Ubiquitous Computing

Smart Devices, Environments and Interactions

Stefan Poslad

WILEY

Ubiquitous Computing

Ubiquitous Computing: Smart Devices, Environments and Interactions Stefan Poslad
 ISBN: 978-0-470-03560-3

Ubiquitous Computing

Smart Devices, Environments and Interactions

Stefan Poslad

Queen Mary, University of London, UK

WILEY

A John Wiley and Sons, Ltd, Publication

This edition first published 2009

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

Library of Congress Cataloging-in-Publication Data

Poslad, Stefan.
Ubiquitous computing : smart device, environment, and interactions / Stefan Poslad.
p. cm.
Includes index.
ISBN 978-0-470-03560-3 (cloth)
1. Ubiquitous computing. 2. Context-aware computing. 3. Human-computer interaction.
I. Title.
QA76.5915.P67 2009
004—dc22
2008052234

A catalogue record for this book is available from the British Library.

ISBN 978-0-470-03560-3 (H/B)

Set in 9/11pt Times by Integra Software Services Pvt. Ltd. Pondicherry, India
Printed and bound in Great Britain by Antony Rowe, Chippenham, UK

To my family, Ros and Ben here, and to friends and family in three wonderful parts of the world, South Wales (UK), Glandorf and Brisbane.

Contents

List of Figures

List of Tables

Preface

Ubiquitous Computing, often also referred to as Pervasive Computing, is a vision for computer systems to infuse the physical world and human and social environments. It is concerned with making computing more physical, in the sense of developing a wider variety of computer devices can be usefully deployed in more of the physical environment. It is concerned with developing situated and pervasive technology that is highly accessible and usable by humans that can be designed to operate in harmony in human and social environments.

Audience

This book is primarily aimed at computer scientists and technologists in education and industry to enable them to keep abreast of the latest developments, across a diverse field of computing, all in one text. Its aim is to also to promote a much more cross-disciplinary exchange of ideas within the sub-fields of computing and between computer science and other associated fields. It interlinks several sub-fields of computing, distributing computing, communication networks, artificial intelligence and human computer interaction at its core, as well as explaining and extending designs which cover mobile services, service-oriented computing, sensor nets, micro-electromechanical systems, context-aware computing, embedded systems and robotics, and new developments in the Internet and the Web. This is a good text to apply models in these fields.

The main prerequisite needed to understand this book is a basic level of understanding of computer science and technology. Parts of the book should be readily understandable by students towards the middle and end of undergraduate courses in computer science, although parts of it may also be used as an introduction textbook to highlight some of the amazing things that are happening in the world of ICT systems. It is also suitable for students at MSc level and for cross-disciplinary use in courses which include computing as just one of the elements of the course. It is the author's hope that this text will contribute to a renewed interest in some of the advanced ideas of computing by a wider audience and will lead to new advanced courses in computing being developed. An overview of the book is found at the end of the first chapter.

Teaching with this Book

The author's website for the book is available at http://www.elec.qmul.ac.uk/people/stefan/ubicom. The website contains PowerPoint slides for the book, additional exercises and selected solutions to exercises, on-line bibliography for the book, etc. The book site also gives advice about how to use this book in different types of educational courses and training programs.

Acknowledgements

Patricia Charlton, Michael Berger and Robert Patton were involved with this book project at the start. In particular, Patricia Charlton contributed many good ideas particularly in the AI chapters, two of which she co-authored. Several international colleagues gave feedback on specific sections of the book: Barbara Schmidt-Belz, Heimo Laamanen, Jigna Chandaria and Steve Mann; as did several colleagues at QMUL: Athen Ma, Chris Phillips, Karen Shoop and Rob Donnan.

The contents of this book arose in part out of teaching various distributed computing, AI, HCI and other applied computing courses at undergraduate level and at MSc at several universities but in particular through teaching the ELEM038, Mobile Services courses to students at Queen Mary, University of London. Second, this book arose out of research in the following projects: AgentCities, CASBAH, Context-aware Worker (an EPSERC Industrial Case-award project with BT, John Shepherdson), CRUMPET, EDEN-IW, iTrust, My e-Director 2012 and from work with my research assistants: uko Asangansi, Ioannis Barakos, Thierry Delaitre, Xuan (Janny) Huang, Kraisak Kesorn, Zekeng Liang, Dejian Meng, Jim Juan Tan, Leonid Titkov, Zhenchen Wang and Landong Zuo. Several of them helped to review parts of this text.

At Wiley, Birgit Gruber instigated this book project. Sarah Hinton and Anna Smart guided the book through the various stages of drafting to the finished product while Susan Dunsmore and Sunita Jayachandran helped apply the finishing touches. Finally, my family offered a high-level of support throughout, encouraging me onwards, to finish it.

1

Ubiquitous Computing: Basics and Vision

1.1 Living in a Digital World

We inhabit an increasingly digital world, populated by a profusion of digital devices designed to assist and automate more human tasks and activities, to enrich human social interaction and enhance physical world[1] interaction. The physical world environment is being increasingly digitally instrumented and strewn with embedded sensor-based and control devices. These can sense our location and can automatically adapt to it, easing access to localised services, e.g., doors open and lights switch on as we approach them. Positioning systems can determine our current location as we move. They can be linked to other information services, i.e., to propose a map of a route to our destination. Devices such as contactless keys and cards can be used to gain access to protected services, situated in the environment. Epaper[2] and ebooks allow us to download current information onto flexible digital paper, over the air, without going into any physical bookshop. Even electronic circuits may be distributed over the air to special printers, enabling electronic circuits to be printed on a paper-like substrate.

In many parts of the world, there are megabits per second speed wired and wireless networks for transferring multimedia (alpha-numeric text, audio and video) content, at work and at home and for use by mobile users and at fixed locations. The increasing use of wireless networks enables more devices and infrastructure to be added piecemeal and less disruptively into the physical environment. Electronic circuits and devices can be manufactured to be smaller, cheaper and can operate more reliably and with less energy. There is a profusion of multi-purpose smart mobile devices to

[1] The physical world is often referred to as the *real-world* or environment in order to distinguish this both from a perceived human view of the world (*imaginary worlds*) not related to reality and basic facts and from computer-generated views of the world (*virtual worlds*).
[2] A distinction needs to be between digital hardware versions of analogue objects, e.g., epaper, versus soft or electronic copies of information held in analogue objects, e.g., etickets, for airlines. The latter type is referred to as vtickets, short for virtual tickets.

Ubiquitous Computing: Smart Devices, Environments and Interactions Stefan Poslad
 ISBN: 978-0-470-03560-3

access local and remote services. Mobile phones can act as multiple audio-video cameras and players, as information appliances and games consoles.[3] Interaction can be personalised and be made user context-aware by sharing personalisation models in our mobile devices with other services as we interact with them, e.g., audio-video devices can be pre-programmed to show only a person's favourite content selections.

Many types of service provision to support everyday human activities concerned with food, energy, water, distribution and transport and health are heavily reliant on computers. Traditionally, service access devices were designed and oriented towards human users who are engaged in activities that access single isolated services, e.g., we access information vs we watch videos vs we speak on the phone. In the past, if we wanted to access and combine multiple services to support multiple activities, we needed to use separate access devices. In contrast, service offerings today can provide more integrated, interoperable and ubiquitous service provision, e.g., use of data networks to also offer video broadcasts and voice services, so-called triple-play service provision. There is great scope to develop these further (Chapter 2).

The term 'ubiquitous', meaning appearing or existing everywhere, combined with computing to form the term Ubiquitous Computing (UbiCom) is used to describe ICT (Information and Communication Technology) systems that enable information and tasks to be made available everywhere, and to support intuitive human usage, appearing invisible to the user.

1.1.1 Chapter Overview

To aid the understanding of Ubiquitous Computing, this introductory chapter continues by describing some illustrative applications of ubiquitous computing. Next the proposed holistic framework at the heart of UbiCom called the Smart DEI (pronounced smart 'day') Framework UbiCom is presented. It is first viewed from the perspective of the core internal properties of UbiCom (Section 1.2). Next UbiCom is viewed from the external interaction of the system across the core system environments (virtual, physical and human) (Section 1.3). Third, UbiCom is viewed in terms of three basic architectural designs or design 'patterns': smart devices, smart environments and smart interaction (Section 1.4). The name of the framework, DEI, derives from the first letters of the terms **D**evices, **E**nvironments and **I**nteraction. The last main section (Section 1.5) of the chapter outlines how the whole book is organised. Each chapter concludes with exercises and references.

1.1.2 Illustrative Ubiquitous Computing Applications

The following applications situated in the human and physical world environments illustrate the range of benefits and challenges for ubiquitous computing. A personal memories scenario focuses on users recording audio-video content, automatically detecting user contexts and annotating the recordings. A twenty-first-century scheduled transport service scenario focuses on the transport schedules, adapting their preset plans to the actual status of the environment and distributing this information more widely. A foodstuff management scenario focuses on how analogue non-electronic objects such as foodstuffs can be digitally interfaced to a computing system in order to monitor their human usage. A fully automated foodstuff management system could involve robots which can move physical objects around and is able to quantify the level of a range of analogue objects. A utility management scenario

[3] And of course there is nothing stopping this happening vice versa – games consoles can act as phones, audio-video players and recorders, etc., and cameras can act as phones, etc.

focuses on how to interface electronic analogue devices to an UbiCom system and to manage their usage in a user-centred way by enabling them to cooperate to achieve common goals.

1.1.2.1 Personal Memories

As a first motivating example, consider recording a personal memory of the physical world (see Figure 1.1). Up until about the 1980s, before the advent of the digital camera, photography would entail manually taking a light reading and then manually setting the aperture and shutter speed of the camera in relation to the light reading so that the light exposure on to a light-sensitive chemical film was correct.[4] It involved manually focusing the lens system of the camera. The camera film behaved as a sequential recording media: a new recording requires winding the film to the next empty section. It involved waiting for the whole film of a set of images, typically 12 to 36, to be completed before sending the recorded film to a specialist film processing company with specialist equipment to convert the film into a specialist format that could be viewed. The creation of additional copies would also require the services of a specialist film processing company.

A digital camera automatically captures a visual of part of the physical world scene on an inbuilt display. The use of digital cameras enables photography to be far less intrusive for the subject than using film cameras.[5] The camera can autofocus and auto-expose recorded images and

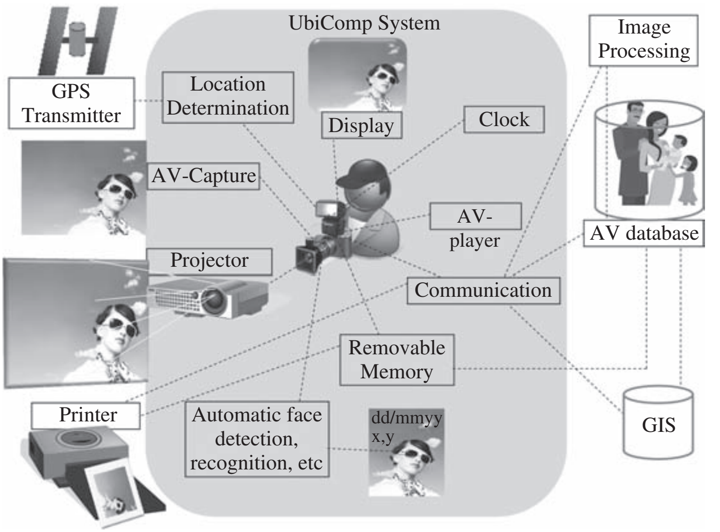

Figure 1.1 Example of a ubiquitous computing application. The AV-recording is person-aware, location-aware (via GPS), time-aware and networked to interact with other ICT devices such as printers and a family-and-friends database

[4] There was an easier-to-operate camera called the compact camera that used a fixed focus (set at infinity) and a fixed exposure for the film.

[5] E.g., digital cameras are less obtrusive: they lessen the need for a photographer to ask the subject to say 'cheese' to make the subject focus on the camera because it is easy and cheap to just shoot a whole series of photographs in quick succession and delete the ones that are not considered aesthetic.

video so that recordings are automatically in focus and selected parts of the scene are lit to the optimum degree. The context of the recording such as the location and date/time is also automatically captured using inbuilt location and clock systems. The camera is aware that the person making a recording is perhaps interested in capturing people in a scene, in focus, even if they are off centre. It uses an enhanced user interface to do this which involves automatically overlaying the view of the physical world, whether on an inbuilt display or through a lens or viewfinder, with markers for parts of the face such as the eyes and mouth. It then automatically focuses the lens so faces are in focus in the visual recording.

The recorded content can be immediately viewed, printed and shared among friends and family using removable memory or exchanged across a communications network. It can be archived in an external audio-visual (AV) content database. When the AV content is stored, it is tagged with the time and location (the GIS database is used to convert the position to a location context). Image processing can be used to perform face recognition to automatically tag any people who can be recognised using the friends and family database. Through the use of micro electromechanical systems (MEMS (Section 6.4) what previously needed to be a separate decimetre-sized device, e.g., a projector, can now be inbuilt. The camera is networked and has the capability to discover other specific types of ICT devices, e.g., printers, to allow printing to be initiated from the camera. Network access, music and video player and video camera functions could also be combined into this single device.

Ubiquitous computing (UbiCom) encompasses a wide spectrum of computers, not just devices that are general purpose computers,[6] multi-function ICT devices such as phones, cameras and games consoles, automatic teller machines (ATMs), vehicle control systems, mobile phones, electronic calculators, household appliances, and computer peripherals such as routers and printers. The characteristics of embedded (computer) systems are that they are self-contained and run specific predefined tasks. Hence, design engineers can optimise them as follows. There is less need for full operating system functionality, e.g., multiple process scheduling and memory management and there is less need for a full CPU, e.g., the simple 4-bit microcontrollers used to play a tune in a greeting card or in a children's toy. This reduces the size and cost of the product so that it can be more economically mass-produced, benefiting from economies of scale. Many objects could be designed to be a multi-function device supporting AV capture, an AV player, communicator, etc. Embedded computing systems may be subject to a real-time constraint, real-time embedded systems, e.g., anti-lock brakes on a vehicle may have a real-time constraint that brakes must be released within a short time to prevent the wheels from locking.

ICT Systems are increasing in complexity because we connect a greater diversity and number of individual systems in multiple dynamic ways. For ICT systems to become more useful, they must in some cases become more strongly interlinked to their physical world locale, i.e., they must be context-aware of their local physical world environment. For ICT systems to become more usable by humans, ICT systems must strike the right balance between acting autonomously and acting under the direction of humans. Currently it is not possible to take humans completely out of the loop when designing and maintaining the operation of significantly complex systems. ICT systems need to be designed in such a way that the responsibilities of the automated ICT systems are not clear and the responsibilities of the human designers, operators and maintainers are clear and in such a way that human cognition and behaviour are not overloaded.

[6] Of the billions of microprocessors manufactured every year, less than 5% of them find their way into multi-application programmable computers, the other 95% or so are deployed in a range of embedded systems and applications.

1.1.2.2 Adaptive Transport Scheduled Service

In a twentieth-century scheduled transport service, timetables for a scheduled transport service, e.g., taxi, bus, train, plane, etc. to pick up passengers or goods at fixed or scheduled point are only accessible at special terminals and locations. Passengers and controllers have a limited view of the actual time when vehicles arrive at designated way-points on the route. Passengers or goods can arrive and wait long times at designated pick-up points. A manual system enables vehicle drivers to radio in to controllers their actual position when there is a deviation from the timetable. Controllers can often only manually notify passengers of delays at the designated pick-up points.

By contrast, in a twenty-first-century scheduled transport service, the position of transport vehicles is determined using automated positioning technology, e.g., GPS. For each vehicle, the time taken to travel to designated pick-up points, e.g., next stop, final stop, is estimated partly based on current vehicle position, progress and historical data of route users. Up-to-date vehicle arrival times can then be accessed ubiquitously using mobile phones enabling JIT (Just-In-Time) arrival at passenger and goods collection points. Vehicles on the route can tag locations that they anticipate will change the schedule of other vehicles in that vicinity. Anticipated schedule change locations can be reviewed by all subsequent vehicles. Vehicles can then be re-routed and re-scheduled dynamically, based upon 'schedule change' locations, current positions and the demand for services. If the capacity of the transport vehicles was extensible, the volume of passengers waiting on route could determine the capacity of the transport service to meet demand. The transport system may need to deal with conflicting goals such as picking up more passengers and goods to generate more revenue for services rendered versus minimising how late the vehicle arrives at pre-set points along its route.

1.1.2.3 Foodstuff Management

A ubiquitous home environment is designed to support healthy eating and weight regulation for food consumers. A conventional system performs this manually. A next generation system (semi-)automates this task using networked physical devices such as fridges and other storage areas for food and drink items which can monitor the food in and out. Sensors are integrated in the system, e.g., to determine the weight of food and of humans. Scanners can be used to scan the packaging of food and drink items for barcodes, text tables, expiry dates and food ingredients and percentages by weight. Hand-held integrated scanners can also select food for purchase in food stores such as supermarkets that should be avoided on health or personal choice grounds. The system can identify who buys which kind of food in the supermarket.

The system enables meal recipes to be automatically configured to adapt to the ingredients in stock. The food in stock can be periodically monitored to alert users when food will becomes out of date and when the supply of main food items is low. The amount of food, at different levels of granularity in terms of the overall amount of food and in terms of weight in grams of fat, salt and sugar, etc, consumed per unit time and per person can be monitored. The system can incorporate policies about eating a balanced diet, e.g., to consume five pieces of fruit or vegetables a day.

System design includes the following components. Scanners are used to identify the types and quantities of ingredients based upon the packaging. This may include a barcode but perhaps not all food has barcodes and can be identified in this way. The home food store can be designed to check when (selected) food items are running low. Food running low can be defined as there is a quantity of one item remaining but items can be large and partially full. The quantity of a foodstuff remaining needs to be measured using a weight transducer but the container weight overhead is needed in order to calculate the weight of the foodstuff. The home food store could be programmed to detect when food is out of date by reading the expiry date and signalling the food as inedible.

Many exceptions or conditions may need to be specified for the system in order to manage the food store. For example, food may still be edible even if its expiry date has past. Food that is frozen and then thawed in the fridge may be past its sell-by date but is still edible. Selected system events could automatically trigger actions, e.g., low quantities of food could trigger actions to automatically purchase food and have it delivered. Operational policies must be linked to context or situation and to the authorisation to act on behalf of owner, e.g., food is not ordered when consumers are absent or consumers specify that they do not want infinite repeat orders of food that has expired or is low in quantity. There can be limitations to full system automation. Unless the system can act on behalf of the human owner to accept delivery, to allow physical access to the home food store and to the building where consumers live, and has robots to move physical objects and to open and close the home food store to maintain temperature controlled environments there, these scenarios will require some human intervention. An important issue in this scenario is balancing what the system can and should do versus what humans can and should do.

1.1.2.4 Utility Regulation

A ubiquitous home environment is designed to regulate the consumption of a utility (such as water, energy or heating) and to improve usage efficiency. For example, currently utility management, e.g., energy management, products are manually configurable by human users, utilise stand-alone devices and are designed to detect local user context changes. User context-aware energy devices can be designed to switch themselves on in a particular way, e.g., a light switches on, heating switches on when it detects the presence of a user otherwise it switches off. These devices must also be aware of environmental conditions so that artificial lights and heating would not switch on if it determines that the natural lighting and heating levels will suffice.

System design includes the following components and usage patterns. Devices that are configured manually may waste energy because users may forget to switch them off. Devices that are set to be active, according to pre-set user policies, e.g., to control a timer, may waste energy because users cannot always schedule their activities to adhere to the static schedule of the timer. Individually, context-aware devices such as lights, can waste energy because several overlapping devices may be activated and switch on, e.g., when a user's presence is detected.

A ubiquitous system can be designed, using multi-agent system and autonomic system models, to operate as a Smart Grid. Multiple devices can self-manage themselves and cooperate to adhere to users' policies such as minimising energy expenditure. For example, if several overlapping devices are deemed to be redundant, the system will decide which individual one to switch on. Energy usage costs will depend upon multiple factors, not just the time a device is switched on, but also upon the energy rating which varies across devices and the tariff, i.e., the cost of energy usage varies according to the time of day. Advanced utility consumption meters can be used to present the consumption per unit-time and per device and can empower customers to see how they are using energy and to manage its use more efficiently. Demand-response designs can adjust energy use in response to dynamic price signals and policies. For example, during peak periods, when prices are higher, energy-consuming devices could be operated more frugally to save money. A direct load control system, a form of demand-response system, can also be used, in which certain customer energy-consuming devices are controlled remotely by the electricity provider or a third party during peak demand periods. Further examples of ubiquitous computing applications are discussed in Chapter 2.

1.1.3 Holistic Framework for UbiCom: Smart DEI

Three approaches to analyse and design UbiCom Systems to form a holistic framework for ubiquitous computing are proposed called the smart DEI[7] framework based upon:

- *Design architectures to apply UbiCom systems*: Three main types of design for UbiCom systems are proposed: smart device, smart environment and smart interaction. These designs are described in more detail in Section 1.4.
- *An internal model of the UbiCom system properties based upon five fundamental properties*: distributed, iHCI, context-awareness, autonomy, and artificial intelligence. There are many possible sub-types of ubiquitous system design depending on the degree to which these five properties are supported and interlinked. This model and these properties are described in Section 1.2.
- *A model of UbiCom system's interaction with its external environments*. In addition to a conventional distributed ICT system device interaction within a virtual[8] environment (C2C), two other types of interaction are highlighted: (a) between computer systems and humans as systems (HCI); (b) between computers and the physical world (CPI). Environment interaction models are described in Section 1.3.

Smart devices, e.g., mobile smart devices, smart cards, etc. (Chapter 4), focus most on interaction within a virtual (computer) world and are less context-aware of the physical world compared to smart environment devices. Smart devices tend to be less autonomous as they often need to directly access external services and act as personal devices that are manually activated by their owner. There is more emphasis on designing these devices to be aware of the human use context. They may incorporate specific types of artificial intelligence, e.g., machine vision allows cameras to recognise elements of human faces in an image, e.g., based upon eyes and mouth detection.

Smart environments consist of devices, such as sensors, controller and computers that are embedded in, or operate in, the physical environment, e.g., robots (Section 6.7). These devices are strongly context-aware of their physical environment in relation to their tasks, e.g., a robot must sense and model the physical world in order for it to avoid obstacles. Smart environment devices can have an awareness of specific user activities, e.g., doors that open as people walk towards them. They often act autonomously without any manual guidance from users. These incorporate specific types of intelligence, e.g., robots may build complex models of physical behaviour and learn to adapt their movement based upon experience.

Smart interaction focuses on more complex models of interaction of distributed software services and hardware resources, dynamic cooperation and completion between multiple entities in multiple devices in order to achieve the goals of individual entities or to achieve some collective goal. For example, an intelligent camera could cooperate with intelligent lighting in a building to optimise the lighting to record an image. Multiple lighting devices in a physical space may cooperate in order to optimise lighting yet minimise the overall energy consumed. Smart interaction focuses less on physical context-awareness and more on user contexts, e.g., user goals such as the need to reduce the overall

---

[7] Smart DEI stands for the Smart Devices, Environments and Interactions model. It is pronounced 'Smart Day' in order to allude to the fact that the model focuses on the use of systems support for daily activities.

[8] A virtual (computing) environment comprises the distributed shared ICT infrastructure in which individual UbiCom system applications operate. Note also there are other sub-types of virtual environment such as *virtual reality* environments in which humans users can interact with computer simulations of parts of imagined worlds using multimodal sensory interfaces.

energy consumption across devices. Smart interaction often uses distributed artificial intelligence and multi-agent system behaviours, e.g., contract net interaction in order to propose tasks.

The Smart DEI model represents a holistic framework to build diverse UbiCom systems based on smart devices, smart environments and smart interaction. These three types of design can also be combined to support different types of smart spaces, e.g., smart mobile devices may combine an awareness of their changing physical environment location in order to optimise the routing of physical assets or the computer environment from a different location. Each smart device is networked and can exchange data and access information services as a core property. A comparison of a type of smart device, smart environment and smart interaction is also made later (see Table 1.6) with respect to their main UbiCom system properties of distributed, context-aware, obtrusive HCI, autonomy and intelligence and with respect to the types of physical world, human and ICT interactions they support.

1.2 Modelling the Key Ubiquitous Computing Properties

A world in which computers disappear into the background of an environment consisting of smart rooms and buildings was first articulated over fifteen years ago in a vision called ubiquitous computing by Mark Weiser (1991). Ubiquitous computing represents a powerful shift in computation, where people live, work, and play in a seamless computer-enabled environment, interleaved into the world. Ubiquitous computing postulates a world where people are surrounded by computing devices and a computing infrastructure that supports us in everything we do.

Conventional networked computer[9] systems[10] or Information Communication Technology (ICT) systems consider themselves to be situated in a virtual world or environment of other ICT systems, forming a system of ICT systems. Computer systems behave as distributed computer systems that are interlinked using a communications network. In conventional ICT systems, the role of the physical environment is restricted, for example, the physical environment acts as a conduit for electronic communication and power and provides the physical resources to store data and to execute electronic instructions, supporting a virtual ICT environment.

Because of the complexity of distributed computing, systems often project various degrees of transparency for their users and providers in order to hide the complexity of the distributed computing model from users, e.g., anywhere, anytime communication transparency and mobility transparency, so that senders can specify who to send to, what to send rather than where to send it to. Human–computer interaction (HCI) with ICT systems has conventionally been structured using a few relatively expensive access points. This primarily uses input from keyboard and pointing devices which are fairly obtrusive to interact with. Weiser's vision focuses on digital technology that is interactive yet more non-obtrusive and pervasive. His main concern was that computer interfaces are too demanding of human attention. Unlike good tools that become an extension of ourselves, computers often do not allow us to focus on the task at hand but rather divert us into figuring out how to get the tool to work properly.

Weiser used the analogy of writing to explain part of his vision of ubiquitous computing. Writing started out requiring experts such as scribes to create the ink and paper used to present the information. Only additional experts such as scholars could understand and interpret the information. Today, hard-copy text (created and formatted with computers) printed on paper and soft-copy

[9] Here the computer is considered to be any device, simple or complex, small or large, that is programmable and has a memory to store data and or code.

[10] A system, at this stage, is defined as a set of interlinked components of interest. Systems are often a system of systems. Everything external to the system's boundary is the system's environment.

text displayed on computer-based devices are very pervasive. Of the two, printed text is still far more pervasive than computer text.[11] In many parts of the world, the majority of people can access and create information without consciously thinking about the processes involved in doing so.[12] Additional visions of Ubiquitous Computing are discussed in Chapter 2 and in the final chapter (Chapter 13).

1.2.1 Core Properties of UbiCom Systems

The features that distinguish UbiCom systems from distributed ICT systems are as follow. First, they are situated in human-centred personalised environments, interacting less obtrusively with humans. Second, UbiCom systems are part of, and used in, physical environments, sensing more of the physical environment. As they are more aware of it, they can adapt to it and are able to act on it and control it. Hence, Weiser's vision for ubiquitous computing can be summarised in three core requirements:

1. Computers need to be networked, distributed and transparently accessible.
2. Human–computer interaction needs to be hidden more.
3. Computers need to be context-aware in order to optimise their operation in their environment.
   It is proposed that there are two additional core types of requirements for UbiCom systems:
4. Computers can operate autonomously, without human intervention, be self-governed, in contrast to pure human–computer interaction (point 2).
5. Computers can handle a multiplicity of dynamic actions and interactions, governed by intelligent decision-making and intelligent organisational interaction. This may entail some form of artificial intelligence in order to handle:

   (a) incomplete and non-deterministic interactions;
   (b) cooperation and competition between members of organisations;
   (c) richer interaction through sharing of context, semantics and goals.

Hence, an extended model of ubiquitous system is proposed. These two additional behaviours enable ubiquitous systems to work in additional environments. These environments are clustered into two groups: (a) human-centred, personal social and economic environments; and (b) physical environments of living things (ecologies) and inanimate physical phenomena. These five UbiCom requirements and three types of environment (ICT, physical and human) are not mutually exclusive, they overlap and they will need to be combined.

1.2.2 Distributed ICT Systems

ICT systems are naturally distributed and interlinked. Multiple systems often behave as and appear as a single system to the user, i.e., multiple systems are transparent or hidden from the user. Individual systems may be heterogeneous and may be able to be attached and detached from the ICT system infrastructure at any time – openness.

---

[11] Many people thought that the rise of computers would lead to a paperless world but this has not happened yet, see Exercises.

[12] Note the ability to read and write text and understand an average vocabulary currently requires several years of training. Hand-writing is inherently dependent on natural language interfaces which have fundamental limitations.

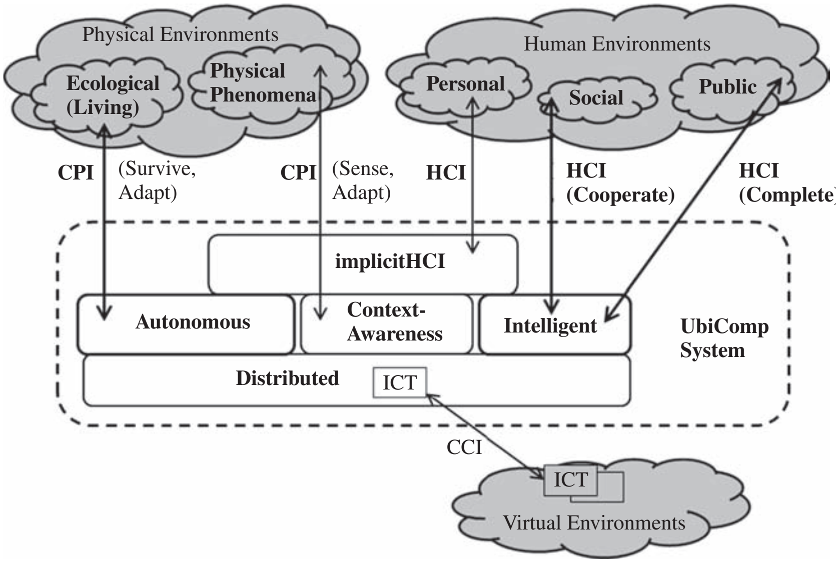

Figure 1.2 A UbiCom system model. The dotted line indicates the UbiCom system boundary

1.2.2.1 Networked ICT Devices

Pervasive computers are networked computers. They offer services that can be locally and remotely accessed. In 1991, Weiser considered that ubiquitous access via 'transparent linking of wired and wireless networks, to be an unsolved problem'. However, since then both the Internet and wireless mobile phones networks have developed to offer seemingly pervasive network access. A range of communication networks exists to support UbiCom interaction with respect to range, power, content, topology and design (Chapter 11).

1.2.2.2 Transparency and Openness

Buxton (1995) considered the core focus of Weiser's vision of ubiquitous computing to be ubiquity (access is everywhere through diverse devices) and transparency (access is hidden, integrated into environments) but that these appear to present an apparent paradox in, how can something be everywhere yet be invisible? The point here is not that one cannot see (hear or touch) the technology but rather that its presence does not intrude into the workplace environment, either in terms of the physical space or the activities being performed. This description of transparency is strongly linked to the notion that devices and functions are embedded and hidden within larger interactive systems. Note also that the vision seems to be associated with a binary classification of system transparency, moving from no transparency to complete transparency. In practice system transparency is often more fuzzy. Systems can have partial connectivity and a limited ability to interoperate with their environment, making transparency more difficult to support. The properties of ubiquity and transparency are core characteristics of types of distributed systems.

A final key property of distributed systems is openness – open distributed systems. Openness allows systems to avoid having to support all their functions at the design time, avoiding closed implementation. Distributed systems can be designed to support different degrees of openness to dynamically discover new external services and to access them. For example, a UbiCom camera can be set to discover printing services and to notify users that these are available. The camera can then transmit its data to the printer for printing.

Openness often introduces complexity and reduces availability. When one function is active, others may need to be deactivated, e.g., some devices cannot record one input while displaying another output. Openness can introduce heterogeneous functions into a system that are incompatible and make the complete system unavailable. Openness can reduce availability because operations can be interrupted when new services and functions are set up. Note many systems are still designed to restrict openness and interoperability even when there appears to be strong benefits not to do so. For example, messages stored in most home answering machines cannot easily be exported, for auditing purposes or as part of a discourse with others. It would be very easy to design phones to share their content via plug and play removable media and a wireless network and to make them more configurable to allow users to customise the amount of message storage they need. Vendors may deliberately and selectively reduce openness, e.g., transparently ignore the presence of another competitor's services, in order to preserve their market share.

Distributed ICT systems are typically designed in terms of a layered model comprising: (1) a hardware resource layer at the bottom, e.g., data source, storage and communication; (2) middleware and operating system services in the middle, e.g., to support data processing and data manipulation; and (3) a human–computer interaction layer at the top. Such a layered ICT model oversimplifies the UbiCom system model because it does not model heterogeneous patterns of systems' interaction. This ICT model typically incorporates only a simple explicit human interaction and simple physical world interaction model. Distributed computer systems are covered in most chapters but in particular in Chapters 3, 4, and 12. Their communications infrastructure is covered in Chapter 11.

1.2.3 Implicit Human–Computer Interaction (iHCI)

Much human–device interaction is designed to support explicit human–computer interaction which is expressed at a syntactical low level, e.g., to activate particular controls in this particular order. In addition, as more tasks are automated, the variety of devices increases and more devices need to interoperate to achieve tasks. The sheer amount of explicit interaction can easily disrupt, distract and overwhelm users. Interactive systems need to be designed to support greater degrees of implicit human–computer interaction or iHCI (Chapter 5).

1.2.3.1 The Calm Computer

The concept of the calm or disappearing computer model has several dimensions. It can mean that programmable computers as we know them today are replaced by something else, e.g., human brain implants, that they are no longer physically visible. It can mean that computers are present but they are hidden, e.g., they are implants or miniature systems. Alternatively, the focus of the disappearing computer can mean that computers are not really hidden; they are visible but are not noticeable as they form part of the peripheral senses. They are not noticeable because of the effective use of implicit human–computer interaction. The forms and modes of interaction to enable computers to disappear will depend in part on the target audience because social and cultural boundaries in relation to technology drivers may have different profile-clustering attributes. For some groups of people, ubiquitous computing is already here. Applications and technologies, such as mobile phones, email and chat messaging systems, are considered as a necessity by some people in order to function on a daily basis.

The promise of ubiquitous computing as technology dissolving into behaviour, invisibly permeating the natural world, is regarded as being unattainable by some researchers, e.g., Rogers (2006). Several reasons are given to support this view. The general use of calm computing removes humans from being proactive – systems are proactive instead of humans. Calm computing is a computationally intractable problem if used generally and ubiquitously. Because technology by its very nature is artificial, it separates the artificial from the natural. What is considered natural is subjective and cultural and to an extent technological. This is blurring the distinction between the means to directly re-engineer nature at the molecular level and the means to influence nature at the macro-level, e.g., pollution and global warming (Chapter 13).

The obtrusiveness of technology depends in part on the user's familiarity and experience with it. Alan Kay[13] is attributed as saying that 'Technology is anything that was invented after you were born.' Everyone considers the technology to be something invented before they were born. If calm computing is used in a more bounded sense in deterministic environments, in limited applications environments and is supported at multiple levels depending on the application requirements, it becomes second nature[14] – calm computing models can then succeed.

1.2.3.2 Implicit Versus Explicit Human–Computer Interaction

The original UbiCom vision focused on making computation and digital information access more seamless and less obtrusive. To achieve this requires in part that systems do not need users to explicitly specify each detail of an interaction to complete a task. For example, using many electronic devices for the first time requires users to explicitly configure some proprietary controls of a timer interface. It should be implicit that if devices use absolute times for scheduling actions, then the first time the device is used, the time should be set. This type of implied computer interaction is referred to as implicit human–computer interaction (iHCI). Schmidt (2000) defines iHCI as 'an action, performed by the user that is not primarily aimed to interact with a computerised system but which such a system understands as input'. Reducing the degree of explicit interaction with computers requires striking a careful balance between several factors. It requires users to become comfortable with giving up increasing control to automated systems that further intrude into their lives, perhaps without the user being aware of it. It requires systems to be able to reliably and accurately detect the user and usage context and to be able to adapt their operation accordingly.

1.2.3.3 Embodied Reality versus Virtual, Augmented and Mediated Reality

Reality refers to the state of actual existence of things in the physical world. This means that things exist in time and space, as experienced by a conscious sense of presence of human beings, and are situated and embodied in the physical world. Human perception of reality can be altered by technology in several ways such as virtual reality, augmented reality, mediated reality and by the hyperreal and telepresence (Section 5.4.4).

Virtual reality (VR) immerses people in a seamless, non-embodied, computer-generated world. VR is often generated by a single system, where time and space are collapsed and exists as a

[13] Kay worked at the Xerox Corporation's Palo Alto Research Center (PARC) in the 1970s and was one of the key researchers who developed early prototypes of networked workstations that were later commercialised by Apple, i.e., the Apple Macintosh.

[14] Second nature is acquired behaviour that has been practised for so long that it seems innate.

separate reality from the physical world. Augmented reality (AR) is characterised as being immersed in a physical environment in which physical objects can be linked to a virtual environment. AR can enhance physical reality by adding virtual views to it e.g., using various techniques such as see-through displays and homographic views. Augmented reality can be considered from both an HCI perspective (Section 5.3.3) and from the perspective of physical world interaction (Section 6.2).

Whereas in augmented reality, computer information is added to augment real world experiences, in the more generic type of mediated reality[15] environment, reality may be reduced or otherwise altered as desired. An example of altering reality rather than augmenting it is, rather than use lenses to correct personal visual deficiencies, is to use them to mask far field vision in order to focus on near field tasks.

Weiser drew a comparison between VR and UbiCom, regarding UbiCom to be the opposite of VR. In contrast to VR, ubiquitous computing puts the use of computing in the physical world with people. Indeed, the contrary notion of ubiquitous, invisible computing compared to virtual reality is so strong that Weiser coined the term 'embodied virtuality'. He used this term to refer to the process of 'drawing computers out of their electronic shells'. Throughout this text, the term 'device' is used to focus on the concept of embodied virtuality rather than the more general term of a virtual service. Multiple devices may also form systems of devices and systems of systems. In very open virtual systems, data and processes can exist anywhere and can be accessed anywhere, leading to a loss of (access) control. The potential for privacy violations increases. In physical and virtual embodied systems, such effects are reduced via the implicit restrictions of the embodiment.

Embodied virtuality has several connotations. In order for computers to be more effectively used in the physical world, they can no longer remain embodied in limited electronic forms such as the personal computer but must exist in a wider range of forms which must be more pervasive, flexible and situated. Hence, the emphasis by Weiser of explicitly depicting a larger range of everyday computer devices in the form of tabs, pad and boards (Section 1.4.1.1). Distributed computing works through its increasing ability to interoperate seamlessly to form a virtual computer out of a group of individual computers; it hides the detailed interaction with the individual computers and hides the embodiment within individual forms forming a virtual embodiment for computing.

The use of many different types of physical (including chemical and biological) mechanisms and virtual assembly and reassembly of nature at different levels, can also change the essence of what is human nature and natural (Sections 5.4, 13.7). Through increasing dependence on seamless virtual computers, UbiCom, humans may also risk the erasure of embodiment (Hayles, 1999).

1.2.4 Context-Awareness

The aim of UbiCom systems is not to support global ubiquity, to interlink all systems to form one omnipresent service domain, but rather to support context-based ubiquity, e.g., situated access versus mass access. The benefits of context-based ubiquity include: (1) limiting the resources needed to deliver ubiquitous services because delivering omnipresent services would be cost-prohibitive; (2) limiting the choice of access from all possible services to only the useful services; (3) avoiding overburdening the user with too much information and decision-making; and (4) supporting a natural locus of attention and calm decision-making by users.

[15] Reality can also be modified by many other mechanisms, not just virtual computer ones, e.g., chemical, biological, psychological, etc.

1.2.4.1 Three Main Types of Environment Context: Physical, User, Virtual

There are three main types of external environment context-awareness[16] supported in UbiCom:

- *Physical environment context*: pertaining to some physical world dimension or phenomena such as location, time, temperature, rainfall, light level, etc.
- *Human context (or user context or person context)*: interaction is usefully constrained by users: in terms of identity; preferences; task requirements; social context and other activities; user experience and prior knowledge and types of user.[17]
- *ICT context or virtual environment context*: a particular component in a distributed system is aware of the services that are available internally and externally, locally and remotely, in the distributed system.

Generally, the context-aware focus of UbiCom systems is on physical world awareness, often in relation to user models and tasks (Section 5.6). Ubiquitous computers can utilise where they are and their physical situation or context in order to optimise their services on behalf of users. This is sometimes referred to as context-awareness in general but more accurately refers to physical context-awareness. A greater awareness of the immediate physical environment could reduce the energy and other costs of physical resource access – making systems more eco-friendly.

Consider the use of the digital camera in the personal visual memories application. It can be aware of its location and time so that it can record where and when a recording is made. Rather than just expressing the location in terms of a set of coordinates, it can also use a Geographical Information System to map these to meaningful physical objects at that location. It can also be aware of its locality so that it can print on the nearest accessible computer.

1.2.4.2 User-Awareness

A camera can be person-aware in a number of ways in order to detect and make sure people are being recorded in focus, so that it configures itself to a person's preferences and interests. These are all specific examples of physical context-awareness.

User context-awareness, also known as person-awareness, refers to ubiquitous services, resources and devices being used to support user-centred tasks and goals. For example, a photographer may be primarily interested in capturing digital memories of people (the user activity goal) rather than capturing memories of places or of people situated in places. For this reason, a UbiCom camera can be automatically configured to detect faces and to put people in focus when taking pictures. In addition, in such a scenario, people in images may be automatically recognised and annotated with names and named human relationships.

Note that the user context-awareness property of a UbiCom system, i.e., being aware of the context of the user, overlaps with the iHCI property. User context-awareness represents one specific sub-type of context-awareness. A context-aware system may be aware of the physical

[16] UbiCom systems may also have an internal system context because a system reflects on its own internal system operation. The internal context may affect adaptation to the external context.

[17] It is not only users who fully determine a system context but other stakeholders such as providers and mediators.

world context, e.g., the location within and the temperature of the environment, and aware of the virtual world or ICT context, e.g., the network bandwidth being consumed for communication (Section 7.6).

In practice, many current devices have little idea of their physical context such as their location and surroundings. The physical context may not be able to be accurately determined or even determined at all, e.g., the camera uses a particular location determination system that does not work indoors. The user context is even harder to determine because the users' goals may not be published and are often weakly defined. For this reason, the user context is often derived from users' actions but these in turn may also be ambiguous and non-deterministic.

1.2.4.3 Active Versus Passive Context-Awareness

A key design issue for context-aware systems is to balance the degree of user control and awareness of their environment (Section 7.2). At one extreme, in a (pure) active context-aware system, the UbiCom system is aware of the environment context on behalf of the user, automatically adjusting the system to the context without the user being aware of it. This may be useful in applications where there are strict time constraints and the user would not otherwise be able to adapt to the context quickly enough. An example of this is a collision avoidance system built into a vehicle to automatically brake when it detects an obstacle in front of it. In contrast, in a (pure) passive context-aware system, the UbiCom system is aware of the environment context on behalf of the user. It just reports the current context to the user without any adaptation, e.g., a positioning system reports the location of a moving object on a map. A passive context-aware system can also be configured to report deviations from a pre-planned context path, e.g., deviations from a pre-planned transport route to a destination. Design issues include how much control or privacy a human subject has over his or her context in terms of whether the subject knows: if his or her context is being acquired, where the context is being kept and to who and what the context is distributed to. Context-awareness is discussed in detail in Chapter 7.

1.2.5 Autonomy

Autonomy refers to the property of a system that enables a system to control its own actions independently. An autonomous system may still be interfaced with other systems and environments. However, it controls its own actions. Autonomous systems are defined as systems that are self-governing and are capable of their own independent decisions and actions. Autonomous systems may be goal- or policy-oriented: they operate primarily to adhere to a policy or to achieve a goal.

There are several different types of autonomous system. On the Internet, an autonomous system is a system which is governed by a router policy for one or more networks, controlled by a common network administrator on behalf of a single administrative entity. A software agent system is often characterised as an autonomous system. Autonomous systems can be designed so that these goals can be assigned to them dynamically, perhaps by users. Thus, rather than users needing to interact and control each low-level task interaction, users only need to interact to specify high-level tasks or goals. The system itself will then automatically plan the set of low-level tasks needed and schedule them automatically, reducing the complexity for the user. The system can also replan in case a particular plan or schedule of tasks to achieve goals cannot be reached. Note the planning problem is often solved using artificial intelligence (AI).

1.2.5.1 Reducing Human Interaction

Much of the ubiquitous system interaction cannot be entirely human-centred even if computers become less obtrusive to interact with, because:

- Human interaction can quickly become a bottleneck to operate a complex system. Systems can be designed to rely on humans being in the control loop. The bottleneck can happen at each step, if the user is required to validate or understand that task step.
- It may not be feasible to make some or much machine interaction intelligible to some humans in specific situations.
- This may overload the cognitive and haptic (touch) capabilities of humans, in part because of the sheer number of decisions and amount of information that occur.
- This original vision needs to be revisited and extended to cover networks of devices that can interact intelligently, for the benefit of people, but without human intervention. These types of systems are called automated systems.

1.2.5.2 Easing System Maintenance Versus Self-Maintaining Systems

Building, maintaining and interlinking individual systems to be larger, more open, more heterogeneous and complex systems is more challenging.[18] Some systems can be relatively simply interlinked at the network layer. However, this does not mean that these can be so easily interlinked at the service layer, e.g., interlinking two independent heterogeneous data sources, defined using different data schemas, so that data from both can be aggregated. Such maintenance requires a lot of additional design in order to develop mapping and mediating data models. Complex system interaction, even for automated systems, reintroduces humans in order to manage and maintain the system.

Rather than design systems to focus on pure automation but which end up requiring manual intervention, systems need to be designed to operate more autonomously, to operate in a self-governed way to achieve operational goals. Autonomous systems are related to both context aware systems and intelligence as follows. System autonomy can improve when a system can determine the state of its environment, when it can create and maintain an intelligent behavioural model of its environment and itself, and when it can adapt its actions to this model and to the context. For example, a printer can estimate the expected time before the printer toner runs out based upon current usage patterns and notify someone to replace the toner.

Note that autonomous behaviour may not necessarily always act in ways that human users expect and understand, e.g., self-upgrading may make some services unresponsive while these management processes are occurring. Users may require further explanation and mediated support because of perceived differences between the system image (how the system actually works) and users' mental model of the system (how users understand the system to work, see Section 5.5.5).

From a software engineering system perspective, autonomous systems are similar to functionally independent systems in which systems are designed to be self-contained, single-minded, functional, systems with high cohesion[19] and that are relatively independent of other systems (low-coupling)

[18] The operating system software alone can contain over 30 million lines of code and require 4000 programmers for development (Horn, 1999). The operating system is just one part of the software for a complex distributed system. There is also the set of operating system utilities to consider.

[19] Cohesion means the ability of multiple systems or system components to behave as a single unit with respect to specific functions.

(Pressman, 1997). Such systems are easier to design to support composition, defined as atomic modules that can be combined into larger, more complex, composite modules. Autonomous system design is covered in part in Chapter 10.

1.2.6 Intelligence

It is possible for UbiCom systems to be context-aware, to be autonomous and for systems to adapt their behaviour in dynamic environments in significant ways, without using any artificial intelligence in the system. Systems could simply use a directory service and simple event condition action rules to identify available resources and to select from them, e.g., to discover local resources such as the nearest printer. There are several ways to characterise intelligent systems (Chapter 8). Intelligence can enable systems to act more proactively and dynamically in order to support the following behaviours in UbiCom systems:

- *Modelling of its physical environment*: an intelligent system (IS) can attune its behaviour to act more effectively by taking into account a model of how its environment changes when deciding how it should act.
- *Modelling and mimicking its human environment*: it is useful for a IS to have a model of a human in order to better support iHCI. IS could enable humans to be able to delegate high-level goals to the system rather than interact with it through specifying the low-level tasks needed to complete the goal.
- *Handling incompleteness*: Systems may also be incomplete because environments are open to change and because system components may fail. AI planning can support re-planning to present alternative plans. Part of the system may only be partially observable. Incomplete knowledge of a system's environment can be supplemented by AI type reasoning about the model of its environment in order to deduce what it cannot see is happening.
- *Handling non-deterministic behaviour*: UbiCom systems can operate in open, service dynamic environments. Actions and goals of users may not be completely determined. System design may need to assume that their environment is a semi-deterministic environment (also referred to as a volatile system environment) and be designed to handle this. Intelligent systems use explicit models to handle uncertainty.
- *Semantic and knowledge-based behaviour*: UbiCom systems are also likely to operate in open and heterogeneous environments. Types of intelligent systems define powerful models to support interoperability between heterogeneous systems and their components, e.g., semantic-based interaction.

Types of intelligence can be divided into individual properties versus multiple entity intelligence properties (see Table 1.5).

1.2.7 Taxonomy of UbiCom Properties

There are many different examples of defining and classifying ubiquitous computing. Weiser (1991) referred to UbiCom by function in terms of being distributed, non-obtrusive to access and context-aware. The concept of UbiCom is related to, and overlaps with, many other concepts, such as pervasive computing, sentient computing, context-aware computing, augmented reality and ambient intelligence. Sentient computing is regarded as a type of UbiCom which uses sensors to perceive its environment and to react accordingly. Chen and Kotz (2000) considers context-awareness use as more specifically applied to mobile computing in which applications can discover and take advantage of contextual information (such as user location, time of day, nearby people and devices, and user activity). Context-aware computing is also similar to sentient computing, as is agent-based computing in which agents construct and maintain a model of their environment to more effectively act in it. Ambient intelligence (ISTAG, 2003) characterises systems in terms of supporting the properties of intelligence using ambience and iHCI. Aarts and Roovers (2003) define the five key features of ambient intelligence to be embedded, context-aware, personalised, adaptive and anticipatory.

Buxton (1995) considers ubiquity and transparency to be the two main properties of UbiCom. Aarts and Roovers (2003) classify ubiquitous systems in terms of disposables (low power, low bandwidth, embedded devices), mobiles (carried by humans, medium bandwidth) and statics (larger, stationary devices with high-speed wired connections. Endres *et al*. (2005) classify three types of UbiCom System: (1) distributed mobile systems; (2) intelligent systems (but their focus here is more on sensor and embedded systems rather than on intelligence *per se*); and (3) augmented reality. Milner (2006) considers the three main characteristics of UbiCom as follows: (1) they are capable of making decisions without humans being aware of them, i.e., they are autonomous systems and support iHCI; (2) as systems increase in size and complexity, systems must adapt their services, and (3) more complex unplanned interaction will arise out of interactions between simple independent components, i.e., emergent behaviour.

Rather than debate the merits or select particular definitions of UbiCom, the main properties are classified into five main types or groups of system properties to support the five main requirements for ubiquitous computing (see Figure 1.2). These groups of properties are not exclusive. Some of these sub-types could appear in multiple types of group. Here are some examples. Affective or emotive computing can be regarded as sub-types of IHCI and as sub-types of human intelligence. There is often a strong notion of autonomy associated with intelligence as well as being a more distributed system notion. Goal-oriented systems can be regarded as a design for intelligence and as a design for iHCI. Orchestrated and choreographed can be regarded as a way to compose distributed services and as a way to support collective rational intelligence. Personalised can be regarded as sub-type of context-awareness and as a sub-type of iHCI.

Different notions and visions for ubiquitous computing overlap. There are often different compositions of more basic types of properties. Ambient intelligence, for example, combines embedded autonomous computing systems, iHCI and social type intelligent system. Asynchronous communication enables the components in distributed systems to be spatially and temporally separated but it also enables automatic systems to do more than simply react to incoming events, to support anytime interaction.

Some properties are similar but are referred to by different terms. The terms pervasive computing and ambient computing are considered to be synonymous with the term ubiquitous computing. Systems are available anywhere and anytime, to anyone, where and when needed. UbiCom is not intended to mean all physical world resources, devices and users are omnipresent, available everywhere, at all times, to everybody, irrespective of whether it is needed or not. Ubiquity to be useful is often context-driven, i.e., local ubiquity or application domain bounded ubiquity.

The taxonomy proposed in this text is defined at three levels of granularity. At the top level five core properties for UbiCom systems are proposed. Each of these core properties is defined in terms of over 70 sub-properties give in Tables 1.1–1.5. These tables describe more finely grained properties of UbiCom systems and similar ones.[20] Thus a type of distributed UbiCom can be defined in terms of being networked and mobile. Several of these sub-properties defined are themselves such rich concepts that they themselves can be considered in terms of sub-sub-properties. For example, communication networks (Chapter 11) include sub-properties such as wired or wireless, service-oriented or network oriented, etc. Mobility (Chapter 4) can be defined

[20] Without formal definitions of terms at this stage, it is not possible to say that terms are equivalent and synonyms.

Table 1.1 Distributed system properties

Distributed System, middleware, set of generic services	
Universal, seamless, heterogeneous	Able to operate across different homogeneous environments, seamless integration of devices and environments, taking on new contexts when new resources become available (Sections 3.2, 3.3)
Networked	UbiCom devices are interlinked using a network which is often wireless (Chapter 11)
Synchronised, coordinated	Multiple entity interaction can be coordinated synchronously or asynchronously over time and space interactions (Section 3.3.3.2)
open	New components can be introduced and accessed, old ones can be modified or retired. Components can be dynamically discovered (Section 3.3.2)
Transparent, virtual	Reduces the operational complexity of computing, acting as a single virtual system even although it is physically distributed (Section 3.4.1)
Mobile, nomadic	Users, services, data, code and devices may be mobile (Sections 4.2, 11.7.5)

Table 1.2 iHCI system properties

Implicit Human–Device Interaction (iHCI)	
Non-intrusive, hidden, invisible, calm computing	ICT is nonintrusive and invisible to the user. It is integrated into the general ecology of the home or workplace and can be used intuitively by users (Section 5.7)
Tangible, natural	Interaction is via natural user interfaces and physical artefact interaction that can involve gestures, touch, voice control, eye gaze control, etc. (Section 5.3)
Anticipatory, speculative, proactive	Improving performance and user experience through anticipated actions and user goals in relation to current context, past user context and group context. This overlaps with user context-awareness (Sections 5.6 and 7.2)
Affective, emotive	Computing that relates to, arises from, or influences human emotions. This is also considered to be a sub-type of human intelligence (Section 5.7.4)
User-aware	ICT is aware of presence of user, user ID, user characteristics, current user tasks in relation to users' goals (as part of iHCI and context-awareness)
Post-human	Sense of being in a world that exists outside ourselves, extending a person's normal experience across space and time (Section 5.4.1)
Sense of presence immersed, virtual, mediated reality	A person is in a real-time interactive environment which experiences an extended sense of presence that combines the virtual and the real, often by overlaying virtual views on real views (Section 5.4.4)

in terms of sub-sub-sub-properties of mobile services mobile code, and mobile hardware resources and devices and in terms of being accompanied, wearable and implanted or embedded into mobile hosts. Over 20 different sub-sub-properties for autonomic and self-star computing are described (Section 10.4).

These groups of properties act to provide a higher level of abstraction of the important characteristics for analyzing and designing ubiquitous systems. It is assumed that generic distributed system services such as directory services and security would also be needed and these may be need to be designed and adapted for ubiquitous computing use.

Table 1.3 Context-aware system properties

Context-aware	
Sentient, unique, localized, situated	Systems can discover and take advantage of the situation or context such as: location, time and user activity. There are three main sub-types of context-awareness; physical-world, user and virtual (ICT) device awareness
Adaptive, active context-aware	Systems actively adapt to context changes in a dynamic environment rather than just present context changes to the user (Section 7.2.4)
Person-aware, user-aware, personalised, tailored,	Tailored to an individual user or type of user, based on personal details or characteristics that a user provides or is gathered about a user. This may trigger system adaptation (Sections 5.6, 5.7)
Environment-aware, context-aware, physical context-aware	Sometimes physical world context-aware awareness is taken by some researchers to mean general context-awareness or general environment awareness. Physical context-awareness includes spatial and temporal awareness (Sections 7.4, and 7.5)
ICT awareness	Awareness of ICT infrastructure in which an UbiCom system exists, e.g., awareness of network QoS when transmitting messages (Section 7.6)

Table 1.4 Autonomous system properties

Autonomous	
Automatic	Operates without human intervention (Section 10.2.1.1)
Embedded, encapsulated embodied	System input-output and computation is completely encapsulated by, or contained in, the device it controls, e.g., a system that acts as a self-contained appliance (Section 6.5)
Resource-constrained	Systems are designed to be constrained in size to be portable or embeddable; to use constrained computation, data storage, input and output and energy (Section 13.5.2)
Untethered, amorphous	Able to operate independently and proactively, free from external authority, external dependencies are minimized (Sections 2.2.3.2, 6.4.4)
Autonomic, self-managing, self-star	Able to support various self-star properties such as self-configuring, self-healing, self-optimising and self-protecting behaviour (Section 10.4)
Emergent, self-organising	More complex behaviour can arise out of multiple simple behaviours (Section 10.5)

Table 1.5 Intelligent system properties

Individual Intelligent Systems	
Reactive, reflex[1]	Environment events are sensed. Events then trigger action selection that may lead to actuators changing their environments (Section 8.3.2)
Model-based,Rule/ policy-based logic/reasoning	Systems use a model of how itself operates and the how the world works (Section 8.3.3), There are many types of model representation such as rule-based, different types of logic-based, etc.
goal-oriented, planned, proactive	User goals can be used to plan actions dynamically rather than pre-programmed actions (Section 8.3.4)
Utility-based, game theoretic	Systems can be designed to handle multiple concurrent goals (Section 8.3.5)
Learning, adaptive	Systems can be designed to improve their own performance (Section 8.3.6)

Table 1.5 (*continued*)

Individual Intelligent Systems	
Multiple Intelligent System, Collective or Social Intelligence	
Cooperative collaborative, benevolent	Multiple agents can share tasks and information in order to achieve shared goals (Section 9.2.3)
Competitive, self-interested, antagonistic, adversarial	Individual agents and organizations have private goals and utility functions that they seek to achieve in a multi-entity setting without requiring collaboration Entities could also act malevolently (Section 9.2.4)
Orchestrated, choreographed, mediated	Multiple interactions can: be controlled and ordered by designating some leader (orchestrated) who acts as a central-planer; allow some freedom of interaction by participants (choreographed) or constrained by the use of some common entity or resource (mediated) (Sections 3.3.2, 9.2.2)
Task-sharing	
Communal, shared meaning	System interaction is sharable, commonly understood within a limited or well-defined domain (Section 8.4)
Shared knowledge	
Speech-act based,[2] intentional, mentalistic.	Multiple agents interact based upon propositional attitudes, i.e., relationships based upon beliefs, desires or wants and intentions[3] (Section 9.3.3.4)
Emergent	Organizations lead to levels of interaction that are not level of the individual interactions (Section 9.2.3.3)

[1] Note a reflex system is different from a reflective system, whereas the former type of system is designed to react to environment stimuli, the latter type of system is designed to think about what it is doing (Section 10.3).
[2] A speech act-based system is different from a speech-based system – whereas the former using a particular linguistic theory to form sentence like structures, the latter is a system that can process human speech input and or convert its output to human-like speech.
[3] Although mental and intentional computing seems like a form of human intelligence, it is usually deployed in terms of a rational model such as such as a BDI type logic (Chapter 7).

Each individual property has its own domain of a more finely grained set of discrete values, rather than being seen as a property that is present or absent. Here are some examples:

- from wireless to wired, ad hoc to fixed and from client-server to P2P communication;
- from full local access only, to partial remote access, to full remote access;
- from asynchronous, to synchronous, to coordinated, to organisational to conventions;
- from mobility ranging from: being static at place of manufacture; moved to the place of installation, e.g., embedded then static; mobile between sessions but static during sessions; mobile (roaming from home) during sessions; to being free roaming without a home, untethered;
- from transportable to portable to hand-held to wearable to implants;
- from macro to micro to nano;
- from fully integrated, to embedded inside, to surface-mounted, to various forms of loose attachments such as amorphous computing;
- from total physical reality, to augmenting reality with virtual reality, to mediated reality, to pure virtual reality to synthetic reality;
- from operating as individuals, to operating as part of societal groups, to globally interacting.

In Section 13.2.2, a multi-lateral model that offers different degrees of support for Ubiquitous computing properties from minimal support to full support is proposed.

There are a few closing remarks about the terminology and meaning of the system properties and concepts. Different fields of computer science may use the same term differently. For example, when HCI refers to the adaptive system, it means the focus of the adaptation is the front end of the system or the UI which is adapting to human behaviour, and the adaptation is driven fully by external concerns. In artificial intelligence, an adaptive system often refers to a system which incorporates machine learning so that the system can improve or adapt its performance over time. There are also many nuances and different contexts of the use of the terms. The grouping of terms in the left column of Tables 1.1–1.5 indicates that these terms have a strong overlap and similarity. This does not necessarily mean that terms within a grouping are fully equivalent to each other.

1.3 Ubiquitous System Environment Interaction

At a high level of abstraction, we can distinguish three types of system environment[21] for each particular UbiCom system: (1) other UbiCom systems which form the ICT infrastructure, supporting services and act as middleware for that particular ICT system applications (virtual worlds);[22] (2) human individuals and human organisations; and (3) physical world[23] systems including ecological and biologic systems. Together, the virtual (computer) environment humans and the physical world can be considered as forming an external environment for UbiCom systems. Note that each of these three main environments appear to have quite different design models and processes. Physical world phenomena are governed by well-known laws of physical forces such as electromagnetism and gravity. Living entities in the physical world are governed by ecological models of their habitat. Human living entities are often governed by economic and social models.

A UbiCom system is often organised conventionally as a layered information system stack with a bottom layer of information resources, a middle layer of processing and a top layer of user information abstractions to view and interact with the information. A common communications pipe allows these to be distributed in different ICT systems.

Humans who own and operate the UbiCom systems and are situated in the physical world regard the physical world and ICT devices as their human environment. Humans perceive and act on their environment, often through visual and touch senses. Their actions can be driven by a world model that guides their actions, consisting of prior experiences that are learnt. The physical environment can be represented using multiple models. In a local physical control model, e.g., lighting controls can sense the existing natural lighting and switch on artificial lighting when the natural light is below a certain threshold. More sophisticated control systems can use feedback control. A second type of physical world model is an ecology system, a self-sustaining system inhabited by multiple autonomous organisms that self-regulate their behaviour in the face of different driving pressures and events in the system.

There are three basic types of environment for UbiCom systems: (1) the infrastructure of other ICT systems; (2) the physical world environment; and (3) the human environment. Several basic types of system environment interaction occur: between humans and ICT systems, HCI (see Figure 1.3); between ICT systems and the physical world, CPI (see Figure 1.4); between ICT systems, C2C or CCI. In addition, interactions can occur between the non-ICT systems such as between different physical world entities and between humans (H2H or HHI), also called social interaction. These types of interaction all coexist. The interrelation and simplification of these interactions is

[21] These three types of environment are also collectively known as *physical space, cyber space and mental space* or as the world of *atoms, bits and minds*.

[22] The ICT system infrastructure is also referred to as a virtual environment or as cyberspace.

[23] Anything that exists in a physical space, natural or artificial, inherently occupies physical space and consumes physical resources, e.g., a desktop computer often rests on a hard physical surface and consumes energy generated by other physical world resources. The nature of the physical world in our model is as an external environment to an UbiCom system which the UbiCom system may sense and control in order to support some specific application or use.

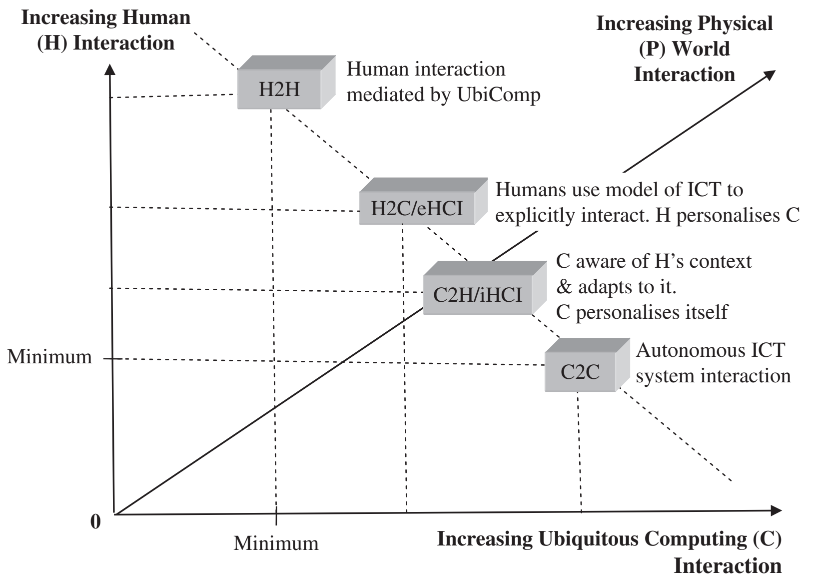

Figure 1.3 Human–ICT device interaction (HCI) is divided into four sub-types of interaction H2H, H2C, C2H and C2C

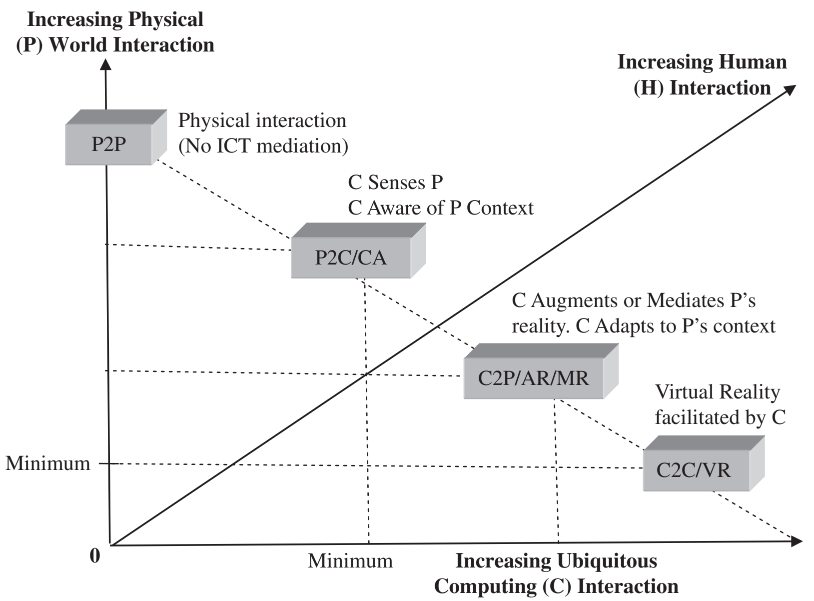

Figure 1.4 ICT device and Physical World Interaction (CPI) is divided into four sub-types of interaction: P2P, P2C, C2P and C2C

discussed further in Section 9.1. Of these three environments, humans have the highest intelligence overall and can act the most autonomously. This needs to be taken into account. Humans are an embodiment of parts of the physical world but also cause the most changes to the physical world. ICT devices are manufactured from the physical world and act in it. ICT devices have a profound effect on humans leading to changes in societal values and norms.[24]

Each of the types of interaction of HCI and CPI is illustrated by describing four degrees of interactions that span their interaction domain. This division of interactions into human–physical, physical–computer and human–computer interaction is a framework to analyse the range of UbiCom systems. Some interactions may span and combine interactions, e.g., mediated reality interaction combines human-physical and human-computer interaction.

1.3.1 Human–ICT Device Interaction (HCI)

For the interaction between two systems, e.g., Humans (H) and ICT systems (C), four characteristic points across the interaction domain are considered, maximum H interaction (H2H or HHI) with minimal C interaction, more H interaction facilitated by C interaction (H2C), more C interaction that leads to human interaction (C2H) and maximum ICT interaction (C2C). Whereas in H2C, Humans have a model of computers, e.g., H has a mental model of C's tasks and goals in order to interact more effectively with the use of C. With C2H, C also has a partial model of H in order to reduce H's interaction (Section 1.3.3). ICT device to physical world interaction and human to physical world interaction are described in a similar way.

ICT device to ICT device (C2C), also called distributed computer systems, is the main focus of computer science, telecoms and networks but we take a much wider perspective here. C2C facilitates all the other types of interaction below, e.g., H2H is often mediated by C2C. C2C is often used to automate tasks and is used for pre-processing to filter out unneeded resources or to filter needed resources transparently to the user.

CCI, in turn, depending on how this is defined, interlinks and requires human interaction. Human interaction is required in different parts of the life-cycle of a CCI system. Humans are involved in the design phase of the UbiCom system, often performing some inspection phase during operations and are involved in the maintenance phase when changes to the design are needed to maintain the system operation (Horn, 2001). Kindberg and Fox (2002) state that in the short term, UbiCom systems will involve humans and that system designers should make clear the system boundary[25] between the ICT system and the human, making clear the responsibilities of both, i.e., what ICT system cannot do and what humans will do.

Humans use multiple devices, explicitly personalised by the user, that are situated in a human's personal space and social space.[26] Humans explicitly access non-interactive and interactive multimedia information services for entertainment and leisure. Humans explicitly access business (enterprise)-

[24] For example, in the 1990s, if two strangers were walking down the street and one starting talking, saying hello, one would think the other was talking to him or her and reply. In the 2000s, if this happens, there is a higher probability that the person is talking on a mobile phone to someone else remotely, so it is more more likely that the other will stay silent.

[25] Kindberg and Fox (2002) refer to this ICT system–human boundary as the *semantic Rubicon*, named after the River Rubicon in Italy in ancient times that marked the boundary of part of what was Italy, where Julius Caesar hesitated before crossing it with his troops into Italy.

[26] See Hall (1996), personal space is the region surrounding each person, or that area which a person considers their domain or territory. It is determined to be up to about 120 cm. This personal space travels with us as we move. These spaces are fluid and multi-functional. Personal space also overlaps with social space (used for interacting with acquaintances, up to about 360 cm) and with public space. The specific dimensions and use of these spaces vary, e.g., with culture and age.

related information away from the office and various other supporting virtual ICT services for education, personal productivity, etc. (Explicit Human to ICT Device Interaction H2C or eHCI).

ICT applications can use a model of the person, perhaps created and maintained based upon observed user interaction, and their activities (Implicit ICT Device to Human Interaction, C2H or iHCI). The C model of H can be used to inform users of timely activities, to automatically filter and adapt information and tasks, and to anticipate human actions and interactions and adapt to them.

Social and organisational interaction can be mediated by ICT devices (Human to Human, Social, Interaction, H2H). Two humans may interact, one to one, e.g., unicast voice calls between two people. Computers may facilitate basic information and task-sharing but computers can also be used to facilitate richer sharing of language, knowledge, experiences and emotions. Humans may interact within social spaces and within enterprise organisational spaces, e.g., to support intra or inter-organisational work-flows and to complete as well as cooperate together to attain resources, e.g., interact within auctions to accrue goods.

1.3.2 ICT Device to Physical World Interaction (CPI)

Physical World to Physical World Interaction (P2P) refers to interactions within nature that are (as yet) not mediated by any significant ICT system. There are a variety of simple animal life interactions used in nature, in contrast to the more complex human to human interaction. These involve shared chemical scents, visual signage and different types of audio signals such as drumming, buzzing and vocal calls. While this type of biological interaction appears to be quite esoteric, models of this interaction can be mimicked in CCI and can be surprisingly effective at solving some interaction problems. In addition, the ways that organisms interact within their natural habitat to maintain a balanced ecosystem are quite effective models for self-regulation of autonomous systems (Chapter 10).

Physical Environment to Computer Device Interaction (P2C) covers context-aware ICT systems. These can be designed to be aware of changes in specific physical world phenomena and to react to this in simple ways, e.g., if the temperature is too low, turn up the heating. ICT systems can also be designed to act on the physical world, changing the state of part of the physical world, according to human goals.

Computer Device to Physical Environment Interaction (C2P) refers to augmented and mediated reality systems. ICT systems are used to augment, to add to, physical reality, e.g., physical world views can be annotated with virtual markers. In the more general mediated reality cases, reality may be diminished and filtered not just enhanced. The interplay between physical world and virtual world reality is a strong theme in electronic games. The term hyperreality is used to characterise the inability of human consciousness to distinguish reality from fantasy, a level of consciousness that can be achieved by some electronic games players.

In pure virtual reality interaction, Computer to Computer Interaction (CCI), the physical world may be used as a conceptual space for virtual interaction. In a virtual reality ICT system, humans can use sensory interfaces such as gloves and goggles to be interfaced to support more natural interaction. Humans may also contain implants for medical conditions that can transmit digital data streams into ICT medical monitoring services. Humans can be represented virtually as avatars in order to explore and interact more richly in virtual ICT worlds.

It is also noted that there is some minimal physical world interaction even with maximum CCI as computers consume physical world resources, e.g., energy. CCI is affected by physical world phenomena, wireless and wired signals will become attenuated to different degrees, partially dependent on their frequency. There is in addition increasing awareness of UbiCom systems operating as part of the physical world ecology, in harmony with it. This must occur throughout the full life-cycle of the UbiCom system including operation (optimising energy use) and destruction (through remanufacturing and recycling).

1.4 Architectural Design for UbiCom Systems: Smart DEI Model

Three basic architectural design patterns for ubiquitous ICT system: smart devices, smart environment[27] and smart interaction are proposed (Figure 1.5). Here the concept smart simply means that the entity is active, digital, networked, can operate to some extent autonomously, is reconfigurable and has local control of the resources it needs such as energy, data storage, etc. It follows that these three main types of system design may themselves contain sub-systems and components at a lower level of granularity that may also be considered smart, e.g., a smart environment device may consist of smart sensors and a smart controller, etc. There is even smart dust (Section 2.2.3.2). An illustrative example of how these three types of models can be deployed is given in Figure 1.5.

These are many examples of sub-types[28] of smarts for each of the three basic types of smarts which are discussed in detail in the later chapters of this book. The three main types of smart design also overlap, they are not mutually exclusive. Smart devices may also support smart interaction. Smart mobile start devices can be used for control in addition to the use of static embedded environment devices. Smart devices may be used to support the virtual viewpoints of smart personal (physical environment) spaces in a personal space that accompanies the user wherever they are.

Satyanarayanan (2001) has also postulated different architectures and paths for developing UbiCom systems, first, to evolve from distributed systems, mobile distributed systems into ubiquitous computing and, second, to develop UbiCom systems from smart spaces characterised by

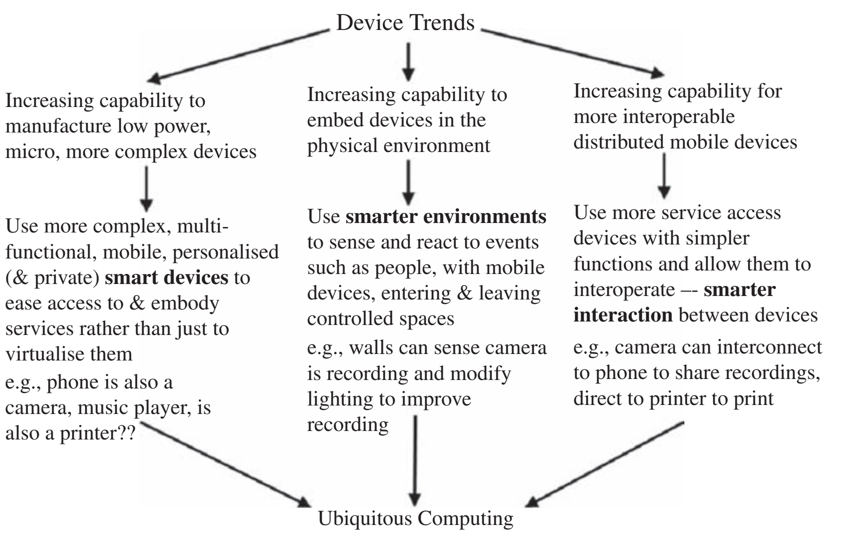

Figure 1.5 Three different models of ubiquitous computing: smart terminal, smart interaction, and smart infrastructure

[27] Note: some people just consider the smart environment model to comprise ubiquitous computing but here ubiquitous computing is also considered to comprise the smart device model, e.g., mobile communicators, and smart interaction model.

[28] Further levels of granularity of the sub-types of smarts could be added, e.g., sub-types of smart embedded environments devices such as implants but these are not indicated in order to simplify Figure 1.6.

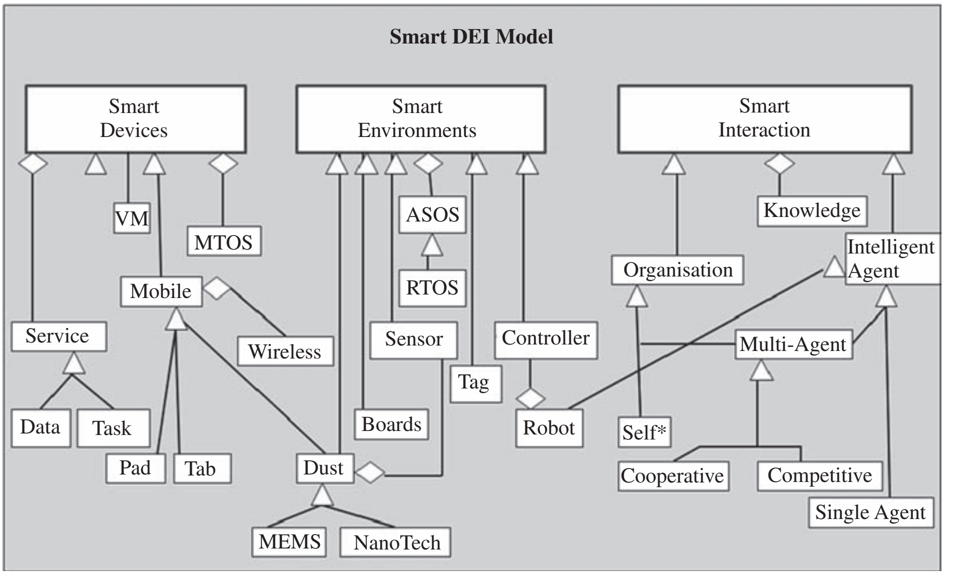

Figure 1.6 Some of the main subtypes (triangle relationships) of smart devices, environments and interactions and some of their main aggregations (diamond relationships) where MTOS is a Multi-Tasking Operating System, VM is a Virtual Machine, ASOS is an Application Specific or embedded system OS, RTOS is a Real-Time OS and MEMS is a Micro ElectroMechanical System

invisibility, localised scalability and uneven conditioning. The Smart DEI model is similar to Satyanarayanan's, except it also incorporates smart interaction. Smart DEI also refers to hybrid models that combine the designs of smart device, smart environments and smart interaction (Figure 1.6). Gillett *et al*. (2000) speculate that general purpose end-user equipment will endure but evolve into a more modular form, driven by user frustration with a proliferation of devices with overlapping functionality and the desire for consistency across multiple environments (such as home, car and office). This is motivation for smart interaction rather the smart device model. However, in practice, users appear to be very tolerant of the vast majority of devices with overlapping functions that are inconsistent and often non interoperable.

1.4.1 Smart Devices

Smart devices, e.g., personal computer, mobile phone, tend to be multi-purpose ICT devices, operating as a single portal to access sets of popular multiple application services that may reside locally on the device or remotely on servers. There is a range of forms for smart devices. Smart devices tend to be personal devices, having a specified owner or user. In the smart device model, the locus of control and user interface reside in the smart device. The main characteristics of smart devices are as follows: mobility, dynamic service discovery and intermittent resource access (concurrency, upgrading, etc.). Devices are often designed to be multi-functional because these ease access to, and simplify the interoperability of, multi-functions at run-time. However, the trade-off is in a decreased openness of the system to maintain (upgrade) hardware components and to support more dynamic flexible run-time interoperability.

1.4.1.1 Weiser's ICT Device Forms: Tabs, Pads and Boards

We often think of computers primarily in terms of the multi-application personal or server computers, as devices with some type of screen display for data output and a keyboard and some sort of pointing devices for data input. As humans, we routinely interact with many more devices that have single embedded computers in them, such as household appliances, and with complex machines[29] that have multiple embedded computers in them. Weiser noted that there was a trend away from many people per computer,[30] to one computer per person,[31] through to many computers per person.

Computer-based devices tend to become smaller and lighter in weight, cheaper to produce. Thus devices can become prevalent, made more portable and can appear less obtrusive. Weiser considered a range of device sizes in his early work from wearable centimetre-sized devices (tabs), to hand-held decimetre-sized devices (pads) to metre-sized (boards) displays. ICT Pads to enable people to access mobile services and ICT tabs to track goods are in widespread use. Wall displays are useful for viewing by multiple people, for collaborative working and for viewing large complex structures such as maps. Board devices may also be used horizontally as surface computers as well used in a vertical position.

1.4.1.2 Extended Forms for ICT Devices: Dust, Skin and Clay

The three forms proposed by Weiser (1991) for devices, tabs, pads and boards, are characterised by: being macro-sized, having a planar form and by incorporating visual output displays. If we relax each of these three characteristics, we can expand this range into a much more diverse and potential more useful range of ubiquitous computing devices.

First, ICT devices can be miniaturised without visual output displays, e.g., Micro Electro-Mechanical Systems (MEMS), ranging from nanometres through micrometers to millimetres (Section 6.4). This form is called Smart Dust. Some of these can combine multiple tiny mechanical and electronic components, enabling an increasing set of functions to be embedded into ICT devices, the physical environment and humans. Today MEMS, such as accelerometers, are incorporated into many devices such as laptops to sense falling and to park moving components such as disk arms, are being increasingly embedded into widely accessed systems. They are also used in many devices to support gesture-based interaction. Miniaturisation accompanied by cheap manufacturing is a core enabler for the vision of ubiquitous computing (Section 6.4).

Second, fabrics based upon light-emitting and conductive polymers, organic computer devices, can be formed into more flexible non-planar display surfaces and products such as clothes and curtains (Section 5.3.4.3). MEMS devices can also be painted onto various surfaces so that a variety of physical world structures can act as networked surfaces of MEMS (Section 6.4.4). This form is called Smart Skins.

Third, ensembles of MEMS can be formed into arbitrary three-dimensional shapes as artefacts resembling many different kinds of physical object (Section 6.4.4). This form is called Smart Clay.

[29] For example, new cars have several tens of embedded computers and sensors to support assisted braking, airbag inflation, etc.

[30] Thomas J. Watson, who led the world's first and largest computer company, IBM, from the 1920s to the 1950s, is alleged to have made the statement in 1943 that: 'I think there is a world market for maybe five computers.' This would mean a ratio of one computer to about a billion people.

[31] The one-computer-to-one-person phase may not have existed for any significant period depending on the definition of a computer. Certainly, people who had personal computers, also had many embedded digital devices at that time too.

1.4.1.3 Mobility

Mobile devices usually refer to communicators, multimedia entertainment and business processing devices designed to be transported by their human owners, e.g., mobile phone, games consoles, etc. There is a range of different types of mobiles as follows:

- *Accompanied*: these are devices that are not worn or implanted. They can either be portable or hand-held, separate from, but carried in clothes or fashion accessories.
- *Portable*: such as laptop computers which are oriented to two-handed operation while seated. These are generally the highest resource devices.
- *Hand-held*: devices are usually operated one handed and on occasion hands-free, combining multiple applications such as communication, audio-video recording and playback and mobile office. These are low resource devices.
- *Wearable*: devices such as accessories and jewellery are usually operated hands-free and operate autonomously, e.g., watches that act as personal information managers, earpieces that act as audio transceivers, glasses that act as visual transceivers and contact lenses. These are low resource devices (Sections 2.2.4.5, 5.4.3).
- *Implanted or embedded*: these are often used for medical reasons to augment human functions, e.g., a heart pacemaker. They may also be used to enhance the abilities of physically and mentally able humans. Implants may be silicon-based macro- or micro-sized integrated circuits or they may be carbon-based, e.g., nanotechnology (Section 6.4).

Static can be regarded as an antonym for mobile. Static devices tend to be moved before installation to a fixed location and then reside there for their full operational life-cycle. They tend to use a continuous network connection (wired or wireless) and fixed energy source. They can incorporate high levels of local computation resources, e.g., personal computer, AV recorders and players, various home and office appliances, etc. The division between statics and mobiles can be more finely grained. For example, statics could move between sessions of usage, e.g., a mobile circus containing different leisure rides in contrast to the rides in a fixed leisure park. Mobile ICT is discussed in detail in Chapter 4.

1.4.1.4 Volatile Service Access

Mobiles tend to use wireless networks. However, mobiles may be intermittently connected to either wireless networks (WAN is not always available) or to wired communications networks (moving from LAN to LAN) or to both. Service access by smart mobile devices is characterised as follows.

Intermittent (service access) devices access software services and hardware intermittently. This may be because resources are finite and demand exceeds supply, e.g., a device runs out of energy and needs to wait for it to be replenished. This may be because resources are not continually accessible.

Service discovery: devices can dynamically discover available services or even changes in the service context. Devices can discover the availability of local access networks and link via core networks to remote network home services. They can discover local resources and balance the cost and availability of local access versus remote access to services. Devices can be designed to access services that they discover on an intermittent basis. Context-aware discovery can improve basic discovery by limiting discovery to the services to the ones of interest, rather than needing to be notified of many services that do not match the context.

With asymmetric remote service access, more downloads than uploads, tends to occur. This is in part due to the limited local resources. For example, because of the greater power needed to transmit rather than receive communication and the limited power capacity, high power consumption results in more received than sent calls. Apart from the ability to create and transmit voice signals, earlier phones were designed to be transreceivers and players. More recently, because of miniaturisation, mobile devices not only act as multimedia players, they can also act as multimedia recorders and as content sources.

1.4.1.5 Situated and Self-Aware

Smart devices although they are capable of remote access to any Internet services, tend to use various contexts to filter information and service access. For examples, devices may operate to focus on local views of the physical environments, maps, and to access local services such as restaurants and hotels.

Mobiles are often designed to work with a reference location in the physical environment called a home location, e.g., mobile network nodes report their temporary location addresses to a home server which is used to help coordinate the mobility. Service providers often charge access to services for mobile service access based upon how remote they are with respect to a reference ICT location, a home ICT location. During transit, mobiles tend to reference a route from a start location to a destination location.

Mobile devices support limited local hardware, physical, and software resources in terms of power, screen, CPU, memory, etc. They are ICT resource constrained. Services that are accessed or pushed to use such devices must be aware of these limitations, otherwise the resource utilisation by services will not be optimal and may be wasted, e.g., receiving content in a format that cannot be played. In the latter case, the mobile device could act as an intermediary to output this content to another device where it can be played.

Mobile devices tend to use a finite internal energy cache in contrast to an external energy supply, enhancing mobility. The internal energy supply may be replenished from a natural renewal external source, e.g., solar power or from an artificial energy gird: energy self-sufficiency. This is particularly important for low-maintenance, tetherless devices. Devices can automatically configure themselves to support different functions based upon the energy available. Without an internal energy cache, the mobility of devices may be limited by the length of a power cable it is connected to.

There is usually a one-to-one relationship between mobiles and their owners. Devices' configuration and operation tends to be personalised, to support the concept of a personal information and service space which accompanies people where ever they are.

1.4.2 Smart Environments

In a smart environment, computation is seamlessly used to enhance ordinary activities (Coen, 1998). Cook and Das (2007) refer to a smart environment as 'one that is able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment'. A smart environment consists of a set of networked devices that have some connection to the physical world. Unlike smart devices, the devices that comprise a smart environment usually execute a single predefined task, e.g., motion or body heat sensors coupled to a door release and lock control. Embedded environment components can be designed to automatically respond to or to anticipate users' interaction using iHCI (implicit human–computer interaction), e.g., a person walks towards a closed door, so the door automatically opens. Hence, smart environments support a bounded, local context of user interaction.

Smart environment devices may also be fixed in the physical world at a location or mobile, e.g., air-born. Smart environments could necessitate novel and revolutionary upgrades to be incorporated into the environment in order to support less obtrusive interaction, e.g., pressure sensors can be incorporated into surfaces to detect when people sit down or walk. A more evolutionary approach could impart minimal modifications to the environment through embedding devices such as surface mounted wireless sensor devices, cameras and microphones.

1.4.2.1 Tagging, Sensing and Controlling Environments

Smart environment devices support several types of interaction with environments such as the physical environment (Chapter 6) as follows:

- *Tagging and annotating the physical environment*: tags, e.g., RFID[32] tags, can be attached to physical objects. Tag readers can be used to find the location of tags and to track them. Virtual tags can be attached to virtual views of the environment, e.g., a tag can be attached to a location in a virtual map.
- *Sensing or monitoring the physical environment*: Transducers take inputs from the physical environment to convert some phenomena in the physical world into electrical signals that can be digitised, e.g., how much ink is in a printer's cartridges. Sensors provide the raw information about the state of the physical environment as input to help determine the context in a context-aware system. Sensing is often a pre-stage to filtering and adapting.
- *Filtering*: a system forms an abstract or virtual view of part of its environment such as the physical world. This reduces the number of features in the view and enables viewers to focus on the features of interest.
- *Adapting*: system behaviour can adapt to the features of interest in the environment of adapt to changes in the environment, e.g., a physical environment route is based upon the relation of the current location to a destination location.
- *Controlling the physical world*. Controllers normally require sensors to determine the state of the physical phenomena e.g., heating or cooling systems that sense the temperature in an environment. Controlling can involve actions to modify the state of environment, to cause it to transition to another state. Control may involve changing the order (assembly) of artefacts in the environment or may involve regulation of the physical environment.
- *Assembling*: robots are used to act on a part of the physical world. There is a variety of robots. They may be pre-programmed to schedule a series of actions in the world to achieve some goal, e.g., a robot can incorporate sensors to detect objects in a source location, move them and stack them in a destination location (palletisation). Robots may be stationary, e.g., a robot arm, or be mobile.
- *Regulating*: Regulators tend to work in a fixed location, e.g., a heating system uses feedback control to regulate the temperature in an environment within a selected range.

1.4.2.2 Embedded Versus Untethered

Smart environments contain components that have different degrees of dependence from their physical and ICT environments. Smart environments may use components that are embedded or untethered. Embedded devices are statics that are embodied in a larger system that may be static or mobile. Embedded systems typically provide control and sensing support to a larger system. Devices may be embedded in: (1) parts of physical environments, e.g., a passenger- or vehicle-controlled area entry system; (2) parts of the human environment, e.g., heart pacemakers; and (3) parts of larger ICT devices, e.g., a location device may be embedded in a phone or camera as opposed to externally connected to it.

Untethererd or amorphous or spray devices are types of environment devices that can be mixed with other particles and spread onto surfaces or scattered into gases and fluids, e.g., smart dust (Sections 2.2.3.2, 6.4.4). They are nomadic or untethered devices that do not need to operate using a home (base) location. They can self-organise themselves to optimise their operation (Section 10.5.1).

[32] RFID tags are also often referred to as smart labels or smart tags, however, smart tags in this text include a much wider range of tags and more specific set of properties (Section 6.2).

1.4.2.3 Device Sizes

Smart environment devices can vary in size. This affects their mobility. Macro-sized devices incorporate a range of device sizes from tab-sized (centimetre-sized) devices, through pad-sized (decimetre-sized) devices, to board-sized (metre-sized) devices (Section 1.2.2.2). Micro Electro Mechanical Systems (MEMS) are fabricated using integrated chip technology. This enables the large-scale cheap manufacture (thousands to millions) production of integrated circuit type devices, to be spread-on surfaces or to be airborne. Nanotechnology is 1 to 100 nanometre-sized devices that are built from molecular components. These are either constructed from larger molecules and materials, not controlled at the atomic level (more feasible) or assemble themselves chemically by principles of molecular recognition (less feasible).

1.4.3 Smart Interaction

In order for smart devices and smart environments to support the core properties of UbiCom, an additional type of design is needed to knit together their many individual activity interactions. Smart interaction is needed to promote a unified and continuous interaction model between UbiCom applications and their UbiCom infrastructure, physical world and human environments.

In the smart interaction design model, system components dynamically organise and interact to achieve shared goals. This organisation may occur internally without any external influence, a self-organising system, or this may be driven in part by external events. Components interact to achieve goals jointly because they are deliberately not designed to execute and complete sets of tasks to achieve goals all by themselves – they are not monolithic system components. There are several benefits to designs based upon sets of interacting components.

A range of levels of interaction between UbiCom system components exists from basic to smart. A distinction is made between (basic) interaction that uses fixed interaction protocols between two statically linked dependent parties versus (smart) interaction that uses richer interaction protocols between multiple dynamic independent parties or entities.

1.4.3.1 Basic Interaction

Basic interaction typically involves two dependent parties: a sender and a receiver. The sender knows the address of the receiver in advance; the structure and meaning of the messages exchanged are agreed in advance, the control of flow, i.e., the sequencing of the individual messages, is known in advance. However, the content, the instances of the message that adhere to the accepted structure and meaning, can vary. There are two main types of basic interaction, synchronous versus asynchronous (Section 3.3.3):

- *Synchronous interaction*: the interaction protocol consists of a flow of control of two messages, a request then a reply or response. The sender sends a request message to the specified receiver and waits for a reply to be received,[33] e.g., a client component makes a request to a server component and gets a response.
- *Asynchronous interaction*: The interaction protocol consists of single messages that have no control of flow, a sender sends a message to a receiver without knowing necessarily if the receivers will receive the message or if there will be a subsequent reply, e.g., an error message is generated but it is not clear if the error will be handled leading to a response message.

[33] This quotes synchronisation at the interaction level, the sender waits for a reply. Synchronisation may also occur at the message level rather than the interaction level. In the case the control of flow of the individual messages is synchronised, the sender waits for some acknowledgement that the receiver has received.

1.4.3.2 Smart Interaction

Asynchronous and synchronous interaction is considered part of the distributed system communication functions (Section 3.3.3.2). In contrast, interactions that are coordinated, conventions-based, semantics and linguistic-based and whose interactions are driven by dynamic organisations are considered to be smart interaction (Section 9.2.3). Hence, smart interaction extends basic interactions as follows:

- *Coordinated interactions*: different components act together to achieve a common goal using explicit communication, e.g., a sender requests a receiver to handle a request to complete a sub-task on the sender's behalf and the interaction is synchronised to achieve this. There are different types of coordination such as orchestration (use of a central coordinator) versus choreography (use of a distributed coordinator).
- *Policy and convention-based interaction*: different components act together to achieve a common organisational goal but it is based upon agreed rules or contractual policies without necessarily requiring significant explicit communication protocols between them. This is based upon previously understood rules to define norms and abnormal behaviour and the use of commitments by members of organisations to adhere to policies or norms, e.g., movement of herds or flocks of animals are coordinated based upon rules such as keeping a minimum distance away from each other and moving with the centre of gravity, etc.
- *Dynamic organisational interaction*: organisations are systems which are an arrangement of relationships (interactions) between individuals so that they produce a system with qualities not present at the level of individuals. Rich types of mediations can be used to engage others in organisations to complete tasks. There are many types of organisational interactional protocol such as auctions, brokers, contract-nets, subscriptions, etc.
- *Semantic and linguistic interactions*: communication, interoperability (shared definitions about the use of the communication) and coordination are enhanced if the components concerned share common meanings of the terms exchanged and share a common language to express basic structures for the semantic terms exchanged.

Consider a scenario in which light resources are designed to be context-aware in order to save energy. They are designed to be actuated by human presence. If they detect a human is present, they automatically switch on. If they detect no one is present, they switch themselves off to save energy. However, if there were several lights in a semi-dark room and they were merely context-sensitive, they would all switch on when someone enters, but this wastes energy unnecessarily. If instead they were designed to support smart interaction, they could decide among themselves which lights to switch on in order to best support particular human activities and goals. Smart interaction requires devices to interact to share resource descriptions (e.g., desk-light, wall light, main ceiling light) and goals (e.g., reading, watching a video, retrieving something). This example is more complex in practice as it may need to support several users and possibly conflicting user-goals. Smart interaction also requires some smart orchestrator (central planner) entity or choreographer (distributed planning) entities to establish goals and be able to plan tasks with the participation of others, directed towards achieving those goals.

Resources and users could compete against each other and participate in market-places in which the use of a resource is assigned a utility value and users are required to make the best bid to acquire the use of a resource (auction interaction). Resources may interact and self-organise themselves to offer a combined service (Chapter 9).

1.5 Discussion

1.5.1 Interlinking System Properties, Environments and Designs

In Table 1.6 (and Figure 1.7) a further comparison is made of the Smart DEI models with respect to the internal system and system environment models. There are different ways to model how smart devices, smart environments and smart interaction interlink

There are relations between each of these smart designs to the three main types of environment, human, physical and virtual. For example, smart environment designs focus on distributing multiple devices in the physical environment which are context aware of their human users and their physical locality. Smart environment devices tend to operate more autonomously, using minimal human operational management, compared to smart devices. Smart devices may also support some

Table 1.6 Comparison of smart device, smart environment and smart interaction

Type	Smart Device	Smart Environment	Smart Interaction
Characteristics	Active multi-function devices based in a virtual computing environment	Active single function devices embedded or scattered in a physical environment	Individual components that must cooperate or compete to achieve their goals
System environment interaction	Weak CPI, strong H2C, weaker C2H and strong C2C	Strong C2P and C2H	Rich H2H, P2P models that apply to HCI and CPI
Distributed system:			
openness, Dynamic services, Volatile ICT Access mobility	Dynamic ICT service, resource discovery	Dynamic physical resource discovery	Dynamic composition of entities and services
Context-awareness: of physical world,	Low-medium	High	Low-medium
user & ICT infrastructure	High, Personalized, 1.1[1] interaction	Shared between users 1-M Interaction	M-M interaction Coordinate, orchestrate
	High	Medium	Low
HCI: locus of control	Some innovative iHCI in smart ICT devices	Very Diverse UIs. iHCI focuses on context-awareness	Language-based interaction and iHCI
	Localized in ICT device	Localized in part(s) of Physical World	Distributed in physical and virtual world
Autonomy	Autonomous control of local ICT resources, less autonomous control of remote services	Autonomous control of local ICT resources	High autonomy of actions and interaction
Intelligence	Low to medium individual rational intelligence	Low to medium individual rational intelligence	High collective intelligence: semantic sharing, social cooperation and competition

[1] 1.1, 1-M and M-M refer to the cardinality of interaction, representing one-to-one, one-to-many and many-to-many interaction respectively.

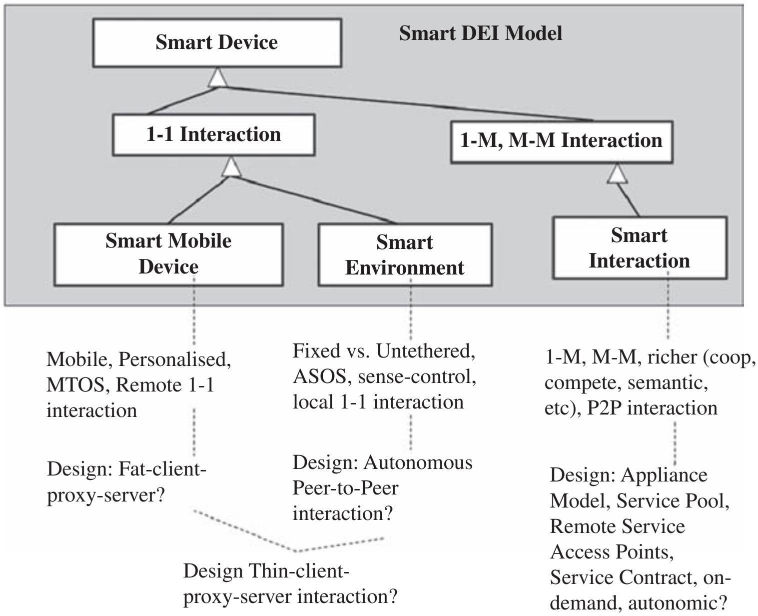

Figure 1.7 Alternate viewpoints and organisations for the device, environment and interaction entities in the Smart DEI model

awareness of the physical environment such as spatial awareness. Smart interaction uses the strongest notion of intelligence compared to the other smart designs in terms of intelligent selection, mediation and composition when multiple cooperating and competing entities interact. One important division[34] the Smart DEI model makes is the separation of the interaction between individual users and UbiCom systems which is modelled as part of smart device design, and the interaction between groups such as social and economic groups of users which is modelled as part of the smart interaction design. Social and economic models of humans may also be used for interaction between multiple ICT devices and services in virtual environments.

1.5.2 Common Myths about Ubiquitous Computing

Ubiquitous computing is quite a broad vision. There is a danger that it becomes too encompassing. Here a few unrealistic expectations about ubiquitous computing are discussed:

- *There is a single definition which accurately characterises ubiquitous computing*: rather there is a range of properties and types for ubiquitous computing which vary according to the application.

[34] This division may seem artificial but is done to separate design concerns. Although many-to-many interactions naturally occur between services, devices and human users, this is often modelled as one-to-one interaction through the use of mediators which act to serialise interactions. In practice, individual interactions naturally overlap or need to be combined with group interactions, e.g., using a phone in a meeting. The smart device and smart environment design may also be combined with the smart interaction design depending on the application.

- *The ideal type of ubiquitous computing is where all the properties of ubiquitous must be fully supported*: it may not be required, useful or usable in many cases in practice, to support the full set of these properties.
- *Ubiquitous computing means making computing services accessible everywhere*: this is unnecessary, too costly and makes smart environments become too cluttered, overloading the user with too many choices and contravening the hidden computer idea. Ubiquitous computing is also about computing being localised within a context and being available only when needed. Hence it is more appropriate to speak of context-aware ubiquity.
- *Ubiquitous computing is boundless computing*: this means that the virtual ICT world can extend fully into the physical world and into the human environment, replacing human and physical world systems and their interactions with computer interaction. But there limits to what computer systems can achieve, at least in the short term, e.g., UbiCom systems are not (yet) capable of completely supplanting human cognition and behaviour. Hence, UbiCom must strike a careful balance between supporting being human and living in harmony and experiencing the physical world, between being designed to give humans more fulfilled control of the their environment and taking away the less fulfilling control of the environment.
- *Ubiquitous computing is just about HCI*: automatic interaction and decisions are also needed in order to reduce human task and cognition overload and to enable tasks to be performed more safely, quicker, repeatedly and accurately. It is also less practical for humans to interact with micro-sized devices in the same way as interacting with macro-sized devices. Human interaction with compositions of multiple devices spatially distributed in shared physical spaces and time cannot be controlled centrally in the same way that humans can control a single device.
- *Calm computing should be used as a model for all HCI*. Calm computing is where the system is active, reducing some decision-making by humans. There are many applications and situations, where human users should clearly lead and control the interaction. Calm computing needs to be selectively used. Degrees of calm computing are needed from weak to strong.
- *Ubiquitous computing is just about augmenting reality*: UbiCom systems may not only enhance human–physical world interaction but it may also change it in wider ways. It may even diminish reality in some ways in order to aid the user in focusing on particular contexts. UbiCom is more about mediated reality.
- *Ubiquitous computing is just distributed or virtual computing*: UbiCom is more than being distributed in terms of interlinked, transparent and open ICT systems. UbiCom also focuses on particular models of human and physical world interaction involving context-awareness of the physical world and human and on supporting implicit human computing interaction.
- *Ubiquitous computing is just mobile wireless computing*: The ability to carry around higher resourced, multi-functional wireless mobile devices is useful but is also limited. Too many functions can cause clutter. Increasing numbers of functions can interfere with each other. It can be complex to make mobile devices strongly locally situated and adapt to the physical world. Ubiquitous computing also concerns being situated and embedded in the physical world.
- *Ubiquitous computing is just about smart environments*: while smarter physical world interaction can be facilitated through embedding active computing in the real world, UbiCom also involves interactions of smart, flexible, mobile devices which are human-centred and which support personal and social interaction spaces.
- *Ubiquitous computing need to be highly autonomous systems*: systems' autonomy is often limited in practice as computers are not able to design themselves, to completely adapt to new environments and user requirements and to maintain themselves in the face of changing requirements.
- *Ubiquitous computing is just about physical world context-awareness*: many types of context-aware systems are episodic, considering only the current physical environment state in order to determine their next actions. This is not effective in a partially observable and

non-deterministic world. In addition, the physical world context needs to be considered as part of the user context.

- *Ubiquitous computing is just distributed intelligence*: action selection and many operations can become overly complex and computationally intractable, requiring substantial computation to enable intelligent deliberation to reach an outcome. Interaction is more effective and easier to compute and execute if it has minimal intelligence, e.g., it is based upon reactive system design, rule-based behaviour as used in self-organising and self-creating systems. However, intelligence is very useful when systems have to deal with uncertainty and to handle autonomous systems that are themselves complex and intelligent.
- *Ubiquitous computing systems can operate effectively in all kinds of environments*: It is unrealistic to expect that ubiquitous computing systems can behave deterministically in non-deterministic, partially observable, etc., human and physical environments. Current systems cannot reliably actively adapt to user contexts where users act in an ad hoc manner. A weather context-aware system cannot reliably and accurately predict which clothes users should wear when the weather itself is unpredictable.

1.5.3 Organisation of the Smart DEI Approach

The book chapters are organised as follows:

- *Basics*: Basic and Vision (Chapter 1); Applications and Requirements (Chapter 2);
- *Smart Devices*: Smart Devices and Services (Section 3); Smart Mobiles, Cards and Device Networks (Chapter 4); Human–computer Interaction (Chapter 5);
- *Smart Environments*: Tagging, Sensing and Controlling (Chapter 6); Context-aware Systems (Chapter 7);
- *Smart Interaction*: Intelligent Systems (IS) (Chapter 8); Intelligent System Interaction (Chapter 9); Autonomous Systems and Artificial Life (Chapter 10).
- *Middleware and Outlook*: Ubiquitous Communication (Chapter 11); Management of Smart Devices (Chapter 12); Ubiquitous System Challenges and Outlook (Chapter 13).

This book can be studied in several ways. The traditional way by starting with the basics part and then reading on to understand more advanced topics that build on these. This tells a story from the more abstract and the underlying concepts to more technology-driven approaches. UbiCom is multi-disciplinary with some common themes across disciplines including: Basics and Vision (Chapter 1); Applications and Requirements (Chapter 2), Ubiquitous System Challenges and Outlook (Chapter 13) (see Table 1.7).

Table 1.7 Book chapters and their relation to the Smart DEI Model (Smart Device, Environment, Interaction), UbiCom system properties, and to system to environment interaction

No	Chapter Title	DEI	UbiCom Property	Environment Interactions
1	Basics and Vision	DEI	All	All
2	Applications and Requirements	DEI	Distributed, iHCI, Context-aware	All
3	Smart Devices and Services	Devices	Distributed	C2C

(*continued overleaf*)

Table 1.7 (*continued*)

No	Chapter Title	DEI	UbiCom Property	Environment Interactions
4	Smart Mobile Devices, Device Networks and Smart Cards	Devices	Distributed	C2C
5	Human–computer interaction	Devices	iHCI	HCI
6	Tagging, Sensing and Controlling	Environment	Context-aware	CPI
7	Context-Awareness	Environment	Context-aware	CPI
8	Intelligent Systems	Interaction	Intelligent	C2C, HCI
9	Intelligent Interaction	Interaction	Intelligent, iHCI	H2H, C2C
10	Autonomous Systems and Artificial Life	Interaction	Autonomy, Intelligence,	C2C
11	Communication Networks	Devices	Distributed	C2C
12	Smart Device Management	Devices	Distributed, iHCI, Context-aware	C2C, HCI, CPI,
13	Ubiquitous System Challenges and Outlook	DEI	All	All

These disciplines in the book can be studied separately. This book is a good way of illustrating their applied use. These discipline specific topics studied in addition to the basic part (Chapters 1 and 2) as follows:

- *UbiCom*: all chapters but in particular HCI (Chapter 5), Tagging, sensing, controlling the physical world (Chapter 6), context-awareness (Chapter 7).
- *ICT*: Smart devices and services (Chapter 3), smart mobile devices, device networks and smart cards (Chapter 4), ubiquitous communication (Chapter 11), smart device management (Chapter 12).
- *HCI*: HCI (Chapter 5), context-awareness (Chapter 6), managing smart devices in human-centred environments (Section 12.3), ubiquitous system challenges and outlook (Sections 13.6, 13.7,13.8).
- *AI*: intelligent systems (Chapter 8), intelligent interaction (Chapter 9). autonomous systems and artificial life (Chapter 10).

EXERCISES

1. Suggest some new ways to advance the personal visual memory application introduced in Section 1.1.1. (Hint: what if the camera was wearable? What if the camera supported other forms of image processing such as text recognition?)
2. Discuss why a paperless information environment in the world has not occurred but why a film-less photography world seems to be occurring. Compare and contrast the use of paper and photographs versus computer pads as information systems with respect to their support for the UbiCom requirements.
3. Give some further examples of ubiquitous computing applications today and propose future ones.
4. Analyse three different definitions of ubiquitous computing and distinguish between them.
5. Debate the benefits for ubiquitous systems: to support a strong notion of autonomy, to support a strong notion of intelligence.

EXERCISES (continued)

6. Debate whether ubiquitous systems must fully support all of the five Ubiquitous system properties and whether or not the five Ubiquitous system properties are independent or are highly interlinked.
7. Debate the need for UbiCom systems to be intelligent, Argue for and against this.
8. Debate the point that whereas mainstream computer science focuses on computer to computer interaction, Ubiquitous computing in addition focuses strongly on ICT device–physical world interaction and on, ICT device–human interaction.
9. Describe the range of interactions between humans and computers, computers and the physical world and humans and the physical world. Illustrate your answer with specific UbiCom system applications. (Additional exercises are available on the book's website.)

References

Aarts, E. and Roovers, R. (2003) Ambient intelligence: challenges in embedded system design. In *Proceedings of Design Automation and Test in Europe*, Munich, pp. 2–7.

Buxton, W. (1995) Ubiquitous media and the active office. Published in Japanese (only) as Buxton, W. (1995). Ubiquitous video, *Nikkei Electronics*, 3.27 (632), 187–195. English translation accessed from http://www.billbuxton.com/UbiCom.html, on April 2007.

Chen, G. and Kotz, D. (2000) A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381. Available from http://citeseer.ist.psu.edu/chen00survey.html. Accessed November 2006.

Coen, M.H. (1998) Design principles for intelligent environments. In *Proceedings of 15th National/10th Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, pp. 547–554.

Cook, D.J. and Das, S.K. (2007) How smart are our environments? An updated look at the state of the art. *Pervasive and Mobile Computing*, 3(2): 53–73.

Endres, C., Butz, A., and MacWilliams, A. (2005) A survey of software infrastructures and frameworks for ubiquitous computing. *Mobile Information Systems*, 1(1): 41–80.

Gillett, S.E., Lehr, W., Wroclawski, J. and Clark, D. (2000) *A Taxonomy of Internet Appliances*. TPRC 2000, Alexandria, VA. Retrieved from http://ebusiness.mit.edu on 2006-09.

Greenfield, A. (2006) *Everyware: The Dawning Age of Ubiquitous Computing*. London: Pearson Education.

Hayles, N.K. (1999) *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics*. Chicago: University of Chicago Press.

Horn, P. (2001) Autonomic Computing: IBM's Perspective on the State of Information Technology, also known as IBM's Autonomic Computing Manifesto. Retrieved from http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf, accessed Nov. 2007.

ISTAG. (2003) IST Advisory Group, Advisory Group to the European Community's Information Society Technology Program. Ambient Intelligence: from Vision to Reality. Retrieved from http://www.cordis.lu/ist/istag.htm. Article on May 2005.

Kindberg, T. and Fox, A. (2002) System software for ubiquitous computing. *IEEE Pervasive Computing*, 1(1): 70–81.

Milner, R. (2006) Ubiquitous computing: shall we understand it? *The Computer Journal*, 49(4): 383–389.

Pressman, R.S. (1997) Design for real-time systems. In *Software Engineering: A Practitioner's Approach*. 4th edn. Maidenhead: McGraw-Hill, pp. 373–377.

Rogers, Y. (2006) Moving on from Weiser's vision of calm computing: engaging UbiComp experiences. In P. Dourish and A. Friday (eds) *Proceedings of Ubicomp 2006, Lecture Notes on Computing Science*, 4206: 404–421.

Russell, S. and Norvig, P. (2003) *Artificial Intelligence: A Modern Approach*. 2nd edn. Englewood Cliffs, NJ: Prentice Hall.

Satyanarayanan, M. (2001) Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8: 10–17.

Schmidt, A. (2000) Implicit human–computer interaction through context. *Personal Technologies*, 4(2&3): 191–199.

Warneke, B., Last, M., Liebowitz, B. and Pister, K.S.J. (2001) Smart dust: communicating with a cubic-millimeter. *Computer*, 34(1): 44–51.

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