Preface

ABSTRACT

The preface provides an introduction to and a definition of ubiquitous computing as a computer science field and relates it to the concept of real time enterprises. We describe the main challenges in ubiquitous computing and introduce the S.C.A.L.E. classification employed to organize the contents of the book. Finally, recommendations about using the book as a reference and teaching resource are given.

OUTLINE AND SUBJECT OF THIS BOOK

On the next couple of pages, we first want to provide an introduction to and a definition of ubiquitous computing (UC)—both as a scientific domain and as a technology area—and relate it to the concept of real time enterprises. We first define the scope of UC as a domain—in particular as covered in the present book. The question is raised whether UC is a research field in its own right; we also explain the required trade-off between breadth and depth of coverage concerning the contents of this book. The S.C.A.L.E. classification is introduced as probably the first attempt to provide a canonical structure and organization of the area. The present preface thus gives the reader a better idea about why this book is structured in a particular way and why it covers the range of topics selected by the editors. A "reader's digest" is provided, both as an overview of the chapters provided in this book and as a guide for readers with different backgrounds and interests.

So far, no single book exists which sufficiently covers ubiquitous computing in a holistic way. Many UC books restrict themselves to combinations of middleware, networking, and security. However, UC has lately extended beyond this focus and even beyond purely technical issues. In particular, understanding current developments in the field requires knowledge about pertinent algorithms and concepts in artificial intelligence and human-computer interaction. All-in-one reference books covering the foundations of ubiquitous computing and the areas mentioned above are missing; therefore, researchers, practitioners, and academics typically use collections of papers from the respective conferences and individual chapters from the books emphasizing a single area. This approach does not provide the target audience with a coherent view of the field. Also, the presentation of materials often presumes too much knowledge about the related topics.

As we will substantiate later, real time enterprises (RTE) are a key application area for UC. In fact, the authors of this book will show that RTE is *the* first application domain when it comes to the economic advantages of UC. Therefore, RTE can be considered a key driving force for large-scale deployment of UC technology. The last part of this book describes a selection of pilot projects and trend analyses concerning UC applied to RTE, provided by *SAP Research.* These chapters will strongly support the

above-mentioned arguments. Note that most UC concepts and technologies described in this book are not restricted to RTE. Rather, readers should consider RTE as the first widespread, commercially successful area - they should also consider our RTE related examples as an aid for a better understanding of the value and usage of UC.

DEFINING UC

According to the Oxford English Dictionary, the word *ubiquitous* has two meanings: the first meaning is *(seemingly) present, everywhere simultaneously,* and the second meaning is *often encountered.* The seminal work by Weiser (1991) introduced the term ubiquitous computing, stating that it:

"represents a powerful shift in computation, where people live, work, and play in a seamlessly interweaving computing environment. Ubiquitous computing postulates a world where people are surrounded by computing devices and a computing infrastructure that supports us in everything we do."

Well, then what is *ubiquitous computing?*

One approach to an answer is the rough division of computer science into three consecutive eras:

1. Era number one was that of *mainframe computers,* where one computer was used by many users (1:N)
2. Era number two is about to end: the era of *personal computers* (PC), where one computer was used (owned) by one user (1:1), and
3. The third, dawning era is one in which many computers surround a single user (N:1)—almost anytime and anywhere.

Based on this approach, ubiquitous computing is nothing else than the third era cited above, that is, it is equivalent to the "Post-PC era."

We all experience this dawning era and realize that it brings about a proliferation of computers, with desktop PCs, laptops, PDAs and cell phones just being examples. The anytime-anywhere availability of computers indicates a shift away from pure desktop—and thereby, "isolated, full-attention"—computing to mobile (or rather, nomadic)—and thereby "integrated, shared-attention" computing. The term *integrated* alludes to the fact that the new computers interact with—or are even perceived as "part of"—their environment; the attribute *shared-attention* emphasizes the fact that users do not devote themselves to "sessions at the computer," but rather conduct an activity during which they "happen to interact with a computer, too."

Accordingly, the computers that surround us fall into two categories:

- Some are *worn or carried* in the sense that they move along with us, and
- Some are *encountered* in the sense that they are either part of the environment of our respective whereabouts, or worn or carried again, namely by people whom we meet.

The first category covers devices denoted as wearables or portable devices, respectively. The second category of devices encountered in the environment shifts from traditional PCs—which were somewhat alien to the environment and general-purpose in nature—to computers that are perceived as an integral part of the environment and that are rather special-purpose in nature (see the next section for concrete examples). More precisely speaking, the boundaries between general-purpose computers (formerly: servers and PCs) and special-purpose computers (formerly: microcontrollers) become blurred. On the

one hand, computers become ever cheaper and can be dedicated to specific tasks. Power constraints and other resource limitations for portable and unattended computers increase this trend towards dedicated devices and favor "right-sizing." On the other hand, the required flexibility, adaptability, sophistication, and maintainability suggest devices that can be easily re-programmed. As a negative side effect, the latter trend introduces the curses of general-purpose computers—vulnerability (e.g., in the IT security sense) and limited reliability due to restricted maturity of devices with fast innovation cycles (cf., the fact that cell phones now have a far greater tendency to "crash," since they are based on full operating systems and software applications).

One is tempted to define UC simply as the era of portable and embedded specialized computers. However, embedded computers have existed for decades and have *already* become ubiquitous: they have become indispensable in washing machines and VCRs, and up to a hundred or even more of them are on duty in modern automobiles. *New* is the fact that these embedded devices are now:

- Networked, that is, cooperation enabled, Internet-enabled, and
- More flexible in terms of both maintenance/evolution and adaptation.

(Rather) new is also the insight that neither PCs—with their still unnatural and non-intuitive interfaces, interaction devices, and modes-of-operation (updates, re-boots, …)—nor embedded devices—with their increasing "featurism" (increasing sophistication that average users don't learn how to exploit)—scale well up to a world where hundreds of them would surround a user. Quantum leaps are required in terms of ease-of-use if ubiquitous computers are not to become a curse.

In summary, we can turn the rough definition "UC is the Post-PC era" into the following more elaborate one:

Ubiquitous computing is the dawning era of computing, in which individuals are surrounded by many networked, spontaneously yet tightly cooperating computers, some of them worn or carried, some of them encountered on the move, many of them serving dedicated purposes as part of physical objects, all of them used in an intuitive, hardly noticeable way with limited attention.

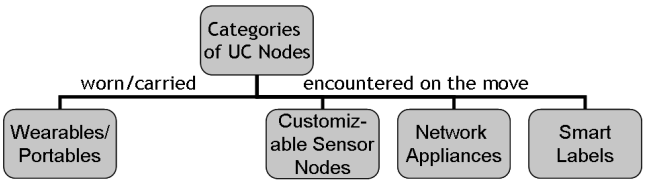
As a link to upcoming sections, readers should note that the definition given above buries two crucial issues:

1. Truly *integrative* cooperation: the path from mere connectivity of the "networked UC nodes" to true cooperation is long and arduous. Rapid deployment of all kind of networked sensors, appliances, labels, and so forth does not by itself lead to a meaningful "whole."
2. As to the last line of the definition above, it was already mentioned that a quantum leap in usability and dependability is required. We want to call this requirement a need for *humane* computing henceforth.

THE SIGNIFICANCE OF UC

The reader may want to get a feeling about the spatial and temporal significance of UC. By "spatial" we mean the spectrum from niche technologies to technologies that deeply influence an era. In this respect, it should soon become clear that UC is going to deeply mark our society over the years to come. By "temporal" we mean how visionary, that is, far off, UC actually is. In this respect, the quick answer, which we are going to substantiate further in the following, is as follows:

Figure 1. Overview of UC nodes



1. On one hand, *UC is already a reality* in the sense that the computer-based devices carried and encountered by users are already—and increasingly—networked.
2. On the other hand, *UC is still a big challenge* with respect to the required quantum leaps in *integrative* cooperation, as mentioned further previously, ("the whole must become way more than its parts") and *humane* behavior.

A few examples are cited for the first aspect, by providing a very selective list of four important categories of "UC nodes" in the global UC network:

1. **Wearables and portable devices** ("networked computers worn or carried" as mentioned in the UC definition), such as handhelds for warehouse picking, washable computer jackets, or companions like the so called *lovegetties* introduced in Japan almost ten years ago. These pocket size devices store their user's profile, that is, dating interests, and beep or vibrate when a "compatible" person—carrying a lovegetty—appears. In contrast to portable device, the term *wearable* denotes a degree of integration with a piece of clothing or clothing accessory that goes beyond that of mobile computing devices, up to a degree where the "computer nature" is hardly noticed by the user. MIT is known for influential research in the field (www.media.mit.edu/wearables). The examples given illustrate that this category of UC nodes is far larger than a simple extrapolation from the common representatives, that is, PDAs, cell phones, MP3 players, laptops, and so forth. Another example, namely body sensors, illustrates the possible seamless integration of "UC nodes worn or carried" and "UC nodes encountered." Body sensors are useful, for example, for activity monitoring (a basis for better context-aware applications) and health condition monitoring.

The last three of these categories are refinements of the "networked computers encountered on the move" from the definition of UC given in the last section (with exceptions like body sensors).

2. **Customizable sensor nodes** like the so-called *motes* developed by UC Berkeley and Intel. As opposed to traditional sensors, these nodes come with a fully programmable microprocessor and micro operating system (called *TinyOS* for *motes*), a variety of sensor options, and low energy networking. Mass production of these customizable nodes is intended to drive down cost such that easy customization and easy assembly of nodes into *sensor networks* are set to simplify application development. The Intel-funded company Crossbow (www.xbow.com) is commercializing motes. Companies like the German start-up *Particle Computer* (www.particle-computer.de) support the IEEE sensor network standard 802.15.4 known as *ZigBee* (www.zigbee.org). In 2003, UC Berkeley built the first single-chip successor to motes nicknamed *Spec*. They represent a major step towards almost invisible sensor networks with large quantities of nodes, often coined as *smart dust*.
3. **Networked appliances,** also called *Internet appliances* or *smart appliances*, which are mainly perceived by their users as tools, machines, devices, furniture, and so forth, rather than computers.

Apart from ever cheaper and more powerful embedded computers, affordable and energy-aware wireless technology is a key enabler not only for hand-held appliances, but also for fixed installations—which are much easier to deploy if network cabling can be spared on and if embedded "clients" can be easily combined with back office servers or gateways inaccessible to the public. One example is *smart vending machines*: the company USA technology (www.usatech.com) developed solutions for supporting online transactions (credit card payment, etc.) and for transmitting various information (fill status or out-of-stock events, out-of-change events, defects, customer behaviour patterns, etc.) to the operating agency. One step further, vending of physical and of digital goods start to converge: Coke vending machines are combined with vending of cell phone credits, ring tones, MP3 music, and so forth.

4. **Smart labels**, which identify physical objects and creatures vis-à-vis an IT-based system. Radio frequency identifiers *(RFIDs)* represent one important technology; *active badges* (e.g., for employees) denote a possible use case. Such labels may be thought of as Web addresses that serve as a "link" to data describing details about the identified object or person. Therefore, their on-board storage and processing capabilities can be very limited, even non-existent. In the application domain of *product identification*, RFIDs are set to complement (and later, replace) barcodes. Relevant barcode standards include the Universal Product Code UPC and (as a superset) the European Article Number EAN. For RFIDs, both are intended to be replaced by EPCglobal's Electronic Product Code EPC (www.epcglobalinc.org), which may contain a serial number in addition to the article number.

These few categories of ubiquitous computing nodes illustrate that specialized networked computers—most of them "hidden" in physical objects—are indeed penetrating the world.

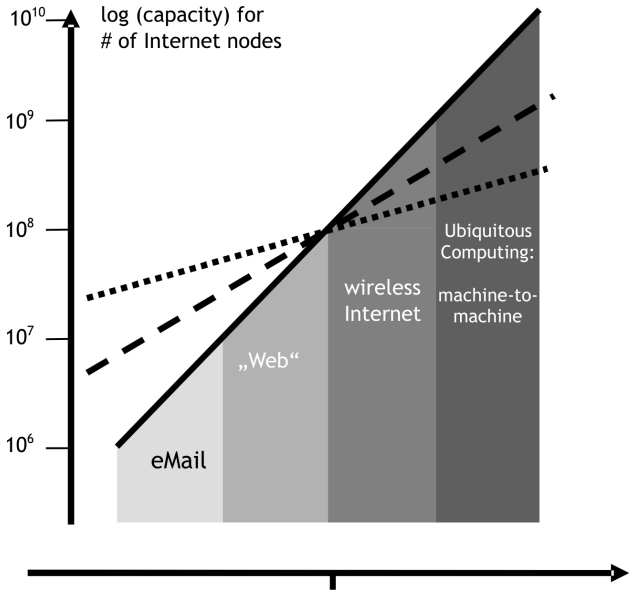
Machine-to-Machine Communication

The example categories cited above illustrate yet another phenomenon of key importance: the advent of large volume machine-to-machine communication. Throughout the history of the Internet, its exponential growth has been boosted by communication with or among people. This development contradicted what protagonists of distributed parallel processing had envisioned, namely a boosting importance of distributed systems as replacements for parallel and super computers. While this use case for distributed systems—and thereby for the Internet—continues to play a non-negligible role (cf., recent advancements in Grid computing), it has never been, and probably will never be a reason for "exploding numbers of Internet nodes." Three "waves of Internet usages" have driven and are driving this explosion to date:

1. "E-mail" as a means for people-to-people communication, the major reason for the Internet to grow up to the order of 10 million nodes.
2. "The Web" as a major use case of people-to-machine communication. Under its increasing importance, the Internet passed the 100 million node mark
3. "The wireless Internet," which—after flops like WAP-based cell phones—currently drives the convergence of the Internet with cell phones (cf., UMTS in Korea), MP3 players, and other portable devices. This wave contributes heavily to the hit of 1 billion Internet nodes around 2008.

As we approach the order of magnitude of the world population, one might expect the Internet (number of nodes) to enter a phase of sub-exponential growth. Yet the predicted shift from people-to-machine to machine-to-machine communication is likely to lead to a continuation of the exponential growth: sensor

Figure 2. Growth rates: Internet vs. CPU power



networks, networked appliances, and smart labels communicate autonomously and forward real-world events through the net.

A second important effect is illustrated in *Figure 2*: the growth rate of the Internet has constantly exceeded that of CPU power—measured, for example, in terms of the time that it takes for key indicators to double, like the number of Internet nodes or the processor speed (18 to 20 months for the latter one according to "Moore's Law").

This difference in growth rate has consequences for the Internet: the *relative* cost of distributed versus local processing decreases. Note, however, that this is only true for the wired Internet, for which the aggregated throughput per second (at least nationwide for the U.S.) and the typical bandwidth are all growing at roughly the same pace as the number of Internet nodes. Two reasons suggest more conservative predictions for wireless nodes: on one hand, there is no long term indication yet that the typical bandwidth will keep pace with the other Internet growth indicators mentioned above. Despite recent boosts in WLAN bandwidth, physical limits and frequency scarcity remain tough obstacles. But even if bandwidth keeps pace, wireless nodes are set to become more and more independent from the power line—and for mass deployment use cases like sensor nodes, even from battery replacement. A question mark must be placed here at least until revolutionary battery technology hits a breakthrough.

In summary, the reader should now understand that UC is the *key technology* that will deeply influence our society for three reasons:

1. UC describes *the* next era of computing. Since we live in the information (i.e., computer) society, the influence will be at least as pervasive as that of computer today. As a side remark, virtually every domain of computer science or IT is potentially impacted. This makes it difficult to be selective and concise for the present book—but not impossible, as we will show.
2. UC has potential impact on every facet of our lives. Computing is no longer "what we do when we sit at the computer" nor "what is encapsulated/hidden deep inside VCRs, and so forth."
3. UC is inevitable and "impossible" at the same time: the components are already developed and massively deployed, consult the four categories of UC nodes described. Since UC use cases are

becoming increasingly profitable, for example, the replacement of barcodes with RFIDs, the industry will push the use of UC technology. Nevertheless, the two top-level challenges remain "integrative cooperation" and "humane computing." Both must be solved in order for UC to become the envisioned helpful anytime-anywhere technology and not a nightmare.

THE CHALLENGES OF UC

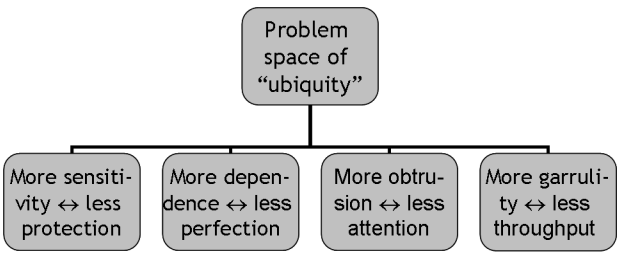
As described in the last sentences above, UC is inevitable and probably very influential, even able to change our society—and it buries unresolved problems. Therefore, UC can be considered one of the biggest challenges of our times—an insight that led to the present book, for instance.

Conflicting Developments Due to Ubiquity of Networked Computers

The above sentences have also pointed out two top-level challenges in UC: "integrative cooperation" and "humane computing." Before we try to describe and define these challenges in more detail, as a basis for the structure and content of this book, we want to mention one more viewpoint of what has been already discussed: we will try to describe the upcoming "ubiquity" of networked computers as a problem space that is becoming increasingly challenging due to conflicting developments on four levels (see Figure 3):

1. **More sensitivity ↔ less protection:** As UC penetrates more and more areas of daily life and work life, more and more sensitive data, but also processes and activities, of privacy critical and liability critical nature depend on computers. As a conflicting development, the established IT security solutions are not fully viable in machine-to-machine scenarios, in particular if a subset of these machines acts autonomously as substitutes for humans or organizations. Further aggravation of the situation is due to the ever widening gap between the "in principle" availability of appropriate security measures and their acceptance, that is, application by users, even more so since IT related concepts—for example, of trust—do not easily match the concepts of trust that users are acquainted with from their real-life experience. Obviously, IT security research—in a broad sense, including trust models and other issues—must be stressed and focused on UC.
2. **More dependence ↔ less perfection:** if many actions and aspects of daily life are supported and controlled by ubiquitous computers, then these actions and aspects (and in turn the average human) become dependent on these computers. Obviously, this dependence becomes as ubiquitous as the computers, and is not restricted to desktop work anymore. As an increasingly critical source of

Figure 3. Overview of conflicting developments



conflict, ubiquitous computing relies on cheap components and close to zero maintenance, a context in which the failure of nodes becomes a regular event as opposed to an exception. Furthermore, overall system dependability declines rapidly with the number of components involved under the assumption that the per-component dependability (and therefore, failure rate) remain constant and that all considered components contribute a crucial portion to the system. Obviously, research on system dependability must emphasize "over-provisioning" approaches where, for instance, a considerable number of components can fail without harmful consequences for system dependability. Complex biological and social systems exhibit this property and can serve as examples.

3. **More obtrusion ↔ less attention:** users are expected to be surrounded by ubiquitous networked computers for many purposes. In principle, each additional computer means an additional user interface. As a source of conflict, the move from desktop use to everyday use means that users have to share attention to computers with attention to their daily lives (human communication, driving, manual work, etc.). Altogether, this issue suggests a dramatic focus on "humane computing" as has been stated several times already. We will see below that such research must be understood in a broader sense than classical human-computer interaction (HCI) research.
4. **More garrulity ↔ less throughput:** the market for sensors is predicted to explode and the replacement of barcodes by RFIDs will lead to zillions of wireless nodes (in the longer term, after intermediate phases where only containers and expensive goods will be tagged). These two effects alone show how dramatically the number of wireless data paths around us will grow—and feed the wired network in turn. One may object to calling this development dramatic, since machine-to-machine communication can often be restricted to short binary messages, whereas computer-to-human communication tends to involve verbose data, media streams in particular. However, increasing performance of computer vision and voice recognition will increase the bandwidth famine on the sensor-to-backend path. Again, one may object that computer vision may be carried out locally and that only reduced information will be sent (intermediate results, i.e., recognized "features" of the scene observed, or the final result, i.e., a "semantic" description of what was recognized). However, computer vision is compute intensive and therefore not likely to be executed entirely on the sensor side. Increased sensor-to-backend bandwidth demands can therefore be expected at least for this kind of sensor.

The critical factor in this fourth dimension is the fact that sensor nodes must be built under extreme energy constraints, such as a grain size battery determining the lifetime of the entire sensor. Since wireless transmission tends to be the most energy hungry part of a sensor and since this hunger increases considerably as bandwidth is increased, we can state that the throughput of many UC nodes will remain far below that of an average PC even today. Similar arguments apply to smart labels: the widely used RFID tags are passive, that is, receive their transmitting energy from the reader (receiver). Portable readers are crucial for many application scenarios, and the required energy is a function of both bandwidth and distance—yet increased reading distance is supposed to be a key advantage over barcodes. All in all, there is a large demand for advancements not only in the energy/bandwidth tradeoff (and in battery capacity, of course, which we consider a non-computer science issue), but also in distributed architectures and distributed algorithms that optimize and trade off minimal resources at sensor nodes (and other UC nodes) and minimal throughput demands.

Integrative Cooperation and Humane Computing

The preceding observations provide further insight into the challenges of UC, which we described as "integrative cooperation" and "humane computing" on the top level. We now want to go one level deeper and divide each of these two challenges into two or three sub-issues.

As to "integrative cooperation," we want to distinguish two issues:

1. **Scalability:** We discussed several times that UC furthers the explosion of the Internet and that even many local networks (such as smart dust and RFID-tagged products in a warehouse) rather represent big assemblies of nodes (in particular in relation to the network bandwidth to be expected). This means that highly scalable solutions are required. The history of the Internet (especially in comparison to less successful networks) is a history of extreme scalability, and all standard textbooks about the Internet state that scalability was and is a key success factor. This applies even more to ubiquitous computing as the future nature of IT systems, depending on the Internet. Note that scalability is to be understood in a broad sense, as will be described further below.
2. **Connectivity:** Again, this term is a brief notion for a broad issue. We have identified scalability as a broad issue in its own right, but everything else that relates to the need to integrate UC nodes and to make them cooperate will be summarized under *connectivity.* In particular, tiny ("dumb," or at least special-purpose) nodes must be integrated into a meaningful whole based on spontaneous discovery and interaction. New, that is unforeseen, components or components that vary over time or usage context must be able to enter into meaningful cooperation with existing nodes. We will see later in the book that such connectivity issues are addressed and solved somewhat differently for different kinds of UC components today; for example, differently for software services than for smart labels. Various chapters of the book will reflect these differences that exist today. When it comes to the vision of "intelligent behavior" of the integrated, cooperating whole, we must deal with approaches that are also related to the scalability issue and to the ease-of-use issues treated below. This indicates that our classification (as always) is good for understanding the topic space, but not "ideal" in the sense that any issue could be classified as belonging to exactly one category (an issue that any taxonomy has to live with).

The residual three issues represent sub-challenges of "humane computing":

3. **Adaptability:** A key to dramatically reduced cognitive load (required in order to counter the "more obtrusiveness, less attention" dilemma described) is found in highly adaptive systems that interact "optimally"—with respect to the use case, user, situation, device or resource constraints, and so forth. A great deal of research in this respect has been carried out lately under the term "context-aware computing;" many of the respective activities tried to adapt software and particularly user interfaces to data retrieved from sensors with the users' locations being a number one context category. Other context types, like activities carried out by the user or data "hidden" in their files, are also important, but their integration into UC solutions is less well understood. Even less emphasized and understood today is adaptation to the users' state of knowledge.
4. **Liability:** As discussed with the issue "more sensitivity, less protection," IT security issues must be revisited under UC requirements and constraints. Moreover, the use of UC technology in everyday life makes UC-based physical and digital components an integral part of our society—and consequently of our economy. A majority of UC components or services will not be available for free. Even if they are free of charge to the end-user, someone will have to pay for their development

and execution. This means that UC services will have to respond to a number of "market rules," for instance: (i) users will want to make sure that "they get what they paid for," (ii) providers will want to make sure that "they get paid for what they provide," and (iii) users will want to be able to truly compare offerings based on prices and a variety of parameters describing the associated "value" of a UC service. Providers will want to be able to offer their "values" to users appropriately, and so forth. Since all these issues are tightly linked to financial, and thereby to legal issues, they are closely intertwined with the security issues mentioned above; altogether, we will summarize the respective issues as "liability."

5. **Ease-of-use:** Under this term, we want to emphasize HCI aspects in the more narrow sense. Maybe the single most important change brought about by UC in this respect is the proliferation of interaction devices and, thereby, modalities. In order to understand this issue, one has to note that throughout the history of computer science, software engineers and programmers were exposed to a single major interaction paradigm at any given time. It started with "holes in paper," that is, punch tapes and punch cards. Later teletype like devices came, followed by full screen terminals like the IBM 3270 and the DEC VT100, both being de facto industry standards. Finally, more than 15 years ago, Windows-based UIs and the WIMPS metaphor (windows, icons, menus, pointers, scrollbars) were introduced. In the UC world of special-purpose nodes, there is no longer any single dominating interaction paradigm: cooperative-use devices like interactive walls, speech-and-graphics devices, soft-key-based cell phones, physical-interaction appliances, and many more must be coped with. Accordingly, multimodal user interfaces—supposed to cater for this variety—are becoming crucial. At the same time, the "classical" graphical (or rather, hands-and-eyes) interfaces are currently experiencing rapid innovation and "voice" is recognized as a most natural way of interaction, in particular for users whose hands and eyes are busy with "real world activities." These and other issues must be addressed under "ease-of-use" in a UC related book.

The five sub-challenges treated above—scalability, connectivity, adaptability, liability, and ease-of-use—can be abbreviated by concatenating the first letter of each sub-challenge, leading to S.C.A.L.E. We will describe below how they represent the backbone structure for the present book—and a suggested structure for the emerging discipline.

THE FOCUS ON REAL TIME ENTERPRISES

This section addresses two questions: "What is a real time enterprise?" And, "Why and how does the present book emphasize this topic?"

First of all, in preparation of a "definition," one must understand that in many domains software has become more and more sophisticated over the years. Whatever domain is supported—it is represented as a machine-readable "digital model" as part of the software. For instance, *virtual reality* applications for the automotive industry, such as "digital mockups," contain not only three-dimensional representations of a yet-to-be-built car but also the materials' characteristics, laws of physics used to model the dynamics—to a degree that supports "virtual crash tests"—and much more. The same applies to enterprise application integration (EAI) software (at the core of real time enterprise approaches): almost the entire enterprise is "digitally modeled," including human resources, workflows, a whole spectrum of financial issues, product lifecycle management, customer relationships, and much more.

Despite the increasing sophistication of "digital enterprise models," the gap between this digital model and the "real enterprise" remains hard to bridge: events and changes in the real enterprise (goods arriv-

ing, stock being manipulated, products being manufactured, etc.) must be reflected in the digital model, and changes and decisions made in the digital model (work orders, production plans, route slips, etc.) must be forwarded to the "real enterprise." The gap is dominated today by *switches in media* (e.g., from machine-readable data to printed paper or vice versa) and "*analog transmission,"* mainly by means of humans and, again, paper as the "media."

This brings about three major "stress points:"

- **Medium cost:** The use of humans as media (i.e., human "manual" intervention) as well as the handling of large amounts of paper represent major cost factors in an enterprise. For instance, paper-based time sheets may be filled using pencils and turned into machine-readable formats afterwards. Even if this cycle is reduced, that is, timesheets are filled at a PDA or laptop, the manual task remains. Only UC technology offers novel ways of entirely avoiding manual entries.
- **Time lag:** The time required for printing and sending paper or for human-based transmission of information causes delays that may represent a major reason for revenue loss (e.g., due to a longer delay between raw material cost incurrence and end product revenue receipt).
- **Inaccuracy (errors, lacking and too coarse-grained information):** Manual processes are error prone. For instance, inventories are regularly made in order to match the "digital" stock data with the "real" one. Misplacements (by customers or employees) and theft are examples of lacking information here, and mistakes in the "analog transmission" are the major source of errors. Both errors and lacking information account for expensive differences between the real and the digital world (cf., the need for more conservative stock planning or missed business due to products out of stock). Moreover, the cost of manual (human) "data elicitation" (counting of goods in stock, provision of time sheets) is prohibitive for more fine-grained updates between real and digital models. For instance, workflow optimization might be achieved with more accurate and more fine-grained data, but the expected benefit may not justify the cost of fine-grained manual acquisition.

Obviously, electronic, that is, automated, ways of bridging the real-to-digital gap have a lot of potential in the business world. Many steps towards this goal have already been taken: a major step was the introduction of barcodes—not only on consumer products, but also on containers, temporarily handled goods like airport baggage, and so forth. Other approaches, like the introduction of handhelds, networked control of production machinery, and paperless business-to-business communication, are still emerging.

While such steps represent a certain degree of real-time behavior, the advancements possible by means of UC technology can be dramatically more far-reaching and more complete (with respect to getting rid of manual/analog gaps). Therefore, the application of UC technology as a means for advancing real time enterprises is extremely promising and buries a high economic potential.

With these explanations in mind, the reader may understand why we consider a definition found on a Microsoft Web site insufficient (http://www.microsoft.com/dynamics/businessneeds/realtime_enterprise.mspx, last retrieved on January 18, 2007): "*A Real Time Enterprise is an organization that leverages technology to reduce the gap between when data is recorded in a system and when it is available for information processing*." This definition focuses only on "time loss" as an optimization dimension and ignores "medium cost" and "inaccuracy" (a deficit probably due to the term "real-time" which buries a temptation to look at the time aspect only—yet a term which we keep using since it is widespread). Moreover, the definition above focuses on the "*inbound"* channel to the digital model—we regard the "*outbound"* channel as equally important: consider, for example, the digital model suggesting spontaneous changes to workflows and parts applied at a production plant, but the realization of these changes

taking too much time and too much effort to cross the digital-to-real gap. In such a case, an electronic and automated transmission would make a huge difference (faster, more accurate forwarding of decisions might be achieved using the most appropriate media, such as computer-generated, detailed individual voice instructions sent to the respective employees' ears).

We will therefore use the following "definition" :

A real time enterprise is an organization that bridges the gap between digital enterprise models and the corresponding real world largely by means of ubiquitous computing technology—in both directions—in order to reduce time lags, medium cost, and inaccuracy.

Now that we have answered the question, "What is a real time enterprise?," we turn to the question "Why and how does the present book emphasize this topic?"

The rationale for emphasizing this application domain is simple: the paragraphs above provide hints about the very considerable and measurable economic potential of UC technology for real time enterprises. The editors have reasons to believe that real time enterprises presently represent the most profitable domain for UC technology. Other areas, such as smart homes or smart travel support, exhibit cost/benefit ratios worse than that. Therefore, the large-scale deployment of UC technology in the domain of real time enterprises will precede deployment in other areas. In turn, many innovations can be expected to lead to a breakthrough in this area, too.

The present book is divided into six parts, the last one of which is dedicated to pilot projects and trends. All short chapters included there treat the use of UC technology in real time enterprises. The pilots were carried out since participating industrial partners had successfully evaluated the related business opportunities; this backs the editors' decision substantially. Looking at *how* the present book emphasizes the topic *real time enterprise*, the core of the answer lies in the last book part: we will look at the most important use cases from this application domain and particular the aspects of UC technology in the real time enterprise context. However, apart from this, we will treat UC technology independent of application domains, such that the book will be useful for the readers interested in either the core technology or its application to any other domain.

THE S.C.A.L.E. CLASSIFICATION USED IN THIS BOOK

UC is the hot topic in computer science, as it describes where computer science as a whole is heading. Lectures and books about UC have outstanding opportunities, since they address the future of computing—but also considerable threats, which the editors face with their carefully devised approach. A major threat is to perceive UC solely as modern distributed computing and not as a vision that needs several major problem domains to be addressed simultaneously—issues of distributed computing, HCI, IT security, and so forth. Since UC draws from and influences many areas of classical computer science, the real challenge is to cover all fundamentals, that is, to avoid being one-sided, without getting lost in a "big world" and remaining shallow. The editors approach this challenge by organizing the book and the pertinent issues into five interrelated categories, which are systematically ordered and addressed under one umbrella:

1. **Scalability:** Solutions for huge networks/applications and support for global spontaneous interoperability;

2. **Connectivity:** Intelligent ad hoc cooperation of a multitude of (specialized) devices over unreliable wireless networks;
3. **Adaptability:** Adaptation to users and context-aware computing;
4. **Liability:** Novel problems in IT security like privacy/traceability tradeoffs, human-understandable trust models, legal binding of service users and providers, and so forth**;**
5. **Ease-of-use:***User-friendliness* in human-computer interaction as a major concern

Another threat is the temptation to draw artificial lines between UC, pervasive computing, and ambient intelligence. Instead, these three terms are considered synonyms, an attitude that is backed by an extremely high degree of overlap in the concrete research agendas of research groups and labs that carry these different names. We will elaborate on the roots and meanings of these three terms further.

From a truly holistic point of view, UC is a multidisciplinary issue that concerns both social sciences and economics in addition to computer science. However, the five categories described above show that an all-encompassing treatment of the computer science issues in UC is already pushing the format of a serious reference book to its limits. Therefore, the present book does not include additional social sciences chapters, but keeps the contents within the limits of computer science, while emphasizing human-centric approaches within these limits. This decision is supported by the fact that several books about the social sciences aspects of UC already exist.

The "Preface," as well as the following "Introduction to Ubiquitous Computing," provides an overview of the history and visions of UC. They explicate the overall structure and the philosophy of the field required for understanding and relating the following in-depth chapters grouped into the five categories as stated above (S.C.A.L.E.) and a final part providing concise descriptions of the pilot projects in UC realized at SAP Research.

As a research field, ubiquitous computing touches almost every area of computer science. More than that, it is closely interwoven with many of them. Then, the question might arise: is it a research and teaching subject in its own right? We believe the answer to this question is "yes." As an analogy, we can look at the area of distributed systems that was at a comparable stage some twenty years ago. The area of distributed systems, too, involves a lot of issues from different computer science fields, coined as distributed simulation, distributed programming, distributed algorithms, distributed databases, distributed artificial intelligence, and so on. For some time, it seemed as if almost every branch of computer science would "go distributed." Today, the area of distributed systems has settled as a teaching and research subject in computer science in its own right. However, the boundaries with many other disciplines remain blurred; for example, distributed software engineering is taught in both software engineering and distributed systems. The analogy shows what the editors expect to happen with ubiquitous computing as a domain "in its own right." For the time being, ubiquitous computing is still an actively evolving topic, that is, not as well established as distributed systems yet.

According to the above analogy, we will have to cover topics in UC that reach out to a number of sub-disciplines within computer science and even beyond. Since different readers will have a background in different sub-disciplines, how can we treat the subjects without having to cover most of computer science, but also without "losing" our readers? In other words, the question arises: how can we achieve the most optimal trade-off between breadth and depth in the coverage of relevant UC topics? The only viable approach here is an attempt to provide the required background knowledge as briefly as possible to our readers, still leaving enough room to covering the UC specific issues. This was one of the major challenges for the contributing authors.

A second and related issue is the fact that UC as a field has never been organized into a canonical discipline before. Therefore, no "pattern" exists for structuring this research area in a concise way that

would help authors and readers to form mindmaps of the discipline and organize the wealth of research issues and related concepts and approaches. We consider the present book as the first one to introduce such a canonical structure: the S.C.A.L.E. classification that was already discussed above. With the help of this taxonomy, we can first organize the problem space of ubiquitous computing in a set of distinct areas. Second, we discuss the contents of particular areas and emphasize some pertinent challenges within each of them. Often, such challenges either represent a truly new aspect in ubiquitous computing or well-known issues, which become particularly relevant in the context of ubiquitous computing.

We expect that a "normal" background in computer science would be sufficient to understand the most of the discussions in this book. Any additional background is hardly needed. However, readers will have to go deeper into particular topics at some places and learn more about the methods and algorithms of general computer science as their application in the context of UC is explained.

The readers might ask the question if this book is going to cover the entire field of ubiquitous computing. The answer to this question is really no, as the field is too broad to be covered completely within the given space limitations. However, this book is more complete than any other reference book on UC known to the editors. It covers not only the issues of networking and mobile computing, but also those of scalability, security and a wide range of topics in human-computer interaction. The editors have made every effort to make the chapters very comprehensive, comprehensible and to provide additional references to advanced research topics related to individual chapters. We also believe that using the book in UC courses will be greatly facilitated due to the organization of materials into distinct areas of the S.C.A.L.E. classification.

As UC is a rather new and rapidly evolving subject, we first tried to organize the problem space into coherent chunks of knowledge. This is approached in two different ways. On one hand, the S.C.A.L.E. classification was devised in order to provide a holistic view of the research areas pertinent to ubiquitous computing. This classification is employed throughout the book for structuring purposes and represents a major "guide" through the remaining book chapters. On the other hand, we attempt to provide a holistic view of the global UC network (or system) by introducing and explaining a reference architecture for designing ubiquitous computing systems. For this purpose, we picked out one particular example of a reference architecture for ubiquitous computing called Mundo, which was used as the design rationale for the MundoCore middleware (Aitenbichler, Kangasharju, & Mühlhäuser, 2006). As a reference architecture, Mundo can be thought of as a rough floorplan for an entire UC system, which helps the readers to understand and organize the field of ubiquitous computing. It can be used to show the differences in the definitions of distributed systems and ubiquitous computing as a field. Those are supposed to be approximated by the differences in their respective reference architectures.

In the following, we describe a set of challenges in ubiquitous computing as defined by the S.C.A.L.E. classification.

Part **S** stands for SCALABILITY in UC systems. It addresses two main issues: (i) Existing UC systems are typically limited to a certain number of components. The question arises how to scale the system to support the cooperation between zillions of components in open and growing ambient environments; (ii) Another aspect is the support of nomadic users around the globe as opposed to a single user interacting with the UC system.

Part **C** stands for CONNECTIVITY in UC systems. This area tries to provide answers to the questions, such as how to easily connect zillions of cooperating components. Here, several levels of abstraction are possible: (i) Wireless networks are a blessing and a curse at the same time: a blessing, due to the unique capabilities of data transmission without hardwiring the networks, a curse, due to the unreliable nature of the connection technologies existing today, posing many challenges for the operation of UC systems. (ii) Still, most issues in connectivity go definitely beyond the wired or wireless nature of the

connection and significantly overlap with the issues in scalability. The questions, such as how to find and understand the peers on the network, how to enable zero configuration, and finally how to design networks for zillions of connections avoiding the bottlenecks of the architectures based on a central server belong to such topics.

Part **A** stands for ADAPTABILITY. This issue is crucial as UC systems are employed by people not only at their leisure, but also during their daily work. Therefore the users may be surrounded by hundreds of computational components, but do not want those components to significantly interfere with their working routines. Such users need as few interactions with the UC system as possible. One major approach to achieve this goal is context-aware computing. Context-awareness is a very important mechanism. It allows the design of a system in such a way that the number of automated tasks is maximized and the number of options that the user has to select explicitly is reduced to the minimum. Beyond that, adaptability means adapting to a particular user interacting with the system on the fly. It involves methods of acquiring and using the data about the user of a UC system. As to adaptability vs. adaptivity, the former comprises both the passive capability of "being adapted" and the active capability of "adapting oneself," while the latter is often associated with the passive variant only.

Part **L** stands for LIABILITY. As the term itself indicates, we must go beyond today's IT security solutions in ubiquitous computing. It should be noted that, while the goals of liability remain the same as the goals of security, special solutions need to be found. Such solutions should: (i) scale, that is, not depend on centralized components, and (ii) be human-centric, for example, flexibly consider conflicting goals, such as privacy and traceability, and related goals, such as dependability.

Part **E** stands for EASE-OF-USE. As stated above, adaptability has to ensure that the amount of interactions between the user and a UC system remain minimal; in contrast to this, ease-of-use means that interactions are optimal, which is related to but not the same as minimal. Optimizing the interaction means, for example, that the modalities are dynamically selected and combined and specific modalities are introduced to meet the requirements of a specific situation. A crucial issue in designing an optimal human-computer interface is understanding the natural input, that is, the ability of the system to derive the meaning of the input and represent it formally for further processing by the system. Mapping the input to such a semantic representation, for example, interpreting the natural language input of the user is a form of computational intelligence. It is currently better studied in limited domains, such as airplane ticket reservation. Scaling this kind of intelligence to unrestricted domains is rather an open research issue.

In the following, we describe each of the **S.C.A.L.E.** components in more detail. This will lay the foundations for understanding our recommendations on how to use the book in ubiquitous computing courses, as a reference, or for self-study.

Part S: Scalability. We consider *scalability* to be a top priority challenge in ubiquitous computing, which is also reflected in the first place it occupies in the acronym S.C.A.L.E. There are several dimensions of scalability that have to be considered in the context of ubiquitous computing for real time enterprises. In this book, we concentrate the discussion along two such dimensions, the technical scalability and the economic scalability.

The *technical scalability* leads to (potential) cooperation of "zillions" of devices. Thus, solutions have to be found that work efficiently with zillions of components. The most relevant areas for this, which are basically alternatives for addressing technical scalability, are:

- *Bionics*, that is, bio-analog computing, including the topics such as (i) neural networks and cooperating robots, which are only of marginal importance for ubiquitous computing, (ii) ant colonies, which are often simulated or executed on a single computer today, swarms and autonomous computing, and (iii) brain-like modeling of the human memory;

- *Event-based computing*, a more conventional paradigm of addressing the technical scalability in UC settings. It is fairly widespread; therefore, we will defer the discussion of it to the section on connectivity.

Economic scalability addresses, in the first place, the issues of global interoperability. It is attached to humans as the components encountered need to cooperate globally. One possible solution to this might be the introduction of standards. However, global standards will be insufficient in many cases and it is often unrealistic to assume that all parties participating in communication will adhere to them. Moreover, interface definition languages, such as remote methods or procedure calls, and so forth, only define the syntax, that is, the formal structure of expressions. Examples of this are typing, names of operations or operands exported in a particular interface. Such languages do not define, for instance, valid call sequences, preconditions, the cost or the quality of the operation and similar things. Though there exist partial solutions to these issues, it is still unclear how to specify the semantics of processes on a broad scale; for example, how to encode formally, that is, in a machine understandable manner, what an operation actually *performs*.

The most relevant areas providing at least partial solutions to the previously described challenges are:

- Web services and business processes, describing service-oriented architectures (SOA) as a promising paradigm to support large-scale UC environments in the business world;
- Ontologies as a key technology to formally capture the meaning of the processes. This is a holy grail for making UC components or services understand each other, giving the hope for a machine to "know" what the specific operation performs;
- Service discovery, taking into account service level agreements, the quality of service in the business sense of "get what you pay for," and further issues.

Part C: Connectivity. The issue of global interconnection of UC components is closely related to scalability. Two important issues that lead from networked computers to truly cooperating nodes are treated in the scalability part, but are related to connectivity: (i) previously unknown components must be able to join a cooperating system, and (ii) a high degree of autonomy of the nodes must be combined with a close cooperation that leads to a "meaningful whole." In the "Connectivity" part of the book, we will discuss further issues of scalable communication infrastructures and scalable cooperation infrastructures, such as spontaneous—that is, ad hoc—communication that must be possible without human intervention, such as configuration. More details follow below.

Scalable communication infrastructures cover wireless networks, which are often a pre-requisite for higher layers, and require the basic understanding of the pertinent technologies such as ZigBee or WiMax. Furthermore, scalable communication infrastructures involve event-based communication praised as the UC approach to connectivity. They operate according to the so-called "push" paradigm, which is an important pre-requisite for the scaleable open cooperation of components superseding the client/server architectural paradigm. Thus, such communication infrastructures constitute a significant contribution to the scalability of ubiquitous computing systems. It may very well be the case in the future that event-based communication will become superseded by the approaches inspired through bionics or socionics. As long as this remains an open research issue, event-based communication certainly represents the best choice. Issues such as advertising of services, the openness of component cooperation and the integration of other paradigms therefore have to be addressed. Appropriate UC middleware should be described including available services and tools.

Scalable cooperation infrastructures focus on issues beyond getting the bits across, such as:

- *Overlay networks,* which are special sub-networks on the Internet that can be classified by at least 3 classes: (i) peer-to-peer networks avoiding centralized bottlenecks and scaling fairly well, (ii) opportunistic networks, trying to correlate the proximity of communication nodes with the proximity of humans, and (iii) smart item networks, AutoID and beyond, representing the networks of "machines;"
- *Service discovery:* a prerequisite for zero configuration, that is, fully automated configuration;
- *Federations* where individual components must be integrated into ambient environments composed of many cooperating components; thereby, many components may be involved, for example, in the interaction of a user with a service or environment.

Scalable cooperation infrastructures may sound like a solution to "economic scalability" already mentioned above, but actually they are not. Many assumptions about the components are made, which really turns this into a connectivity issue on the whole.

Part A: Adaptability. The capability of a UC system to dynamically adapt its behavior according to the state of the environment is structured along two main dimensions: context awareness and user awareness. *Context awareness* is a term that means the adaptation of a system to the situation of use. The situation of use can thus be characterized by the different types of context, such as sensed context, modeled context and inferred context. Sensed context is represented by the data acquired with the help of various sensors. Some examples of measurements performed with the help of sensors are, for example, temperature, shock, location, and so forth. Modeled context can be obtained from other software or databases, where it must have been previously specified, for example, models of tasks or activities fall into this category. Finally, the context can be inferred, which means that some inferences are made and conclusions are drawn based on possibly multiple sources of contextual knowledge, either sensed, modeled or both. A GPS component, for instance, may sense the location of a particular entity to be a specific street in the town (sensed context). Another component consults the knowledge base with modeled contextual knowledge and determines that there is a co-located chemical plant. From these facts, it can be inferred that the entity is in a dangerous location (inferred context), so that appropriate actions can be taken to resolve this undesirable situation.

It should be noted that contextual models are subject to "aging," which means that they have to be continuously updated. Sometimes, the contextual evidence is uncertain, that is, it is provided given specific probabilities. The evidence obtained from different information sources may even be contradictory. For instance, the sensors may be imprecise, so that a calendar entry reports a different location as it is sensed by the GPS component. The most well investigated type of context is location. Therefore it will be described in particular detail in a special chapter of the book. This type of contextual information may be absolute ("X is room 253"), or relative ("voltmeter is with Mr. X").

User awareness denotes the ability of a UC system to adapt to human users. The notion of user should not be understood in the narrow sense of a system user. In the future, this could be, for example, also a provider of specific services in a UC environment. We will describe the technologies pertinent to user awareness, such as user models, user profiles and preferences, and user agents. A challenge here is to design models supporting a huge range of possible new types of users. For example, the users may be inexperienced, they may be limited in terms of interaction capabilities, such as hands/eyes free environments, or display restricted cognitive capabilities, such as limited attention. Appropriately supporting interaction with these kinds of users requires the modeling and understanding of their actions in an appropriate way. Specifically for UC applications, user models should become ubiquitous. Current UC systems contribute and use only a small fraction of this information to date.

Part L: Liability. This part discusses the protection of actors, that is, users of UC systems, and those concerned by actions, that is, peers, third parties and the society. Protection has to be realized in the presence of zillions of peers, which requires special extensions of the conventional concepts of security to make them scalable with respect to the number of parties and the open nature of communication. Scalable security pursues the old goals, for example, of ensuring privacy and authentication, but involves a set of new aspects discussed below.

Machine-to-machine communication & ad hoc (spontaneous) communication and interaction: *A priori*, UC settings are *un*managed domains. One cannot assume that in each and every setting hierarchical and managed security infrastructures, for example a PKI (public key infrastructure), are in place. Thus, to support ad hoc communication, including machine-to-machine communication without users involved, PKIs and the like are impractical. First, a PKI require powerful hardware to complete the cryptographic operations in a reasonable time. Second, every centralized approach scales badly in respect of zillions of peers, and the pre-requisite of being always reliably connected to a central and trusted third party is not given. An early approach to support secure ad hoc machine-to-machine interaction is the resurrecting duckling protocol (Stajano & Anderson, 1999).

End-to-end encryption is very hard to achieve in UC settings. One major question is: "How do we define an endpoint?" At one extreme a user and her interaction with the UC environment is the endpoint. But this requires the user to trust the very first device she is interacting with. Whether or not work on trusted computing helps is still unclear today.

Since UC places the human at its centre, we need anthropomorphic security methods and solutions that comply with human intuition while a user interacts with UC technology in her everyday life. This suggests intuitive and human understandable models of trust, risk, and recommendation, to name but a few concepts. In addition, security measures need to be accepted by a user. Therefore, a focus on user-friendliness (ease-of-use) is necessary.

Taking society as a whole into account, liability has to deal with conflicting goals and forces. For example, protection of an individual's privacy may conflict with a public/society goal to secure living together. This involves several scientific disciplines (law, society, computer science), but liability in UC needs to provide flexible solutions that comply with society rules and laws as well with individual needs. These solutions will always come with a tradeoff between the concerned parties.

In this context, liability is also related to the rules of an economy, or market: UC services are offered and used based on currencies, which both users and providers have to associate with the value of these services. Users want "to get what they pay for," providers want to "get paid for what they provide"—a matter of contracts *prior to* service use, and guarantees plus enforcement means *during* service use. A free economy must be based on competition, which in turn can only work if values and prices associated with goods (here: services) can be compared. The key question is the extent to which these economic concepts can be formalized and automated for software. There are concepts and mechanisms in current networks (telephony networks in particular), which provide a certain degree of contract/guarantee/enforcement-means and of comparability as described, either for Internet-based services (such as those following the Web service paradigm and standards) or for services offered in operator owned (e.g., telephony) networks. Major concepts to mention in this respect comprise: (i) the accounting and billing concepts used in telephony networks, (ii) "service level agreements" (SLAs) between service providers and users (a wealth of non-formal approaches to SLA exist for non-software, for example, call center or help-desk services, but some concepts have been formalized for software-based negotiation and control in the network services context), (iii) "quality-of-service" (QoS) concepts known from multimedia (transmission) services, and (iv) semantics-based descriptions of Web services based on, for example, the Web Service Modeling Language WSML.

Part E: Ease-of-Use. User-friendliness considers the optimal use and combination of modalities as well as the advancement of specific modalities given a particular application setting. The readers should note that, partially, ease-of-use is treated in the "Adaptability" part of the present book. In particular, the adaptability of the user interface treated there belongs within the scope of ease-of-use. In a nutshell, the user interface is reduced to what "makes sense" in a given situation.

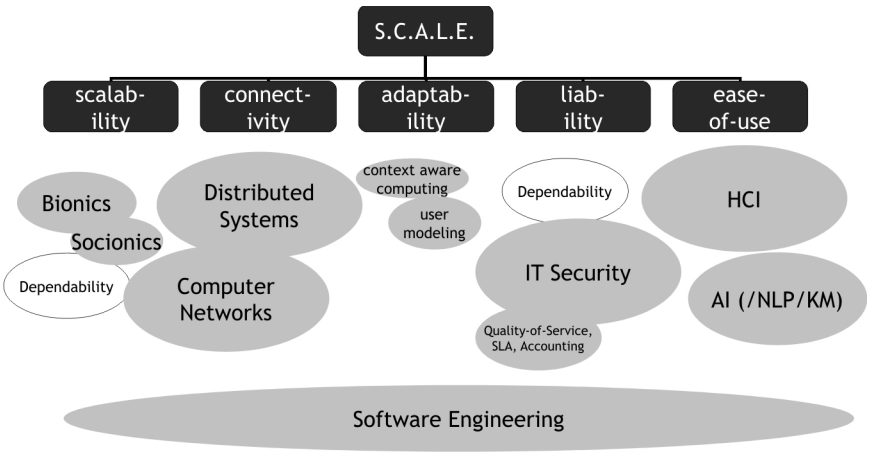
We further discuss a variety of input and output devices in the ubiquitous computing world. Such devices are designed to enable multimodal interaction. A simple distinction in this context is made between hands and eyes and mouth and ear interaction. In advanced hands and eyes interaction, graphical displays and GUIs are typically predominant, though further developments are still needed at this point. Examples of hands and eyes interaction are focus + context displays, 3rd dimension (VR), 4th dimension (dynamic displays), immersion and narration. Mouth and ear interaction bears great potential as it allows the users to operate in a hands/eyes free manner. However, voice processing is still underdeveloped today. It requires understanding speech input, which is challenging due to multiple reasons, such as the quality of speech recognition, the necessity to adapt to a multitude of speakers, and generally the difficulty of generating semantic representations for unrestricted language.

Generally, a better integration of human-computer interaction and software engineering is desirable. So far, these two strands of computer science research have developed in parallel. Another important issue is multimodality, which goes beyond syntactic transformation (transcoding) of XML representations with the help of XSLT to generate device-specific variants of these XML representations. A more abstract interaction layer has to be introduced, so that the details concerning the use of a specific modality are decoupled from the core program. The advantages concerning the use of this particular modality, however, cannot yet be fully exploited. True multimodality also implies the use of multiple modalities simultaneously, giving the user a lot of freedom while interacting with the system. In this case, the inputs are subject to so-called fusion, that is, they are mapped to a single unified semantic representation of the user's input, while the outputs have to be fissioned, that is, distributed over a set of appropriate modalities for presentation. Furthermore, multimodality can be integrated with advanced user interfaces, which are context-sensitive, adapt to the user and behave pro-actively, making suggestions for actions to be taken in a specific situation. Finally, multimodality has to be enabled for federated devices, which involves determining the optimal use of multiple input and output devices currently surrounding the user. In this case, liability explained above becomes an important issue.

Ease-of-use in UC systems should approach human intuition. Thereby, a cross-cutting concern is to design post-desktop metaphors utilizing many of the aspects discussed above. For example, the task of sorting e-mails or documents into specific folders or directories is carried out by the user. In the future, all applications will be likely to adapt to the user's favorite structure. In a similar way, as metaphors have been invented for graphical user interfaces, appropriate ones have to be designed and introduced for voice, and so forth.

Users of UC systems should get used to and understand the probabilistic behavior of such systems as their inherent characteristic. Some amount of uncertainty is a natural consequence of scalability. Mass data emerge as the result of ubiquity; for example, digital recordings are swamping user disks due to digital cameras. The models of dealing with this problem today involve either reducing the amount of data, that is, deleting some pictures, classifying the data manually, or accepting the chaos. The future will make use of more intelligent methods, for example, modeling the human brain, whereby the data is classified and ranked automatically. During this process, it has to be (i) evaluated with respect to existing priorities, (ii) summarized, that is, reduced to a set of compact statements, and (iii) even forgotten, that is, deleted, if the data is no longer relevant.

Figure 4. Relationship to computer science research



A further approach to ease-of-use is modeling human-computer interaction on human dialogues. Intelligent user interfaces should be capable of talking with the user in such a way as people would talk to each other. For this purpose, computational linguistics and natural language processing techniques have to be employed. Determining the meaning of language is, however, not a trivial task as stated above. Grammars used for natural language analysis are often limited to specific domains. To enable natural language understanding beyond phrases, large databases with world knowledge are required, which are difficult to obtain. Therefore, natural dialogues with computers are still a matter of ongoing research.

Another issue involving understanding natural language in UC systems involves integrating formal knowledge with informal knowledge sources. Formal knowledge is represented in structured semantically annotated documents, while informal knowledge is rather scattered over a huge number of information repositories. Examples of the information repositories with informal knowledge are e-mails, forums, Web sites, blogs, and so on. Analyzing such informal knowledge documents is especially challenging, as the language employed in electronic communication is often informal. For instance, it typically contains abbreviated expressions, which have to be resolved automatically.

To enable multimodal interaction with UC systems, the results of natural language analysis have to be combined with the analysis of other modalities, such as gesture or facial expressions. Note that the latter even involves the analysis of human emotions, whose interpretation is challenging even for people, let alone machines. Therefore, natural language processing alone would not suffice to enable natural human-computer interaction. Instead, it should be enriched with additional techniques to cover all forms of input.

As became evident from the previous discussion, the S.C.A.L.E. classification defining the scope of ubiquitous computing as a field is related to multiple areas of computer science. These relations are shown in the Figure 4.

- **Scalability**, although an issue in its own right, is not considered a scientific (sub-) discipline; therefore, only bionics and socionics are mentioned here as recognized disciplines. A number of other issues will be treated in this part, some of which have a long tradition (but are rooted in different areas that would really be a distraction if listed here). Dependability is marked as "half related" in the sense that new, scalable approaches must be looked for which depart from the idea that the dependability of individual components can ever reach a level of "perfection."

- **Connectivity** is largely rooted in computer networks and distributed systems—areas that some consider as the roots of UC as a whole, but that would underemphasize success critical issues such as "humane computing."
- **Adaptability** is a superset of the newly established area "context-aware computing." The book emphasizes the need for more elaborate user modeling than what usual context-aware computing approaches foresee, thus the inclusion of this second research domain in the figure.
- **Liability** as a major part of the book is largely dominated by IT security issues. More appropriate solutions for them must be found, such as anthropomorphic concepts and models, scalable schemes, and so forth. For turning UC into a global market, economic aspects must be handled, leveraging off sub-aspects of computer networks/distributed systems, such as QoS, SLA, and accounting/billing concepts.
- **Ease-of-use** finally is dominated by HCI research, but is also greatly influenced by the AI concepts used in natural language processing and computational linguistics, and by the concepts of knowledge management.

Software engineering is the basis for efficient and effective development of UC applications. Therefore it influences UC greatly—but must also itself be influenced by UC, since many software engineering methods, notations, and concepts must reflect the requirements imposed by UC.

READER'S DIGEST

The book will be interesting to many researchers and practitioners in industry and academia, whose interests lie in one or several of the five different axes of the S.C.A.L.E. classification, that is, scalability (AI algorithms, ontologies, services), connectivity (networks, peer-to-peer), adaptability (context and user models), liability (trust, security), and ease-of-use (multimodal interaction, intelligent user interfaces). In this case, they can learn the foundations of ubiquitous computing and how their area of interest can be readied for ubiquitous computing.

We believe that the book will serve not only as a practical reference book, but also as a reference book for academics in the area of ubiquitous computing. For students in advanced undergraduate and graduate programs, the book will be suitable as a course book in computer science, electrical and computer engineering, and information technology. It can be used in courses like distributed systems, computer networks, human-computer interaction, and IT security. Additionally, students doing specialized studies in areas like software engineering, media and communications engineering, ubiquitous computing/pervasive computing/ambient intelligence, or business process modeling will be able to use the book as supplementary material. It will provide not only foundational material and a description of the state-of-the-art, but bibliographic pointers for further readings and a discussion of the research issues, trends and the development of the field.

As to the introduction and five main parts, the book covers different aspects of ubiquitous computing in a manner that makes the book suitable for undergraduate courses. Concepts, methods and algorithms of ubiquitous computing are presented such that the book is "self-contained." This means that students can learn the foundations of ubiquitous computing from the book itself. In addition, each chapter provides an extensive list of bibliographic references and a discussion of research issues complementing the foundational material. This makes the book suitable as a reference for computer science practitioners and also as a reference book in advanced, for example, graduate, courses. For each part, a short introduction prepares the ground for the reader to understand and interrelate the following chapters. For all

chapters, special care is taken not to remain shallow, but to cover long-term methodological knowledge (overview of the state-of-the-art with selected fundamental methods, algorithms, or concepts in detail) and to describe current problems and innovative solutions.

Finally, a supplementary part with the descriptions of ubiquitous computing projects at SAP Research complements the treatment of the subject by providing insight into real-life projects and problems that have to be dealt with while transferring UC technologies into industrial practice. SAP Research is the research division of SAP, the world's leading producer of business software (used, incidentally, by the top 100 companies worldwide). The main charter of SAP Research is the realization of the vision "real time enterprise." Thus, the digital world (company software) is connected online to the physical world: events and actions in one of these worlds are reflected without error-prone intermediate manual steps or switches in media in the other world in real time. This final part of the book is intended as the "grounding" that illustrates to what extent the approaches and research findings from the previous chapters are already on their way to practical use.

We believe that we have provided the first systematic, full-coverage book about ubiquitous computing (a.k.a., pervasive computing, a.k.a., ambient intelligence) that can serve as a reference book and a researcher and practitioner's guide.

REFERENCES

Aitenbichler, E., Kangasharju, J., & Mühlhäuser, M. (2006). MundoCore: A lightweight infrastructure for pervasive computing. *Pervasive and Mobile Computing, 3*(4), 332-361.

Stajano, F., & Anderson, R.J. (1999). The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols, 7th International Workshop*, Cambridge, UK (pp. 172-194).

Weiser, M. (1991). The computer for the twenty first century. *Scientific American*, 265(3), 94-104.