Chapter 14

Sociotechnical Ambient Systems: From Test Scenario to Scientific Obstacles

14.1. Introduction

Scientific advances have lead to an explosion in the number and functions of electronic devices used in everyday lives, throwing us into the world of ambient intelligence, as defined by Weiser 20 years ago in 1991 [WEI 91]. In this context, systems' designs presuppose movement from a centered point on multifunction machines toward a set of devices with varying interactions that are distributed and dispersed in the environment, which can be accessed by interfaces, thrusting the user into increasingly realistic and mixed worlds. As a result, the individual and her/his social, physical, and organizational contexts are at the heart of considerations in the design of these systems, which must be adapted to users' needs and behaviors. This human-machine coupling must be intuitive in correlation with the content of exchanges and multimodality. Some processes may also need to occur on demand in the network (system coupling) without the users' outside control and must, in addition, be implemented by embedded autonomous entities acting collectively. These systems, consisting of human beings and components continuously interacting, may be physical entities or distributed devices. They are autonomous and have the ability to adapt to a human being's task and to the available digital physical resources.

In the field of ambient systems, the current design covers a spectrum ranging from dedicated *ad hoc* systems, to open and evolving systems. On the one hand, a system specific to one application may be implemented by an ambient system, like for example, the followup of a package by an RFID tag. A standard design focused around

---

Chapter written by Georges DA COSTA, Jean-Pierre GEORGÉ, and Marie-Pierre GLEIZES.

this kind of scenario results in an *ad hoc* system, which is highly efficient but may be difficult to use in another context. On the other hand, ambient systems in the future will not be completely specified but complex, distributed, and open. Designers will no longer have the ability to completely control them and create them in the standard descending way.

To relieve the challenge of designing these systems, interdependent and interdisciplinary research must solve these related problems

– At a collective level: this entails studying interactions (between artificial and/or human agents) and the means of obtaining coherent and adapted collective behavior despite complexity and dynamics.

– At the human level: we need to understand the user's needs and behavior.

– At the level of the artificial agent: it is necessary to study its different aspects such as decision-making, autonomy, and adaptation.

– At the environmental level: we need to know how to represent and recognize the environment to adapt to it.

– Design methods and tools: future designers need to have access to a series of models, methods, and tools to deploy these systems.

This chapter focuses on the preliminary design stage for ambient socio-technical systems that are not designed for a specific application, but a more general application that is described by one or more scenarios. The aim here is partly to examine tests led in the workshop on ambient systems, which have been undertaken at ETIA (Ecole Thématique Intelligence Ambiante, Villeneuve d'Ascq, 2009)[1]. The 21 participants in the workshop mostly consisted of researchers (PhD students and researchers) in computer science working on different sub-fields such as human–machine interaction, information systems, networking, etc. The aim of the workshop was to examine a mockup of an ambient socio-technical system in the form of software and/or material building blocks and to define the main challenges facing researchers in ambient intelligence today.

14.2. Definitions and characteristics

Ambient systems are designed to provide adapted services that respond to an individual, collective, and social requirements [COU 08]. Their "ambient" components must consist of physical entities (PDAs, sensors, etc.) or distributed software. These entities are known as AmID (*Ambient Intelligence Devices*) in this chapter. AmIDs, have, by default, the ability to interact (perception and action),

1 www.univ-valenciennes.fr/congres/etia09/

a more or less “intelligent” autonomous behaviors and may have the ability to adapt themselves to the user’s current task and the available digital and physical resources.

The development of ambient systems requires the use of different technologies and/or tools. The latter are described in the form of a stack (Figure 14.1), called AmiLab (*Ambient Laboratory*) where the lowest levels represent the most basic components and the highest levels describe more complex elements designed using lower levels. In AmiLab, considered levels are as follows

– components that are basic electronic entities;

– sensors and effectors capable of collecting data and adapting to their environment, respectively;

– AmIDs that correspond to an augmented entity (either sensor or effectors, which are, for example, capable of perception, decision-making, and action);

– network functions that allow an AmID to communicate with at least one other AmID;

– *middleware* that provides functions relating to discovery, interaction, and dynamic composition of functions and sensory-motor devices;

– user services that correspond to the “intelligent” level where the most relevant function to the user is carried out, the interaction with the user.

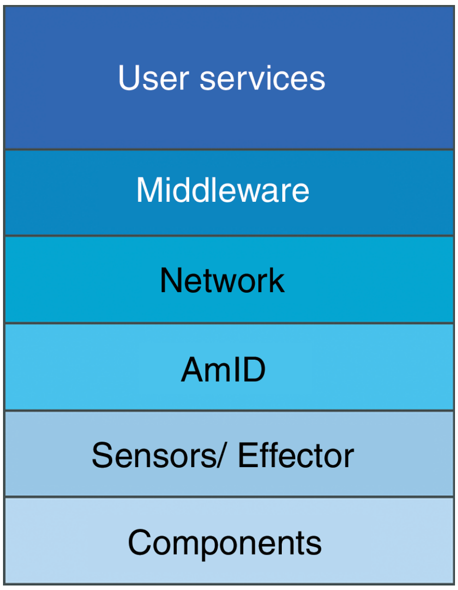

Figure 14.1. *Amilab structure*

The basis of ambient systems relies on the notion of interaction (human-system or system-to-system) and on the local autonomous component of AmIDs. Indeed, interaction increases the action capabilities of system components. As a result, the challenge of designing such systems is based on the hyper-interaction capabilities between human and/or artificial components. It implies that components in these systems must be able to interact with other components that are not known in advance by humans/users. These interactions can take place between digital entities (agent, processing capabilities, etc.), physical entities (such as medication, patient beds, PDAs, classrooms, etc.) and environmental elements (such as light levels and temperature in the home, weather on campus, etc.) or a collection of all or some of these entities and environmental elements. In addition, because components are autonomous and mobile, they must be able to judge the most relevant means of interaction and find other entities with which they can interact according to their environment.

The central and original point here is to provide AmIDs with the means of adapting to their context, either by knowing to choose the adapted interface, or to carry out the correct action to satisfy the needs of users while being transparent for the user. To respond to these challenges, among others, new grounds must be explored. It is necessary to gradually abandon finality during the AmID design stage. The AmID designer therefore does not know the role the design will play in its projected ambient system and its initial function may be changed. The mobile phone, for example, is used today as a mirror or flashlight that is completely different to its original function. At the design stage, it is necessary to account for essential conditions according to which AmIDs must function:

– An AmID ignores the explicit collective finality of the AmID network in which it is located. This sets it apart from standard programming for which the adjustment of a collective finality assimilate to *feed-back* in cybernetics. An AmID's specification is, as a result, incomplete.

– An AmID ignores the languages of others, that are not of the same kind and ontologies are not initially shared. This is one of the presumptions in ambient systems because a component at its design phase will not have been specified to interact during its life with all other AmIDs.

– Each AmID is generally able to perceive actions in the environment (including humans or artificial systems) by itself via an interface. This interface must adapt to the context.

– Each AmID must have action capabilities in the environment using effectors or communicating via interfaces.

– An AmID must be informed of the actions of AmIDs around it, notably via messages and sensors.

The design of these systems requires a different approach to that of standard systems and software because, as they are open, incompletely specified, complex, and distributed, designers therefore no longer have the ability to completely control them and design them in descending order. Collective functions must therefore be developed [DIM 05].

In addition, systems can no longer be designed by a single design team at the same time. This indicates that the designer no longer has access to completely formalized final needs of a client. The previously cited design approach for ambient socio-technical systems entails defining and modeling a general basic inoperable architecture, which is open and can be extended. This technological basis enables the introduction and development of complementary, multi-use, and reusable building blocks that are vital for the creation of innovative applications for roaming users. A primary example is that of collecting environmental information and the user's context, security, and protection of private life are essential points of this.

14.3. Real-life scenario: *Ambient Campus*

To design this kind of system, it is tempting to simply combine the different technologies related to ambient systems. It is more pertinent to base this on different scenarios, even if they are currently prospective to more precisely evaluate genuine needs and constraints. The *Ambient Campus* scenario shows what we can reasonably expect from the development of these technologies in just a few years time. A university campus, for example, integrates multiple infrastructures and interdependent services that come into their own when they have a continuous and considerable flow of information. The campus is a large-scale ambient system whose infrastructure includes rooms, buildings, and services in close interaction with users such as students, teachers, administrators, or management.

5th September 2029. Neo has just arrived on Paul Sabatier Corp (PSC) campus. His personal digital AMI has taken care of every aspect of administration and Neo only has send his electronic signature to PSC to confirm as well as accept his course fees. For this verification, as for any transaction over $1,000, Neo's identity is biometrically confirmed.

When he exits the metro, his AMI sends a campus map to his mobile phone. This map, made available by BIG, the campus information system, is specifically generated for Neo when his location on the campus and profile are detected, and specific elements have been highlighted (its the first inscription of Neo in License). BIG contacted AMI to transmit this map.

The screen on Neo's phone is quite small and retinal projection systems are still expensive. One of the campus information screens offers him to display the map between two advertisements. Neo accepts and without any effort on his part, the map is displayed. In addition, these screens are fitted with intuitive control interfaces and Neo can easily use this map for additional information, etc.

AQ1

Figure 14.2. *View of an ambient campus*

19th October. Neo is outside his next class with the students from his group. The secretariat changes the room the last minute. The students are notified of this change and an itinerary is provided to go to the new room as quickly as possible. This information is also sent to Professor Smith, their lecturer. Unfortunately, the second room is already being used (after all, room planning in 2029 is not so different from 2012). This time, PRO, Professor Smith's personal agent, searches for an alternative. A neighboring room, detecting itself empty and having all the necessary requirements, responds to PRO.

The lesson starts and Neo comes to the board to solve an exercise. Meanwhile, Professor Smith studies the information BIG has sent him about Neo. He can see his mixed track record (after all a student in 2029 is not so different from the one in 2012), as well as his former education that explains a gap in his knowledge, making the exercise more difficult. Professor Smith compares this with the other students and decides to slightly change the lesson to adjust and provide the missing knowledge.

Unfortunately, a fire alarm goes off. Is it real or a drill? As a research team from the computing research laboratory is testing a crisis management system, alerts increase but in cases of doubt, it is always best to evacuate. Students have also received a map on their phone based on their current position, which indicates the evacuation route toward the nearest assembly point.

2nd December. Neo is having a few difficulties with one subject, but is not alone. AMI puts him in contact with four other students experiencing the same problems (in the same way Neo has been put in contact with a number of friends). They meet in a revision room reserved for them by their AMIs. These rooms are particularly well technologically equipped for learning and the real work can begin. Their activity is analyzed and compared with a database of work sessions. The system gradually suggests supporting materials and other electronic resources.

At lunch time, the students decide to eat. Their AMIs notify them like usually the occupation of the three university cafes. This data collected in real time reflects the estimated waiting time fairly precisely. One of the cafes is less busy and their AMIs suggest this. The same mechanisms are also used to collect statistics of campus use to foresee new investments. The university is therefore one of the first to have paved paths, which students actually walk on!

Neo's friends from his favorite online role-playing game also appear on his screen with their position. If authorized, Neo can find them quickly and know their availability and the best means of contacting them. The authorizations can be managed according to categories (friends, colleagues, students) and can be dynamic (i.e. change over time, workload, etc.). Information is collected as and when by their AMIs, location via Wifi (average precision), RFID in rooms (high precision), availability given per person (interface on a mobile phone or PDA), or deduced (class room, IT rooms, meeting rooms, offices, on the phone, at the computer, etc.). However, unfortunately, the location alert functions ("you are 20 m from X who you urgently AQ2 need to see to get a signature") and the prediction of path overlaps must be paid for. However, this was the precondition for SmallBrother investing in PSC campus (as well as sitting on the board of directors).

Neo can manage a list of people in this same way. It has therefore given his mobile phone the ability to go into "available" mode (where he can meet anyone) to "ninja" mode (where he is invisible on campus, particularly useful when he wants to see his girlfriend).

13th February. Neo is surprised when his AMI suggests a new service, apparently, produced independently by a collection of services by the AMI/PRO/BIG collective on campus. Finally, the "killer-app" he has been waiting for and will push him to finish his studies!

14.4. Intuitive architectures

In an ambient system, AmIDs interact and even cooperate together with their environment. In the almost opposite manner to the completely distributed and ubiquitous entities that are AmIDs, this environment can be either completely centralized (such as the management of geographic data in Android [BIS 03]) or non-existent (if we consider AmIDs to be self sufficient). To work on these different hypotheses during the workshop, the *Ambient Campus* scenario (section 14.3) was given to participants. Their goal was to propose a supporting infrastructure for this scenario. Objectives were also to understand the multi-disciplinary aspect of these problems and to highlight scientific challenges.

14.4.1. *The building blocks of the Ambient Campus scenario*

As a first stage, the majority of groups in the workshop identified the need to define the entities involved in the scenario. The main types of non-human entities collected were: BIG, Paul Sabatier Corporation, Rooms, AMI (one or more per student), and PRO. AMIs, and PRO were entities that enter and exit the ambient system considered.

The groups then focused on the concept of services. Two types of services were identified in this scenario:

– services specific to the application such as subscription management, map management, room bookings;

– generic services such as locating an individual via his/her *smart phone*, authentication of new arrivals, management of access rights, data security (payments), respect for private life, managing profiles and roles, adapting information content to means of communication between entities, implementing information systems, and managing communication between entities.

These different services and entities use shared information of different types, ranging from the state of a room light to the set of students' trajectories during a crisis situation. The central question concerns the type and form of infrastructure allowing the management and communication of this information between services and entities.

14.4.2. *Limitations of simplistic infrastructures*

Initially, it appears that the two extreme solutions (AmIDs alone or a completely centralized structure) were unusable. As we will see in the next section, a shared infrastructure is necessary to enable AmIDs to offer authorizations (building and data access for instance). In addition a totally centralized version is impossible as it would entail too much communication when deployed on a large scale.

Let us first consider the limitations of systems without a support infrastructure. To operate, AmIDs must collect information on their environment (temperature, proximity to other AmIDs, etc.). It is therefore necessary to be able to identify AmIDs to manage their access rights to this information, but also actions they may make. A student can therefore not cancel all her/his lessons for a day, but a personal tutor can. It should be noted that there is a fine line between some fixed AmIDs and some infrastructure elements. A door can therefore be considered in the two roles depending on the exact definition of the chosen infrastructure.

With regard to centralized systems, they can only be used when we want a localized and autonomous system, such as a museum, for example. If we want an open system such as on a campus, it is costly and complex to centralize and manage all the information produced by the different elements in the infrastructure. In any case, centralizing all the information in a system is not useful in itself. While it is interesting for an AmID managing a blind to know the exact light levels inside and outside a room, centralizing the two values for all the rooms on a campus has little meaning and is highly costly in terms of communications.

14.4.3. *Context and role bubbles*

AmIDs are situated in a complex environment, with each in turn interacting with different elements. In one room, interaction can take place with other AmIDs as well as with another AmID on the same floor (a coffee machine or printer, for example) or even in the same country (verification of the status of a bank account, for example).

Context bubbles can be defined as the set of coherent contexts (in terms of scope and meaning). For example, a person is her/his own closest bubble. A second bubble is the room in which s/he is in. Its building is then next, followed by its campus and so on, etc.

Equally, role bubbles refer to AmID roles in relation to their context. A user can have a role as a *teacher* as well as another role as a *student*. However, away from their establishment, these roles do not define users. Roles are strongly linked to context bubbles and it is possible to have several roles simultaneously. A student can therefore be a first aider and thus can become the person responsible if someone is feeling faint during a course.

Every AmID is therefore immersed in a constant interaction between context and roles:

– a context is an overlap of several different contexts (interrelated geographic bubbles);

– a change of context is therefore a geographic movement;

– a change of role follows a change in context, but can also a be a logical change (i.e. obtaining a first-aid certificate);

– infrastructure (centralized or even completely distributed) manages roles and transmits context.

Infrastructure therefore manages data effectively, whether on a global or local level. However, managing data does not necessarily involve transmitting them. A fundamental objective of infrastructure is the detailed management of data by:

– Adaptation: when AmIDs do not have the same needs and characteristics (more specifically from the perspective of communications), it is necessary to adapt data before sending it to AmIDs. A *smartphone* does not need the same video flow as a television and transmitting a non-adapted HD video flow to a *smartphone* would generate a network overload and reduce its autonomy while still displaying a miniature version. Equally, the service needs of different AmIDs are different and adaptation means that these can be adapted as needed. One of the most common uses is that of marketing, service research, or personalization.

– Aggregation: transmitting aggregated data reduces the amount of processing needed for an AmID while reducing the number of communication resources used. As a result, to find a free room with a video projector, it is simpler for an AmID to directly ask for the aggregated response rather than ask for the state of all rooms. The same also applies when we are interested in an average value such as the temperature or humidity of a building.

– Synthesis: this is one of the most complex aspects because it is related to the effective use of data introducing a vast range of possibilities. Adaptation and aggregation are elements that react to the infrastructure while synthesis can be seen as a more proactive aspect. Depending on its context and specific rules, it can produce new data. Synthesis can therefore detect several students having the same difficulties and put them into contact with one another.

This logical view of hierarchical context bubbles allows us to design an ambient supportive infrastructure away from technical constraints.

14.5. Scientific challenges

This logical vision is not directly and simply transposable to reality. There are two kinds of scientific challenges that emerge from a logical perspective concerning bubbles and roles' parameter, and from a technical perspective with regard to the hardware and software structure, which provides this logical infrastructure.

However, ambient intelligence is, by its very nature, multi-disciplinary and raises a number of other scientific challenges in each of the disciplines concerned within social and human sciences or ICT. We will focus here on the levels involved in software research ranging from AmIDs to services within the field of ITC. Difficult challenges will be treated at each level, between AmIDs, the network, and the aspects of behavior, and the decisions to examine the challenges of the ambient systems presented in section 14.2. The most significant scientific challenges for an ambient system are:

– System acceptability and respect for private life are challenges that must condition any development of the application and must be ensured at the four levels of Amilab: AmID, networks, *middleware*, and services. In the *Ambient Campus* scenario, people on campus must be able to easily use this new system. However, students' private data such as their location outside of class hours must not be accessible by teachers.

– Reliability entails ensuring that the system functions correctly. The mechanisms implemented must be reliable at all levels. This problem is not specific to ambient computing, but is dominant because these applications are centered around humans and must be reliable at all levels.

– The system's reactivity allows it to intervene in real time within an acceptable timeframe for the situation in question. For instance, Neo, arriving for the first time on campus, must have a map sent to him within a reasonable timeframe so that he can use it to get around.

– Discretion supports an embedded system, which is less intrusive and therefore more acceptable. This challenge requires the system to have self-* properties [KEP 03], which ensure system autonomy and therefore as little intervention as possible by the user. The ambient campus applications therefore do not mean that students have to wear special clothing filled with sensors. In this scenario, the technology exists and is familiar to users.

– Adaptation and auto-adaptation allow a system to adapt independently to context and to its user(s). Adaptation can be treated at the AmID level which, through its learning capabilities, can modify its future behavior to adapt itself. The network can adapt to maintain its level of service by, for example, ensuring connections. In terms of *middleware*, adaptation allows components to exploit their environment. Human-machine interaction must also be able to adapt to user's conditions. Adaptation refers to the system's cognitive abilities or the system's learning capacity, i.e. the ability to test and improve its behavior. Adaptation results in a more robust system, i.e. it is more resistant to malfunction. In this scenario, adaptation is necessary to allow adapted labeling to support interaction with the campus map. It is also required to manage an event such as changing rooms and lack of availability. Auto-adaptation is manifested by creating new services proposed by the collective at the end of the scenario.

– The management of context relates to the representation, memorization, and use of elements (environment, other human or artificial actors, etc) with a potential impact on the execution of one or several tasks. The notion of context is important at all four levels of Amilab. Its model is complex since its environment with constantly changing actors is highly dynamic. Context is necessary due to its use in processes such as adaptation, personalization, information transmission, and interaction. Managing context is necessary throughout the scenario so that entities consider the most pertinent decisions such as, for instance, helping students by suggesting supporting materials based on students' current activity and their educational profile.

– Optimality in the system's function is linked to its ability to efficiently manage resources such as energy, memory, or resources related to the application and therefore is evaluated at all levels of Amilab. Resources such as university cafes are more commonly used and provide users with more satisfaction.

– The opening of the system by accepting new entities in the system. The arrival of new AmIDs into the ambient system requires the design of models and mechanisms for introducing new elements into the network, *middleware*, and services. The system must be able to deal with a large number of situations that are unforeseen during design. On a campus, it is clear that entities such as mobile telephones and laptops must enter and exit, as is the case when Neo arrives on campus for the first time.

– Interoperability between all the systems is also necessary to obtain an open system, knowing that the whole system is either designed by a single person nor with a single technology. This challenge strongly affects the deployment of such systems and must be taken into consideration at the four levels of Amilab. There are different types of AmIDs in this scenario: mobile phones, laptops, billboards, library management systems, etc. and they must be able to interact with one another.

These several transverse challenges are not the only scientific challenges raised by ambient system. Many other, more specific challenges are involved in the development of well-performing ambient systems. The rest of this section will therefore focus on detailing the major challenges in computing research highlighted in the workshop. These scientific issues are described in terms of Amilab's major aspects.

14.5.1. *AmID*

The aim is to create AmID entities using low-level sensors and effectors (see Figure 14.1). We will focus on the perception aspect of the AmID, which has two large challenges: processing information and analyzing and structuring video data.

14.5.1.1. *Data processing*

This consists of processing, in real time, flows of sensory data (video, audio, RFID, laser, etc.) collected by fixed or embedded sensors. The capabilities of fixed sensors must be improved and provided with processing capabilities to limit network bandwidth [BER 11]; each sub-system receives data and analyzes it to detect events (such as a person or vehicle) and some behavior (human interactive gestures, dangerous situations such as falls, risks of collision, aggressive attitudes, etc.) and sends alarms on the network to neighboring sub-systems (to control it) and to a control station (to merge with other information, etc.)

14.5.1.2. *Analysis and structure of video data*

The analysis and structure of video data can entail camera data, which may be equipped with micros. There are two types of cameras used: fixed (*webcam style*) or hand held (as in the ANR Blanc 2009 IMMED project:Indexing Multimedia Embedded Data for diagnosing and monitoring dementia treatment, which studies the behavior of patients predicted to develop Alzheimer's). It is today possible to extract and identify the different people appearing in a video [ZHA 10]. To achieve this, the visual and audio data associated with these people is used. However, this raises a number of challenges, which include:

– Adaptation to context: recognizing people (or objects) is made difficult in low light, long distance from the camera (problems of scale), and camera angle. In the case of hand-held cameras, there is the additional problem of the carrier's movement. Audio sensors are also occasionally unusable (due to too much noise). This therefore requires additional post-production to adapt the signal (controlling the camera according to the quality of images, detection, location, and extraction of audio sources, etc.).

– Real-time processing: applications in robotics, transport, and video-surveillance require reactivity. A number of functions (detection, monitoring and identification of people, detecting obstacles, etc.) are already executed in real time (from 10 to 20 Hz). For more complex algorithms, the transition from flow processing to real time processing is a veritable challenge because it requires re-thinking parameterizations and modeling methods to ensure both quality and density.

– Decentralization: the ability to interpret complex situations with a series of heterogeneous "intelligent" sensors and ensure that sensors cooperate and coordinate with de-centralized filtering to model imprecision and uncertainty, etc.

– Integration: this requires the design and development of communicating integrating wireless sensors, which are compact and energy independent (e.g. batteries) so that they can be easily deployed.

14.5.2. *Network level*

Networks are a basic element of embedded distributed systems and can be classified according to two sets, core networks and access networks. Access networks are networks of sensors or *ad hoc* networks, i.e. dynamic networks with mobile routers. A significant challenge is that of the management and security of open networks. How can AmIDs obtain and verify the identity and certification of other previously unknown AmIDs [GAM 11]? At the root of this question is the problem of ensuring trust in a constantly changing environment. Current solutions [STR 11] hypothesize on the prior knowledge of identity/accreditation providers, which cannot be guaranteed in an ambient context. Communication management must therefore be seen from both an individual and collective perspectives:

– Identifying and characterizing local surveillance and analysis mechanisms in a communication context continuously. An AmID, in particular, must be able to deduce and characterize its potential dependencies with regard to other AmIDs. For example, two AmIDs may communicate relying on one or several other AmIDs, which relay their communication. Equally, such an AmID relay must be aware of its role.

– Providing some of the information that will be used in the inference of AmID behavior and its interactions, therefore favoring adaptation to user context.

– Envisaging control and communication mechanisms that adapt to interaction priorities with a view to increasing the quality and continuity of users' preferred services and potentially adapting a cooperative attitude with other AmIDs to maintain a communication context as well as manage some aspects specific to security:

- Before being able to communicate, each AmID must prove its identity to others. It is therefore necessary to provide relevant authentication mechanisms in a context where only the interaction between agents can provide a solution.

- Communication security mechanisms also require research. Indeed, each AmID has different abilities in terms of processing power (like the processor, memory), autonomy (power, battery), supported security protocols, etc. It is therefore important to study whether standard security protocols (IPsec, SSL, etc.) can be used in this kind of environment or if a specific approach should be developed.

14.5.3. *Middleware level*

To design ambient applications, the *middleware* layer is central. *Middleware* represents the level of abstraction between applications and mechanisms (operating systems, networks) charged with executing them. The main challenges are modularity and extendability, on the one hand, because applications are not completely specified and evolutive, and on the other, accounting for software and material heterogeneity [GRA 11]. Heterogeneity is notably considered in the FP7 ICT FET CONNECT project [GRA 11].

14.5.3.1. *Automatic code generation*

Software platforms allow increasingly heterogeneous applications to be executed (virtual machines, Java machines, mobile OS, etc.) and evolve highly rapidly. Software developers want to move away from this evolution to allow the optimal and rapid adaptation of their applications. In this context, model-driven engineering must be further improved at the level of abstraction required when developing the system. This technological challenge therefore entails using languages, models, and approaches based on models allowing the automatic generation of specific "on the fly" codes, i.e. auto-adaptive codes.

14.5.3.2. *Engineering models*

Model-driven engineering (MDE) bases system design on the notion of models. As a result, it defines modeling languages (DSML, *Domain-Specific Modeling Language*)

[FUE 06] specific to the problem while being sufficiently flexible to integrate the necessary tools into the development process to account for global problems (system approach).

With regard to identifying specific modeling languages, the difficulty lies in the capacity to define AmId models and architectures adapted to ambient intelligence, which can be interoperated with heterogeneous AmIDs. Current models essentially account for task aspects while neglecting adaptation-related aspects. The highly dynamic and heterogeneous nature of these applications raises modeling problems that MDE models can effectively solve. The second point relates to the systematic approach for designing ambient intelligence applications [VER 11]. In this kind of application, a model must be produced to account for system heterogeneity, i.e. both the material environment (sensor, effectors, machines, etc.), the virtual environment (network, operating system, communication, etc.) as well as entities within the application (AmID). The contribution of MDE integrates methods and tools related to each point of view (human-machine interaction (HMI), ergonomy, architecture, etc.) or task (specification, design, simulation, prototyping, etc.) within a complete and homogeneous development.

14.5.4. *User service level*

In the context of mobility and volatility in ambient systems, the availability and quality of services varies. Due to the simultaneous and co-localized presence of component services, new (emerging) services must be constructed automatically and dynamically via composition and adaptation. The ambient system must therefore adapt and control the component to disturb its users as little as possible.

14.5.4.1. *Service auto-adaptation and autocomposition*

There is a certain complexity in the definition, experimentation, and evaluation of software technologies in the auto-adaptation and auto-composition of ambient services. Within this context, the following scientific challenges can be highlighted:

– When should a service be adapted? Identification, characterization, and strong evolution in services' context of use. The context use by a service may be acquired directly by this service (direct context) or constructed dynamically by a combination of contexts directly provided by a series of other services. The latter case introduces a challenge of context interoperability as well as respect for data confidentiality. Interoperability must allow each service the possibility of choosing how it wants to describe and model its own *direct* context. The challenge of using these contexts often raises the issue of modeling and the dynamics of the elements composing this context (modeling other services, users, resources, etc.).

– What should be adapted? Identifying the elements in a service that could be adapted in a given context. This adaptation can lead to a delegation to one or several other services in all or part of the initial service so that it corresponds as best as possible to all constraints. It is also necessary to refine automatic services and dynamically and automatically compose these services within ambient systems by allowing services to organize themselves to produce an "emerging" service.

– How should it be adapted? Identifying the mechanisms used to compose, delegate, and identify services allows us to respond to a specific need. As a result, an orchestration problem arises due to the power that some services can have in relation to others. Services must improve over time and therefore adapt by considering previous experience. This can raise problems directly related to service heterogeneity and their volatility and therefore requires deep reflection on what can be exchanged as characteristics on different accessible services and is part of the challenge related to the exchange of data between services.

For the design of services for users, we will focus on the following fields of research in computing: *context-aware computing*, multi-agent systems, and human–system interaction.

14.5.4.2. *Context-aware computing*

The notion of context is not new, but the uncertainty that is prevalent in ambient intelligence brings new challenges. Context is generally defined as any information that could be used to characterize entities at a given moment (people, places, objects) that have or could have an effect on the interaction between user and system, which are themselves part of context. Data can concern the place of interaction, the identity, and the state of groups/individuals and objects (whether computing or not) in the environment. To sumarize, context can be defined as the range of information that may impact on the execution of a task by an entity. A number of contextual models rely on centralized conceptualization such as Context Toolkit [DEY 01, CHE 04] or ontology management [CHE 04, EUZ 08]. Other distributed approaches cannot adapt to context during execution such as the COSMOS system [ROU 08]. The previously proposed models are not completely adapted to use within an ambient system because a contextual model for ambient systems needs to respect distribution criteria, openness, and accounting for the dynamics autonomously. Context management requires overcoming challenges regarding the following points, firstly the design of acquisition capacities for contextual information and modeling this information, i.e. their potential transformation, representation, update and, lastly, their use.

14.5.4.3. *Multi-agent systems*

Multi-agent systems [FER 99] are systems consisting of autonomous agents that interact in a shared environment to solve a shared task. The agents in these systems can be assiciated to the AmIDs [GEO 07]. The different challenges in multi-agent systems in designing an ambient application are as follows:

– Interoperability between the autonomous and heterogeneous agents (in terms of type, aim, and capability) in an ambient intelligence system. The whole range of electronic devices possible as well as humans within the system introduce a vast amount of diversity (objectives, limitations).

– The ability to account for a lack of knowledge about the design of the system's global aim. This lack of knowledge is due to the multi-objective, fluid, and therefore widely and incompletely specifiable nature of this aim.

– The fact that the system is located in a real and mixed environment (human and machines), that it is physically distributed and has specific limitations (response time, connectivity) and that it must be able to withstand being subject to failure and malfunction.

– Openness in the system with the introduction and disappearance of agents during operation as well as the adaptation of agents to their environment.

– The dynamics of the environment and the need to treat situations that were unpredicted by the designer, creating continually adaptive systems.

– The automatic deployment of such systems and the ability to scaleup.

The main aims of the projects in the FP7 Proactive Initiative: *Self-Awareness in Autonomic Systems* (AWARENESS) consists of creating systems capable of optimizing resources and their performance by adapting to context and internal changes [2].

14.5.4.4. *Human-machine interaction*

In ambient systems, HMI is not only limited to studying the basic keyboard–mouse–screen interaction. HMI currently focuses on three major aspects:

– multi-modality for effective and relevent communication between a system and user by using several interaction modalities;

– adaptation to account for changes in the interaction environment (task, platform, user) while guaranteeing usability of the interactive system. The adaptation of the interface to the user is an as-yet unresolved problem [CAL 03, KAM 11];

– mixed systems that facilitate human activities overlapping physical objects in daily life and computing capabilities.

Since technologies in interactions are no longer limited to a simple technical device, their description is complex. They should, in particular, integrate a description of the role of the user's physical environment, potential side effects, and specific

2 cordis.europa.eu/fp7/ict/fet-proactive/aware_en.html

constraints. The definition of a context-specific interaction model is therefore required to provide a concise and usable description of these interaction agents.

Studying the characterization of the juxtaposition of different interaction techniques throughout an activity constitute a second challenge to guaranteeing coherence throughout interaction techniques. It is therefore necessary to rely on an interaction model and complementary considerations in terms of HMI.

Finally, the last challenge identified lies in the implementation of mechanisms that can deduce a software structure from this model, which allows this technique to be implemented as well as possible anchor points with the agents composing the application's functional node.

As Joëlle Coutaz has said: "the new challenge is no longer simply providing finalized products for tasks and given activities, but to provide tools so that the user becomes the designer, 'creator of its own objects' and, by extension, the inventor of her/his own ambient system" [COU 08].

14.6. Conclusion

This chapter has traced the experimentation of initial forays into scenario-based ambient systems in the form of the ambient campus. The first stage of this process entailed identifying the software and/or material building blocks involved in the system. The briefness of the workshop (a few hours only) did not provide highly detailed design blocks. However, it did raise a number of questions and highlighted a number of scientific challenges. The main issues centered around the level of centralization or decentralization required to create an ambient system and, on the other hand, the notion of context.

The diversity of scientific fields involved in ambient systems has been clearly identified and are found at different levels in Amilab, as shown in section 14.2. The major scientific challenges raised such as acceptability, respect for private life, reliability, openness, and interoperability, reflect the complexity of designing these systems. In addition, these challenges characterize future ambient systems.

Overcoming these challenges requires integrated inter-disciplinary research ranging from HSS, ICT, and the areas of application concerned. Design must be multidisciplinary to construct not only reliable and effective ambient systems that are accepted and therefore used by people, but it is also important to implement and formalize a methodology that could take the form of a collaborative and technological platform between all the disciplines concerned. Its role is to ensure coordination of the greatest possible number of those involved in the project, supported by a design and decision-making system to ensure that:

– Needs are well identified.

– The implementation of technical processes is carried out well, is economical, and conforms to techno-scientific, legal, and ethical constraints.

– Decisions are taken on the basis of sufficient and objective knowledge.

– Evaluation is carried out on the basis of recordings and predefined guidelines.

The collaborative aspect is ensured by experts from different disciplines and systems' end users. This platform must breath life into inter-disciplinary research using a range of methods such as different technologies while ensuring that the issue is thoroughly examined and that the solutions are studied from as many perspectives as possible. The technological aspect corresponds to a series of software and/or material building blocks available to designers. It is important to reduce the complexity and length of the design process so that designers of ambient applications have access to general building blocks at all levels of Amilab.

The design of systems and software is not a new activity but the creation of ambient systems introduces other constraints. For example, design is, by its very nature, multi-disciplinary and is not based on the whole system since we do not know its final use, which therefore introduces new issues during its design.

14.7. Acknowledgments

We would like to thank GDR I3 and ASR from the CNRS for organizing the ETIA (Ecole Thématique Intelligence Ambiante) summer school as well as Paul Sabatier University for their support with the AmIE (*Ambiant Intelligent Entities*) project and the IRIT laboratory for their support with the strategic "ambient socio technical systems" aspect. We would also like to particularly thank Rémi Bastide, Marie-Françoise Canut, Emmanuel Dubois, Pierre Glize, Christine Sénac, and Philippe Truillet, and researchers at IRIT, for their participation in examining the scenario used in this chapter.

14.8. Bibliography

[BER 11] BERGÉ L. P., BRIAND C., DE BONNEVAL A., LEFEBVRE O., TAÏX M., TRUILLET P., "SysCARE : Système de Communication et d'Assistance Robotisée (poster)", *Journées francophones Mobilité et Ubiquité (UBIMOB)*, Toulouse, France, 6–8 June 2011.

[BIS 03] BISWAS P., CHATTERJEE R., "Location aware application development framework", US Patent 6,594,666, 15 July 2003.

[CAL 03] CALVARY G., COUTAZ J., THEVENIN D., LIMBOURG Q., BOUILLON L., VANDERDONCKT J., "A Unifying reference framework for multi-target user interfaces", *Interacting with Computers, Special Issue on Computer-Aided Design of User Interface*, vol. 15, no. 3, pp. 289–308, 2003.

[CHE 04] CHEN H., FININ T., JOSHI A., "A context-broker for building smart meeting rooms", *Proceedings of the Knowledge Representation and Ontology for Autonomous Systems Symposium*, AAAI Press, 2004.

[COU 08] COUTAZ J., CROWLEY J., Plan "Intelligence Ambiante": Défis et Opportunités, Document de réflexion conjoint du comité d'experts "Informatique Ambiante" du département ST2I du CNRS et du Groupe de Travail "Intelligence Ambiante" du groupe de concertation sectoriel (GCS3) du ministère de l'Enseignement supérieur et de la Recherche, DGRI A3., 2008.

[DEY 01] DEY A., ABOWD G., SALBER D., "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications", *Human-Computer Interaction*, vol. 16, no. 2, pp. 97–166, 2001.

[DIM 05] DI MARZO SERUGENDO G., GLEIZES M., KARAGEORGOS A., "Self-organization in multi-agent systems", *The Knowledge Engineering Review*, vol. 20, no. 2, pp. 165–189, 2005.

[EUZ 08] EUZENAT J., PIERSON J., RAMPARANY F., "Dynamic context management for pervasive applications", *The Knowledge Engineering Review*, vol. 23, pp. 21–49, 2008.

[FER 99] FERBER J., *Multi-Agent Systems – An Introduction to Distributed Artifical Intelligence*, Addison-Wesley, Boston, 1999.

[FUE 06] FUENTES L., JIMÉNEZ D., "Combining components, aspects, domain specific languages and product lines for ambient intelligent application development", *International Conference on Pervasive Computing*, Dublin, Irlande, 2006.

[GAM 11] GAMBS S., KILLIJIAN M. O., DEL PRADO CORTEZ M. N., "GEPETO*: towards a platform for evaluating privacy aspects of mobility data", *Journées francophones Mobilité et Ubiquité (UBIMOB)*, Toulouse, France, 6–8 June 2011.

[GEO 07] GEORGÉ J., CAMPS V., GLEIZES M., GLIZE P., "Ambient intelligence as a neverEnding self-organizing process: analysis and experiments", *Artificial and Ambient Intelligence convention (Artificial Societies for Ambient Intelligence) (AISB-ASAMi 2007)*, Newcastle, Royaume-Uni, 2007.

[GRA 11] GRACE P., GEORGANTAS N., BENNACEUR A., BLAIR G., CHAUVEL F., ISSARNY V., PAOLUCCI M., SAADI R., SOUVILLE B., SYKES D., "The CONNECT architecture", *11th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Connectors for Eternal Networked Software Systems*, vol. 20, no. 2, Bertinoro, Italie, pp. 27–52, 2011.

[KAM 11] KAMMOUN S., ORIOLA B., TRUILLET P., JOUFFRAIS C., "Conception et Evaluation d'un Système de Dialogue pour un Dispositif de Suppléance pour Déficients Visuels", *Journées francophones Mobilité et Ubiquité (UBIMOB)*, Toulouse, France, 6–8 June 2011.

[KEP 03] KEPHART J., CHESS D., "The vision of autonomic computing", *Computer*, vol. 36, no. 1, pp. 41–50, IEEE, 2003.

[ROU 08] ROUVOY R., CONAN D., SEINTURIE L., “Software architecture patterns for a context-processing middleware framework.”, *IEEE Distributed Systems Online*, vol. 9, no. 6, 2008.

[STR 11] STREITZ N., “Smart cities, ambient intelligence and universal access”, *Universal Access in Human-Computer Interaction. Context Diversity*, vol. 6767 of *Lecture Notes in Computer Science*, Springer Berlin, Heidelberg, pp. 425–432, 2011.

[VER 11] VERGONI C., TIGLI J. Y., LAVIROTTE S., REY G., RIVEILL M., “Construction bottom-up d’applications ambiantes en environnements partiellement connus a priori”, *Journées francophones Mobilité et Ubiquité (UBIMOB)*, Toulouse, France, 6–8 June 2011.

[WEI 91] WEISER M., “The computer for the twenty-first century”, *Scientific American*, AQ3
vol. 265, 1991.

[ZHA 10] ZHANG P., “Complex wavelet feature extraction for video-based face recognition”,
Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon), March 2010. AQ4

Author Queries

[AQ1] Please provide Figures in gray scale since this book is going to print in single color (i.e., gray scale).
[AQ2] Please confirm if the usage of X is appropriate here.
[AQ3] Please provide page range for reference [WEI 91].
[AQ4] Please provide location for reference [ZHA 10].