Noname manuscript No.
(will be inserted by the editor)

Conflict Detection and Resolution in Home and Building Automation Systems

A literature review

Sílvia Resendes · Paulo Carreira · André C. Santos

Received: date / Accepted: date

Abstract The evolution and increasing commoditization of Home and Building Automation Systems (HBAS) is contributing to their widespread adoption. However, an effort must still be made to render them usable, intelligent, highly adaptive and able to fulfill users' needs. When distinct users interact with such a system, their intentions are likely to be different, often resulting in conflicting situations, which the system ought to recognize and, if possible, resolve automatically. However, conflict detection and resolution in HBAS are not yet fully understood. This work aims at investigating conflict in Ambient Intelligence systems, namely those supported by HBAS. Our main contribution is a systematization and review of existing literature concerning conflict detection and resolution in these systems.

Keywords Ambient Intelligence · Home and Building Automation Systems · Conflict Detection · Conflict Resolution

S. Resendes
IST - Technical University of Lisbon, Avenida Prof. Dr. Cavaco Silva, Taguspark 2744-016 Porto Salvo, Portugal
Tel.: +351-926067493
E-mail: silvia.resendes@tagus.ist.utl.pt

P. Carreira
INESC-ID and Technical University of Lisbon, Avenida Prof. Dr. Cavaco Silva, Taguspark 2744-016 Porto Salvo, Portugal
Tel.: +351-966716670
E-mail: paulo.carreira@ist.utl.pt

A. C. Santos
INESC-ID and Technical University of Lisbon, Avenida Prof. Dr. Cavaco Silva, Taguspark 2744-016 Porto Salvo, Portugal
Tel.: +351-914007809
E-mail: acoelhosantos@ist.utl.pt

1 Introduction

We assist to a commoditization of intelligent control hardware (Smartthings 2012; LIFX 2012) which, along with the introduction of a large variety of intelligent consumer electronics equipment (Merz et al. 2009), has been fueling a fast adoption of home and building automation systems (HBAS). It is safe to expect that these systems become ever more popular and offer increasingly sophisticated behaviour, even anticipating the users' intentions and taking proactive action to assist them in their activities. However, the vision of these truly intelligent home and building automation systems has been largely unmet, since many systems hardly go beyond scheduled sequences of actions and very simple sensor-actuator rules, e.g., (Bucceri 2006).

The technological advancements in consumer electronics have also caused people to adapt and get used to a variety of equipment, being smartphones the most notable example, used in an almost unconscious way, given the habitude and naturality with which this device is rooted in our daily lives. In a landmark text, Weiser stated that *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it"* (Weiser 1991). Given their constant presence and participation on user's daily routines, it is of utmost importance that HBAS manage to be, as much as possible, non-intrusive and non-disruptive, i.e., these systems should not interfere or interpose in the normal course of their users' activities.

The main working blocks of an intelligent environment, controlled by a HBAS, are depicted in Figure 1. Occupants interact with different objects and devices, within a space equipped with different sensors and actuators. An automatic controller system implements some

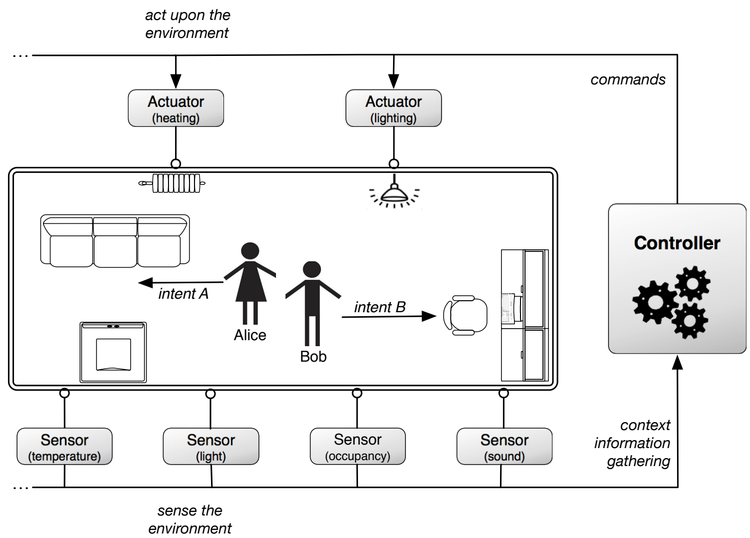

Fig. 1 Overview of an intelligent environment. The diagram depicts a smart-home with an HBAS consisting of sensors, actuators and controller modules. Within the home, users Alice and Bob have intents that enact distinct (and thus conflicting) actuations from the HBAS.

sort of intelligence, and controls these devices, by interpreting the environment information gathered by sensors, determining what action to take, and commanding the actuators towards taking appropriate action upon the surrounding environment.

In order to be non-intrusive and non-disruptive, intelligent environments take action based on the state of devices, objects, places and people. This abstract notion of state is commonly referred to as *context* (Dey 2001). Context detection refers to the collection and analysis of data from sensors placed on the user's environment, and the system's inference of the current scenario. This inference is based on pattern detection of sensor readings from previous user actions and feedback, and even on mathematical predictions. Upon detecting a change in context, this system reacts based on a set of previously defined policies and rules. The system's behaviour is based on its Context-Awareness and results in an automated response to context changes.

In an intelligent environment, however, multiple contexts may coexist, entailing distinct reactions, which can *conflict* with each other. For example, a context may imply a lamp to be turned on, while another may imply it to be turned off. Ideally, these conflicting situations should be automatically resolved by the HBAS.

With that aim, we will herein motivate the automatic conflict detection and resolution problem, and the potential of Ambient Intelligence (AmI) developments towards more effective and user-centred HBAS.

1.1 Motivation

In their everyday life, both at home and at work, people will most likely share space and other resources, and thus the systems that control them. When considering a shared space, the actions taken by one user may affect others. At a given instant, each user's intentions can be different. As a result, each user will want a different scenario setup. However, these different contexts require corresponding scenarios that frequently cannot be activated simultaneously on the same space. In this case, we can say that we have a *conflict situation*.

As an example, consider a meeting room in an intelligent office building, where occupant A is initially sitting alone, and has adjusted the air temperature setpoint to 20°C. Moreover, it is known that this occupant prefers temperatures between 19°C and 22°C. Suppose that occupant A wants to have a meeting with a colleague. When the second occupant arrives at the space, having a different preference, how should the system adjust? In view of the described situation, Figure 2 shows three possible scenarios, each corresponding to the entrance of an occupant in the room, with different preferences, eliciting a distinct system behaviour.

- *Occupant B*, with a tolerance of 19.5°C to 22.5°C, enters the room. There is no conflict, therefore no action is needed from the system;
- *Occupant C*, with a tolerance of 21°C to 23.5°C, enters the room. A conflict exists, since the tem-

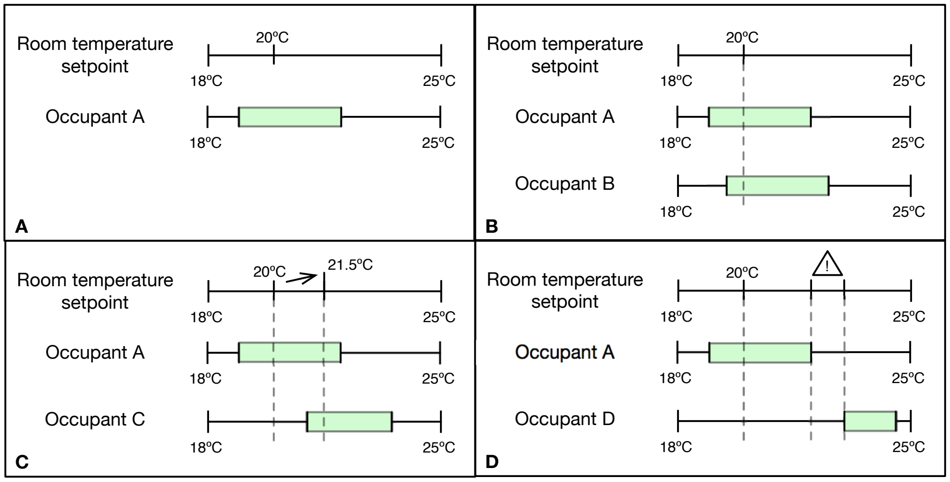

Fig. 2 Example scenarios, depicting the room temperature environment variable setting and the possible system responses to the entrance of a second occupant with a different preference for the temperature variable. The darkened bars represent user tolerances. In Scenario A, occupant A is alone in the room, with the temperature set to 20°C. Scenarios B to D represent the entrance, respectively, of occupants B to D, along with the possible system responses.

perature is set to a value below user C's tolerance. However, there is a possible system actuation that resolves it, since changing the setpoint to 21.5°C accommodates the preferences of both occupants;

- *Occupant D*, with a tolerance of 23°C to 24.5°C, enters the room. A conflict exists, but in this case the system has no way of resolving it, since there is no temperature range that would accommodate the preferences of both occupants. Possible system actions would be: *(i)* maintaining its state; *(ii)* adjusting to an intermediate value; *(iii)* informing occupants of its inability to solve the conflict.

The situation depicted in scenario D shows a very common case of conflict, i.e., a preference conflict involving concurrency over resources, therefore any system aiming at conflict resolution must cope with the notions of *ownership* and *priority* among users, in order to appropriately adjust its behaviour. However, conflict can be an even more multi-dimensional problem. For example, consider the example scenario given in Figure 1, that shows how conflict arises between two users, and at the same time between energy saving and comfort, in a HBAS: suppose that at the end of the day, Alice and Bob arrive home and enter their empty living room. The system is programmed to turn on the ceiling lamp, when placed sensors detect someone's presence, dimming it either at full capacity or at half capacity. This means that there are two illumination scenarios that are to be selected based on the inferred context. The first context is "Alice wants to watch tv"; the second context is "Bob wants to do his homework". In both contexts, half capacity is more energy-saving. However, while in the first context half capacity is also more comfortable, since it avoids glare; in the second context, full brightness is more comfortable for reading. This means that in the first case, energy saving and comfort are side to side, but in the second case they are conflicting. Moreover, these users' intended contexts are conflicting, since the two illumination scenarios cannot happen simultaneously, hence the system cannot fully fulfill the needs of both users with just one illumination configuration. In this scenario and faced with these possibilities, how should the system dim the lamp, and thus resolve the newly arisen conflict?

Besides a preference and resource concurrency conflict, this example shows conflict on comfort versus energy saving. As expectable, the interaction of applications in an intelligent environment also creates room for more conflict. A plethora of conflict types emerge, and systems must have the ability to resolve them on behalf of users, or otherwise acknowledge their own limitations and inform users that the conflict should be explicitly resolved. The fact that each person is essentially unique and ever changing, subject to the influence of a large set of factors, results in different goals and mental models, thus less predictable and, in many cases, conflicting.

In multi-user scenarios, context inference becomes even harder. It raises challenges such as distinguishing the preference of each user, as well as resolving the conflicts among different user preferences (Hasan et al. 2006). The different dimensions of conflict will be detailed later. Moreover, the difficulty of tracking the intentions and thus the context of each user greatly limits the ability of the system to respond appropriately. Several systems, such as Gaia (Ranganathan and Campbell 2003) and CARISMA (Capra et al. 2003) consider single and multi-user conflicts regarding a single application and leave out multi-user conflicts in a multi-applicational environment. However, as we can see, multi-user conflict resolution is more complex than single user's. The more users, the harder are context inference and conflict resolution.

Existing literature on automatic conflict detection and resolution is considerably scarce and disorganized. The present work aims at investigating the nature of conflicts in AmI systems, and at developing a conflict classification, grounded on relevant literature on the subject, with the intent of understanding their sources and traits, to determine which are amenable to automatic resolution. A further contribution of this paper will be a survey and systematization of existing work on this topic and its most directly related areas.

1.2 Organization

This paper is structured as follows. Section 2 reviews home and building automation systems. In Section 3, we introduce relevant concepts and review the aspects of AmI that we believe can most significantly contribute to the development of HBAS, namely Context-Awareness and Multi-Modal Interaction. Section 4 discusses conflicts. Based on existing literature, a taxonomy of conflicts is suggested, along with strategies to deal with and resolve conflict. This is done both at the interpersonal context and in AmI environments. Finally, Section 5 presents the attained conclusions.

2 Home and Building Automation Systems

The basis of any HBAS is a network of sensors and actuators, connected to an intelligent automation control system (Merz et al. 2009). Sensors periodically collect information from the surrounding environment to feed the control system, and may be of various types and purposes: sound, motion, temperature, humidity, luminosity, air flow, among others. Conceptually, the information read by sensors is continuously aggregated and analyzed, to determine what activities are taking place on a given place at a given point in time, i.e., to determine the contexts that apply to each entity. Using that information, an intelligent control system orchestrates all distributed devices, sending them commands, in order for the space to comply with a set of pre-defined and pre-programmed policies, thus carrying out users' preferences and demands. Actuators, as the name implies, receive instructions from the control system and act upon devices, thus producing an effect on the environment. Automated doors and windows, motorized blinds and shades, lamps, Heating-Ventilation and Air Conditioning (HVAC) systems, multimedia and consumer electronics equipment are examples of generally controllable devices that can be used as actuators. It is important to note that an actuator is not always binary (on/off) nor bounded to the room division where it is installed. On the contrary, many devices influence adjacent spaces and the users therein. Therefore, in order to reason about conflict, it is important to specify the possible device states and their effect on the surrounding environment. For example, audio equipment can be on or off, but different volume levels have different effect radii, and some levels may influence the comfort of users within their radii.

2.1 Applications of HBAS

Having been initially meant for commercial and corporate ends, BAS have resource management and optimization as a core goal, not only of energy, but also of other resource types. An interesting application of a BAS is to predict energy-related needs (for lighting and heating) based on occupancy patterns, thus allowing the creation and scheduling of useful scenarios with adequate configurations, adjusted to the energetic needs of each phase of a work day. A more concrete approach, that can be considered as a subset of a BAS, is room automation. When a single room has a particularly large number of devices, it is common to centralize their automation and make their control exclusive to that room. Good examples are corporate boardrooms, where projecting equipment and individual displays are only useful for that particular room (including for privacy and security reasons), or research labs, with specialized and often expensive or sensitive equipment.

Building Automation (BA) also applies to home environments, creating what we now know as Home Automation, Domotics or Smart-Homes. However, while in commercial or corporate settings the main goals are productivity and profit, at home other priorities stand out, such as safety, leisure and comfort. Also, home occupants have distinct goals, that change frequently and vary in importance, both among occupants and through time. Moreover, at home people need the most to feel in control. In contrast, in a work environment, occupants are frequently subject to hierarchy. Any discomfort caused by the BAS, perhaps due to a misconfiguration or incorrect context inference, may be attributed to superior decisions or overrides, thus resignedly tolerated. However, at home this does not happen, therefore we may say that tolerance towards erroneous interference from a HAS is inferior to that of a BAS.

Despite of the architectural similarities between HAS and BAS, their goal, priority and requirement divergences make imperative the use of different design principles. Consequently, conflict situations that arise are also likely to be solved in different ways.

2.2 Towards smart HBAS

In the field of Home and Building Automation, many topics have earned the attention of research groups.

There are two main visions regarding further development of these systems: *(i)* evolution via *user approaching*: many advocate intelligent systems should evolve through the development of human- computer interaction, and the increase of user ability to tailor the system; *(ii)* evolution via *user understanding*: others argue evolution should be taken by increasing system's intelligence, through better profiling and Machine Learning, and more effective context detection and inference, thus reducing interaction to a minimum.

Regarding user approaching, several authors propose more adequate, adaptable and innovative user interfaces and interaction modes for this kind of systems (Ghanam et al. 2009; Nichols et al. 2002; Findlater and Gajos 2009; Satoh 2007; Yamazaki 2007; Ballagas et al. 2003); while others focus on the development of methods to allow the personalization of these systems' applications (Kawsar and Nakajima 2007; Coutaz et al. 2010; García-Herranz et al. 2007).

Regarding user understanding, an obvious path is improving sensor data attainment, analysis and storage (Abadi et al. 2005; Ledlie et al. 2005; Szewcyzk et al. 2009). This sensor data is the main basis for user profiling (Aipperspach et al. 2006) and profile processing and evolution throughout time (Schaefer et al. 2006).

For instance, Abielmona et al. (Abielmona et al. 2010) proposed a MAS consisting of fixed and mobile intelligent sensor agents that explore a natural and continuously changing environment, with the goal of minimizing its mapping uncertainty, i.e., entropy, which rises from the fact that agents can have complementary or competitive (conflicting) readings compared to one another, varying through time. The authors compared different agent deployment, sensor data collection and fusion strategies, ultimately demonstrating the efficiency of a technique to produce a multi-resolution and multi-dimensional view of the environment: the tree-in-motion mapping (TIMM).

Some works, however, take on a more personal approach by using questionnaires to model user behaviour and to find behaviour patterns (Ha et al. 2006).

Moreover, given the increasing mobility of both users and devices, the addition of sensors based on wireless technologies, such as those available on mobile devices, show great potential as another promising way of improving context inference abilities (Welbourne et al. 2005; Santos et al. 2009).

Korpipaa et al. (Korpipaa et al. 2003), regardless of intentionality, demonstrated how the two paths can be successfully merged, by presenting a mobile terminal software framework that provides systematic methods for acquiring and processing useful context information from a user's surroundings and giving it to applications.

As we may conclude from what has been seen so far, these systems have much to win with the development of Context-Awareness and interaction technologies and capabilities. Many challenges still lie ahead, regarding both the software infrastructure (Kindberg and Fox 2002) and more social domains (Edwards and Grinter 2001). However, there are very interesting works providing design guidelines (Davidoff et al. 2006), architectures (Bottaro and Gerodolle 2008; Rashidi and Cook 2009; Granzer et al. 2006) and implementation prototypes (Spinellis 2003) for smart spaces. Fortunately, over time, on one thing all come to agree: the crucial importance of user-centrality in these systems.

2.3 Towards energy-efficient HBAS

Making buildings greener has also been a subject of crescent interest. The use of occupancy sensors to increase energy saving (Garg and Bansal 2000; Agarwal et al. 2010) and the development of more efficient lighting and HVAC control techniques are the focus of most approaches (Cziker et al. 2007; Epstein et al. 2003; Lah et al. 2005, 2006; Miki et al. 2004; Zhou et al. 1993). Examples and estimates regarding the practical application of such techniques can be found in (Richman et al. 1995; Kovach-Hebling et al. 1997; Doulos et al. 2007; Floyd and Parker 1995; Galasiu et al. 2004; Lee et al. 1998; Littlefair et al. 2010; Newsham and Mancini 2006; Park and Hong 2006).

Furey et al. (Furey et al. 2012) propose a system to overcome RTLS (Real-Time Location System) weaknesses by applying artificial intelligence approaches to historical user location data, towards smart prediction of future locations. Their system, HABITS (acronym for History Aware Based Indoor Tracking System), was able to improve an existing RTLS system in terms of yield, accuracy, latency, cost and predictive ability, simply by processing and analysing historical user location. Such a system could be particularly useful to a HBAS, in the sense that it might contribute significantly to an effective context prediction, either in conjunction or as a replacement of movement sensors.

Reinisch et al. (Reinisch et al. 2011) propose an interesting and comprehensive multi-agent system concept for an intelligent home, addressing the reduction of energy consumption without neglecting user comfort.

None of these works, however, truly consider the notion of conflict. Conflict detection and resolution will benefit from the developments of HBAS in the multiple areas mentioned above, since these systems will then be able to become closer to their users, as well as more knowledgeable of them.

3 Ambient Intelligence

AmI refers to the paradigm of creating environments that are able to, as silently and imperceptibly as possible, detect and respond to users' needs, without requiring any explicit orders from them. This paradigm envisions an environment consisting of a pervasive network of small electronic devices with minimum processing capability, that act as either sensors or actuators, and therefore obtain information or act upon the environment, thus changing it (Chalmers 2011). The effective commissioning and orchestration of these devices results in a smart environment, characterized by systems and technologies that incorporate the following features: *(i)* embedded in the environment, in the sense that technology is ubiquitous and as indistinguishable from the surroundings as possible; *(ii)* context-aware, i.e., capable of obtaining information about the user and ongoing activities to infer the current context; *(iii)* personalizable, in the sense that users can establish a set of preferences that tailor the system to their needs; *(iv)* adaptive to user and environment changes; *(v)* predictive, capable of anticipating of users' wishes based on their past behaviour, incorporating either implicit or explicit feedback.

In other words, AmI aims at achieving an environment with *"machines that fit the human environment instead of forcing humans to enter theirs"* (Weiser 1991). To that end, it integrates knowledge from several areas, mainly: *Context-Awareness*, as introduced before; *user profiling*, i.e., the construction of user profiles using mathematical techniques that enable the discovery of patterns and correlations in large amounts of stored data, e.g., log data from which user actions, routines and preferences may be inferable; and *human-computer interaction design*, since the interaction modes influence the ubiquity notion of such systems.

Nowadays, a large variety of disciplines can be considered under the umbrella of AmI: distributed intelligence, data and information communication, software and hardware design, robotics, information fusion, computer vision, speech recognition, social sciences, ethics and law (Remagnino and Foresti 2005). Moreover, AmI is a continuously evolving area, as new visions and theories appear, towards more aware and user-centred systems (Nakashima et al. 2010). The awareness aspect of AmI has also evolved, partly due to the increasing possibilities offered by recent technological advancements. Three main phases can be considered, towards the actual level of user-centeredness: *(i)* context-aware distributed systems, able to obtain relatively accurate context data; *(ii)* the ability to infer valuable information from the attained context data; *(iii)* actually providing social value to users.

Unfortunately, despite active research, a decade ago the technological infrastructure required to support the ubiquitous world didn't yet exist. Much work had to be done, and Weiser wisely pointed out the three main technology-related challenges that still had to be surpassed: *(i)* creating the infrastructure for pervasive wireless networking, with enough bandwidth to support communication between hundreds of wireless computers; *(ii)* enhancing networking protocols, otherwise unable to handle the desired infrastructure mobility; *(iii)* development of windows systems, aiming at window mobility over a network (Weiser 1993).

Nowadays, technology is advanced enough to allow such high expectations. Example of such is a work by Tapia et al. (Tapia et al. 2010a), who argue in favour of implementing MAS to build AmI-based systems, due to the autonomy, reasoning, reactivity, pro-activity and social abilities of agents. They suggest wireless sensor networks (WSNs), based on technologies such as RFID, ZigBee, Bluetooth, Wi-Fi and GPRS, to be embedded in agents, allowing them to obtain and process context information. Their AmI-based architecture, FUSION@, integrates a Service-Oriented Architecture (SOA) approach with intelligent agents, while allowing the interconnection of such a heterogeneous WSN, further embedded in users' applications and executable in a large variety of platforms.

In the remainder of this section, relevant aspects regarding Context-Awareness (Section 3.1) and human-computer interaction (Section 3.2) will be detailed.

3.1 Context-Awareness

In a common system, users can directly interact through what we call *explicit user orders*, for instance, turning on a radio. However, in a smart environment, such as a HBAS, there are other ways of controlling the system. In fact, the notion of smart or intelligent environment has to do with the system's ability to autonomously acquire and apply knowledge about an environment, and to adapt to its inhabitants (Cook and Das 2004), i.e., the system can infer *implicit user orders* and adjust accordingly without requiring user intervention. This environment knowledge is what we call *context information*, which Dey defines as *"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 application themselves"* (Dey 2001). In this paper, application stands for control application, which is any sensor or actuator that can provide information or change the environment on behalf of the user.

A major difficulty is how implicit user orders and intentions are inferred from context changes, and how the system reacts to explicit user orders. Inferring user orders and intentions implicitly is difficult since user preferences change over time or based on situation (Hasan et al. 2006). On any given moment, these preferences are influenced by aspects such as mood, motivations, goals and needs. This kind of information is highly subjective and inconstant, and mostly unattainable by common non-intrusive sensors. Moreover, each actuator on a pervasive system may affect its surrounding physical environment, thus influencing the context. This poses several challenges when it comes to inferring context, e.g., problems regarding context validity (Zimmer 2004) or quality-of-context (QoC) (Neisse et al. 2008), hence to determining the appropriate action.

Therefore, there are two main reasons for inappropriate context based system behaviour: either the system infers and acts upon a wrongly assumed context; or the system does not detect a context and thus does not take any action.

Even considering the fact that these systems are intended to automatically infer context and respond to implicit rather than explicit commands, it is essential that any system behaviour be overridable. This is especially true in a home control system.

Complementary to the notion of Context-Awareness is Situation-Awareness (SAW), described by Endsley as the *"perception of elements in the environment within a volume of time and space, the comprehension of their meanings and the projection of their status in the near future."* (Endsley 1988), i.e., the awareness of what is occurring and the ability to predict possible developments and consequences of such situation.

Buchmayr et al. (Buchmayr and Kurschl 2011) surveyed and compared situation-aware AmI in terms of SAW, regarding criteria such as technology, data processing, decision making approaches and prediction mechanisms. The attained results clearly show that SAW is an issue in all the evaluated projects, despite of the current maturity and reliability of sensor infrastructures, because AmI applications still face a huge gap regarding high level data fusion, mostly caused by interoperability problems between different devices and data systems.

In this type of system (i.e., a distributed system), the possibility of conflict is higher than in other systems, mainly due to a number of contexts and services used, and the mobility of entities (Syukur et al. 2005). Thus, the notion of context is wider than that of the physical space zone the user is in: it has to do with the settings that entail the user's level of *comfort*, such as the noise, illumination, temperature and air flow conditions, adequate to the user's intentions in a given moment. For example, if a user wants to read, a comfort context would entail low noise, medium to high illumination, mild temperature, and low air flow.

Tuttlies defines conflict with respect to a user or application as: *"...a context change that leads to a state of the environment which is considered inadmissible by the application or user"* (Tuttlies et al. 2007). However, the definition of conflict varies from context-aware application to application (Park et al. 2005), as well as with the system's layer in which it arises. Context-aware systems' architecture is a complex topic, object of several studies. A thorough survey and analysis of existing systems gave Baldauf (Baldauf et al. 2007) the foundations to depict a layered conceptual framework for such systems, shown in Figure 3, in which the base layers are more hardware-oriented and the top ones correspond to the actual system logic and intelligent services. It is important to note that this is merely a conceptual and functional separation, since systems usually aggregate some layers in their implementation.

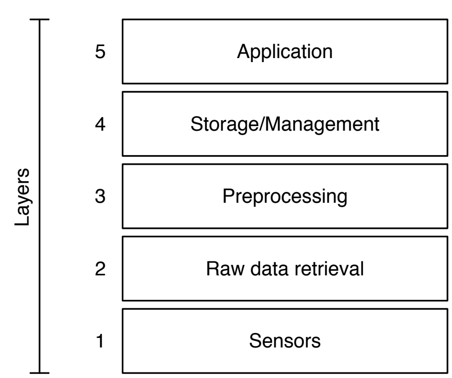

Fig. 3 Baldauf's conceptual framework for context-aware systems, displaying a layered organization of information processing components underlying context aware systems. Adapted from Baldauf et al., 2007 (Baldauf et al. 2007).

Our choice of Baldauf's framework was based on the fact that his survey on Context-Awareness was the most referenced and the only one that presented a conceptual framework in addition to comparing existing systems. As this framework is of interest to our work, we point out some of the most relevant aspects of each layer. The first layer consists of *sensors*, that may be either physical, i.e., hardware sensors that capture physical data; virtual sensors that obtain context data from other sources, e.g., using mouse movements to determine if someone is at a given place; or logical, which are a combination of physical and virtual sensors with additional information, e.g., information from databases. The *raw data retrieval* layer deals with obtaining raw data from sensors and offering a unified API for accessing sensor data. The *preprocessing* layer takes sensor raw data to infer the contextual data. At this level, distinct data sources may provide contradicting information about the same context. These context sensing conflicts arise from data quality problems, hence we will not pursue them in this paper. The *storage and management* layer enables storing and querying context data. Problems in this layer are limited to informational incoherence, i.e., short periods of time, between two information requests, when the user has outdated context information. Finally, the *application* layer is where conflict detection and resolution are realized, since this layer implements the actual system intelligence, embodied in some application that reasons about the context information and reacts to context change events. Therefore, this layer will be the focus of our work. However, as usual in layered systems, we must consider that some conflicts arisen in lower layers may propagate up, or even pass undetected and unresolved as a result of the lower layers' limitations.

3.2 Human-Computer Interaction

The usual, WIMP-based (Windows, Icons, Mouse and Point) interaction modes, popular in computer interfaces are, in many situations, inappropriate and inadequate to interact with ubiquitous systems, since they contradict the ideal of having technology vanish into the environment. To this end, Multi-Modal Interaction refers to a form of human-computer interaction that involves more than one interaction modality, and therefore is better suited for ubiquity, as multi-modality is applied and useful both in information input (human-computer) and output (computer-human). Moreover, depending on each system's goals, some forms of interaction will likely be more effective than others.

As Gross (Gross 2008) points out, there has been an evolution of foci towards interactive systems: from single-user WIMP, to cooperative systems, to single-user ubiquitous computing (ubicomp), to single user AmI, and most recently to cooperative AmI; the latter of which can be defined as environments that aim to improve users' work and private life by analysing and adapting to the current situation with a special focus on interaction among users. Therefore, AmI systems should ideally include the use of intelligent (natural) user interfaces (Alcañiz and Rey 2005).

3.2.1 Acting as Multimodal Output and Sensing as Multimodal Input

In multi-modal output, through which the computer can send information to the user, a modality refers to the human senses employed to process incoming information (Maybury and Wahlster 1998). To this end, the most commonly used modalities are vision and audition. Visual information can be received either in textual, graphical or mixed form, and through various types of displays. Auditive information ranges from simple beeps to more complex formats, e.g., speech synthesis.

In AmI systems, existing actuators take action upon the environment, therefore affecting nearby users. Nevertheless, their output is not limited to explicit visual or auditive informational content. Instead, actuators in a smart environment act on temperature, humidity or other environment parameters, which are often not immediately and consciously perceived by users. However, since the actions of these devices directly affect users' sensory receptors, they may well be considered a slight extension of the usual notion of multi-modal output.

Multi-modal input has evolved further, both at the physical level (variety and ergonomy of used devices) and in depth (possible actions). This is partly due to the need for a better adaptation to the users and their ability to more effectively and naturally interact with available systems. In fact, as Markopoulos (Markopoulos 2005) states, it is expected that people will interact continuously with computation, in an ever-increasing range of forms, situations and locations; and the set of devices and services that a user might use to access a system is likely to be numerous, diverse and expanding over time. Therefore, contrarily to traditional interaction design, ubiquitous computer-human interaction should be designed to be extended and to be combined with other, initially unknown, forms of interaction.

Users can provide information to the system either with or without directly manipulating input devices. Interaction through device manipulation can be divided according to the type of information provided: *(i)* text, through physical or virtual keyboards; *(ii)* spatial information, through pointing devices, such as mice, trackpads, light pens and others; *(iii)* audio, through microphones, MIDI keyboards or other digital musical instruments; *(iv)* imaging and video, through digital cameras, scanners and other imaging equipment. Some of these devices enable the input of more than one of the above information types. However, the richest and most interesting interaction forms are those that require no explicit device manipulation, thus making the interaction more natural. This is the case of gesture recognition, which allows users to interact with the system through a large range of predefined gestures, from simple pointing to more complex and composite ones, e.g., gestures used to play tennis on a Wii™console. Another interesting form is the application of eye gaze tracking systems to ease the input of spatial information, thus improving the interaction.

In AmI, to a large extent, explicit knowledge about the existence of the input devices by the users is not mandatory. We will consider interaction through sensors as *implicit interaction*, in the sense that it provides information input without requiring direct commands. The intended ubiquity and "invisibility" of HBAS bring new challenges to their interaction design, and can only win with exploring various forms of obtaining user information without directly requesting it.

Ideally, a conflict resolution system should aim at offering multi-modal input and output, in order to take advantage of the synergies existing between the various interaction modes, providing the user a means for interacting with the system in a more natural way, thus offering a smoother and more pleasant user experience.

3.2.2 Social Intelligence, Perception and Persuasion

The potential of currently developing AmI technologies has leveraged the investment of several companies and the interest of international research centers (Sorrentino 2009). Markopoulos et al. argue in favour of considering intelligence beyond its narrow sense of problem solving, learning, and system adaptation; to cover the ability of a system to interact socially with people and become a socially competent agent in the group interactions it supports (Markopoulos et al. 2005). This notion is aligned with that of *social intelligence*, long defined by Vernon as a person's ability to *"...get along with people in general, social technique or ease in society, knowledge of social matters, susceptibility to stimuli from other members of a group, as well as insight into the temporary moods or underlying personality traits of strangers"* (Vernon 1933). Ideally, a system supporting effective conflict resolution should intervene and communicate with users in a manner that is perceived by them as socially competent.

Along with other authors in the area, such as de Ruyter and Saini, Markopoulos also points out that this requirement of social intelligence presents challenging research problems to the human-computer interaction community. Although measuring a system's perceptiveness, success and value addition is difficult (Venkatesh et al. 2003), experiments show that it is possible to build socially intelligent home dialogue systems, which create a positive perception of technology and elicit an overall greater user interest, acceptance and satisfaction, regarding their social intelligence aspect (de Ruyter et al. 2005; Saini et al. 2005).

A good example of the use of AmI towards providing social value is the ALZ-MAS, an AmI-based multi-agent system presented by Tapia et al. (Tapia et al. 2010b), which aims at enhancing the assistance, healthcare and safety of Alzheimer patients living in geriatric residences. It uses complex reasoning and planning mechanisms to dynamically schedule the medical staff daily tasks, which can be done through an interface that also displays basic information about nurses, patients and the building. By taking advantage of the cooperation among autonomous agents and the use of context-aware and wireless technologies, ALZ-MAS obtains real-time environment information and allows its users to control and manage existing physical services, thus providing a ubiquitous, non-invasive, high-level interaction among users, system and environment.

Li (Li 2012) acknowledged the essential role of smart environments in making home healthcare possible. However, he argues that several challenges still need to be addressed towards user acceptance of these systems, mainly system reliability, data privacy and security, high device interoperability, extensibility to support future medical devices, and a design based on medical service models and best practices. Regarding user interaction, the author refers that modifying the patient's behaviour through feedback, by promoting activities and behaviours with positive health impact, is becoming a trend in home healthcare systems.

This is another interesting nuance of AmI that has been a subject of interest to researchers and can be proven an asset to future work on our topic is *persuasion*. Captology (as the area of persuasive technology is called), explores the theory, design, and analysis of computers for planned persuasive effects. Fogg layed the foundations for this area, arguing that people inferred about social presence in a computing product when subject to five primary types of social cues: physical, psychological, language, social dynamics and social roles (Fogg 2003). In social sciences, Cialdini is well-known for having introduced the six principles of persuasion: liking, reciprocity, social proof, consistency, authority and scarcity (Cialdini 2001). These principles are still the basis for other works on the area. For instance, Kaptein et al. explore the idea that for persuasive technologies to be effective, their adaptivity to individual user susceptibility is necessary (Kaptein et al. 2009). Their results show that including persuasive cues increases user compliance to requests, and also that incorporating a user profile of susceptibility to specific cues, and adopting the right persuasive strategy, could greatly enhance a persuasive system's effectiveness.

The usefulness of perception and persuasion aspects for conflict resolution stems from the fact that some conflicts cannot be resolved without explicit user intervention. Therefore, a persuasive HBAS could be useful in two ways: *(i)* in persuading users to lightly resolve the conflict (and possibly suggest the most energy-efficient resolution); *(ii)* in occasionally persuading users to try environment conditions slightly different from their defined preferences, e.g., to improve energy-saving or to minimize future conflicts.

4 Conflict

Conflict is a natural disagreement between different attitudes, beliefs, values, or needs (Wang and Ting 2011).

Smart environments, such as HBAS, with systems that perform automatic control of different types of equipment, present new services and new intentions towards them, that can cause new types of interpersonal conflict to arise. The smart environment itself may consist of applications that have their own goals and intentions (e.g., saving energy), therefore giving way for more conflicts. It is expectable that these systems end up having an intermediating role in the automatic resolution of any arising conflicts.

In this section, we will be interested in analysing conflict as a disagreement of goals and intentions towards the environment both of human agents and applications. To that end, we survey relevant literature regarding different dimensions of conflict, including their sources and solvability. Based on this analysis, we develop a taxonomy of conflict, aiming at unifying interpersonal conflict and conflict in AmI.

4.1 Classifying Conflicts

As we have previously mentioned, Tuttlies defines conflict with respect to a user or application as *"...a context change that leads to a state of the environment which is considered inadmissible by the application or user"* (Tuttlies et al. 2007). To better understand the nature of conflicts in AmI systems, a brief explanation follows, regarding the various conflict types that will be considered throughout this paper. Our taxonomy classifies conflicts according to four different dimensions: *(i)* source, *(ii)* intervenients, *(iii)* time of detection and *(iv)* solvability. A summary of this classification is presented in Table 1.

Relatively to the *source*, a conflict can happen either at: *resource level*, i.e., when multiple users concur over a given resource, e.g., a tv; *application level*, when multiple applications concur over a resource, e.g., building management applications controlling a room's lighting; *policy level*, when conflict arises due to conflicting policies in a given context, e.g., a user enters a library, and although his smartphone allows him to listen to music through his speakers, the space has a silence policy; or *profile (or role) level*, when there are conflicting user profiles and preferences in the same context, e.g., one user prefers the lights at full capacity to read, and another user prefers them at half capacity to watch tv.

Conflicts can also be classified regarding their *intervenients*. A conflict situation can be either: *single-user*, in which one user has conflicting intentions, for example comfort and energy saving; *user vs. user*, when more than one user concur over a given resource, application, or environment state; *user vs. space*, when user actions conflict with any established space policies, e.g., a user's smartphone ringing in a room with a silence policy.

The *time of detection* is another way of classifying conflicts. A conflict can be detected *a priori*, i.e., predicted to happen. In some literature, this is called a potential conflict (Syukur et al. 2005). More particularly, it can be either a *definite potential conflict*, i.e. a conflict that will definitely occur if the user is in the right context; or a *possible potential conflict*, if the possibility of occurrence is less than that of the definite potential conflict, since it may still not happen, even if the user is in the right context and time. A conflict can also be detected while it is happening (also called actual conflict), either through the system's Context-Awareness capabilities, or through user feedback. The last case is when a conflict is only detected after a reasonable resolution opportunity has passed, likely due to Context-Awareness limitations or sensing delays, as we have previously explained.

Finally, conflicts can be distinguished by their *solvability*. The best case scenario is *conflict avoidance*, when detection happens before occurrence and a conflict is solved before it actually happens. More commonly, a conflict is detected during its actual occurrence, and conflict resolution takes place. Otherwise, the system may *acknowledge its inability* to deal with the conflict. Another possibility is that the system, not detecting a conflict soon enough to resolve it, realizes

Table 1 A four-dimensional taxonomy of conflicts, organized according to their classification dimensions, possible types and respective meaning.

Classification dimension	Possible types	Meaning
Source	resource	concurrency over a resource
	application	concurrency over an application
	policy	conflicting policies in a context
	role	conflicting profiles in a context
Intervenients	single user	conflicting user intentions
	user vs. user	concurrency over a resource or application
	user vs. space	user conflicts with space constraints
Time of Detection	*a priori*	detected before its occurrence (predicted)
	when it occurs	detected during occurrence
	a posteriori	detected after it has occurred
Solvability	conflict avoidance	the system resolves the conflict before its occurrence
	conflict resolution	the system resolves the conflict during its occurrence
	acknowledge inability	the system recognizes its inability to resolve conflict
	acknowledge occurrence	the system acknowledges too late that a conflict occurred

later that it happened (e.g., due to delayed sensor information) and can only *acknowledge its occurrence*, possibly informing the users or system administrators of the occurred conflict situation.

As we have seen, conflicts may be of several types and arise from a number of dimensions, thus requiring different detection and resolution mechanisms. Given the predictive nature of HBAS, conflicts can sometimes be anticipated before they even happen. In that case, the system may take action against the actual occurrence of the conflict, thus resolving it. We will call this conflict resolution through avoidance, more commonly known as *conflict avoidance*. Another possibility is that the system only detects the conflict when a new context is identified, either by the application or through user feedback. In that case, action may be taken in order to eliminate the conflict. The system may: *(i)* restore the previous context; *(ii)* adjust the new context towards general desirability or tolerance; *(iii)* inform users that they should explicitly resolve the conflict. Several conflict resolution approaches for AmI have been developed so far, and will be described further.

4.2 Origins of Conflict

Conflict occurs in all kinds of human relationships. The absence of conflict usually signals the absence of meaningful interaction (Fisher 2000). Interpersonal conflict arises when incompatible goals develop between persons or groups, and its origins vary. The Circle of Conflict Model, originally developed by Moore (Moore 2003) and adapted by Furlong (Furlong 2005), looks at conflicts in the perspective of their origins. This model postulates five main underlying causes to conflict: *values*: all values, morals, ethics and beliefs; *relationships*: negative past history or experiences; *externals/moods*: external factors not directly related to the situation, but that still contribute to the conflict; *data*: when parties have incorrect, incomplete or different data or interpretations of it; *structure*: system structure problems, such as limited resources, authority problems, and organizational structures (Furlong 2005). Moreover, unresolved or inadequately resolved conflicts tend to result in tension, which may trigger or intensify posterior conflicts.

4.3 Interpersonal Conflict Resolution

Interpersonal conflict is something humans learn to deal with, more or less effectively, from an early age. Conflict by itself is neither good nor bad. The manner in which it is handled, or responded to, is what determines whether it is constructive or destructive (Sandole et al. 2009).

Thomas (Thomas 1992) adapted a previously presented graphical view of interpersonal conflict handling modes (Thomas and Kilmann 1978) following a two-dimensional taxonomy: Assertiveness, i.e., the extent to which the person attempts to satisfy his own concerns; and Cooperativeness, the extent to which the person attempts to satisfy the other person's concerns. In this taxonomy, five modes are presented and described, and remain well accepted and studied. We present an adaptation of this taxonomy in Figure 4. Furthermore, the Thomas-Kilmann Conflict Mode Instrument (TKI) is a broadly used tool to help identify which style one tends towards when conflict arises. The TKI has also been recently adapted to serve as basis for a computational simulation model for interpersonal conflict management in team building, using an agent-based modeling method (Wang and Ting 2011), which shows both its currency and overall acceptance.

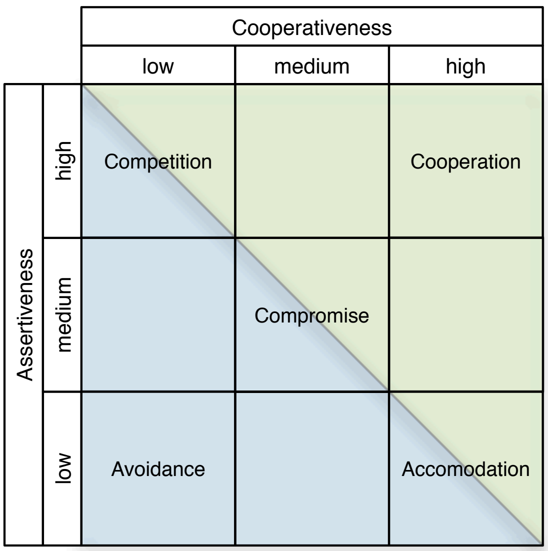

Fig. 4 Two-dimensional taxonomy, displaying the five types of interpersonal conflict handling modes in terms of assertiveness and cooperativeness levels. The diagonal represents the threshold between non-constructive or destructive (left) and constructive (right) interpersonal conflict resolution. Adapted from Thomas, 1992 (Thomas 1992).

These two basic dimensions of behaviour define five different modes for responding to conflict situations: *Competition*: power oriented mode, in which an individual pursues his own concerns at the other person's expense, resulting in a "win-lose" situation, because only one can win; *Accommodation*: an individual self-sacrifices by neglecting his own concerns to satisfy the other's, also resulting in a "win-lose" situation; *Avoidance*: the person neither pursues his own concerns nor the other's, thus the conflict isn't actually dealt with, resulting in a "lose-lose" situation; *Cooperation*: involves an attempt to work with others to find some solution that fully satisfies their concerns, by digging into an issue and trying to find a solution to an interpersonal problem, therefore both parties win; *Compromise*: attempt to find some convenient and mutually acceptable "win-win" solution that partially satisfies both parties, possibly exchanging concessions, or seeking a quick and middle-ground solution.

4.4 Formal Models of Conflict

Several facets of human behaviour, e.g. language and social relations, have been studied and mathematically modeled for quite some time, as they were of interest for various areas (Casti 1989). Conflict was no exception to this sort of modeling. In fact, Coombs attempted to "bring some order to the chaotic variety of conflict" (Coombs and Avrunin 1988), by describing some of its structural aspects and proposing a conflict classification that considers three types of conflict: *Type I*: conflict within a user, who must make an option; *Type II*: conflict between users who must settle for the same thing; and *Type III*: conflict between users who want the same thing but must settle for different ones. In our conflict taxonomy, their Type I conflict corresponds to the single user conflict and their Type II conflict is our user vs. user conflict (see Figure 4).

Among other authors, Pawlak (Pawlak 1984) presented a mathematical model of interpersonal conflict situations, that enabled a graph representation of conflicts, and consequently, their computer simulation. He based his model on binary relations of alliance, conflict and neutrality among conflicting entities, as well as their strength, thus indirectly including the notion of hierarchy. Pawlak's approach can be considered an extension of ideas presented a decade earlier (Roberts 1976) and many authors further extended and applied his model. Deja (Deja and Slezak 2001) enhanced Pawlak's model with local aspects of conflicts, namely: *(i)* the notion of a local state enabling a subjective perception of the world by each agent; *(ii)* the objective evaluation of the global situation by an expert (represented by a quality function), leading to a global metric of a conflict's strength; and *(iii)* constraints that restrict the set of possible situations to admissible ones. He further proposes a conflict resolution strategy based on Boolean reasoning and rough set theory (Slezak et al. 2005), which applied to these local aspects, enable a more effective way to reach consensus among agents.

An example application of the rough set approach to a high-level requirements negotiation for an automated lighting system is presented in (Skowron et al. 2006). Skowron and Deja (Skowron and Deja 2002) further allow a tolerance relation, specified as a distance function, to denote similarity between local agent states.

Models similar to the ones presented above have also been applied in other Computer Science areas. For instance, Maeda (Maeda et al. 1999) proposes an approach to expert's knowledge representation based on weighed interval importances. In this approach, expert's knowledge is given as a relative importance for each attribute. Moreover, Maeda shows how this representation can be applied to conflict analysis, by considering that when there are plural experts for a given issue, their knowledge may be formulated using interval density functions, and that then conflict degrees between two agents can be obtained as an interval value.

4.5 Conflict Resolution in AmI

Conflict resolution in smart environments is still a relatively new topic. Several works on intelligent environments and resource management describe their conflict detection and resolution methods (see Table 2).

4.5.1 Conflict resolution in multi-agent systems

Alshabi et al. (Alshabi et al. 2007) present a survey on agent cooperation models and conflict resolution methods in multi-agent systems, comparing three existing architectures, namely HOPES (Bell and Grimson 1992), HECODES (Bell and Grimson 1992) and MAGIC (Bensaid and Mathieu 1997). They propose their own multi-agent architecture, in which each agent independently chooses its balance between cooperation and autonomy, and in case of conflict, the agents adopt one of three negotiation strategies: *(i)* arbitrary leader election, *(ii)* chaining or *(iii)* cloning. These strategies are detailed in (Ramachandran et al. 2001).

Jacak and Pröll present a heuristic approach for conflict management in multi-agent systems (Jacak and Proll 2007). Each agent is able to perceive and react to the environment, to plan and execute an action, and to negotiate with other agents. This allows agents to coordinate and negotiate their actions towards achieving a common global goal, which happens by a correct sequencing of each agent's local goals, resulting in conflict avoidance through negotiation.

Kung and Lin propose and design the Context-Aware Embedded Multimedia Presentation System (CEMP), which offers *context adaptation control*, provided by an adaptation mechanism and a learning and prediction module that adapt to user's needs based on context information, and offers steady multimedia stream services according to the user's context information (Kung and Lin 2006). It is based on a context vocabulary ontology, expressed with the Web Ontology Language (OWL), that provides the formal explicit description about the multimedia information domain and the formal user context. The system provides a mechanism for adaptation reasoning, which infers the best adaptation control, whenever contexts change. In CEMP, conflict resolution is performed through a mechanism that calculates each context's priority and weight, to get the best context results, and thus adapt towards meeting the multimedia service's quality requirements.

4.5.2 Interest/intention conflict resolution

COMITY is an application framework that implements *a priori* conflict detection and resolution through avoidance, of conflicts in the environment resulting from applications' responses to different user interests (Tuttlies et al. 2007). The approach is based on the PCOM (Becker et al. 2004) component model, which proposes the idea of *context contracts* to specify each component's requirements and effects on the environment. Application in COMITY is a coordination of components which can be changed at runtime, to de liver the functionality required by the application under potentially new *context constraints*. The conflict manager component performs conflict detection and adapts the application at run-time, based on the information available in a database with a context model representing the environment state, and a database of conflicting situations.

Armac et al. (Armac et al. 2006) propose to specify components as services, where each service consists of actions that denote state transitions on resources. The admissible state transitions are captured by an ω -automaton associated with each resource. A conflict situation is detected when a service attempts to transition a resource from a state previously set by another service. Conflict resolution is handled by a rule-based mechanism and handled with priorities.

In (Park et al. 2005), Park et al. propose a dynamic conflict resolution scheme for resolving conflicts between different context-aware applications in smart environments, which incorporates users' intentions and preferences. They model user intentions as the value assigned to a context attribute by the actions they requested from applications, and express user preferences as cost functions over the distance between user intentions and the resolved value. Based on this information, the resolved value is determined to the one which minimizes the cost of all users involved in conflicts.

In (Silva et al. 2010), Silva et al. define a collective conflict as an inconsistent state that a collective application may reach while evaluating collective contexts, in which the application becomes unable to satisfy, simultaneously, divergent individual interests. They propose a conflict detection and resolution methodology that uses a client-server architecture model to select and configure the current most appropriate conflict resolution algorithm available. This decision is made considering the application's demands for quality of services (QoS) and resources consumption. The QoS criteria consid-

Table 2 Summary of the related work on conflict resolution

Conflict resolution topic	References
Conflict resolution in multi-agent systems	(Alshabi et al. 2007) (Jacak and Proll 2007)
Interest/intention conflict resolution	(Tuttlies et al. 2007) (Park et al. 2005)
Resource conflict resolution	(Armac et al. 2006) (Retkowitz and Kulle 2009) (Huerta-Canepa and Lee 2008)
Policy conflict resolution	(Syukur et al. 2005) (Lupu and Sloman 1997) (Jiao et al. 2009) (Jiao et al. 2008) (Capra et al. 2003)
Authorization conflict resolution	(Masoumzadeh et al. 2007)

ered in their work is the collective satisfaction of users regarding the attained resolution results. This framework is further detailed in (Silva et al. 2011), where the authors present a case study regarding a collective tourist guide application to demonstrate the developed methodology's effectiveness.

4.5.3 Resource conflict resolution

Retkowitz et al. (Retkowitz and Kulle 2009) argue that in most real-life scenarios, conflicts among users require manual resolution by the users, therefore, they focus primarily resource concurrency conflicts, and present an approach that attempts to avoid such conflicts through a more dynamic and effective resource sharing. Their approach considers dependencies between services, that are realized as bindings. A configuration process tries to derive a service composition that simultaneously matches user requirements, device environment and all service dependencies, respecting previously defined binding policies and constraints, and handling service bindings and access control based on priority groups.

Huerta-Canepa et al. (Huerta-Canepa and Lee 2008) also propose a resource management scheme for smart spaces based on ad-hoc interaction, assuming that only two jobs can be executed at each time. Conflict avoidance is performed through area device control, and conflict resolution is done at the resource level, based on the utility and cost of executing a given job instead of the current one, and on user priorities.

4.5.4 Policy conflict resolution

In conflict resolution, a policy is a principle of behaviour, that establishes and delimits acceptable system states and system behaviour for given situations. Policies, as well as rules with priority schemes among entities, are the most common conflict resolution techniques (Kawsar and Nakajima 2007). In (Syukur et al. 2005), Syukur et al. investigated conflict detection and avoidance strategies based on context changes, assuming a policy-based application model in which policies are used to explicitly control the behaviour of an application. This paper classifies policy conflicts in terms of their sources, detection and resolution techniques, and different timing approaches for resolution. The authors argue that each strategy's suitability depends on system situations (e.g., number of users, considered contexts, types of services), on system goals (e.g., high performance or low cost) and on the types of conflicts that the system has to detect or resolve. Nevertheless, their results pointed that the best conflict detection was achieved with a hybrid of offline (*a priori*) and dynamic (run-time) detection, since it helped improve users' wait time; and that *a priori* conflict resolution (and consequent conflict avoidance) of all conflicts (including potential conflicts) as soon as they were detected, was the best resolution technique, as it improved the system's performance by promptly responding to user requests.

Lupu et al. (Lupu and Sloman 1997) also consider policy conflicts, and although assuming that some conflicts can only be detected at runtime, they rather focus on techniques for offline conflict detection and offline conflict resolution (which ultimately is conflict avoidance), that assist the users in specifying policies, roles and relationships. The authors consider that management policies are specified with regard to domains of objects and that conflicts arise when these domains overlap. Moreover, they describe a conflict analysis tool which forms part of a Role Based Management framework, in which management policies are specified regarding object domains. They consider that conflicts potentially arise when overlaps occur between these object domains, i.e., when ⟨subject,action,target⟩ tuples have different policies applying to them. Their approach uses roles and inter-role relationships to reduce the scope of policies that need to be examined, and applies a precedence strategy based in domain nesting to reduce the number of overlaps presented to users.

Jiao et al. present another policy conflict detection algorithm in (Jiao et al. 2009), that aims to improve the time that it takes for a routine to search every rule in a policy rule set with conventional algorithms, to see if conflict occurs before a new rule is added to the rule set. In order to address this problem and thus increase the efficiency of policy conflict detection, the authors start by formalizing context, policy rules and policy conflicts; and proceed by constructing a formal concept lattice through formal concept analysis, based on which policies are later grouped into the associated formal concepts. These formalizations were then used to propose an algorithm for conflict detection, which was subject to a performance analysis and simulation. Results show that concept lattice is an effective support, both in determining whether policy attribute values, and in reducing the number of policy rules to be tested during conflict detection.

CARISMA (Capra et al. 2003) supports runtime dynamic policy conflict detection and resolution. Conflicts are defined as the inconsistencies that arise when contradictory behaviours are requested by the same application as a reaction to a context change, or when cooperating applications don't agree on a common behaviour to be applied. They argue that conflicts cannot be resolved statically at the applications' design time, but rather need to be resolved at run-time. This approach defines an application profile as the application's current configuration. Based on this notion of profile, CARISMA defines and handles two types of conflicts: *(i)* intraprofile conflicts, in which a conflict exists inside the profile of an application running on a given device; and *(ii)* inter-profile conflicts, which exist between the profiles of applications running on different devices. Moreover, it uses a microeconomics technique that relies on a type of sealed-bid auction between applications, in order to achieve a relatively fair conflict resolution method.

4.5.5 Authorization conflict resolution

Masoumzadeh et al. (Masoumzadeh et al. 2007) refer to authorization conflicts, which are a particular case of policy conflicts, namely when two or more different policies both permit and forbid an access, i.e., give and deny authorization, regarding a given situation, e.g., a policy allows an occupant to turn on the radio, and another policy imposes silence. The authors propose conflict detection to be performed almost statically, and conflict resolution to be left for run-time. They consider that a practical solution for this is establishing a precedence among conflicting policies. To that aim, they formalize the use of context constraints in a rule-based context-aware multi-authority policy model, i.e., a model that allows the establishment of precedence principles for policies. Moreover, they analyze timing strategies and resolution algorithms, and propose a comprehensive graph-based approach to enable precedence establishment among authorizations in a conflict situation. In the detection phase, a potential conflict graph is almost statically constructed, and in an actual conflict situation, using this previously created graph provides the resolution.

5 Conclusions

The development of usable, intelligent and highly adaptive Home and Building Automation Systems (HBAS), is a multi-domain problem that has been receiving increasing attention. We presented an overview on existing HBAS, and their multidisciplinary developments and evolution perspectives.

Moreover, AmI was addressed, with respect to its potential contributions to the development of more intelligent, adaptive and user-centred HBAS, particularly regarding the areas of Context-Awareness and human-computer interaction.

Performing automatic conflict resolution requires a context-aware system that takes information regarding the context of the environment as input, checks for conflicting situations and then produces a new context in the environment, in order to conciliate conflicting requirements or inform about the conflicting situation. In other words, the automatic resolution of conflicts consists of creating actuations to accommodate requirements of new activities carried out by occupants.

Many gaps still remain regarding automatic adaptation to the user, one of which is the use of effective conflict detection and resolution mechanisms, able to respond automatically and appropriately to a variety of decision-demanding scenarios. Context inference is possible through the analysis of sensor data and the use of profiling and Machine Learning techniques on this data. Conflicts occur when a user or application changes the environment state, causing an undesired context. Conflict detection is based on this context analysis. A conflict taxonomy was presented in this paper, regarding different classification dimensions, for clarity. We further surveyed conflicts in the interpersonal sense and with regard to existing conflict detection and resolution methods in AmI systems.

Another contribution was the systematization of existing automatic conflict detection and resolution approaches. We finished with the exposition of possible conflict detection and resolution methods, which we believe will likely have practical applicability in HBAS.

Acknowledgements The work reported in this article was supported by national *Fundação para a Ciência e a Tecnolo-*

gia (FCT) funds under contract PEst-OE/EEI/LA0021/2011 and grant SFRH/BD/47409/2008.

References

Abadi D, Madden S, Lindner W (2005) REED: Robust, Efficient Filtering and Event Detection in Sensor Networks. In: Proceedings of the 31st International Conference on Very Large Data Bases, VLDB Endowment, VLDB '05, pp 769–780

Abielmona R, Petriu E, Whalen T (2010) Distributed intelligent sensor agent system for environment mapping. Journal of Ambient Intelligence and Humanized Computing 1(2):95–110

Agarwal Y, Balaji B, Gupta R, Lyles J, Wei M, Weng T (2010) Occupancy-driven energy management for smart building automation. In: Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, ACM, BuildSys '10, pp 1–6

Aipperspach R, Cohen E, Canny J (2006) Modeling Human Behavior from Simple Sensors in the Home. In: Pervasive Computing, Lecture Notes in Computer Science, vol 3968, Springer, pp 337–348

Alcañiz M, Rey B (2005) New technologies for ambient intelligence. Ambient Intelligence 6(1):3–15

Alshabi W, Ramaswamy S, Itmi M, Abdulrab H (2007) Coordination, Cooperation and Conflict Resolution in Multi-Agent Systems. In: Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, Springer, pp 495–500

Armac I, Kirchhof M, Manolescu L (2006) Modeling and analysis of functionality in eHome systems: dynamic rule-based conflict detection. In: 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems, 2006. ECBS 2006.

Baldauf M, Dustdar S, Rosenberg F (2007) A survey on context-aware systems. International Journal of Ad Hoc Ubiquitous Computing (IJAHUC) 2(4):263–277

Ballagas R, Ringel M, Stone M, Borchers J (2003) iStuff: a physical user interface toolkit for ubiquitous computing environments. In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, CHI '03, pp 537–544

Becker C, Handte M, Schiele G, Rothermel K (2004) PCOM - a component system for pervasive computing. In: Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004, pp 67–76

Bell D, Grimson J (1992) Distributed Database Systems. Addison-Wesley

Bensaid N, Mathieu P (1997) A Framework for Cooperation in Hierarchical Multi-Agent Systems

Bottaro A, Gerodolle A (2008) Home SOA: facing protocol heterogeneity in pervasive applications. In: Proceedings of the 5th IEEE International Conference on Pervasive Services, ACM, ICPS '08, pp 73–80

Bucceri R (2006) Latest Technology in Automated Home Control: System Design Manual Using X-10 & Hardwired Protocols. Silent Servant Inc.

Buchmayr M, Kurschl W (2011) A survey on situation-aware ambient intelligence systems. Journal of Ambient Intelligence and Humanized Computing 2(3):175–183

Capra L, Emmerich W, Mascolo C (2003) CARISMA: context-aware reflective middleware system for mobile applications. IEEE Transactions on Software Engineering 29(10):929–945

Casti J (1989) Alternate realities: mathematical models of nature and man. Wiley

Chalmers D (2011) Sensing and Systems in Pervasive Computing: Engineering Context Aware Systems. Springer

Cialdini R (2001) Harnessing the Science of Persuasion. Harvard Business Review 79(9):72–81

Cook D, Das S (2004) Smart Environments: Technology, Protocols and Applications. Wiley Series on Parallel and Distributed Computing, John Wiley & Sons

Coombs C, Avrunin G (1988) The structure of conflict. L. Erlbaum Associates

Coutaz J, Fontaine E, Mandran N, Demeure A (2010) About composing our own smart home. In: Proceedings of the International Conference on Advanced Visual Interfaces, ACM, AVI '10, pp 405–406

Cziker A, Chindris M, Miron A (2007) Implementation of Fuzzy Logic in Daylighting Control. In: 11th International Conference on Intelligent Engineering Systems, 2007. INES 2007., pp 195–200

Davidoff S, Lee M, Yiu C, Zimmerman J, Dey A (2006) Principles of Smart Home Control. In: UbiComp 2006: Ubiquitous Computing, Lecture Notes in Computer Science, vol 4206, Springer, pp 19–34

Deja R, Slezak D (2001) Rough Set Theory in Conflict Analysis. In: New Frontiers in Artificial Intelligence, Lecture Notes in Computer Science, vol 2253, Springer, pp 349–353

Dey A (2001) Understanding and Using Context. Personal and Ubiquitous Computing 5(1):4–7

Doulos L, Tsangrassoulis A, Topalis F (2007) The impact of colored glazing and spectral response of photosensors in the estimation of daylighting energy savings. In: Proceedings of the 2nd PALENC Conference and the 28th AIVC Conference

Edwards W, Grinter R (2001) At Home with Ubiquitous Computing: Seven Challenges. In: Ubicomp 2001: Ubiquitous Computing, Lecture Notes in Computer Science, vol 2201, Springer, pp 256–272

Endsley M (1988) Design and evaluation for situation awareness enhancement. In: Proceedings of the Human Factors and Ergonomics Society 32nd Annual Meeting, Human Factors Society, pp 97–101

Epstein G, McCowan B, Birleanu D (2003) Integrating Daylighting and Electrical Lighting for Premium Efficiency and Performance

Findlater L, Gajos K (2009) Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces. AI Magazine 30(4):68–73

Fisher R (2000) Sources of Conflict and Methods of Resolution. The American University

Floyd D, Parker D (1995) Field Commissioning of a Daylight-Dimming Lighting System. In: Right light 3: proceedings of the 3rd European Conference on Energy-Efficient Lighting

Fogg B (2003) Persuasive technology: using computers to change what we think and do. Morgan Kaufmann series in interactive technologies, Morgan Kaufmann Publishers

Furey E, Curran K, Kevitt P (2012) Probabilistic indoor human movement modeling to aid first responders. Journal of Ambient Intelligence and Humanized Computing pp 1–11

Furlong G (2005) The Conflict Resolution Toolbox: Models and Maps for Analyzing, Diagnosing and Resolving Conflict. John Wiley & Sons Canada, Ltd.

Galasiu A, Atif M, MacDonald R (2004) Impact of window blinds on daylight-linked dimming and automatic on/off lighting controls. Solar Energy 76(5):523–544

García-Herranz M, Haya P, Alamán X, Martín P (2007) Easing the Smart Home: augmenting devices and defining scenario. In: 2nd International Symposium on Ubiquitous Computing & Ambient Intelligence, Thomson editorial, pp 67–74

Garg V, Bansal N (2000) Smart occupancy sensors to reduce energy consumption. Energy and Buildings 32(1):81–87

Ghanam Y, Shouman M, Greenberg S, Maurer F (2009) Object-Specific Interfaces in Smart Homes. Research report 2009-937-16, Department of Computer Science, University of Calgary, Calgary, Canada

Granzer W, Kastner W, Neugschwandtner G, Praus F (2006) A modular architecture for building automation systems. In: IEEE International Workshop on Factory Communication Systems, 2006, pp 99–102

Gross T (2008) Cooperative ambient intelligence: towards autonomous and adaptive cooperative ubiquitous environments. International Journal of Autonomous and Adaptive Communication Systems 1(2):270–278

Ha T, Jung J, Oh S (2006) Method to analyze user behavior in home environment. Personal and Ubiquitous Computing 10(2):110–121

Hasan M, Anh K, Mehedy L, Lee Y, Lee S (2006) Conflict resolution and preference learning in ubiquitous environment. In: Computational Intelligence, Lecture Notes in Computer Science, vol 4114, Springer, pp 355–366

Huerta-Canepa G, Lee D (2008) A multi-user ad-hoc resource manager for smart spaces. In: International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2008, pp 1–6

Jacak W, Proll K (2007) Heuristic Approach to Conflict Problem Solving in an Intelligent Multiagent System. In: Computer Aided Systems Theory – EUROCAST 2007, Lecture Notes in Computer Science, vol 4739, Springer, pp 772–779

Jiao S, Liu Y, Hu H, Wei D, Zhang Y (2008) Dynamic Policy Access Model Based on Formal Concept Analysis. In: 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08., pp 1–5

Jiao S, Liu Y, Qi X, Zhu Y, Wang J (2009) Detecting Conflict Policy Rules with Concept Lattice. In: 5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09., pp 1–4

Kaptein M, Markopoulos P, de Ruyter B, Aarts E (2009) Can You Be Persuaded? Individual Differences in Susceptibility to Persuasion. In: Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I, Springer, INTERACT '09, pp 115–118

Kawsar F, Nakajima T (2007) Persona: a portable tool for augmenting proactive applications with multimodal personalization support. In: Proceedings of the 6th international conference on Mobile and Ubiquitous Multimedia, ACM, MUM '07, pp 160–168

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

Korpipaa P, Mantyjarvi J, Kela J, Keranen H, Malm E (2003) Managing context information in mobile devices. Pervasive Computing, IEEE 2(3):42–51

Kovach-Hebling A, Goller M, Herkel S, Wienold J (1997) Assessing the Energy Saving Potential of Daylighting Technologies for Non-residential Buildings in Germany. In: Right light 4: proceedings of the 4th European Conference on Energy-Efficient Lighting

Kung H, Lin C (2006) Application-Layer Context-Aware Services for Pervasive Computing Environments. In: First International Conference on Innovative Computing, Information and Control, 2006. ICICIC '06., vol 3, pp 229–232

Lah M, Zupancic B, Krainer A (2005) Fuzzy control for the illumination and temperature comfort in a test chamber. Building and Environment 40(12):1626–1637

Lah M, Zupancic B, Peternelj J, Krainer A (2006) Daylight illuminance control with fuzzy logic. Solar Energy 80(3):307–321

Ledlie J, Ng C, Holland D (2005) Provenance-Aware Sensor Data Storage. In: 21st International Conference on Data Engineering Workshops, 2005.

Lee E, DiBartolomeo D, Vine E, Selkowitz S (1998) Integrated Performance of an Automated Venetian Blind/Electric Lighting System in a Full-Scale Private Office. In: Proceedings of the ASHRAE/DOE/BTECC Conference, Thermal Performance of the Exterior Envelopes of Buildings VII

Li K (2012) Smart home technology for telemedicine and emergency management. Journal of Ambient Intelligence and Humanized Computing pp 1–12

LIFX (2012) Lifx: The light bulb re-invented. `http://lifx.co` (Last accessed on October, 2012)

Littlefair P, Ortiz J, Bhaumik C (2010) A simulation of solar shading control on UK office energy use. Building Research & Information 38(6):638–646

Lupu E, Sloman M (1997) Conflict analysis for management policies. In: Proceedings of the 5th IFIP/IEEE international symposium on Integrated network management V: integrated management in a virtual world, Chapman & Hall, Ltd., pp 430–443

Maeda Y, Senoo K, Tanaka H (1999) Interval Density Functions in Conflict Analysis. In: New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Lecture Notes in Computer Science, vol 1711, Springer, pp 382–389

Markopoulos P (2005) Designing ubiquitous computer human interaction: The case of the connected family. In: Future Interaction Design, Springer, pp 125–149

Markopoulos P, de Ruyter B, Privender S, van Breemen A (2005) Case study: Bringing social intelligence into home dialogue systems. Interactions 12(4):37–44

Masoumzadeh A, Amini M, Jalili R (2007) Conflict Detection and Resolution in Context-Aware Authorization. In: 21st International Conference on Advanced Information Networking and Applications Workshops, 2007. AINAW '07., IEEE Computer Society, AINAW '07, pp 505–511

Maybury M, Wahlster W (1998) Intelligent user interfaces: an introduction. In: Readings in Intelligent User Interfaces, Morgan Kaufmann Publishers Inc., pp 1–13

Merz H, Hansemann T, Hübner C (2009) Building automation: communication systems with EIB/KNX, LON und BACnet. Signals and communication technology, Springer

Miki M, Hiroyasu T, Imazato K (2004) Proposal for an intelligent lighting system, and verification of control method effectiveness. In: IEEE Conference on Cybernetics and Intelligent Systems, 2004, vol 1, pp 520–525

Moore C (2003) The Mediation Process: Practical Strategies for Resolving Conflict. Jossey-Bass Publishers

Nakashima H, Aghajan H, Augusto J (eds) (2010) Handbook of Ambient Intelligence and Smart Environments. Springer

Neisse R, Wegdam M, van Sinderen M (2008) Trustworthiness and quality of context information. In: The 9th International Conference for Young Computer Scientists, 2008. ICYCS 2008., pp 1925–1931

Newsham G, Mancini S (2006) The Potential for demand responsive lighting in non-daylit offices. LEUKOS, The Journal of the Illuminating Engineering Society 3(2):105–120

Nichols J, Myers B, Higgins M, Hughes J, Harris T, Rosenfeld R, Pignol M (2002) Generating remote control interfaces for complex appliances. In: Proceedings of the 15th annual ACM symposium on User Interface Software and Technology, ACM, UIST '02, pp 161–170

Park I, Lee D, Hyun S (2005) A Dynamic Context-Conflict Management Scheme for Group-Aware Ubiquitous Computing Environments. In: 29th Annual International Computer Software and Applications Conference, 2005. COMPSAC 2005., IEEE Computer Society, vol 1, pp 359–364

Park T, Hong S (2006) Development of an Experimental Model of BACnet-based Lighting Control System. In: IEEE International Conference on Industrial Informatics, 2006, pp 114–119

Pawlak Z (1984) On conflicts. International Journal of Man-Machine Studies 21(2):127–134

Ramachandran V, Ramaswamy S, Rajan P (2001) Complex Negotiation Protocols For A Distributed Simulation

Ranganathan A, Campbell R (2003) An infrastructure for context-awareness based on first order logic. Personal and Ubiquitous Computing 7(6):353–364

Rashidi P, Cook D (2009) Keeping the Resident in the Loop: Adapting the Smart Home to the User. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 39(5):949–959

Reinisch C, Kofler M, Iglesias F, Kastner W (2011) ThinkHome Energy Efficiency in Future Smart Homes. EURASIP Journal on Embedded Systems

Remagnino P, Foresti G (2005) Ambient Intelligence: A New Multidisciplinary Paradigm. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 35(1):1–6

Retkowitz D, Kulle S (2009) Dependency Management in Smart Homes. In: Distributed Applications and Interoperable Systems, Lecture Notes in Computer Science, vol 5523, Springer, pp 143–156

Richman E, Dittmer A, Keller J (1995) Field Analysis of Occupancy Sensor Operation: Parameters affecting lighting energy savings. In: Proceedings of the Illuminating Engineering Society of North America 1995 Annual Conference

Roberts F (1976) Discrete mathematical models, with applications to social, biological, and environmental problems. Prentice-Hall

de Ruyter B, Saini P, Markopoulos P, van Breemen A (2005) Assessing the effects of building social intelligence in a robotic interface for the home. Interacting with Computers 17(5):522–541

Saini P, de Ruyter B, Markopoulos P, van Breemen A (2005) Benefits of Social Intelligence in Home Dialogue Systems. In: Human-Computer Interaction - INTERACT 2005, Lecture Notes in Computer Science, vol 3585, Springer, pp 510–521

Sandole D, Byrne S, Sandole-Staroste I, Senehi J (2009) Handbook of Conflict Analysis and Resolution. Routledge

Santos A, Tarrataca L, Cardoso J, Ferreira D, Diniz P, Chainho P (2009) Context Inference for Mobile Applications in the UPCASE Project. In: MobileWireless Middleware, Operating Systems, and Applications, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 7, Springer, pp 352–365

Satoh I (2007) Visual Components for Pervasive Computing Management. In: IEEE International Conference on Pervasive Services, pp 19–28

Schaefer R, Muller W, Groppe J (2006) Profile Processing and Evolution for Smart Environments. In: Ubiquitous Intelligence and Computing, Lecture Notes in Computer Science, vol 4159, Springer, pp 746–755

Silva T, Ruiz L, Loureiro A (2010) How to conciliate conflicting users' interests for different collective, ubiquitous and context-aware applications? In: IEEE 35th Conference on Local Computer Networks (LCN), 2010, pp 288–291

Silva T, Ruiz L, Loureiro A (2011) Conflicts treatment for ubiquitous collective and context-aware applications. Journal of Applied Computing Research 1(1):33–47

Skowron A, Deja R (2002) On Some Conflict Models and Conflict Resolutions. Romanian Journal of Information Science and Technology 5(1–2):69–82

Skowron A, Ramanna S, Peters J (2006) Conflict Analysis and Information Systems: A Rough Set Approach. In: Rough Sets and Knowledge Technology, Lecture Notes in Computer Science, vol 4062, Springer, pp 233–240

Slezak D, Wang G, Szczuka M, Düntsch I, Yao Y (2005) Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing - 10th International Conference, RSFDGrC 2005. Lecture Notes in Computer Science, Springer

Smartthings (2012) Smarthings: Make your world smarter. http://smartthings.com (Last accessed on October, 2012)

Sorrentino F (2009) Ambient Intelligence. In: Encyclopedia of Information Communication Technology, IGI Global, pp 32–39

Spinellis D (2003) The information furnace: consolidated home control. Personal and Ubiquitous Computing 7(1):53–69

Syukur E, Loke S, Stanski P (2005) Methods for Policy Conflict Detection and Resolution in Pervasive Computing Environments. In: Policy Management for Web workshop in conjunction with WWW'05 Conference, ACM, pp 10–14

Szewcyzk S, Dwan K, Minor B, Swedlove B, Cook D (2009) Annotating smart environment sensor data for activity learning. Technology and Health Care 17(3):161–169

Tapia D, Abraham A, Corchado J, Alonso R (2010a) Agents and ambient intelligence: case studies. Journal of Ambient Intelligence and Humanized Computing 1(2):85–93

Tapia D, Rodríguez S, Corchado J (2010b) A Distributed Ambient Intelligence Based Multi-Agent System for Alzheimer Health Care. In: Pervasive Computing, Computer Communications and Networks, Springer, pp 181–199

Thomas K (1992) Conflict and conflict management: Reflections and update. Journal of Organizational Behavior 13(3):265–274

Thomas K, Kilmann R (1978) Comparison of four instruments for measuring conflict behavior. Psychological Reports, Vol 42 (3, Pt 2)

Tuttlies V, Schiele G, Becker C (2007) COMITY: Conflict Avoidance in Pervasive Computing Environments. In: On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops, Lecture Notes in Computer Science, vol 4806, Springer, pp 763–772

Venkatesh V, Morris M, Davis G, Davis F (2003) User acceptance of information technology: toward a unified view.

MIS Q 27(3):425–478
Vernon P (1933) Some Characteristics of the Good Judge of Personality. The Journal of Social Psychology 4(1):42–57
Wang W, Ting S (2011) Development of a Computational Simulation Model for Conflict Management in Team Building. International Journal of Engineering Business Management 3(2):9–15
Weiser M (1991) The Computer for the Twenty-First Century. Scientific American 3
Weiser M (1993) Hot Topics: Ubiquitous Computing. IEEE Computer 26(10):71–72
Welbourne E, Lester J, LaMarca A, Borriello G (2005) Mobile Context Inference Using Low-Cost sensors. In: Location- and Context-Awareness, Lecture Notes in Computer Science, vol 3479, Springer, pp 95–127
Yamazaki T (2007) The Ubiquitous Home. International Journal of Smart Home (IJSH) 1(1):17–22
Zhou H, Rao M, Chuang K (1993) Artificial Intelligence Approach to Energy Management and Control in the HVAC Process: An Evaluation, Development and Discussion. Developments in Chemical Engineering and Mineral Processing 1(1):42–51
Zimmer T (2004) Towards a better understanding of context attributes. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004., pp 23–27