TU100 course on My Digital Life [37]. Sustainability and health are poised for similar kinds of contributions from ubicomp researchers. But even as I describe this level of applications research as the highest achievement for the ubicomp community, it comes at a cost. These results will be published *outside* of our community and, unlike the example of My Digital Life, may not entail a specific reference to ubicomp. SMS texting is beginning to show impact in chronic disease management [51], but if these preliminary results bear out in larger clinical studies, we will see the results in medical journals. Who will know, and who will care, that the ubicomp research community played a role in these important domain contributions? We should!

In summary, ubicomp research can be applications agnostic. When motivated by a specific application domain, the connection to the domain should be authentic. A promising connection between ubicomp and a specific domain will then "disappear" into the domain, but we should not lose sight of that contribution.

## SO, WHAT'S NEXT?

I have mainly critiqued where we are today as a research community, and the only concrete suggestion about where we go next is to disappear into the larger computing research agenda, or into the research literature of other domains, and cease to be a niche topic. However, the analogy to the personal computing generation reveals that there is more work to be done. I will now frame a research challenge that will help to establish the maturity of the ubicomp generation of computing.

### Unfinished business: Simplifying development of ubicomp applications

It is still too difficult to build ubicomp applications and experiences. This difficulty limits progress, particularly for applications driven research that should be controlled by non-ubicomp researchers and designers. By contrast, since the mid 1990's it has not been difficult for non-experts to build complex applications for personal computers. What is the difference? Let's look back at the history of the personal computing generation, with a specific focus on programmability.

#### *From prototypes to tools to designers: The PC story*

The vision of personal computing started in the mid-late 1960's, the brainchild of a few talented leaders—Ivan Sutherland, Douglas Engelbart, and Alan Kay. Those early visions were made concrete by novel demonstration systems—SketchPad, NLS/Augment and the Xerox Alto—developed by very gifted researchers. Englebart's work introduced the notion of producing not only a working demonstration, but also a toolkit so that others might explore. Kay's group introduced the desktop metaphor as a simpler way to engage humans in the kinds of information processing tasks the personal computer could support. While that metaphor was very appealing, it was not easy to create applications for the personal computer in the 1970's.

The 1980's was marked by both the emergence of standard personal computing platforms, the IBM PC and the Apple Macintosh. Killer applications—the first being the electronic spreadsheet—provided the motivation for investing in personal computers (companies first buying them for each employee at work and then individuals purchasing them for their own use at home). Once enough people owned personal computers, there was further motivation to continue to develop more killer apps (word processors, database management systems, and later email clients and WWW browsers). Having a base of personal computers created a market for software applications, which further motivated important software developments aimed at simplifying the engineering of personal computing applications (see Myers et al.'s historical account [31]). Architectural models that separated application semantics from presentation details, and event-based programming toolkits with reusable interaction widgets made it easier for software engineers to build increasingly complex and usable software. In the mid-1980', Apple released HyperCard. Wikipedia aptly describes the impact of HyperCard:

*"Many people who thought they would never be able to program a computer started using HyperCard for all sorts of automation and prototyping tasks, a surprise even to its creator."*

A variety of programming and authoring tools followed in the successful footsteps of HyperCard, such as Macromedia Director, Visual Basic and Processing. By the mid-1990's, 30 years after the original visions, programming the personal computer was no longer the privilege of the software engineers or computing savvy. Development opened up to a much more creative population of designers who could take the user experience in more imaginative directions. Personal computing, and the development of applications for the personal computing platform, had become mainstream and simple.

#### *From prototypes to tools to designers: The ubicomp story*

The story for ubicomp is strikingly similar, to a point. The vision of ubicomp started in the late 1980's, the brainchild of a few talented leaders—Mark Weiser, Andy Hopper, and Ken Sakamura. Those early visions were made concrete by novel demonstration systems—PARCTab, mPad, Liveboard, ActiveBadge, and TRON—developed by very gifted researchers. Within the PARCTab project, Bill Schilit introduced a rule-based tool for colleagues to author their own context-aware rules that would offer services based on the location of the PARCTab device. While this vision was very appealing, it was not easy to create applications for ubicomp devices in the 1990's.

By the late 1990's and early 2000's, we started to see the emergence of standard "inch-scale" ubicomp platforms, most notably PDAs and mobile phones. While ubicomp researchers in the late 1990's and 2000's sought the elusive "killer app" that would spur investment in the infrastructure, industry found it—human–human communication either by voice or texting via the mobile phone. Individuals have purchased billions of mobile phones, and many companies provide the data and communications infrastructure that supports this first and dominant ubicomp platform. Location-aware applications are another class of killer apps that have benefitted from the proliferation of mobile phones.

Building on the work of Schilit in context-aware computing, architectural models offered important separation of concerns between physical sensing and the use of the interpreted contextual information by applications. One of the earliest and most influential examples was Dey and colleagues' Context Toolkit [14]. On the "inch scale," Fitchett and Greenberg introduced Phidgets, a collection of sensors and actuators that could be programmed with Visual Basic [23]. Arduino improved upon the Phidgets idea and has become a dominant platform for sensor/actuator embedded applications development. When Apple introduced the iPhone, and more importantly the App Store, there was an explosion of development of smartphone apps. The Android platform and its Market (now Google Play) also fed this explosion. On the "foot-scale," the introduction of the tablet/pad has gained traction, spurred by the desire to consume rather than produce media. On the "yard-scale," we have seen adoption of electronic whiteboards in schools, particularly elementary schools, interactive tabletops (e.g., DiamondTouch and Microsoft Surface) and programmable billboards delivering increased advertising revenue through programmable billboards.

By 2012, nearly 25 years after the original ubicomp visions, development in the ubicomp space is certainly much simpler than it was for the founding visionaries, but it has yet to take the leap that personal computing did. Ubicomp development is still mostly in the hands of the software engineers and tech-savvy individuals. If the ubicomp generation is to mirror the PC generation, then development has to be opened up for the creative designers. The challenge remains; what is ubicomp's HyperCard? To answer this question, we must first understand a fundamental difference between programming for the PC and programming for ubicomp.

### *From <u>programming</u> environments to programming <u>environments</u>*

The user experience of the personal computer is a 2-dimensional graphical user interface. The development environment in which creative designers produce those user experiences is also a 2-dimensional graphical user interface. The *programming* environments of the personal computing generation consist of interactive development environments that exactly match the characteristics of the end user experience.

In contrast, the user experience for ubicomp is the 3-dimensional physical world. However, the dominant development environment remains the 2-dimensional graphical user interface. A creative designer in the ubicomp space should be programming *environments* using tools that exactly match the characteristics of the end user experience. Until that happens, these designers will always be faced with the challenge of imagining how the experiences created in the development environment will play out in the actual execution environment.

There has been progress towards bridging this design-time/run-time environment gap. Mobile phone development has flourished, partly because we can run phone simulators on the development platforms. However, the majority of the mobile phone development experience is still being limited to its 2-dimensional GUI characteristics, a subset of its full capabilities. Mobile phone development environments do not make it easy to program the capabilities of embedded sensors (e.g., the accelerometers, GPS sensors, and cameras), important elements of the run-time experience on a mobile phone. It is not even easy to simulate the "seamful" experience of spotty network connectivity that Chalmers and colleagues recommend be exploited rather than wished away [10].

Augmented reality is an emerging capability on mobile phones, but there are limited tools that allow the authoring of augmented reality experiences. The gaming industry has developed sophisticated digital-world engines that provide programmable models of imagined spaces, but these are purposely placed in parallel to the physical world, not in alignment with the world in the way that ubicomp would desire. 3-dimensional programming environments, such as Google's SketchUp, make it easier to develop models of the physical world, but still leave a gap between that model and the actual world. Some very promising work that links the physical world and a 3D model has emerged through clever interactive techniques with the Kinect sensor, demonstrating that at least the construction of a 3D model can be done within that space [19,27].

Commodity sensor/actuator programming platforms, most notably Arduino and its variants, have dramatically lowered the barrier to entry for embedded applications development. Microsoft Research's amazing efforts with .NET Gadgeteer allows a programmer to model and "print" an entire custom integrated circuit and housing [26], but that is not the same thing as programming that device to behave by manipulating it in the physical world. For many years, programming as an educational activity has spurred simpler models of development, and efforts like LEGO Mindstorms and robotic platforms like PLEO begin to bring the development environment at least partially out from the 2D GUI.

Probably most close in spirit to the idea of programming environments, is the merger between end user programming, a long-explored topic in the 2D GUI development world, and context aware computing. Dey and colleagues began to explore that combination with efforts like iCAP and a CAPpella [15,16]. Mysteriously, this trend has not continued. Perhaps with more commodity sensing hardware in the gaming world (e.g., the Nintendo Wii and the Microsoft Kinect sensor), the time has come to revisit end user programming for ubicomp.

There are threads of research that begin to go in the direction that will better match the ubicomp development environment to its user experience environment. But until normal designers have affordable and simple tools that approach the capabilities that James Cameron had at his disposal in the development of the 3D experience of the movie Avatar, we cannot declare that ubicomp development is mainstream.[1] The bar has been set.

**A fourth generation of computing?**

If you accept that ubicomp as a niche research field has disappeared, then you probably have already been wondering about what the next generation of computing will be and how it will inspire future generations of researchers. I am afraid my thoughts on that topic are not as clear cut. The visions of the personal computer and ubicomp were fuelled by masterful prototype demonstrations; I have no such prototypes to offer. Nor do I have prototypical "day in the life" scenarios of an everyday figure, similar to Weiser's Sal, to inspire the reader's imagination. Instead, I offer an alternative way for us to contemplate computing's fourth generation.

There are common themes that span across the dreams of Vannevar Bush, J.C. Licklider and the many others who have written about how we can unlock the potential of computing to better serve our human experience. Whenever we are inspired by those visions, we act upon them in the context of what is available to us at the time. When Weiser authored his vision of ubicomp, it was possible to think of a world in which we could build computational devices of different sizes, connected to each other and to the physical world. That future is very much a reality today, despite Bell and Dourish's correct observation that the underlying infrastructure is and always will be a messy socio-technical system.

What else is available to us today that can change our expectations for what we can build in the coming years? We can assume a world of near-instantaneous answers to just about every question we can ponder. While the instant answers are not always the best ones, they are increasingly sufficient. Weiser's Sal went through her busy life with easy access to experiences that she had lived through. Today, Sal, can benefit from everyone's collective experiences, even those that are going on simultaneously. Crowdsourcing, inspired by von Ahn's notion of human computation [4], can answer complex questions in real-time [8]. Sal can gain instantaneous advice for an adolescent child with developmental disabilities who needs explicit advice on how to respond to an unfamiliar social situation. Rogers rightly cautioned against a ubicomp vision of perfectly functioning autonomous computing services [38]. Her vision of "soft ubicomp," in which humans are engaged through technology to support their life's activities, is eminently attainable through the power of crowdsourcing.

Hidden in Weiser's vision of a world with different-sized devices with differing levels of permanence and ownership, is the assumption (hope?) that these devices would interoperate. One of the reasons the smartphone has become even more intimate a device than the PC, a reality that Weiser did not envision, is because it is simply difficult to take all of the information that resides on a single device and instantly transfer it to another device. We cling to our smartphones because they are too closely tied to the information on them. But that is changing as well with cloud computing, the commoditization of reliable storage and computing cycles that reside outside the devices we use to summon their services. Sal's family today can share a family digital assistant, a single device that immediately adapts to being "owned" by whoever is holding it while simultaneously remembering all of the people in the family and how they and their information relates to the current holder.

Why be limited to thinking about computing as being delivered through single devices? Computing, and the instruments that deliver its services do not have to be intimately connected. When someone calls Sal for a conversation, she can pick up a handset by the bedside, converse through a laptop in the office, or speak through a watch that delivers music and coaching advice while exercising. Computing cycles and storage are effectively being divorced from our interaction devices. Input and output will certainly follow suit and be freed from each other.

I am motivated to make at least one bold prediction for the fourth generation of computing as a natural progression from the previous three generations. The first generation provided one computer to many individuals. The second generation suggested one computer per individual. The third generation promised, and delivered, many computers per individual. All of these generations suggest a division between the computing device and the individual. The fourth generation need no longer abide by that division. The human–computer experience will be more conjoined than ever before. Weiser wrote about the opportunities and

[1] See Karen Holtzblatt's CHI 2010 Lifetime Achievement award talk, excerpted at http://www.youtube.com/watch?v=5CUnePCP1Uo) for more on Avatar and using the right tools at the right time.

challenges of not being able to distinguish computing artifacts from other physical artifacts in our environment. The coming decades will see us being inspired and challenged as our own physical being and our sense of identity is no longer easily distinguished from elements of computing. Twenty years ago, these ideas were part of the wearable/cyborg movement. Today, this vision may have a greater chance of gaining traction.

## CONCLUSIONS

The past 21 years have been exciting for ubicomp. We have gone from the inspiring visions of a few pioneers, through a period of formative and foundational research findings. The success of ubicomp as the third generation of computing cannot be denied, and we should celebrate past achievements and the inevitable near-future advances that will finally put the creative controls into the hands of designers and domain experts. Ubicomp in the 2010's will be mainstream and mundane, just as the PC was by the 1990's.

I claim that ubicomp is no longer a niche research topic, but is best seen as the intellectual domain of all of computing. As such, it is difficult to see what the role is for the various formal ubicomp communities that exist today. From 1991-1999, researchers evangelized the opportunities of ubicomp to their separate research communities. Between 1999-2005, ubicomp crystallized into a small set of venues that best portrayed the intellectual agenda of ubicomp. I trumpeted the unique contributions of that agenda: hacking in the real world; practical deployment of new services; and being truly multidisciplinary. Since 2005, the intellectual agenda has been subsumed into many different communities, both new and old, to the point that it is impossible to know where to look for the best research results that impact the intellectual agenda. This spread cannot be reversed, and it results in the disappearance of ubicomp's intellectual agenda as it seeps into almost all aspects of the computing intellectual agenda.

It may be hard for longstanding communities to disappear the way I have suggested the intellectual agenda of ubicomp has disappeared. But those communities can adapt their practices to better support the intellectual goals. I implore the ubicomp community to not only curate intellectual content that it solicits for its annual meetings, but to also expend the necessary effort to seek out and advertise the results from communities that do not currently see the ubicomp community as their intellectual home. We should also seek ways to better document how our accomplishments have influenced other research communities, computing and otherwise.

Finally, as we contemplate the fourth generation of computing, we should do so with a keen understanding of how our computing climate today is different than it has ever been. Those differences allow us to revisit the dreams of the past and realize them in creatively productive new ways.

I hope I have given you reason to reflect on where ubicomp is as a uniquely identifiable and sustainable research agenda. I also hope that you will respond to the challenges being presented.

## ACKNOWLEDGMENTS

The opinions expressed in this document are mine, but they have been greatly influenced over the years by the many students and colleagues with whom I have collaborated. I thank all of you who have pointed out the inaccuracies in previous drafts. I particularly thank those of you who will read this paper after it is published and take the time to respond in constructive ways to help shape the future of our research community.

## REFERENCES

1. Abowd, G.D. Software engineering issues for ubiquitous computing. *ICSE 1999: Proceedings of the 1999 International Conference on Software Engineering*, (1999), 75–84.
2. Abowd, G.D. Classroom 2000: an experiment with the instrumentation of a living educational environment. *IBM Systems Journal 38*, 4 (1999), 508–530.
3. Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., and Pinkerton, M. Cyberguide: A mobile context-aware tour guide. *Wireless Networks 3*, 5 (1997), 433.
4. Ahn, von, L. *Human computation*. Carnegie Mellon University, 2005.
5. Albinali, F., Goodwin, M.S., and Intille, S.S. Recognizing stereotypical motor movements in the laboratory and classroom: a case study with children on the autism spectrum. *Ubicomp '09: Proceedings of the 11th international conference on Ubiquitous computing*, ACM Request Permissions (2009).
6. Bahl, P. and Padmanabhan, V.N. RADAR: an in-building RF-based user location and tracking system. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, (2000), 775–784.
7. Bell, G. and Dourish, P. Yesterday's tomorrows: notes on ubiquitous computing's dominant vision. *Personal and Ubiquitous Computing 11*, 2 (2007).
8. Bernstein, M.S., Brandt, J., Miller, R.C., and Karger, D.R. Crowds in two seconds: enabling realtime crowd-powered interfaces. *UIST '11: Proceedings of the 24th annual ACM symposium on User interface software and technology*, ACM Request Permissions (2011).
9. Brotherton, J.A. and Abowd, G.D. Lessons learned from eClass: Assessing automated capture and access in the classroom. *ACM Transactions on Computer-Human Interaction (TOCHI) 11*, 2 (2004), 121–155.
10. Chalmers, M., MacColl, I., and Bell, M. Seamful design: showing the seams in wearable computing. *Eurowearable, 2003. IEE*, (2003), 11–16.

11. Cohn, G., Gupta, S., Froehlich, J., Larson, E., and Patel, S. GasSense: Appliance-level, single-point sensing of gas activity in the home. *Pervasive Computing*, (2010), 265–282.
12. Consolvo, S., McDonald, D.W., Toscos, T., et al. Activity sensing in the wild: a field trial of ubifit garden. *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ACM Request Permissions (2008).
13. Davies, N., Cheverst, K., Mitchell, K., and Efrat, A. Using and Determining Location in a Context-Sensitive Tour Guide. *Computer 34*, 8 (2001), 35–41.
14. Dey, A.K., Abowd, G.D., and Salber, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction 16*, 2 (2001), 97–166.
15. Dey, A.K., Hamid, R., Beckmann, C., Li, I., and Hsu, D. a CAPpella: programming by demonstration of context-aware applications. *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Request Permissions (2004).
16. Dey, A.K., Sohn, T., Streng, S., and Kodama, J. iCAP: interactive prototyping of context-aware applications. *PERVASIVE'06: Proceedings of the 4th international conference on Pervasive Computing*, Springer-Verlag (2006).
17. Dietz, P. and Leigh, D. DiamondTouch: a multi-user touch technology. *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, ACM Request Permissions (2001).
18. Dietz, P., Yerazunis, W., and Leigh, D. Very low-cost sensing and communication using bidirectional LEDs. *UbiComp 2003: Ubiquitous Computing*, (2003), 175–191.
19. Du, H., Henry, P., Ren, X., et al. Interactive 3D modeling of indoor environments with a consumer depth camera. *UbiComp '11: Proceedings of the 13th international conference on Ubiquitous computing*, ACM Request Permissions (2011).
20. Edwards, W.K. and Grinter, R.E. At Home with Ubiquitous Computing: Seven Challenges. *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, Springer-Verlag (2001).
21. Froehlich, J., Larson, E., Campbell, T., Haggerty, C., Fogarty, J., and Patel, S. HydroSense: infrastructure-mediated single-point sensing of whole-home water activity. *Ubicomp '09: Proceedings of the 11th international conference on Ubiquitous computing*, (2009), 235–244.
22. Gellersen, H.-W., Beigl, M., and Krull, H. The MediaCup: Awareness Technology Embedded in a Everyday Object. *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, Springer-Verlag (1999).
23. Greenberg, S. and Fitchett, C. Phidgets: easy development of physical interfaces through physical widgets. *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, ACM Request Permissions (2001).
24. Harrison, C., Tan, D., and Morris, D. Skinput: appropriating the body as an input surface. *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, ACM Request Permissions (2010).
25. Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. Sensing techniques for mobile interaction. *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, ACM Request Permissions (2000).
26. Hodges, S., Villar, N., Scott, J., and Schmidt, A. A New Era for Ubicomp Development. *Pervasive Computing, IEEE 11*, 1 (2012), 5–9.
27. Izadi, S., Kim, D., Hilliges, O., et al. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. *UIST '11: Proceedings of the 24th annual ACM symposium on User interface software and technology*, ACM Request Permissions (2011).
28. Krumm, J. and Cermak, G. Rightspot: A novel sense of location for a smart personal object. *UbiComp '03: Proceedings of hthe 5th International Conference on Ubiquitous Computing*, (2003).
29. Langheinrich, M. Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems. *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, Springer-Verlag (2001).
30. Mankoff, J., Kravets, R., and Blevis, E. Some Computer Science Issues in Creating a Sustainable World. *Computer 41*, 8 (2008).
31. Myers, B., Hudson, S.E., and Pausch, R. Past, present, and future of user interface software tools. *Transactions on Computer-Human Interaction (TOCHI 7*, 1 (2000).
32. Otsason, V., Varshavsky, A., LaMarca, A., and de Lara, E. Accurate GSM indoor localization. *UbiComp'05: Proceedings of the 7th international conference on Ubiquitous Computing*, Springer-Verlag (2005).
33. Patel, S., Truong, K., and Abowd, G. Powerline positioning: A practical sub-room-level indoor location system for domestic use. *UbiComp '06: Proceedings of the 8th International Conference on Ubiquitous Computing*, (2006), 441–458.
34. Patel, S.N., Reynolds, M.S., and Abowd, G.D. Detecting Human Movement by Differential Air Pressure Sensing in HVAC System Ductwork: An Exploration in Infrastructure Mediated Sensing. *Pervasive '08: Proceedings of the 6th International Conference on Pervasive Computing*, Springer-Verlag (2009).

35. Patel, S.N., Robertson, T., Kientz, J.A., Reynolds, M.S., and Abowd, G.D. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. *Proceedings of the 9th international conference on Ubiquitous computing*, Springer-Verlag (2007), 271–288.

36. Rhodes, B. and Starner, T. Remembrance Agent: A continuously running automated information retrieval system. *The Proceedings of The First International Conference on The Practical Application Of Intelligent Agents and Multi Agent Technology*, (1996), 487–495.

37. Richards, M., Petre, M., and Bandara, A. Starting with Ubicomp: using the SenseBoard to introduce computing. *43rd ACM Technical Symposium on Computer Science Education (SIGCSE 2012)*, (2012).

38. Rogers, Y. Moving on from weiser's vision of calm computing: Engaging ubicomp experiences. *UbiComp '06: Proceedings of the 8th International Conference on Ubiquitous Computing*, (2006), 404–421.

39. Satyanarayanan, M. Pervasive computing: Vision and challenges. *Personal Communications*, (2001).

40. Stuntebeck, E.P., Patel, S.N., Robertson, T., Reynolds, M.S., and Abowd, G.D. Wideband powerline positioning for indoor localization. *Ubicomp '08: Proceedings of the 10th international conference on Ubiquitous computing*, (2008), 94–103.

41. Theimer, M., Demers, A., and Welch, B. Operating system issues for PDAs. *Proceedings of the Fourth Workshop on Workstation Operating Systems*, (1993), 2–8.

42. Tolmie, P., Pycock, J., Diggins, T., MacLean, A., and Karsenty, A. Unremarkable computing. *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Request Permissions (2002).

43. Varshavsky, A., de Lara, E., Hightower, J., LaMarca, A., and Otsason, V. GSM indoor localization. *Pervasive and Mobile Computing 3*, 6 (2007).

44. Want, R. An Introduction to Ubiquitous Computing. In J. Krumm, ed. Ubiquitous Computing Fundamentals, 2010.

45. Weiser, M. The computer for the 21st century. *Scientific American*, (1991).

46. Weiser, M. Some computer science issues in ubiquitous computing. *Communications of the ACM 36*, 7 (1993).

47. Weiser, M. The computer for the 21st Century. *IEEE Pervasive Computing 99*, 1, 19–25.

48. Woodruff, A. and Mankoff, J. Environmental Sustainability. *IEEE Pervasive Computing 8*, 1 (2009).

49. Wyche, S.P., Magnus, C.M., and Grinter, R.E. Broadening Ubicomp's vision: an exploratory study of charismatic pentecostals and technology use in Brazil. *Ubicomp '09: Proceedings of the 11th international conference on Ubiquitous computing*, ACM Request Permissions (2009).

50. Youssef, A., Krumm, J., Miller, E., Cermak, G., and Horvitz, E. Computing location from ambient FM radio signals [commercial radio station signals]. *Wireless Communications and Networking Conference, 2005 IEEE*, (2005), 824–829.

51. Yun, T.-J., Jeong, H.Y., Hill, T.D., et al. Using SMS to provide continuous assessment and improve health outcomes for children with asthma. *IHI '12: Proceedings of the 2nd ACM SIGHIT symposium on International health informatics*, ACM Request Permissions (2012).