

Uncertainty and Error Handling in Pervasive Computing: A User's Perspective

Marie-Luce Bourguet

*School of Electronic Engineering and Computer Science,
Queen Mary, University of London
U.K.*

1. Introduction

From this chapter's perspective, pervasive computing is a new class of multimodal systems, which employs passive types of interaction modalities, based on perception, context, environment and ambience (Abowd & Mynatt, 2000; Feki, 2004; Oikonomopoulos et al., 2006). By contrast, early multimodal systems were mostly based on the recognition of active modes of interaction, for example speech, handwriting and direct manipulation. The emergence of novel pervasive computing applications, which combine active interaction modes with passive modality channels raises new challenges for the handling of uncertainty and errors. For example, context-aware pervasive systems can sense and incorporate data about lighting, noise level, location, time, people other than the user, as well as many other pieces of information to adjust their model of the user's environment. In affective computing, sensors that can capture data about the user's physical state or behaviour, are used to gather cues which can help the system perceive the user's emotions (Kapoor & Picard, 2005; Pantic, 2005). In the absence of recognition or perception error, more robust interaction is then obtained by fusing explicit user inputs (the active modes) and implicit contextual information (the passive modes). However, in the presence of errors, the invisibility of the devices that make up the pervasive environment and the general lack of user's awareness of the devices and collected data properties render error handling very difficult, if not impossible.

Despite recent advances in computer vision techniques and multi-sensor systems, designing and implementing successful multimodal and ubiquitous computing applications remain difficult. This is mainly because our lack of understanding of how these technologies can be best used and combined in the user interface often leads to interface designs with poor usability and low robustness. Moreover, even in more traditional multimodal interfaces (such as speech and pen interfaces) technical issues remain. Speech recognition systems, for example, are still error-prone. Their accuracy and robustness depends on the size of the application's vocabulary, the quality of the audio signal and the variability of the voice parameters. Signal and noise separation also remains a major challenge in speech recognition technology.

Recognition-based multimodal interaction is thus still error prone, but in pervasive computing applications, where the capture and the analysis of passive modes are key, the possibilities of errors and misinterpretations are even greater. Furthermore, in pervasive

computing applications, the computing devices have become invisible and the users may not be aware of their behaviour that is captured by the system. They may also have a wrong understanding of what data is captured by the various devices, and how it is used. In most cases, they do not receive any feedback about the system's status and beliefs. As a result, many traditional methods of multimodal error correction become ill adapted to pervasive computing applications. When faced with errors, users encounter a number of new challenges: understanding the computer's responses or change of behaviour; analysing the cause of the system's changed behaviour; and devising ways to correct the system's wrong beliefs.

This chapter addresses these problems. It exposes the new challenges raised by novel pervasive computing applications for the handling of uncertainty and errors, and it discusses the inadequacies of known multimodal error handling strategies for this type of applications. It is organised as follows. In the next section, we explain our usage of the words multimodal and pervasive computing and we propose our own definitions, which are based on the notions of active and passive modes of interaction. We also describe a number of pervasive computing applications, which will serve in the remainder of the chapter to illustrate the new challenges raised by this type of applications. In section 3 of the chapter, we briefly review the various recognition error handling strategies that can be found in the multimodal interaction literature, then in section 4, we show that many of these multimodal error handling strategies, where active modes only are used, are ill adapted to pervasive computing applications. We also discuss the new challenges arising from the deployment of novel pervasive computing applications for error correction. In section 5, we suggest that promoting users' correct mental models of the devices and data properties that make up a pervasive computing environment can render error handling more effective. Section 6, finally, concludes the chapter.

2. Multimodal and pervasive computing

The concepts of multimodal interaction and of pervasive computing share many commonalities, one of which is the lack of an accepted definition.

2.1 Active and passive modes

The concept of multimodal interaction partly arose from the difficulties met by the speech recognition research community, in the early eighties, to implement satisfactorily robust speech-based interfaces. The idea was to complement the error prone voice inputs with more deterministic ones, such as mouse and keyboard inputs (i.e. direct manipulation and typing). In parallel with the deployment of more robust touch screen and pen input technologies, alternative recognition-based modalities of interaction (pen-based 2D gestures and hand-writing) also started to emerge. Vision-based recognition modalities, i.e. inputs captured by a camera, soon followed, complementing the list of recognition-based input modalities, while at the same time opening new possibilities for context awareness and the perception of passive modes of interaction (e.g. gaze and facial expressions). Until now, the concept of multimodal interaction has never stopped evolving, encompassing yet more input modes, which are increasingly based on perception and sensory information. The various mission statements made by the Multimodal Interaction Working Group (MIWG) of the W3C (World Wide Web Consortium) offer a good evidence of this evolution. In 2002 (W3C, 2002), the MIWG aimed to develop new technology to create "web pages you can

speak to and gesture at". According to its charter, the MIWG was "tasked with the development of specifications covering the following goals: To support a combination of input modes, including speech, keyboards, pointing devices, touch pads and electronic pens; To support the combination of aural and visual output modes; To support a combination of local and remote speech recognition services; To support the use of servers for speech synthesis and for streaming pre-recorded speech and music; To support a combination of local and remote processing of ink entered using a stylus, electronic pen or imaging device; To support varying degrees of coupling between visual and aural interaction modes; To support the use of a remote voice dialog engine, e.g. a voice gateway running VoiceXML; To support the coordination of interaction across more than one device, e.g. cell phone and wall mounted display." The MIWG's charter, in 2002, was still resolutely geared towards speech and pen interaction. In 2003, however, the same Working Group (W3C, 2003), in a document dedicated to their multimodal interaction framework, was listing the following input modes: speech, handwriting, keyboarding and pointing device, but with the assumption that "other input recognition components may include vision, sign language, DTMF (Dual-tone multi-frequency signaling), biometrics, tactile input, speaker verification, handwritten identification, and other input modes yet to be invented". Beside the recognition component, The W3C also included a "System and Environment component", which "enables the interaction manager to find out about and respond to changes in device capabilities, user preferences and environmental conditions", hence incorporating in their system specifications some of the aims of a pervasive computing system.

Nowadays, in the HCI literature, the expression "pervasive computing" is mostly used to describe connected computing devices in the environment. For example, the following description of pervasive computing has been proposed by TechTarget©: in pervasive computing, "the goal of researchers is to create a system that is pervasively and unobtrusively embedded in the environment, completely connected, intuitive, effortlessly portable, and constantly available"; and also: pervasive computing is the "possible future state in which we will be surrounded by computers everywhere in the environment that respond to our needs without our conscious use." In these statements, the notions of being "unobtrusive" and "without our conscious use" add a dimension which was not present in earlier (often speech-based) multimodal systems. To be unobtrusive and to not require our conscious use, pervasive computing applications must rely on novel sources of information which are called "passive" modes of interaction. Examples of passive modes include vision-based modes captured by cameras (e.g. gaze and facial recognition for affective computing), sensory information (levels of lighting, temperature, noise, as well as biometrics information), GPS (Global positioning System, for positioning and time information) and tag technologies such as RFID (Radio Frequency Identification). In other words, the term pervasive is used to qualify the system (the devices, their type and their connectivity), whereas the term multimodal is often used to qualify the interaction. In this context, the multimodal interface becomes the mean to interact with the pervasive system.

In this chapter, we will use the term "multimodal" to qualify systems, which make use of several active modes of interaction, at least one of them being recognition-based (failing that, we will simply talk of "interactive" systems). We will use the word "pervasive" to qualify systems, which combine both active and passive modes of interaction. The next section provides some examples of pervasive computing applications.

2.2 Pervasive computing applications

Novel pervasive computing applications have started to emerge, which address one or more aims of a pervasive computing environment: being unobtrusive, being invisible and not requiring users' conscious interaction with the system. We describe briefly here five types of pervasive computing applications: context-aware multimodal interaction systems, location-aware systems, affective computing, smart home applications and wearable computers.

2.2.1 Context-aware multimodal interaction systems

Context-aware multimodal interaction systems make up one class of pervasive computing applications where the emphasis is on using passive modalities (the context) to enhance the interaction, especially its efficiency and robustness. (Dey, 2001) provides a good definition of the word context: "context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." Still according to (Dey, 2001), a "system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task". In a context-aware multimodal interaction system, perceived contextual information is often used to complement or disambiguate an active mode of interaction, such as speech (Stillman & Essa, 2001; Macho et al., 2005). For example, (Yoshimi & Pingali, 2002) describe a video conferencing application, which combines carefully placed multiple distributed microphone pairs with calibrated cameras to identify the current speaker and their location, in order to achieve a finer control of the speech recognition process. More recently, (Luo et al., 2009) show that in a classroom environment, sources of contextual information are abundant (people, activities, location, physical environment and computing entities) that can influence the result of otherwise ambiguous multimodal fusion. Some contextual information like illumination, temperature and noise level is obtained directly from sensors, while other like user activities and location is dynamically obtained from a Virtual Interactive Classroom (VIC) software platform. The VIC platform can display to the lecturer, in real-time, emotion data, through attention detection, facial expression recognition, physiological feature detection and speech emotion recognition processes for every student in the virtual classroom and for groups of students engaged in discussion. In this application, context is used to augment lecturer-student interaction with additional information and communication opportunities. (Yue et al., 2005) describe an hypermedia mobile system able to provide users with geographic information services at any time from anywhere. This system encompasses a context-sensitive multimodal module in which explicit multimodal user inputs and implicit contextual knowledge are integrated. Context sensitive information is used to evaluate users cognitive load and attention interference from the mobile environment, in order to adapt the interaction between the user and the mobile device (e.g. adaptation of the level of complexity and detail of the displayed information) as well as some aspects of the device's interface (e.g. orientation-aware adaptation of the display mode). Finally, (Crowley, 2006) proposes a framework for context aware observation of human activity, in which a situation (i.e. the current state of the environment) is described by a configuration of relations for observed entities. The stated aim of such a framework is to provide a foundation for the design of systems that act as a "silent partner" to assist humans in their activities in order to provide appropriate services without explicit commands and configuration (i.e. unobtrusive systems that do not require users conscious intervention).

2.2.2 Location-aware systems

Location-aware systems are a special type of context-aware systems where the emphasis is on adapting a service to the user's current location. Location-aware systems are typically deployed in pervasive environments with highly mobile users. Their aim is to make mobile devices know where they are and automatically do the right thing for that location, for example automatically reconfiguring themselves, adapting their security level, or being able to share information with nearby devices. (Khoury & Kamat, 2009) describe a system that can dynamically track a user's viewpoint and identify the objects and artefacts visible in the mobile user's field of view. In this system, the user's spatial context is defined both by their location (captured by GPS) and by their three-dimensional head orientation (captured by a magnetic orientation tracking device). Indoors, GPS technology is of no use, but RFID tags can help establish the location of a particular object or spot within the range of a few centimeters. An alternative to expensive RFID tags is to use the Wireless Local Area Network (WLAN) technique to sense and detect a location, as explained in (Tsai et al., 2010). (Tsai et al., 2010) describe a location-aware tour guide system for museums where a location position agent can sense the strengths of the signals from all the access points to which the mobile devices can be linked. If the visitor changes location, a context-aware agent matches the new coordinates on the museum map, which is shown on the visitor's PDA. Nowadays, advanced web browsers have also become location-aware and allow developers to find a computer's location by looking at the surrounding WiFi networks (which is not as precise as using a GPS, but more precise than relying on a user's IP address). Websites that use location-aware browsing aim at bringing more relevant information, or saving users time while searching.

2.2.3 Affective computing

Affective computing applications, another class of pervasive computing applications, can sense users' physiological state, track their activities and perceive their behaviour to infer their psychological state, mood and level of satisfaction or dissatisfaction. The virtual classroom application (Luo et al., 2009) mentioned already is an example of context-aware affective computing application. (Kapoor & Picard, 2005) describe a framework that can automatically extract non-verbal behaviours and features from face and postures, to detect affective states associated with interest and boredom in children, which occur during natural learning situations. Features are extracted from four different channels: the upper face (brow and eye shape, likelihood of nod, shake and blink), the lower face (probability of fidget and smile), the posture (current posture and level of activity), as well as information from the status of the application (in this case an educational game). In (Benoit et al., 2007) a driving assistant system is described that relies on passive modalities only (facial expression, head movement and eye tracking) to capture the driver's focus of attention and predict their fatigue state. The driver's face is monitored with a video camera and three signs of hypo-vigilance are tracked: yawning, blinking (or eyes closure) and head motion. Complex bio-inspired algorithms are then used in order to analyse the data and predict attention and fatigue.

2.2.4 Smart homes

Smart home applications are becoming increasingly popular and an area of rapid expansion for research and development. According to the Smart Homes Association, smart home

technology is: “the integration of technology and services through home networking for a better quality of living”. The idea is that anything in the home that uses electricity can be put on the home network and when commands are given, either by voice, remote control, computer or mobile phone, the home reacts. Most applications relate to lighting, home security, home theater and entertainment and thermostat regulation. A smart home system may also automatically decide to turn off lights and TV sets when the user leaves the house, adjust artificial lighting levels according to changes in day light, add items to an electronic shopping list when the house fridge gets empty, etc. According to (Edmonds, 2010), “Microsoft Chairman Bill Gates' home might be the most famous smart home to date. Everyone in the home is pinned with an electronic tracking chip. As you move through the rooms, lights come on ahead of you and fade behind you. Your favorite songs will follow you throughout the house, as will whatever you're watching on television. The chip keeps track of all that you do and makes adjustments as it learns your preferences. When two different chips enter the same room, the system tries to compromise on something that both people will like.”

2.2.5 Wearable computers

Finally, wearable computer applications, where the computing devices are worn on the body, implies context recognition with on-body sensors. (Ferscha & Zia, 2009) present a wearable computer, which aims at providing directional guidance for crowd evacuation, by means of a belt worn by individuals in the crowd. The vibrotactile belt notifies individuals in panic about exits. It has the capability to sense the neighbourhood, to extract the relative spatial relations (distance and orientation) of all neighbours, and to interact with the person wearing it in a natural and personal way. Smart textiles, used for example in fashion (Vieroth et al., 2009) offer an extreme example of wearable computing where the pervasive computing environment is the clothing.

3. Multimodal error Handling Strategies

This section of the chapter offers a brief review of error handling strategies in multimodal interaction systems. We will then examine in section 4 how these strategies can or cannot be applied to pervasive computing applications such as the ones presented in the previous section.

Despite recent progress in recognition-based technologies for human-computer interaction (speech, gestures, handwriting, etc.), recognition errors still occur and have been shown to reduce the effectiveness of natural input modalities (Halverson et al., 1999; Karat et al., 1999; Suhm et al., 1999). The impact of recognition errors on multimodal systems' usability varies according to the application and depends upon a number of factors such as the amount of input required, the acceptability of uncorrected errors, the benefits of using recognition-based modalities (as compared with other interaction means), the availability of adequate error handling mechanisms, etc. (Bourguet, 2006) has listed thirty different error handling strategies in recognition-based multimodal interfaces and proposed a classification (see Fig. 1). According to this classification, an error handling strategy can either be the responsibility of the machine (i.e. the multimodal interface) or of the user, and can fulfil one of the three following purposes: error prevention, error discovery and error correction.

Purpose Actor	Error Prevention	Error Discovery	Error Correction
MACHINE	<ul style="list-style-type: none"> - <i>Error reduction by design</i> (1) Isolated characters (2) Single-stroke alphabets (3) Tap-to-speak interfaces (4) Restricted grammars (5) Guided dialogues (6) Context-sensitive cues (7) Consistency and symmetry (8) Structured graphics (9) Adapted modalities <ul style="list-style-type: none"> - <i>Error reduction by context</i> (10) Feature level integration (11) Context-aware systems 	<ul style="list-style-type: none"> - <i>Automatic detection</i> (15) Thresholding (16) Semantic, pragmatic and common sense knowledge (17) Mutual disambiguation (18) Synchronisation models 	<ul style="list-style-type: none"> - <i>Automatic correction</i> (16) Semantic, pragmatic and common sense knowledge (17) Mutual disambiguation (18) Synchronisation models
USER	<ul style="list-style-type: none"> - <i>User prevention</i> (12) Modality selection (13) Complementary inputs (14) Redundant inputs 	<ul style="list-style-type: none"> - <i>Machine-led discovery</i> (19) Implicit confirmation (20) Explicit confirmation (21) Mediation strategies (22) Visual display of results (23) List of alternative hypotheses 	<ul style="list-style-type: none"> - <i>User correction</i> (24) Repeat (25) Spell out (26) Rephrase (27) Contradict (28) Modality switch (29) Cross-modal correction (30) Correction marks

Fig. 1. Taxonomy of multimodal error-handling strategies (adapted from (Bourguet, 2006))

3.1 Machine error handling

It appears from Fig. 1 that most strategies for error prevention can be attributed to the machine. They work in two possible ways: either the interface is designed to influence or constrain user behaviour into less error-prone interaction (i.e. “error reduction by design”), or greater recognition accuracy is achieved through the use of additional or contextual information (i.e. “error reduction by context”).

Error reduction by design techniques achieve error prevention by leading users towards the production of inputs that are easier to recognise. The different techniques differ in the level of constraints they impose on user behaviour and actions, and the degree of control the user has on the interaction. For example, “Tap-to-speak interfaces” are interfaces in which users must indicate to the system by a brief signal that they are going to talk before each utterance (Oviatt et al., 1994). Another technique consists in implementing “guided dialogues” where users are prompted to say or do something from a limited set of possible responses. Another, less constraining technique, consists in controlling the system’s responses and discourse level throughout the dialogue (“consistency and symmetry”) in order to shape the users’ speech and actions to match that of the system’s (Heer et al., 2004).

Error reduction by context techniques achieve error reduction by augmenting users’ inputs with redundant or contextual information. Context-aware systems, in particular, make use of passive modalities and, according to our definitions, thus belong to the domain of pervasive computing. Similarly, “Feature level integration” exploits intrinsic properties of tightly coupled modalities such as speech and lip movements and does not require users to consciously act multimodally.

Error discovery by machine works in three possible ways: by using statistical data (“thresholding”), by exploiting cross-modal information (“mutual disambiguation” and “synchronisation models”), or by applying knowledge-based rules (Baber & Hone, 1993). (Bourguet & Ando, 1998) have shown that to be effective at disambiguating interaction, cross-modal information must be complementary but not always semantically rich. They show for example that timing information from hand gestures (i.e. speech and gesture synchronisation data) can be used to locate in the speech signal the parts that are more semantically significant, such as important nominal expressions. In (Holle & Gunter, 2007) a series of experiments are presented, which explore the extent to which iconic gestures convey information not found in speech. The results suggest that listeners can use gestural information to disambiguate speech. For example, an iconic gesture can facilitate the processing of a lesser frequent word meaning.

Finally, with knowledge-based and cross-modal strategies, the automatic discovery of recognition errors can sometimes lead to automatic correction as well. This is generally true if the correct output figures in the list of alternative hypotheses produced during the recognition process.

3.2 User error handling

On the users’ side, user prevention strategies rely on users’ spontaneous change of behaviour to prevent errors. This is facilitated in natural multimodal interfaces by the availability of multiple modalities of interaction, which allows users to exercise their natural intelligence about when and how to deploy input modalities effectively (Oviatt, 1999).

When a recognition error occurs, users are normally in charge of notifying the machine. It is important, however, that the machine facilitates error discovery. Machine-led discovery techniques include implicit confirmation (Narayanan, 2002), explicit confirmation (e.g. when in safety critical systems users are asked to confirm that what has been recognised or understood is correct), visually displaying recognition results, and allowing the selection of the correct result from a list of alternative hypotheses.

Once errors have been found, users can effectively help the machine resolve them, usually by producing additional inputs. Studies of speech interfaces have found that the most instinctive way for users to correct mistakes is to repeat (Suhm et al., 2002). However, although repeating might be the most obvious way to correct when the system mishears, it is often the worse for the system (Frankish et al., 1992). The main reason for this is that when repeating, users tend to adjust their way of speaking (e.g. by over-articulating) to what they believe is easier for the recogniser to interpret, which often has the opposite effect. In handwriting, a similar strategy to repeating is to overwrite a misrecognised word. Linguistic adaptation is another strategy that has been observed where users choose to rephrase their speech, in the belief that it can influence error resolution: a word may be substituted for another, or a simpler syntactic structure may be chosen (Oviatt, 2000). In multimodal systems, it has been suggested that users are willing to repeat their input at least once, after which they will tend to switch to another modality (Oviatt & van Gent, 1996). For example, if speech input failed repeatedly when entering data in a form, users may switch to the keyboard in order to type their entry. Alternative strategies include locating a recognition error by touching a misrecognised word on a writing-sensitive screen where recognition output is displayed, then correcting the error by choosing from a list of alternative words, typing, handwriting, or editing using gestures drawn on the display (Suhm et al., 1999).

To summarize this section it can be said that an extensive body of work exists in multimodal error handling and that a large number of strategies have been proposed and tried. However, most of these strategies assume the use of active modalities. In the next section of the paper, we show that many of these strategies are ill adapted to pervasive computing applications, where passive modalities play an important role.

4. Error handling in pervasive computing

4.1 Machine error handling

In "traditional" multimodal user interfaces, machine error handling plays an important role in error prevention. In particular, *error reduction by design* (see Fig. 1) is a major error handling strategy, which aims at preventing interaction errors by influencing or guiding users' behaviour. In pervasive computing, where, in general, the devices must not interfere with social interaction and human behaviour, error reduction by design goes against the fundamental principle of unobtrusiveness. In many pervasive computing applications, especially context and location aware applications (e.g. the hypermedia mobile system (Yue et al., 2005), the silent partner concept (Crowley, 2006) and the dynamic user-viewpoint tracking system (Khoury & Kamat, 2009)) as well as in smart home applications, the aim is indeed of anticipating and not influencing users' needs and actions. (Matsumiya et al., 2003) specifically address the problem of unobtrusiveness and "zero disturbance" in pervasive computing. The authors present an authentication model, which aims at authenticating mobile users without interfering with their mobile behaviour. They describe a "zero-stop" authentication model that can actively authenticate users in an environment populated with various mobile and embedded devices without disturbing users' movements. In pervasive applications, where unobtrusiveness and zero disturbance is a major concern, one of the most important error handling strategy class (*error reduction by design*) thus becomes inapplicable.

In contrast, the second class of machine error handling strategies for error prevention: *error reduction by context* (see Fig. 1), plays an important role in pervasive computing applications. The use of context and multi-sensors information to render user-system interaction more robust and efficient is one major aim of context-aware multimodal applications (e.g. the video conferencing application (Yoshimi & Pingali, 2002) and the virtual classroom application (Luo et al., 2009)). Another example of such strategy is provided by "machine lip reading" techniques which consist in combining acoustic information from the speech signal with visual information captured from the shapes of the speaker's lips to achieve more robust speech recognition without requiring any additional user inputs (Meier et al., 2000).

As far as the automatic correction of recognition errors is concerned, it can be achieved in current multimodal interfaces, by using *semantic, pragmatic, and common sense knowledge* (see Fig. 1). (Singh, 2002) has collected common sense statements from the public for the Open Mind Common Sense Project, resulting in a database that currently contains more than 700,000 facts. The common sense statements have been used to reorder the recognition hypotheses returned by a speech recogniser and filter out possibilities that "don't make sense". In pervasive computing, the approach can be taken further by incorporating knowledge about human behavioural and social signalling. As the field matures, such knowledge will undoubtedly become invaluable to allow machines to automatically detect and correct errors. For example, the understanding of users emotions through the analysis of facial expressions (see the affective computing applications) will allow machines to disambiguate between literal and ironic statements.

4.2 User error handling

According to the definitions we proposed in section 2.1, the main difference between a multimodal system and a pervasive one, is the nature of the interaction modes: active versus passive modes. All the user strategies for error handling described in section 3, assume active modes of interaction and users complete awareness (but not necessarily control) of all aspects of the interaction. In fact, they also assume that the source and cause of the error can somehow be located and identified by the user or by the machine. For example, to be able to “exercise their natural intelligence about when and how to deploy input modalities effectively” (user error prevention strategies), users must have a good understanding of the properties and characteristics of each modality. They must also be able to anticipate the performance quality they are likely to obtain from the systems that recognise and interpret these modalities. Similarly, user error correction strategies such as *modality switch* and *cross-modal correction* require users to be able to identify the faulty recognition process in order to choose a more appropriate alternative.

Fig. 2 summarises some characteristics of the devices that are used in pervasive computing applications as well as some properties of the data that these devices capture (passive modalities), which are likely to make error handling by users difficult and render known user error handling strategies, such as the ones presented in section 3.2, impractical.

Devices Characteristics		Data Properties	
Invisibility	Where is the device / sensor located?	What data	What is captured ?
Multiplicity	Which device captured the data ?	Use	How is the data used ?
Sensitivity	How sensitive is the device ?	Trustworthiness	How well is the data used ?
Disparity	How disparate are the different devices ?	Accuracy	How accurate is the data?
		Combination	How is the data combined with other ?

Fig. 2. Device and data issues in pervasive computing applications (adapted from (Bourguet, 2008))

First of all, many of the devices and sensors have become invisible (*Invisibility*). Several devices and sensors are connected in the pervasive environment, and it may not be possible to know which of them has collected data (*Multiplicity*). Given the multiplicity of the devices, it may also be difficult to know the specific properties of each of them, in particular how sensitive a particular device is (*Sensitivity*) and how similar or different all the devices are in their characteristics (*Disparity*). From the user’s perspective, when the system seems to behave abnormally (following a recognition or sensing error), the invisibility of the devices implies that the user cannot locate the faulty system (in order, for example, to avoid using

it). The multiplicity of the devices means that even if they can be located, which device(s) caused the error may not be obvious. The unknown sensitivity of the devices does not allow users to adapt their behaviour in order to provide better quality inputs, and the disparity of the devices does not allow users to devise and re-use appropriate strategies for error handling.

As far as data properties are concerned, the user may not know which data is captured by the pervasive system (*What data*) and for what purpose it has been captured (*Use*). These two properties are inherent to most affective computing applications. Not knowing what is captured and for what purpose may let the user wonder about the appropriateness of the data use and give rise to problems of miss-trust (*Trustworthiness*). Questions can also arise about the performance quality of the various recognition and sensing processes, and in particular about the accuracy of the collected data (*Accuracy*). Finally, in relation with the multiplicity of the devices, how is the data combined with other to build complex representations of the environment and of the user and make inferences is another source of uncertainty (*Combination*).

Let's imagine that an affective computing application has wrongly inferred from the analysis of an image of the user's face, combined with a low level of user's activity, that the user is anxious or in difficulty. The system may then embark on trying to comfort the user by accordingly changing its behaviour and response mode (for example by lowering the level of difficulty of the user's current activity). In this situation, the user is faced with a number of challenges: (1) understanding the computer's change of behaviour (e.g. "the system is trying to comfort me"); (2) analysing possible causes of the system's changed behaviour (e.g. "the system believes I am unhappy"); and (3) devising ways of correcting the system's wrong belief (e.g. "I should smile more!"). However, in order to devise an error correction strategy, the user must know the type of data or data combination that is at the origin of the wrong inference (*What data* and *Combination*), by which device it has been captured (*Multiplicity*), where the device is located (*Invisibility*), how to provide better inputs (*Sensitivity* and *Accuracy*). In other words, when data has been wrongly interpreted in a pervasive environment, error correction is difficult because it may be impossible to know which device is responsible, and what combination of data contributed to the wrong interpretation. Even when users are aware of what data is captured, for example images of their face, it may not be clear how the data is used by the system, and how accurate is the data. Finally, when it comes to try and influence system's behaviour and beliefs, it will be necessary to understand how sensitive the devices are, and how disparate or homogeneous they are in their properties and characteristics.

Another issue in pervasive computing, is that users may not always be aware of their own actions, which have been captured and exploited by the system to enhance the interaction (see for example the "virtual classroom" application (Luo et al., 2009)). The use of passive modalities, for example through the capture of spontaneous gestures and facial expressions, is an important property of pervasive environments. However, the shift from an environment where the user is always the conscious actor of every input received by the system, to an environment where the user is only one possible source of inputs among others (see context and location aware applications), or where the inputs produced by the user are produced unconsciously (see affective computing applications), is a dramatic one. For example, the "driving assistant" application (Benoit et al., 2007) explicitly relies on the fact that users have little or no control on the data captured by the system, so it can detect dangerous behaviour in driving conditions.

Furthermore, the invisibility of the devices in pervasive computing raises one of the most important challenges for error discovery by users. This is because when the devices responsible for capturing and analysing interaction data become invisible, it becomes increasingly difficult for users to identify the causes of the errors. In multimodal interfaces, the machine is primarily in charge of enabling error discovery by providing adequate feedback on its status and beliefs (*machine-led discovery*). In pervasive computing, the additional challenge is thus to devise ways of providing the necessary feedback while remaining invisible and unobtrusive. The users' ability to devise error handling strategies is generally dependent on the availability of system's feedback about its current status and beliefs. (Bourguet, 2008) describes an experimental study that aims to test users spontaneous change of behaviour in situation of error correction, when the cause and source of the error cannot easily be identified. The context of the experiment is multimodal (speech and pen interaction) and not pervasive, but it gives an insight into users' opportunistic error handling strategies in complex error situations. The study is designed to verify if users are likely to modify some aspects of their input when repeating a complex multimodal command (a command that combines speech and gestures), in the belief that it can help error resolution. In particular, the study aims at comparing users modality synchronisation patterns in normal situations of interaction, and in situations of error correction. It was found that when repeating a multimodal command, users are likely to use different modality synchronisation patterns to try and influence the performance of recognition-based modalities, but only if the source of the error can be identified. Synchronisation patterns that significantly depart from typical patterns should thus be interpreted with in view the possibility that the user is in error recovery mode, and modality integration techniques should be able to adapt to changing synchronisation patterns. However, users only seem to be able to adapt their behaviour when they can identify the source and nature of the system error. In absence of cues about the origin of the error, they either choose to repeat the command in the same way it was originally entered or they give up on the interaction. This result let us foresee new challenges for handling errors in pervasive computing applications, where the cause and nature of the errors are likely to be difficult to anticipate and identify.

Users' ability to understand the systems and devices used in human computer interaction, allow them to make prediction about future events, which in turns allow them to devise appropriate strategies for system error handling. The "invisibility", "what data", and "use" properties shown in Fig. 2 particularly affect the ability of users to predict future events and to prevent errors from occurring. In other words, error handling necessitates accurate users' mental models of the multimodal and pervasive computing systems. In the next section, we discuss the merits and difficulties of promoting through system design good users' mental models in pervasive computing applications.

5. Towards more usable pervasive computing applications

According to (Norman, 1988), a mental model is "the model people have of themselves, others, the environment, and the things with which they interact." Mental models allow us to make predictions before carrying out an action about its possible effects. When they are correct or sufficiently accurate, we can use them to solve unexpected problems, if however they are inadequate, they can lead to difficulties. When interacting with devices, users build

and employ two types of mental models: structural and functional models (Preece et al., 1994).

Users can build a structural model of a system when they have grasped, understood and stored in memory the structure of how the devices work. Typically, structural models are simplified models that enable the person using them to make predictions about the behaviour of the devices they represent. In other words, a structural model is a representation of "how-it-works". The advantage of structural models is that by explaining how a device works, they allow a user to predict the effects of any possible sequence of actions, and hence to work out how to achieve most tasks possible with the device (Preece et al., 1994). They are particularly useful when a device breaks down or, by extension, when it commits errors. However, constructing a structural model is difficult and often requires a great deal of effort.

A model that represents "how to do it" is a functional model. To build a functional model, users must have acquired procedural knowledge about how to use the devices. Functional models are normally constructed using existing knowledge of similar domains and situations, but in desktop and mobile HCI, the widely used visual interface metaphors (e.g. the office desk metaphor with its file and folder icons) have become the models that users learn. Most of the time, functional models are sufficient and people seem to get by without using structural models (very few computer users know about the internals of a computer and how it works, but every regular computer user knows how to use it in order to accomplish their task). Indeed, according to (Preece et al., 1994), users tend to develop functional-based mental models of systems while remaining largely unaware of their structural aspects.

During multimodal error handling, however, both structural and functional mental models are useful. For example, users achieve error prevention by effectively allocating inputs to modalities, sometimes producing complementary or redundant inputs. The allocation of inputs to modalities necessitates a good understanding of the devices used for data capture, of the nature of the captured data, and of the use that is made of it. In other words, it necessitates a good mental representation of "how it works" in order to predict the system's responses to a planned sequence of actions. Similarly, user correction strategies require adequate knowledge about "how to do things" in order to come up with alternative ways of inputting information, which will effectively repair system's errors.

In pervasive computing, the invisibility and unobtrusiveness requirements make it impossible to develop visual interface metaphors, which have become so familiar in more traditional computing applications. Hence even functional mental models are difficult to convey and build. Generally, users get to find out about a system through its physical interface and its behaviour, i.e. what is called the system image. In pervasive computing, the system's physical interface may have completely disappeared, rendering the system image evasive, to say the least. The system image also includes the system's behaviour, i.e. the way it responds. The difficulty in pervasive computing is that the system's response may not be in relation with any user's conscious action but with environmental changes, and hence appear to be unpredictable or incomprehensible. If the system image is not able to convey to the users the designer's model in a clear and obvious way, then it is likely that the users will develop incorrect mental models. Consequently, they will experience great difficulties in understanding the system, using the system and knowing what to do when the system doesn't behave in the way they assumed it would.

One fundamental advantage of structural models is that they allow a user to predict the effects of any possible sequence of actions. As mentioned already, an additional difficulty in pervasive computing is that users are not always aware of the actions that have been captured, and the effects that can be observed (system's response) may have been triggered by environmental changes that users have not perceived or paid attention to. Here the "what data" property, once more, is the main obstacle. Structural models, in principle, also allow to work out how to achieve most tasks possible with the device. However, in pervasive computing, the notion of task is not always relevant, as the pervasive system is sometimes working on our behalf (see the smart home applications) or is trying to automatically adapt to our needs and affective state (see location-aware systems and affective computing applications).

Some work has highlighted the importance of a user-centred approach to the design of pervasive computing applications. (Ciarletta & Dima, 2000) have adapted the OSI reference model (Open Systems Interconnection model) to pervasive computing, adding a model of the user to their pervasive computing conceptual model (see Fig. 3). In particular, the abstract layer formalizes the necessity of maintaining consistency between the user's reasoning and expectations (Mental Models) and the logic and state of the pervasive computing application (Application). The intention is that, given the limited techniques that pervasive computing applications developers can use to communicate the state of the application, the proposed conceptual model will "force pervasive developers to consider the user's point-of-view much more than developers in traditional environments".

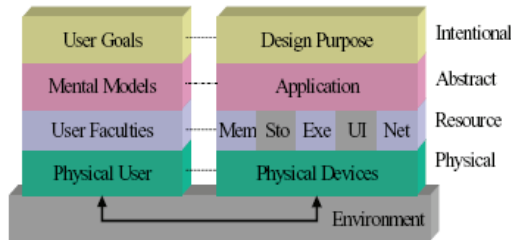


Fig. 3. Pervasive computing conceptual model (reproduced from (Ciarletta & Dima, 2000))

(Dobson & Nixon, 2004) clearly state that it is vitally important that users can predict when and how pervasive systems will adapt (i.e. respond to inputs and environmental changes), and can perceive why a particular adaptation has occurred. Arbitrary behavioural changes are incomprehensible to users and make a pervasive system completely unusable; on the other hand, single behaviour is unattractive in that it prevents a system from adapting to context. The difficulty is thus to find the optimal balance between adaptability (reactivity to contextual changes) and comprehensibility (leading to predictability). They conclude that predictability can arise in pervasive computing applications from having a close, structured and easily grasped relationship between the context and the behavioural change that context engenders. In other words, an application's behavioural variation should emerge "naturally" from the context that causes it to adapt, and any change in behaviour should be accompanied by a perceptible change in the context that "makes sense" for the application at hand. Moreover, the changes should correspond to external contextual cues that convey the need for the behavioural change to the user. This way, users should be able to build

functional mental models that allow them to use pervasive computing applications in most normal situations. Functional models might not be enough, however, to cope with abnormal situations where error handling has become necessary.

More recently, (Leichtenstern & Andre, 2008) explore the idea of using mobile phones as interfaces with pervasive computing environments, as they are devices that most users are already familiar with. The mobile interface is designed following a usage model which specifies various contexts, users and environment, as well as the user's goals and mental model. The mobile interface, while remaining familiar, is adaptable according to the current state of the usage model.

6. Conclusion

In this chapter we addressed the new challenges raised by novel pervasive computing applications for the handling of uncertainty and errors. We exposed the differences between "traditional" multimodal systems and pervasive computing applications, such as context and location aware systems, affective computing applications, smart homes and wearable computers. In particular, we discussed the inadequacies of known multimodal error handling strategies in pervasive environments, where the devices are heterogeneous and have become invisible, and where users largely remain unaware of the types and properties of the data that these devices capture and exploit. We observed that most traditional error strategies for error prevention have become impractical because they are irreconcilable with the fundamental principle of unobtrusiveness in pervasive computing. We also observed that most user strategies for handling errors were dependent on users being able to identify the source and cause of the error and on users having good structural and functional mental models of the interactive systems.

With the increasing diversity of devices, contexts of use, and users, the design of effective means of error prevention, detection, and correction will be a determinant factor of usability and users' acceptance of pervasive computing applications. This chapter has highlighted the necessity of providing users with appropriate support to allow them to devise and deploy adequate strategies for handling errors. However, error handling in pervasive computing applications is more complex than in current multimodal interfaces. In pervasive computing, it will be of paramount importance that users are supported in their forming of adequate mental models of the system. These mental models should provide users with the correct knowledge of what data is captured and recorded, and how it is used. Because of the invisibility of the devices and the necessity of being unobtrusive, supporting the development of adequate mental models is more challenging than in traditional interfaces. Provided that the pervasive computing application successfully promotes adequate mental models, it can be anticipated that users will develop whole new strategies to cope with errors in pervasive computing applications, and research to gain a better understanding of these strategies will be needed in order to devise appropriate interface designs and techniques to support them.

7. References

- Abowd, G.D. & Mynatt, E.D. (2000). Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer Human Interaction*, Vol. 7, No. 1, (2000) 29–58

- Baber, C. & Hone, K.S. (1993). Modelling error recovery and repair in automatic speech recognition. *International Journal of Man-Machine Studies*, Vol. 39, No. 3, (1993) 495-515
- Benoit, A.; Bonnaud, L.; Caplier, A.; Damousis, I.; Tzovaras, D.; Jourde, F.; Nigay, L.; Serrano, M. & Lawson, J.L. (2007). Multimodal signal processing and interaction for a driving simulator: component-based architecture. *Journal on Multimodal User Interfaces*, Vol. 1, No. 1, (2007)
- Bourguet, M.L. & Ando, A. (1998). Synchronisation of speech and hand gestures during multimodal human-computer interaction, *Extended Abstracts CHI 1998*, pp. 241-242, Los Angeles, USA, April 1998, ACM Press
- Bourguet, M.L. (2006). Towards a taxonomy of error handling strategies in recognition based multimodal human-computer interfaces. *Signal Processing journal*, Vol. 86, No.12, (December 2006) 3625-3643
- Bourguet, M.L. (2008). Handling uncertainty in pervasive computing applications. *Computer Communications journal*, Vol. 31, No. 18, (December 2008) 4234-4241
- Ciarletta, L. & Dima, A. (2000). A Conceptual Model for Pervasive Computing, *Proceedings of the 2000 International Workshop on Parallel Processing*, Washington, DC, USA, August 2000
- Crowley, J.L. (2006). Things that See: Context-Aware Multi-modal Interaction. *Lecture Notes in Computer Science*, Vol. 3948, (2006) 183-198
- Dey, A.K. (2001). Understanding and Using Context. *Journal of Personal and Ubiquitous Computing*, vol 25, (2001) 4-7
- Dobson, S. & Nixon, P. (2004). More principled design of pervasive computing systems, *Proceedings of the Engineering for Human-Computer Interaction and Design, Specification and Verification of Interactive Systems*, pp. 292-305, Hamburg, Germany, July 2004
- Edmonds, M. (2010). How Smart Homes Work, Copyright © 2010 Discovery Communications Inc. <http://tfc.howstuffworks.com/home/smart-home1.htm>
- Feki, M.A.; Renouard, S.; Abdulrazak, B.; Chollet, G. & Mokhtari, M. (2004). Coupling context awareness and multimodality in smart homes concept. *Lecture Notes in Computer Science*, Vol. 3118 (2004) 906-913
- Ferscha, A. & Zia, K. (2009). LifeBelt: Silent Directional Guidance for Crowd Evacuation, *Proceedings of 13th International Symposium on Wearable Computers (ISWC 09)*, pp. 19-26, Linz, Austria, September 2009
- Frankish, C.; Jones, D. & Hapeshi, K. (1992). Decline in accuracy of automatic speech recognition as a function of time on task: fatigue or voice drift? *International Journal of Man-Machine Studies*, Vol. 36, (1992) 797-816
- Halverson, C.; Horn, D.; Karat, C. & Karat, J. (1999). The beauty of errors: patterns of error correction in desktop speech systems, *Proceedings of the INTERACT 99*, pp. 133-140, Edinburgh, UK, September 1999, IOS Press
- Heer, J.; Good, N.; Ramirez, A.; Davis, M. & Mankoff, J. (2004). Presiding over accidents: system direction of human action, *Proceedings of the ACM CHI'04*, pp. 463-470, Vienna, Austria, April 2004
- Holle, H. & Gunter, T.C. (2007). The role of iconic gestures in speech disambiguation: ERP evidence. *Journal of Cognitive Neuroscience*, Vol. 19, No. 7, (2007) 1175-1192

- Kapoor, A. & Picard, R.W. (2005). Multimodal affect recognition in learning environments, *Proceedings of the ACM MM'05*, Singapore, November 2005
- Karat, C.; Halverson, C.; Horn, D. & Karat, J. (1999). Patterns of entry and correction in large vocabulary contentious speech recognition systems, *Proceedings of the ACM CHI'99*, pp. 568-575, Pittsburgh, Pennsylvania, May 1999
- Khoury, H.M. & Kamat, V.R. (2009). High-precision identification of contextual information in location-aware engineering applications . *Advanced Engineering Informatics journal*, Vol. 23, No. 4 (October 2009) 483-496
- Leichtenstern, K. & Andre, E. (2008). User-centred development of mobile interfaces to a pervasive computing environment, *Proceedings of the First International Conference on Advances in Computer-Human Interaction*, pp. 114-119, Sainte Luce, Martinique, February 2008
- Luo, A.; Zhou, J.; Wang F. & Shen, L. (2009). Context Aware Multimodal Interaction Model in Standard Natural Classroom. *Lecture Notes in Computer Science*, Vol. 5685, (2009) 13-23
- Macho, D.; Padrell, J.; Abad, A.; Nadeu, C.; Hernando, J.; McDonough, J.; Wolfel, M.; Klee, U.; Omologo, M.; Brutti, A.; Svaizer, P.; Potamianos G. & Chu, S.M. (2005). Automatic speech activity detection, source localization, and speech recognition on the chil seminar corpus, *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 876-879, Amsterdam, July 2005
- Matsumiya, K.; Aoki, S.; Murase, M. & Tokuda, H. (2003). Active authentication for pervasive computing environments. *Lecture Notes in Computer Science*, Vol. 2609, (2003) 267-273
- Meier, U.; Stiefelhagen, R.; Yang, J. & Waibel, A. (2000). Towards unrestricted lipreading. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 14 (2000) 571-585
- Narayanan, S. (2002). Towards modeling user behavior in human-machine interactions: effect of errors and emotions, *Proceedings of the ISLE Workshop on Dialogue Tagging for Multi-modal Human Computer Interaction*, Edinburgh, UK, December 2002
- Norman, D.A. (1988). *The Psychology of Everyday Things*, Basic Books, New York
- Oikonomopoulos, A.; Patras, I. & Pantic, M. (2006). Human action recognition with spatiotemporal salient points. *IEEE Transactions on Systems, Man and Cybernetics Part B*, Vol. 36, No.3, (2006) 710-719
- Oviatt, S.; Cohen, P. & Wang, M. (1994). Toward interface design for human language technology: modality and structure as determinants of linguistic complexity. *Speech Communication*, Vol. 15, (1994) 283-300
- Oviatt, S. & van Gent, R. (1996). Error resolution during multimodal human-computer interaction, *Proceedings of the Fourth International Conference on Spoken Language Processing*, pp. 204-207, Philadelphia, Pennsylvania, October 1996
- Oviatt, S. (1999). Ten myths of multimodal interaction. *Communication of the ACM*, Vol. 42, No. 11, (1999) 74-81
- Oviatt, S. (2000). Taming recognition errors with a multimodal interface. *Communications of the ACM*, Vol. 43, No. 9, (2000) 45-51
- Pantic, M. (2005). Affective Computing, In: *Encyclopedia of Multimedia Technology and Networking*, vol. 1, pp. 8-14, Hershy, PA, USA, Idea Group Reference

- Preece, J.; Rogers, Y.; Sharp, H.; Benyon, D.; Holland, S. & Carey, T. (1994). Knowledge and Mental Models, In: *Human-Computer Interaction*, pp. 123-139, Addison-Wesley
- Singh, P. (2002). The public acquisition of commonsense knowledge, *Proceedings of the AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, Palo Alto, California, 2002
- Stillman, S. & Essa, I. (2001). Towards reliable multimodal sensing in aware environments, *Proceedings of the PUI 2001*, Orlando, Florida, November 2001
- Suhm, B.; Myers, B. & Waibel, A. (1999). Model-based and empirical evaluation of multimodal interactive error correction, *Proceedings of the ACM CHI'99*, pp. 584-591, Pittsburgh, Pennsylvania, May 1999
- Suhm, B.; Myers, B. & Waibel, A. (2001). Multimodal error correction for speech user interfaces. *ACM Transactions on Computer-Human Interaction*, Vol. 8, No. 1, (2001) 60-98
- Tsai, C.W.; Chou, S.Y. & Lin, S.W. (2010). Location-aware tour guide systems in museums. *Scientific Research and Essays journal*, Vol. 5, No. 8, (April 2010) 714-720
- Vieroth, R.; Löher, T.; Seckel, M.; Dils, C.; Kallmayer, C.; Ostmann, A. & Reichl, H. (2009). Stretchable Circuit Board Technology and Application, *Proceedings of 13th International Symposium on Wearable Computers (ISWC 09)*, pp. 33-36, Linz, Austria, September 2009
- W3C. (2003) <http://www.w3.org/TR/mmi-framework/>
- W3C. (2002) <http://www.w3.org/2002/01/multimodal-charter.html>
- Yoshimi, B.H. & Pingali, G.S. (2002). A multi-modal speaker detection and tracking system for teleconferencing, *Proceedings of the ACM Conference on Multimedia*, pp. 427-428, Juan-les-Pin, France, December 2002
- Yue, W.; Mu, S.; Wang, H. & Wang, G. (2005). TGH: A case study of designing natural interaction for mobile guide systems, *Proceedings of Mobile HCI'05*, pp. 199-206, Salzburg, Austria, September 2005



Ubiquitous Computing

Edited by Prof. Eduard Babkin

ISBN 978-953-307-409-2

Hard cover, 248 pages

Publisher InTech

Published online 10, February, 2011

Published in print edition February, 2011

The aim of this book is to give a treatment of the actively developed domain of Ubiquitous computing. Originally proposed by Mark D. Weiser, the concept of Ubiquitous computing enables a real-time global sensing, context-aware informational retrieval, multi-modal interaction with the user and enhanced visualization capabilities. In effect, Ubiquitous computing environments give extremely new and futuristic abilities to look at and interact with our habitat at any time and from anywhere. In that domain, researchers are confronted with many foundational, technological and engineering issues which were not known before. Detailed cross-disciplinary coverage of these issues is really needed today for further progress and widening of application range. This book collects twelve original works of researchers from eleven countries, which are clustered into four sections: Foundations, Security and Privacy, Integration and Middleware, Practical Applications.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Marie-Luce Bourguet (2011). Uncertainty and Error Handling in Pervasive Computing: A User's Perspective, Ubiquitous Computing, Prof. Eduard Babkin (Ed.), ISBN: 978-953-307-409-2, InTech, Available from: <http://www.intechopen.com/books/ubiquitous-computing/uncertainty-and-error-handling-in-pervasive-computing-a-user-s-perspective>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.