Chapter I
Introduction to Ubiquitous Computing

Max Mühlhäuser
Technische Universität Darmstadt, Germany

Iryna Gurevych
Technische Universität Darmstadt, Germany

ABSTRACT

The present chapter is intended as a lightweight introduction to ubiquitous computing as a whole, in preparation for the more specific book parts and chapters that cover selected aspects. This chapter thus assumes the preface of this book to be prior knowledge. In the following, a brief history of ubiquitous computing (UC) is given first, concentrating on selected facts considered as necessary background for understanding the rest of the book. Some terms and a few important standards are subsequently mentioned that are considered necessary for understanding related literature. For traditional standards like those widespread in the computer networks world, at least superficial knowledge must be assumed since their coverage is impractical for a field with such diverse roots as UC. In the last part of this chapter, we will discuss two kinds of reference architectures, explain why they are important for the furthering of Ubiquitous Computing and for the reader's understanding, and briefly sketch a few of these architectures by way of example.

A BRIEF HISTORY OF UBIQUITOUS COMPUTING

Mark Weiser

The term ubiquitous computing was coined and introduced by the late Mark Weiser (1952-1999). He worked at the Xerox Palo Alto Research Center (PARC, now an independent organization). PARC was more or less the birthplace of many developments that marked the PC era, such as the mouse, windows-based user interfaces, and the desktop metaphor (note that Xerox STAR preceded the Apple Lisa, which again preceded Microsoft Windows), laser printers, many concepts of computer supported cooperative work

(CSCW) and media spaces, and much more. This success is contributed (among other reasons) to the fact that PARC managed to integrate technology research and humanities research (computer science and "human factors" in particular) in a truly interdisciplinary way. This is important to bear in mind since a considerable number of publications argue that the difference between UC and Ambient Intelligence was the more technology/networks-centered focus of the former and the more interdisciplinary nature of the latter that considered human and societal factors. We do not agree with this argument, in particular due to the nature of the original UC research at PARC—and the fact that quite a number of UC research labs worldwide try to follow the PARC mindset. Indeed, Mark Weiser concentrated so much on user aspects that quite a number of his first prototypes were mere mockups: during corresponding user studies, users had to imagine the technology side of the devices investigated and focus on use cases, ideal form factors and desired features, integration into a pretend intelligent environment, and so forth.

Weiser's Vision of UC

Mark Weiser's ideas were first exposed to a large worldwide audience by way of his famous article *The Computer of the 21st Century,* published in *Scientific American* in 1991. A preprint version of this article is publicly available at: http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html.

Maybe the most frequently cited quotation from this article reads as follows: "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." This was Mark's vision for the final step in a development away from "standard PCs", towards a proliferation and diversification of interconnected computer-based devices. A deeper understanding of Mark Weiser's visions can be drawn from his position towards three dominant, maybe overhyped trends in computer science at his time: virtual reality, artificial intelligence, and user agents. With a good sense for how to raise public attention, Mark criticized these three trends as leading in the wrong direction and positioned UC as a kind of "opposite trend". We will follow Mark's arguments for a short while and take a less dramatic view afterwards.

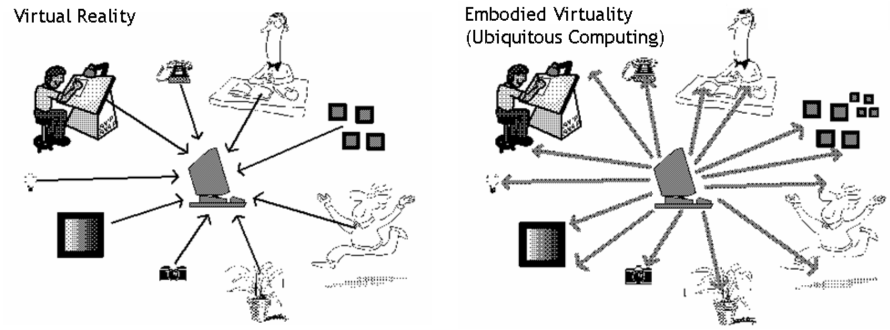
UC vs. Virtual Reality (VR)

According to Mark, VR "brings the world into the computer", whereas UC "brings the computer into the world". What he meant was that VR technology is generally based on elaborate models of an existing or imagined (excerpt of the) world. This model contains not only 3D (geometric) aspects but many more static and dynamic descriptions of what is modeled. For instance, digital mockups of cars have been pushed to the point of simulating crash tests based on the car /obstacle geometry, static, and dynamic material characteristics, laws of physics, and so forth. As the sophistication of models grows, more and more aspects of the world are entered into the computer, finally almost everything happens in the virtual space and even the human becomes a peripheral device for the computer, attached via data gloves and head-mounted displays. Mark Weiser criticized mainly the central and peripheral roles of computers and humans, respectively. He proposed to follow the UC vision in order to *invert* these roles: by abandoning the central role of computers and by embedding them in the environment (in physical objects, in particular), room is made for the *human in the center.* In this context, he used the term "embodied virtuality" as a synonym for UC. The cartoons in Figure 1 were made by Mark Weiser and provided by courtesy of PARC, the Palo Alto Research Center, Inc.

UC vs. Artificial Intelligence (AI)

In essence, Mark Weiser criticized the overly high expectations associated with AI in the 1980's. In

Figure 1. Mark Weiser's cartoons about UC vs. virtual reality



the late 1980's and early 1990's, that is, at the time when he developed his UC vision, AI research had to undergo a serious confidence crisis. The term AI had not been associated with a commonly accepted, reasonably realistic definition, so that the association with human intelligence (or the human brain) was destined to lead to disappointments. The AI hype had provided researchers with considerable funds—but only for a while. Mark Weiser proposed to take a different approach towards a higher level of sophistication of computer-based solutions (which had been the goal of AI at large). He considered it a more reasonable objective to concentrate on small subsets of "intelligent behavior" and to dedicate each computer to such a subset. Higher sophistication would be fostered by interconnecting the special-purpose computers and by making them cooperate. This reasoning lead to the term *smart,* considered more modest than the term *intelligent.* Sensor technology plays an important role in dedicating computers to a small subset of "understanding the world around us" (a key element of intelligent behavior). By widely deploying and interconnecting sensor-based tiny computers, one would be able to integrate environmental data (location, temperature, lighting, movement, etc.) and use this information to produce smart behavior of computers and computerized physical objects.

UC vs. User Agents (UA)

In contrast to virtual reality and artificial intelligence, the term *user agent* is not very prominent in the general public. At the time referred to, UAs were thought as intelligent intermediaries between the user and the computer world, that is, as an approach towards increased ease-of-use or better human-computer interaction. User agents were often compared to the common perception of British butlers who are very discreet and unobtrusive, but always at disposal and extremely knowledgeable about the wishes and habits of their employers. Following this analogy, UAs were installed as autonomous software components between applications and users, inspecting and learning from the user-software application. Mark Weiser challenged five requirements usually derived from this analogy for user agents and proposed UA as a better alternative for the first three; as to the last two, he judged the necessary base technology as immature:

1. UAs were supposed to give advice to their users based on what they had learned. Mark Weiser asked, in essence, why they would not do the job themselves—a promise that UC should fulfill;
2. UAs were supposed to obey the user, for example, by applying planning algorithms to basic operations with the aim to fulfill

the goals set by a user. In contrast to this approach, UC was intended to behave rather proactively, that is, to propose and even act in advance as opposed to *re*acting on command;

3. A third widespread requirement suggested that UAs should intercept the user-application interface. UC in contrast should be more radical and take over the interaction or carry out functions on its own—an approach presumed by Mark Weiser to be the only viable one if humans were to be surrounded by hundreds of computers;
4. A basic assumption about UAs was that they would listen to the (interactions of) the user. Mark Weiser considered natural language processing technology and speech recognition technology at his time to be far too immature to promise satisfying results in this respect;
5. UAs should learn the users' preferences, wishes, and so forth by observation. Again, the necessary (machine learning) technology was judged to be too immature to live up to this promise.

We will resume the VR / AI / UA discussion in the next large section.

Mark Weiser's Three Key Devices

We want to finish this lengthy but still extremely compressed, much simplifying and abstracting treatment of Mark Weiser's contributions by looking at three devices. These complementary UC devices were prototyped at his lab; investigated in the context of PARC's typical creative, team-oriented setting, all three were thought as electronic replacements for the common "analog" information appliances.

The *Xerox "Pad"* can be considered to be the prototype and father of present PDA's, introduced even before Apple's Newton appeared in 1993. The initial concept was that of an electronic equivalent to "inch-size" information bearers, namely "PostIt Notes": easy to create and to stick almost everywhere, available in large quantities. As the PDA analogy suggests, the prototypes had a lot more functionality than PostIt Notes—but were also a lot more expensive and cumbersome to handle by design (not only due to short and mid-term technology limitations).

The *Xerox "Tab"* can be considered to be the prototype and father of present Tablet PC's. The analogy from the traditional world was that of a "foot-size" information bearer, namely a notebook or notepad. One may infer from the rather stalling market penetration of Tablet PC's that technology is still not ready for mass market "Tabs" today, but one may also expect to find a pen centric, foot size, handheld computer to become very successful any time soon. An interesting facet of the original Tab concept was the idea that Tabs would in the future lay around for free use pretty much as one finds paper notebooks today, for example, as part of the complementary stationery offered to meeting participants.

The *Xerox "Liveboard"* was the prototype of present electronic whiteboards. A PARC spinoff company designed and marketed such boards, and today many companies like Calgary-based SmartTechnologies Inc. still sell such devices. Liveboards represented the "yard-size" information bearers in the family of cooperating devices for cooperating people. In contrast to many devices sold today, Liveboards supported multi-user input pretty early on.

The developments and studies conducted at Mark Weiser's lab emphasized the combination of the three device types for computer supported cooperation, and cooperative knowledge work in particular.

While Mark Weiser was a truly outstanding visionary person with respect to predicting the future of hardware, that is, UC nodes (proliferation of worn and embedded networked devices, specialized instead of personal general-purpose computers, numbers by far exceeding the number of human users), two other people were more instrumental in generating awareness for the two remaining big challenges mentioned in the preface of this book, namely *integrative cooperation* and *humane computing*; the former of these challenges was emphasized by Kevin Kelly, the latter by Don Norman. A deeper analysis reveals that for the second aspect, humane computing, it is very difficult to argue about the true protagonists. Readers remember that Mark Weiser was actually placing a lot of emphasis on usability, by virtue of his education and mindset and in the context of the human focus of PARC. He also coined the exaggerated term "invisible" for mature technology. On the other hand, Don Norman was not advocating the humane computing challenge in all its facets yet. Nevertheless, we want to highlight him next as maybe the single most important advocate of this challenge.

The Book *Out of Control* by Kevin Kelly

In 1994, K. Kelly published a book entitled *Out of Control*. The thoughts expressed by Kelly were an excellent complement to Mark Weiser's publications. While the latter emphasized the emergence of networked small "neuron like" (i.e., smart) UC nodes, Kelly emphasized the integrated whole that these neurons should form. His starting argument was the substantiated observation that the *complexity of the made,* that is, of human-made systems or technology, approached the *complexity of the born*, that is, of "nature-made" systems, such as human or biological organisms, human or biological societies (cf. ant colonies), and so forth.

This observation led to the obvious requirement to investigate the intrinsic principles and mechanisms of how the born organized, evolved, and so forth. By properly adopting these principles to '"the made", this complexity might be coped with. Research about the organization and evolution of the born should be particularly concerned with questions such as: how do they cope with errors, with change, with control, with goals, and so forth. For instance, beehives were found *not* to follow a controlling head (the queen bee does *not* fulfill this function), and it is often very difficult to discern primary from subordinate goals and to find out how goals of the whole are realized as goals of the individuals in a totally decentralized setting.

Kevin Kelly summarizes central findings and laws of nature several times with different foci. Therefore, it is not possible to list and discuss these partly conflicting findings here in detail. An incomplete list of *perceived central laws* "of God" reads as follows: (1) give away control: make individuals autonomous, endow them with responsible behavior as parts of the whole, (2) *accept* errors, even "build it in" as an essential means for selection and constant adaptation and optimization, (3) distribute control *truly*, that is, try to live with no central instance at all, (4) promote *chunks* of different kinds (e.g., hierarchies) for taming complexity, and (5) accept heterogeneity and disequilibrium as sound bases for survival.

The Book *The Invisible Computer* by Donald Norman

Don Norman emphasized the "humane computing" grand challenge described in the preface of this book. World renowned as an expert on usability and user-centered design, he published *The Invisible Computer* in 1999. He considered the usability problems of PC's to be intrinsically related to their general-purpose nature and thus perceived the dawning UC era more as a chance than a risk for humane computing. The intrinsic usability problems that he attributed to PCs were rooted in two main anomalies, according to Don Norman: (1) PCs try to be all-purpose and all-user devices—a fact that makes them overly complex, and (2) PC's are isolated and separated from daily work and life; truly intuitive use—in the context of known daily tasks—is therefore hardly possible. From this analysis, Norman derived various *design guidelines*, *patterns*, and *methodological implications*, which we will summarize again at an extremely coarse level:

1. He advocated UC nodes using the term "information appliances": dedicated to a specific task or problem, they can be far simpler and more optimized;
2. He further advocated user-centered development: especially with a specific user group in mind, "information appliances" as described previously can be further tailored to optimally support their users;
3. Norman stated three key axioms, that is, basic goals to be pursued during design and development: simplicity (a drastic contrast to the epidemic "featurism" of PC software), versatility, and pleasurability as an often forgotten yet success critical factor;
4. As a cross-reference to the second big UC challenge (integrative cooperation), he advocated "families of appliances" that can be easily and very flexibly composed into systems.

History Revised

The preceding paragraphs are important to know for a deeper understanding of the mindset and roots of UC. However, about 15 years after the time when the corresponding arguments were exchanged, it is important to review them critically in the light of what has happened since. We will first revise the three "religious disputes" that Mark Weiser conducted against AI, VR, and UAs. To put the bottom line first, the word "versus" should rather be replaced by "and" today, meaning that the scientific disciplines mentioned should be (and have, mostly) reconciled:

As to *UC and VR*, specialized nodes in a global UC network can only contribute to a meaningful holistic purpose if models exist that help to cooperatively process the many specialist purposes of the UC nodes. In other words, we need the computer embedded into the world and the world embedded in the computer. Real Time Enterprises are a good example for very complex models—in this case, of enterprises—for which the large-scale deployment of UC technology provides online connectivity to the computers embedded into the world, that is, specialized nodes (appliances, smart labels, etc.). In this case, the complex models are usually not considered VR models, but they play the same role as VR models in Mark Weiser's arguments. The progress made in the area of augmented reality is another excellent example of the benefit of reconciliation between UC and VR: in corresponding applications, real-world vision and virtual (graphical) worlds are tightly synchronized and overlaid.

As to *UC and AI*, Mark Weiser had not addressed the issue of how interconnected, smart, that is, "modest", specialized nodes would be integrated into a sophisticated holistic solution. If the difference between AI and the functionality of a single smart UC node (e.g., temperature sensor) was comparable to the difference between a brain and a few neurons, then how can the equivalent of the transition (evolution) from five pounds of neurons to a well-functioning brain be achieved? Mark Weiser did not have a good answer to that question—such an answer would have "sounded like AI" anyway.

Today, there is still not a simple answer yet. The most sophisticated computer science technology is needed in order to meet the integration challenge of how to make a meaningful whole out of the interconnected UC nodes. However, the state of the art has advanced a lot and our understanding for what can be achieved and what not (in short term) has improved. For instance, socionic and bionic approaches have become recognized research areas. A mature set of methods and algorithms is taught in typical "Introduction to AI" classes today and has replaced the ill-defined, fuzzy former understanding of the area. Thus the boundaries between AI and computer science are more blurred than ever and their discussion is left to the public and press.

As to *UC and UAs*, remember that Mark Weiser considered UAs as "too little" in terms of what they attempted (at least too little for the UC world envisioned by him), yet "too much" in terms of what the underlying technology was able to provide. This left doubts about how the even more ambitious goals of UC could be met, namely active (proactive, autonomous, even responsible) rather than reactive (obeying) behavior. In other words, Mark Weiser was right when he advocated active as opposed to reactive behavior, but he had little to offer for getting there. Luckily, the technologies that he had then considered immature (e.g., speech processing, NLP, machine learning) have advanced a lot since.

All in all, Mark Weiser's arguments from 15 years ago (1) provide a deep understanding of the field, (2) should be modified towards a more conciliatory attitude (in particular with respect to AI and VR / complex "world models"), and (3) have become more substantiated in certain respects since technology advancements make some of his more audacious assumptions more realistic (but most visions of his "opponents", too). In other words, Mark Weiser's visions were and still are *marking* the research and developments made by the UC community. His concepts and predictions were accurate to a degree that was hardly paralleled by any other visionary person. Restrictions apply as to his overly drastic opposition to VR, AI, and UAs: some of the exaggerated promises of these were repeated by him in the UC context - right when he denounced the over-expectations raised by AI and UAs! VR and AI in particular should be reconciled with UC. Maybe Weiser underestimated the two grand challenges of the UC era, namely "integrative cooperation" and "humane computing".

Kevin Kelly and Donald Norman emphasized these two challenges, respectively. Looking at the advancements in totally decentralized systems, Kelly's promises can be evaluated as too extreme today: bionics social science inspired, and autonomic (or autonomous) computing have advanced a lot. However, two restrictions still apply: (1) less decentralized systems still prove to be extremely viable in daily operation—it will be hard for fully decentralized systems to really prove their superiority in practice; (2) system-wide goals must still be planned by some centralized authority and—to a certain extent manually—translated into methods for fully decentralized goal pursuit; evolution-like approaches that would generate optimization rules and their pursuit automatically in a fully decentralized systems are still hardly viable. As a consequence, the present book will not only describe the above-mentioned computing approaches in part "Scalability", but also other aspects of scalability.

As to Don Norman, he was right to advocate simplicity as a primary and key challenge. However, he maybe underestimated the 'humane computing' problems associated with the nomadic characteristics and 'integrative cooperation' challenge of the UC era. The usability of the integrated whole that we advocate to build out of UC nodes is by far not automatically endowed with easy-to-use user interaction just because the participating appliances exhibit a high degree of usability. On the other hand, only the integration that is, federation of miniature appliances with large interaction devices (wall displays, room surround sound, etc.) may be able to provide the usability desired for an individual device.

As we conclude this section, we should not forget to mention that the UC era was of course not only marked by just three visionary people.

TERMS AND SELECTED STANDARDS

While there is a lot of agreement among researchers and practitioners worldwide that the third era of computing is dawning as the era of networked, worn/portable and embedded computers, there is not so much agreement about what to *call* that era. This fact is something of an obstacle, for instance for wider recognition in politics (the crowd does not scream the same name as one may put it). This situation is aggravated by the fact that partial issues and aspects of Ubiquitous Computing are also suffering from buzzword inflation. With this background in mind, one may understand why we list a considerable number of these buzzwords below and provide a short explanation, rather than swapping this issue out into a glossary alone. Knowledge of the following terms is indeed necessary for attaining a decent level of "UC literacy".

Synonyms for Ubiquitous Computing

First, we want to look at the terms that describe—more or less—the third era of computing as introduced:

- **Post-PC era:** The root of this term is obvious, it describes 'the era that comes after the second, that is, the PC era. We suggest avoiding this term since it points at what it is not (PC's) rather than at what it actually is.
- **Pervasive computing:** A distinction between the word ubiquitous and pervasive is difficult if not artificial. One could argue that the term pervasive eludes more to the process of penetration (i.e., to the verb pervade) whereas ubiquitous eludes more to the final state of this process. We suggest that pervasive computing and ubiquitous computing are synonyms, one (pervasive) being slightly more common in industry (its origin has been attributed to IBM), the other one (UC) being slightly more common in academia.
- **Ubiquitous computing:** The term may be interpreted as "computers everywhere". We are using it as the notion for the third era of computing throughout the book and prefer it, among others, because we try to fight buzzword mania and dislike the invention of additional terms for a named concept. We therefore propose to stick to the first (reasonable) term invented and somewhat broadly accepted; since Mark Weiser is the first visionary person who sketched essential characteristics of the dawning era and since he invented the term UC, the question of what is the oldest well-known term should not be questionable.
- **Ambient intelligence:** This term was invented in particular in the context of the European Union's research framework programs (5, 6, 7). As a positive argument, one may say that the two words reflect the grand challenges of UC as stated in this book: ambient may be associated with the challenge of humane computing, making UC systems an integral part of our daily life. Intelligence may be interpreted as the challenge of integrative cooperation of the whole that consists of myriads of interconnected UC nodes. On the downside, one should remember that Mark Weiser had intentionally avoided the term "intelligence" due to the over-expectations that AI had raised. We suggest avoiding this term, too, because it is still burdened with these over-expectations and because it is still ill defined.
- **Disappearing / invisible / calm computing:** All three terms are less common than UC and pervasive computing. Their roots have been discussed in the historical context above. Obviously, disappearing describes again a process while "invisible" describes a final state. "Calm" emphasizes hearing as opposed to vision like the other two. In any case, the terms "invisible" and "disappearing" are not very well chosen (despite our tribute to Don Norman) since computers and interfaces that have totally disappeared cannot be commanded or controlled by hu-

mans any more. Since we doubt that 100% satisfactory service to the user can be paid at all without leaving the customer, that is the user, the option to explicitly influence the service behavior, we consider the term misleading. We favor again Mark Weiser's notion of computers that are so well interwoven with the fabric of our lives that we hardly notice them.

- **Mixed-mode systems:** This is a term used to describe the heterogeneity of UC nodes, in contrast to the rather resource rich, general purpose PC's of the last era. This term is even less common, but pops up every now and then like those previously discussed, and should not be used to describe UC as a whole since it emphasizes a particular aspect.
- **Tangible bits:** This term has found some currency in the Netherlands and Japan, but remained rather uncommon in general. It refers mainly to the fact that networked computers are becoming part of the physical world.
- **Real time enterprise:** This term has been explained in the preface of the book and is not thought as a synonym for UC, but rather as a very important and cutting-edge application domain that may drive down the learning curve, that is, prices of UC hardware and solutions.

It was mentioned in the preface that some authors argued in favor of one or the other of the UC synonyms, saying that their choice was more far-reaching in time (the other ones being intermediate steps) or space (the other ones only comprising a subset of the relevant issues). However, we cannot follow these arguments, mainly because research labs and projects around the world work on the same subjects, some more advanced or holistic, some less ambitious or more specialized, carrying the names UC, pervasive computing, and ambient intelligence rather randomly.

Towards a Taxonomy of UC Nodes

Throughout this book, UC nodes will be categorized according to different aspects. In the context of reference architectures further below, we will emphasize the role of UC nodes in a holistic picture. In the present paragraph, we want to try categorizing them as devices. It should be noted that in the preface of the book, we already provided a preliminary, light weight introduction. The difference between carried (worn, portable) and encountered nodes was emphasized and four preliminary categories (wearables, sensors, appliances, and smart labels) were briefly described. It soon became clear that smart labels attached to goods must be distinguished again from those attached to humans, although the base technology may be the same.

In a second, more serious attempt to categorize UC nodes as device categories, we propose the following distinction (see Figure 2):

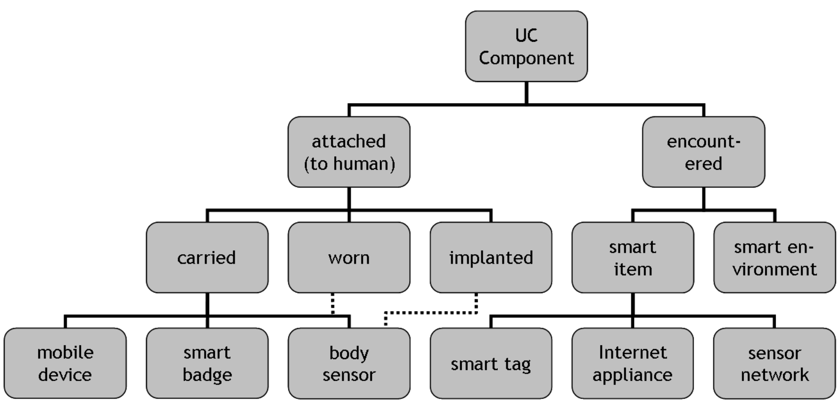
1. **Devices attached to humans**
    a. **Devices carried:** Here we further distinguish three subcategories: (1) *mobile devices*, synonymous with portable devices, contain rather general purpose computers and range from laptops via PDA's to mobile phones and the like, (2) *smart badges*, that is, smart labels serve for identification, authentication and authorization of humans and possibly further purposes, and (3) *body sensors* of all kinds play an increasingly important role in particular in the fitness and health context;
    b. **Devices worn:** These wearables range from truly sophisticated, computer-augmented cloths and accessories to prototypes that are built from standard components (PDA in a holster with headset, etc.). A further categorization is not attempted since the spectrum is rather blurred;

   c. **Devices implanted:** while there is a lot of hype about implanted RFID tags and networked health implants, the many issues (e.g., health, privacy, or dependability) around the necessary device-environment communication have not permitted this category to become widespread.
2. **Devices encountered**
   a. **Smart items** denote computer-augmented physical objects. The terms “smart object” and “smart product” are used with subtle differences depending on the context to denote more sophisticated variants of smart items, such as smart items that proactively communicate with the users. We suggest treating smart items as the most general term and to distinguish the following subcategories: (1) *smart tags* as the least sophisticated variant: they can be considered to be mimicry for embedded computers: by attaching a smart tag to a physical object, a physically remote computer (often in proximity, though) can take over some of the functionality that would be embedded otherwise. This approach opens the door for turning even the cheapest products into UC nodes. The term “smart label” is sometimes used synonymously; sometimes it is used as the comprehensive term for smart tags and smart badges (attached to humans, see earlier discussion). We suggest sticking to the term smart tag for the smart item sub-category described here; (2) *networked sensor nodes*, and (3) *networked appliances* denote the other subcategories of smart items. They were already introduced in the preface of this book.
   b. **Smart environments** denote the surroundings of smart items, that is, the additional communication and compute power installed in order to turn an assembly of smart items into a local, meaningful whole.

The reader must be aware that all terms arranged in the taxonomy are not settled yet for a common understanding. For instance, one might argue whether a sensor network that computes context information for networked appliances and users should be considered a set of smart items (as we defined it) or a part of the smart environment. Nevertheless, we find it useful to associate a well-defined meaning with these terms and to apply it throughout the book (see Figure 2).

In addition, it should be noted that *smart environments* (with integrated smart items) constitute a particularly important research

Figure 2. Taxonomy of UC devices



area—maybe because they permit researchers and project leaders to implement self-contained "little UC worlds" without a need for multiparty agreements about interoperability standards. In particular, "smart homes" were among the first subjects of investigation in the young history of UC. Prestigious projects in the smart home area were and are conducted by industry (Microsoft eHome, Philips AmbientIntelligence initiative, etc.) and academia (GeorgiaTech AwareHome, MIT House, etc.). HP made an early attempt to overcome the isolation of such incompatible islands by emphasizing standard middleware in the *Cooltown* project). Quite a number of projects about smart homes terminated without exciting results, not to the least due to insufficient business impact (note our argument in favor of Real Time Enterprises as a more promising subject). More recently, smart homes projects have focused on issues considered to be particularly promising, as was discussed in the preface to this book. Important areas comprise home security, energy conservation, home entertainment, and particularly assisted living for the aging society—a topic considered particularly interesting in Europe (1 year prolongation of independepnt living saving about half a billion Euros in Germany alone). Renowned large-scale projects were carried out, for example, in Zwijndrecht (Belgium) and Tønsberg (Norway) in this respect.

A Few More Relevant Terms

A few more UC terms—and sometimes, corresponding concepts—are worth mentioning.

- **Smart dust** is a term used for sensor networks if the emphasis is on miniaturization and the concept is based on one-time deployment and zero maintenance. Environment data sensors are often cited as an example, the vision then is to deploy them, for instance, from an aircraft, and let them monitor the environment until they fail. Environment-friendly degradation is a major issue here, of course.
- **Things that think** was the name of an early UC project led by Nicholas Negroponte at the MIT media lab. Other authors have since hijacked the term.
- **Smart paper** denotes the vision of a display device that would exhibit characteristics comparable to traditional paper in terms of weight, robustness, readability, and so forth, and loadable with the content of newspapers, journals, books and so forth,. it would help to save paper and revolutionize the press distribution channels and more. Many projects that were not even close to this vision had, and continue to have, the name "smart paper".
- **Smart wallpaper** is a similar term to *smart paper* in that it extrapolates the above mentioned characteristics to wall-size devices.
- **Smart <you-name-it>:** virtually every noun has been associated with the attribute smart recently, not always alluding to the characteristics of UC nodes. For instance, smart materials are supposed to adapt to the context of use, with no IT involved. Most of the time though, smart <something> alludes to a physical object that has been augmented with an embedded computer.
- **The Internet of things** is a term favored by the press. It is not considered appropriate as a term for UC as a whole by the authors since it emphasizes the hardware side of UC as opposed to the human side, which was already described as crucial and as a major challenge (cf. humane computing). Most publications that favor this term concentrate on the two standards discussed in the following section.

The EPCglobal Standard

As mentioned at the beginning, we will only sketch two important standards in the UC context. Other standards are too unimportant, too immature, too specific (they might be treated in one of the focused parts of this book), or part of the background knowledge about well-established technology that this book cannot cover. The first standard to mention is EPCglobal and was mentioned in the preface of this book. As mentioned, it is meant to succeed the barcodes that encode the European article number or universal product code on current consumer products. The 96-bit Electronic Product Code EPC is usually stored on RFIDs (a subcategory of smart tags, as we can now say) and can be read:

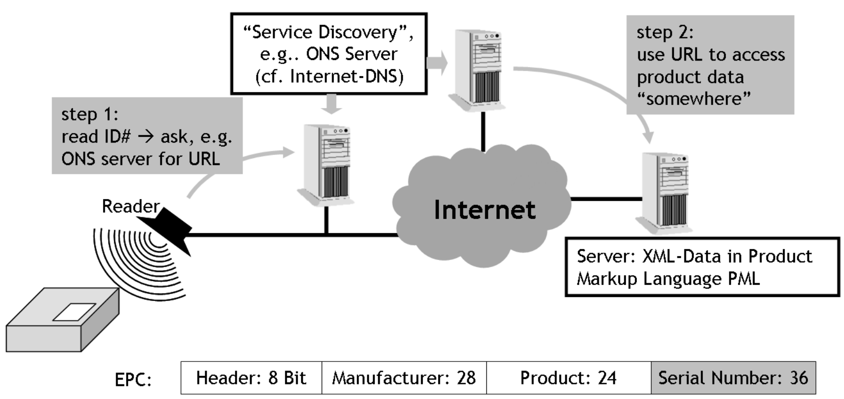
- From a greater distance (e.g., 10m)
- With better reading accuracy
- With much less effort (e.g., en-passant by a RFID reader gate as opposed to carefully with line-of-sight connection by a barcode scanner)
- In bulk (RFID readers can read, for example, a hundred tags at once)

Since the EPC contains a 36-bit serial number, individual items can be tracked and traced. For instance, theft can be much more easily attributed to criminals, product life cycles can be recorded more accurately, product lots with manufacturing errors can be called back more specifically, etc. On the other hand, the serial number may in principle be used to trace an individual, too, if she carries around an RFID tagged product. This privacy issue has raised many concerns in recent years and amplified the decision of the whole sales and retail industry to focus on tagging their containers, palettes, cases, etc., for a start. So-called item level tagging is only envisioned for highly valuable goods initially; it may enter the mass market when tag prices and system costs have come down and after settling the privacy issues.

Figure 3 depicts the functioning of EPC smart tags in an overall IT infrastructure. In step 1, an EPC code is read from a product. In the example, each carton on the palette could contain a number of tagged products. The residual example would then explain the action for just one such tag. Usually prior to reading the tag, the system has already searched and discovered servers capable of 'resolving' certain ranges of EPC code. Based on the results of this discovery process, the appropriate 'resolution node', called an ONS server, is asked to resolve the EPC code, that is to translate it into a global Internet address where the relevant product information is actually stored. The product information is encoded in a standardized way, using the so-called product markup language PML, an XML derivate.

The second generation of RFID tags introduced in 2006 features improved bulk reading (hundreds of tags simultaneously), size and cost improvements. "Printable" tags have become common: these paper labels with embedded RFID chips can

Figure 3. RFID / EPC scheme



be custom imprinted with custom human-readable information. The chips themselves are not altered in the printer and they come with pre-assigned EPC codes from the manufacturer.

The OSGi Standard

The Open Services Gateway Initiative (OSGi) is an industry driven nonprofit consortium. OSGi standardized a Java virtual machine (JVM). This JVM can be considered a standardized virtual 'computer' that runs on any real computer and is capable of executing programs that are transmitted to it, so-called *bundles*. OSGi standardizes not only the format for bundles, but also the necessary protocols and procedures for authenticating and authorizing senders of bundles, for replacing and updating bundles (remote maintenance), for discovering other bundles, and so forth. OSGi bundles are particularly useful for controlling the functionality of networked appliances. Possible use cases include SetTopBoxes, Vehicles (note that car electronics today requires much shorter maintenance cycles than the mechanical parts, especially for software updates!), consumer electronics, and so forth. As to smart homes, the favored concept is that of a residential gateway that is connected to the global Internet and receives updates for smart home appliances via OSGi. The residential gateway may then forward bundle updates and so forth to the relevant appliances if needed.

OSGi has a number of deficiencies. For instance, it is not considered to be very resource effective. Nevertheless, it has tremendous impact as a de facto standard for dealing with some of the elementary aspects of coping with global UC systems in a platform and vendor independent way.

REFERENCE ARCHITECTURES FOR UBIQUITOUS COMPUTING

The Importance and Role of a Reference Architecture

A sophisticated distributed infrastructure is needed in order to make a myriad of networked UC nodes communicate and cooperate. If interoperability is to take on a worldwide scale, means for agreement among arbitrary participants must be provided. Ideally, the move from isolated proprietary UC solutions to a world of cooperating UC components is driven by so-called *reference architectures* which establish several *levels* of agreement: on level one, a common terminology and conceptualization of UC systems is established in order for researchers and practitioners to speak the same language and to work on the same global UC vision. On the second level, a common understanding of the ensemble and components of a typical UC system is established, including the potential roles of the components. On level three, basic functional principles can then be agreed upon. A fourth level is desirable but beyond the scope of reference architectures, that is concrete standards for intercomponent cooperation. This level is discussed in the introduction to the part Scalability.

Reference Architectures in a More Realistic World

In reality, a worldwide common understanding and corresponding standards have to be developed in a struggle for the best solution. Real life has a large impact on what becomes widespread. By "real life" we mean breaking research results, industry practice, experiences gained with proprietary prototypes and realizations, user acceptance, and not least business interests defended by global industrial players. Nevertheless, the exercise of proposing and refining reference architectures—in communication with the stakeholders mentioned—plays a key role in a struggle for globally interoperable solutions. Here reference architectures must be invented and published and then consolidated and reiterated based on feedback by the stakeholders.

Prominent Examples from the Past

The ISO reference architecture for open systems interconnection (OSI) was developed in the 1970s as an important step towards global networks. OSI was very successful in that it led to a common terminology and a common understanding of the components of computer networks including their roles. The fourth level aforementioned above: ISO standards for communication protocol, were not nearly as successful as the reference architecture itself. Rather, the Internet protocols TCP and IP took over almost the entire market. Nevertheless, the OSI reference architecture was extremely influential on the computer networking community as a whole and on the Internet in particular. Another ISO reference architecture is ODP (open distributed processing). It emphasizes complex distributed systems and applications. An influential contribution of ODP is its support for different viewpoints of various stakeholders. In particular, ODP emphasized the importance of enterprise modeling for application development. All too often, applications are modeled and built with a technology focus and thus neglect the (dynamically changing) organization they should support. ODP addresses important issues, but came at a time when distributed applications were usually rather simple: ODP was considered overkill.

Layered Architectures vs. Component Architectures

Before we introduce concrete reference architectures, it is worth recalling the two complementary flavors:

- **Layered reference architectures** serve as a blueprint for layered software architectures. Both arrange sets of functions into layers that act as virtual machines: only the "what" (provided functionality and how to access it) must be known to users in higher layers, whereas the internal "how" (realization) is hidden and can be independently modified. The layer stack represents the range from higher to lower function sets, where higher means "closer to what users and applications need" and lower means "closer to what hardware provides". Strict variants preclude higher layer components to access lower layers except for the one immediately below. Recent research has concentrated on approaches for automatic, selective custom configuration of the entire layer stack, according to the needs of applications—this trend is important in the UC world where dedicated, resource-poor UC nodes cannot host fat all-purpose layers.
- **Component reference architectures** take a birds-eye view on the world addressed. They define a number of cooperating components or rather component types, and specify inter-component cooperation at a certain level of detail. Again, a kind of art of right-sizing exists: too few component types do not really help to understand and discern relevant roles and specializations common to the world addressed, too many component types lead to overly complex architectures and problems in matching reference and reality.

Although we focus on the Computer Networks / Distributed Systems aspects of UC in the remainder of this chapter, readers should note that the entire book represents a holistic approach.

Why Component Reference Architectures are Important for UC

The OSI reference architecture assumes a network consisting of rather homogeneous nodes, namely general-purpose computers with 'sufficient' CPU and memory capacity. Accordingly, a common definition of a computer network reads as follows:

A computer network CN is a set of autonomous nodes AN, each of which disposes of CPU(s) and memory, plus a Communication Subsystem CSS capable of exchanging messages between any of the nodes: CN :== {AN} ∪ CSS.

In the definition, "all nodes are created equal". At a closer look, computer networks rely on four mandatory constituents of nodes (ANs):

1. **Communication capability:** The capacity of exchanging messages with other nodes through the CSS.
2. **Address:** A unique identifier that can be used to specify the recipient or sender of messages.
3. **Processor:** A general purpose CPU.
4. **Memory:** Means for storing—at least—incoming messages.

In a UC world, resource scarcity and the special-purpose nature of many nodes are key issues.

A holistic UC approach must scale from servers to sensors and support the consideration of smart labels etc. The definition of a UC node must be different from the one above—the four constituents now read as follows:

1. **Communication** is mandatory, but may be passive (cf. passive RFID tags)
2. **Address** is not necessarily a unique identifier; for example, in a sensor network, a random node out of a redundant set with identical address may provide a certain functionality
3. **Processor** becomes an *optional* constituent
4. **Memory** becomes an *optional* constituent, too

With the above modifications, not all nodes are *autonomous (ANs)* any more.

Proposed UC Component Reference Architectures

The definition introduces a first possibility for distinguishing nodes as components of an application, that is, from the component architecture point of view. However, it only discerns between existing versus missing fundamental characteristics. More interesting is the aspect of different roles that nodes can play in the network—not application specific roles, but fundamental roles in the set of cooperating resources. Thus UC systems will take on more complex node topologies than what was considered in the eras of simple interprocess communication and client-server computing. In addition, a holistic approach needed for UC systems raises issues such as security, which are important when trying to find important node types at different levels of granularity.

One of the first proposals for a UC component reference architecture was made by the Fraunhofer research institute FOKUS in Berlin. They did not distinguish different node types that would assume different roles, but identified important roles that *each* UC node may potentially assume. Their concept is coined *I-Centric Services* and achieved a certain level of influence on the industrial Object Management Group (OMG). In their view, a UC node (usually a software service) should provide standard interfaces for four major issues:

1. **Discovery** of peers in a spontaneous, configuration-free manner
2. **Maintainance**, i.e., software update and revision

3. **Reservation***,* that is, pre-allocation of some of the node's resources as a basis for service guarantees
4. **Configuration** as a means for customizing the service for a dedicated role

Nodes that conform to these interfaces are called *super distributed objects (SDO)* in this proposal.

We will discuss another component architecture in some more detail since it attempts to discern between more specific roles of UC nodes. It was developed in the Telecooperation Group at the Technische Universität Darmstadt and is called Mundo, see Figure 4. Mundo distinguishes five different node types: Me, Us, It, We, and They.

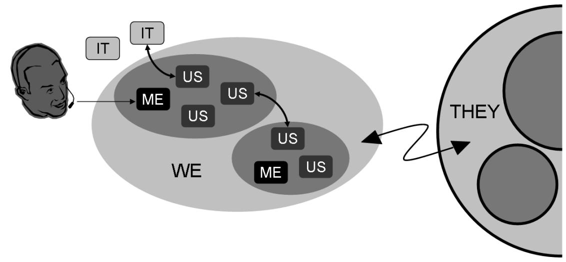
Me (Minimal Entity): Mundo emphasizes the importance of a distinct personal UC node, that is the device tightly associated with its user: the Me. Every user uses exactly only one Me at any time. The rationale is rooted in the envisioned ubiquity of computer support in everyday life: if every step that one takes is potentially computer supported and controlled, then humans need a high level of trust that the computers "do the right thing". For instance, users will want to make sure that their actions are only recorded and disclosed to the degree they consent to or that is legally imposed. As another example, they want to be sure that they only trigger actions which they understand in their legal, financial, and other consequences and that they agree to. To this end, the Mundo researchers propose to conceptualize a single, truly owned UC node type that acts in the user's stead and controls when, how, and to what extent other UC node types are invited or chartered to participate in actions. Since computer use becomes ubiquitous, such a personally-owned node type must be carried along virtually at all times. This imposes strong requirements with respect to miniaturization, robustness, and the conflicting goals of (a) the impossibility to falsify or duplicate such a node, and (b) the possibility to replace it easily in case of theft or failure. An important research questions is concerned with the minimum functionality of a Me.

Me nodes are considered as the representation of their users in the digital world—a digital persona involved in all user activities. It is a small wearable computer with minimal functionality. In order to support interaction with UC environments in a sensible way, the term *minimal* must be associated with a set of specific requirements regarding *size*, *identity*, *security*, *interaction*, *context awareness*, and *networking*. The design was guided by the principle that the minimal feature set of a system is determined by the worst-case environmental conditions under which the application must run satisfactorily (Satyanarayanan, 2001). This leads to a focus on speech based interaction and it is described in detail by Aitenbichler, Kangasharju, and Mühlhäuser (2004). Any Me can augment its capabilities through *association* with other entities of the Mundo architecture as described next.

Us (Ubiquitous aSsociable object): Minimization pressure will not permit feature-rich Mes. Hence, they must be able to connect to other mobile devices or devices embedded into the environment to offer more powerful services to their users, such as large display space. This process is called *association* and such devices are called *ubiquitous associable objects* (Us). A Us is a computing device that extends the user's *personal environment* by adding storage, processing capacity, displays, interaction devices, and so forth. During association, the Me sends authentication information to the Us, sets up a secure communication link, and personalizes the Us to suit the user's preferences and needs. For privacy reasons, any personalization of a Us becomes automatically unavailable if it is out of range of the user's Me.

It (smart ITem): There are also numerous smart items that do not support association that would classify them as Us. Vending machines, goods equipped with radio frequency IDs, and landmarks with "what is" functionality are just

Figure 4. Mundo reference architecture



a few examples. Such devices are called *smart items* (ITS). An IT is any digital or real entity that has an identity and can communicate with a US or the ME. Communication may be active or passive. Memory and computation capabilities are optional (cf. the four constituents of a UC node described previously).

WE (Wireless group Environment): Ad-hoc networking is restricted to an area near to the user of a ME device, as connections with remote services will involve a non ad hoc network infrastructure. The functionality of a wireless group environment is to bring together two or more personal environments consisting of a ME and arbitrary US entities each. It enables cooperation between the devices and also allows for sharing and transferring hardware (e.g., US devices) and software or data between WE users.

THEY (Telecooperative Hierarchical ovErlaY) stands for the backbone infrastructure as part of the Mundo component architecture. It connects users to the (nonlocal) world, and delivers services and information to the user. The THEY integrates different physical networks and provides transparent data access to users. Frequently used data may be cached on US devices.

UC Layered Reference Architectures

Many actual UC projects are based on a layered architecture. Most of them are just first approaches to software architectures, only a few of them are intended to serve as a crystallization point for the community and future standards. Nevertheless, one of them may turn out to be so successful that a future reference architecture will evolve from it. We will concentrate on a small selection of the few projects that have a general reference model in mind. They concentrate on different challenges or foci, that is their findings will have to be merged if a holistic layered architecture is to be derived.

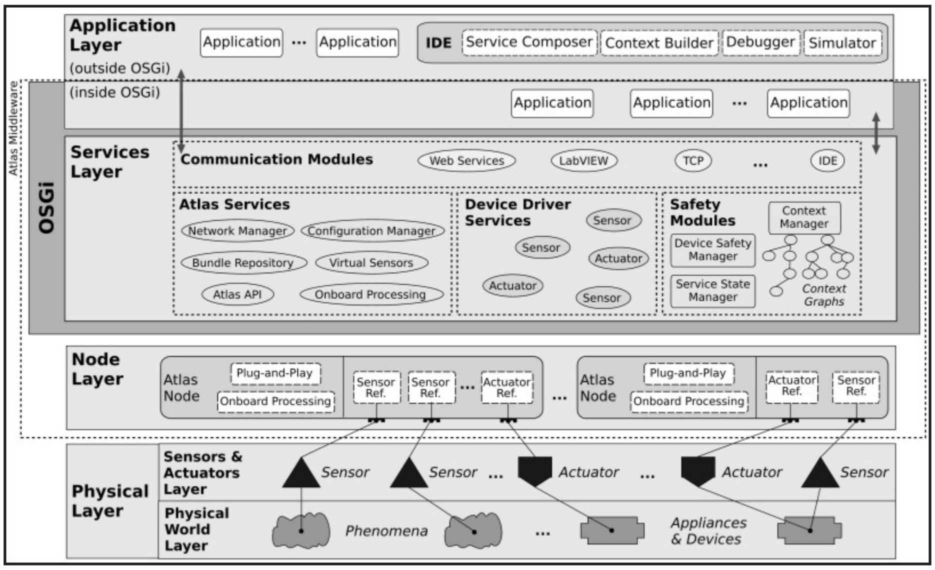
A first focus is the enterprise modeling that ODP already addressed. A reference architecture worth mentioning here is ODSI, the so-called open distributed services infrastructure (Bond, 2001). Although already outdated, ODSI was influential since it fostered the move away from ODP's more top-down approach to a component-based, that is service based approach that supports the concept of applications being compositions of services.

Other reference architectures emphasize Smart Environments. Two facets are important and investigated—still—in different camps even as to the work on reference architectures: smart information spaces and smart physical spaces. By smart information spaces, we mean environments which concentrate on cooperative treatment of IT- and data/media centric work (cf. Mark Weiser's three initial UC devices). Smart physical spaces are often called smart spaces or more specifically smart houses, labs, offices, homes etc. Work on these kinds of environments emphasizes the tangible, physical (computer-augmented) objects to be handled.

As for smart information spaces, an interesting reference architecture was proposed in the LifeSpaces project in South Australia (Bright & Vernik, 2004). Their architecture incorporates some of the findings from ODSI and distinguishes four layers:

1. **Enterprise model:** This layer supports rules, processes, and organizational models of roles and services in the enterprise.
2. **Coordination and control including interaction support:** On this layer, a shared and persistent event space of limited capacity,

Figure 5. SmartSpace middleware layered reference architecture



and an agent-based workspace infrastructure are offered.

3. **Enterprise bus:** This term refers to a communication layer based on the publish/subscribe paradigm.
4. **The service layer:** Here, the core functionality is represented by easily composable services. An *enterprise bus* is offered for services to communicate and cooperate; this bus connects so-called *peers* which host the services.

As for smart physical spaces, a prominent example is the reference architecture developed by the Gator Tech Smart House project of the University of Florida (see Figure 5). The reference architecture depicted is a more recent version of what was published by Helal, Mann, El-Zabadani, King, Kaddoura, and Jansen (2005) and is included by courtesy of the authors (the commercial version is called *Atlas* now). For more information, the reader may consult the group's Web Site at *www.icta.ufl.edu* or the Atlas Web site at www.pervasa.com. The architecture emphasizes sensors (plus actuators) and networked embedded devices at the lowest layer as the hardware foundation of UC applications. The OSGI standard is exploited for customizing and maintaining these sensors and embedded devices in a dedicated second layer. The third layer contains three large parts which reflect major insights into the nature of the UC world (note that these insights have a large influence on the present book, too):

- **The context management layer** reflects the importance of context-awareness for UC as a whole, as discussed in the preface of the book;
- **The service layer** reflects services (and service-oriented architectures, SOA) as the dominating paradigm for building autonomous software components in a UC setting;

- **The knowledge layer** reflects the fact that large-scale service composition cannot rely on standardized interfaces that are distributed prior to software (service) development; rather, service discovery and service interaction must rely on machine readable descriptions of the service semantics available at runtime;
- Due to the strictly service-oriented concept used, application development boils down to service composition; **the top layer** offers corresponding tools.

In conclusion, it should have become clear that both a component based and a layered reference architecture, if widely accepted, would be important steps from UC islands towards truly global UC. The reference architectures presented could serve as a basis for better communication among the UC protagonists and for the necessary standards.

REFERENCES

Aitenbichler, E., Kangasharju, J., & Mühlhäuser, M. (2004). Talking assistant headset: A smart digital identity for ubiquitous computing. *Advances in pervasive computing* (pp. 279-284). Austrian Computer Society.

Bond, A. (2001). *ODSI: Enterprise service co-ordination*. In *Proceedings of the 3rd International Symposium on Distributed Objects and Applications DOA'01* (pp. 156-164). IEEE Press.

Bright, D., & Vernik, R. (2004). *LiveSpaces: An interactive ubiquitous workspace architecture for the enterprise in embedded and ubiquitous computing*. Springer (pp. 982-993).

Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y., & Jansen, E. (2005, March). *The Gator Tech Smart House: A programmable pervasive space*. *IEEE Computer, 38*(3), 64-74.

Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *IEEE Personal Communications* (pp. 10-17). IEEE Press.

ADDITIONAL READING

Aarts, E., & Encarnaco J. L. (Eds.). (2006). *True visions. The emergence of ambient intelligenc.* Berlin, Germany: Springer.

Adelstein, F., Gupta, S. K. S. et al. (2004). *Fundamentals of mobile and pervasive computing.* New York: McGraw-Hill Professional Publishing.

Antoniou, G., & van Harmelen, F. (2004). *A semantic web primer.* Massachusetts: MIT Press.

Hansmann, U., Merk, L. et al. (2003). *Pervasive computing handbook. The mobile world.* Berlin, Germany: Springer.

Hedgepeth, W.O. (2006): *RFID metrics: Decision making tools for today's supply chains*. University of Alaska, Anchorage, USA

Helal, A.A., Haskell, B., Carter, J.L., Brice, R., Woelk, D., & Rusinkiewicz, M. (1999). *Any time, anywhere computing: Mobile computing concepts and technology*. Springer.

Huber, A. J. F. & Huber, J.F. (2002): *UMTS and mobile computing*. Artech House.

Jurafsky, D., & Martin, J. H. (2000). *Speech und language processing.* Upper Saddle River, NJ: Prentice Hall.

Lu, Y., Staff, L.Y., Zhang, Y., Yang, L.T., & Ning, H. (2008). *The Internet of things*. Taylor & Francis Group

McTear, M. F. (2004). *Spoken dialogue technolog.* London: Springer.

Moreville, P. (2005): *Ambient findability.* O'Reilly.

Riva, G., Vatalaro, F., Davide, F., & Alcañiz, M. (2005): *Ambient intelligence*. Amsterdam: IOS Press.

Sharp, H., Rogers, Y., Preece, J. (2002). *Interaction design: Beyond human-computer interaction*. J. Wiley & Sons.

Stajano, F. (2002). *Security for ubiquitous computing*. Cambridge: John Wiley & Sons, Ltd.

Weber, W., Rabaey, J. M., & Aarts, E. (Eds.). (2005). *Ambient intelligence*, Berlin, Germany: Springer.