Ambient-Intelligence Trigger Markup Language
A new approach to Ambient Intelligence rule definition

Juan Manuel Fernández, Sergi Torrellas, Stefan Dauwalder, Marc Solà, Eloisa Vargiu and Felip Miralles

Barcelona Digital Technology Center, `jmfernandez@bdigital.org`, `storellas@gmail.com`, {`sdauwalder, msola, evargiu, fmiralles`}`@bdigital.org`

Abstract. Assistive technologies need to constantly adapt and react to user needs. To this end, ambient intelligence techniques could be adopted. One approach consists of defining suitable rules to trigger actions or suggestions to the users. In this paper, ATML (Ambient intelligence Trigger Markup Language), a novel suitable language, is presented and described. ATML is aimed at defining and describing actions and rules in the field of ambient intelligence and context-awareness. To show how useful ATML is, we briefly introduce its current implementation in BackHome, an EU project concerning physical and social autonomy of people with disabilities.

1 Introduction

Today's appliances have successfully become integrated to such an extent that we use them without consciously thinking about them. Computing devices have transitioned in this past half a century from big mainframes to small chips that can be embedded in a variety of places ranging from communication appliances (e.g. mobile phones) to simple applications (e.g. weather sensors). In this setting, various industries have silently distribute computing devices all around us, often without us even noticing, both in public spaces and in our more private surroundings with small amounts of intelligence providing them autonomy to perform small-scale decisions to modify the environment.

The advances in the miniaturization of electronics allow purchasing sensors, actuators and processing units at very affordable prices [4] favouring the inclusion of these elements in our normal day activities and houses. This novel approach can be networked with the coordination of highly intelligent software applications to understand the events and relevant context of a specific environment. This knowledge enables to take sensible decisions in real-time or a posteriori to adapt the features of applications to the real setting in which they are.

These elements are to be coordinated by intelligent systems that integrate the available resources to provide an intelligent environment. This confluence of topics has led to the introduction of the area of Ambient Intelligence (AmI) that is defined as a digital environment that proactively, but sensibly, supports people in their daily lives [3]. AmI is aligned with the concept of the disappearing computer [29][25]: "*Technologies that disappear weave themselves into everyday life to the point that they are indistinguishable*".

Networks, sensors, human-computer interfaces, pervasive computing and artificial intelligence are all relevant but none of them conceptually fully cover AmI. It is, though, AmI which brings all these together to provide flexible and intelligent services to users acting in their environments. Indeed, AmI relies in the application of artificial intelligence techniques to provide added value services to the end-users.

Ambient Assisted Living (AAL) fosters the provision of these intelligent environments for the independent or more autonomous living of people with disabilities, via the seamless integration of info-communication technologies in homes and residences [18]. Assistive Technologies (ATs) are becoming of crucial importance in this AAL scenario as they are often used to provide support at home to engage and promote independence [6].

Among other application fields, let us recall here the importance that AmI takes on personalized assistance of people with disabilities and the improvement of AT to their impairments and needs. In particular, AmI helps monitoring the context features and behaviour and facilitates the control on environmental appliances (e.g. lights, windows). Those systems combine all the above features with the understanding of situations in which to perform actions pro-actively, such as triggering emergency alarms (e.g. fire, gas leak).

In order to adapt the AT and to react properly to the different situations that a user may need, several AmI techniques should be developed. One approach can be the use of rules which should trigger actions, or suggestions to the user, in order to react properly to the situation. To fulfill the need to express these rules we present, in this paper, a suitable language called ATML (Ambient intelligence Trigger Markup Language). Based on RuleML [19], ATML is aimed at defining and describing actions and rules in the field of AmI and context-awareness. The underlying idea is to adapt the RuleML to the AmI systems and applications in the area of ATs.

As a practical demonstration of the usefulness of ATML, we present its current implementation in BackHome[1], an EU project concerning physical and social autonomy of people with disabilities, by using mainly Brain-Neural Computer Interfaces (BNCIs) and integrating other assistive technologies as well.

The paper is organized as follows: Section 2 briefly introduces AmI. Section 3 discusses the representation of rules and actions in the field of AmI. Section 4 presents the proposed language and its fundamentals. Section 5 illustrates the benefits of ATML in BackHome through an example of usage. Section 6 ends the paper with conclusions.

2 Ambient Intelligence

According to Augusto and McCullagh [4] we may see Ambient Intelligence (AmI) as the confluence of Pervasive Computing (aka, Ubiquitous Computing), Artificial Intelligence (AI), Human Computer Interaction (HCI), Sensors, and Networks. First, an AmI system pervasively senses the environment by relying on a network of sensors. The gathered information is, then, processed by AI techniques to provide suitable actions to be performed on the environment through controllers and/or specialized HCI.

---

[1] www.Backhome-FP7.eu

According to [10], the AmI is placed in the confluence of a multi-disciplinary and heterogeneous ecosystem. This position allows the AmI applications to get the information of the surroundings, actuate and change the environment, the different human interfaces available, as well as apply some reasoning techniques. The conjunction of all these fields is used by the AmI systems always respecting the privacy of the user. Following this description, a system incorporates AmI principles if the following characteristics are met: *Sensitive*, AmI systems have to incorporate the ability to perceive their immediate surroundings; *Responsive*, AmI systems have to be able to react in front of the context occurring; *Adaptive*, AmI systems are to be flexible enough to accommodate the responses along the time; *Transparent*, AmI systems have to be designed to be unobtrusive; *Ubiquitous*, AmI systems have to be concealed so as to minimize the impact of bulky and tedious appliances; and *Intelligent*, AmI systems have to incorporate intelligent algorithms to react in front of specific scenarios. These principles can be applied to several fields ranging from education to health or security. As we have already commented, among others, this work focusses on ATs. The capability of AmI techniques for recognizing activities [5] [20], monitoring diet and exercise [15], and detecting changes or anomalies [11] support the key idea of providing help to individuals with cognitive or physical impairments. For instance, AmI techniques can be used to provide reminders of normal tasks or the step sequences to properly realize and complete these tasks. For those with physical limitations, automation and inclusion of AI to their home and work environment may become a response for independent living at home [30].

Several artefacts and items in a house can be enriched with sensors to gather information about their use and in some cases even to act independently without human intervention. Some examples of such devices are white goods (e.g., oven and fridge), household items (e.g., taps, bed and sofa) and temperature handling devices (e.g., air conditioning and radiators). Applying AmI in this scenario may: (a) increase safety (e.g., by monitoring lifestyle patterns or the latest activities and providing assistance when a possibly harmful situation is developing) and/or (b) improve comfort (e.g., by adjusting temperature automatically).

In addition, AmI allows the home itself to take decisions regarding its state and interactions with its residents. There are several physical smart homes that have been designed with this theme in mind. For instance, the MavHome project treats an environment as an intelligent agent, which perceives it using sensors and acts on the environment using powerline controllers [13].

3 Ambient Intelligence and Triggered Actions

Being implemented in real-world environments, AmI involves problems such as incompleteness and uncertainty of the information available about the user and the environment. In fact, we generally deal with information that might be in some way correct, in somewhere incorrect, and in some part missing. Thus, an elaborated reasoning process that deals with those information drawbacks might be performed to successfully define an accurate knowledge representation. To this end, AmI relies on the context as a model of the current situation of the user and its immediate environment [14].

In order to use context effectively, we must understand what context is and how it can be used. A precise notion of context is essential in an intelligent environment, even more in assistive applications since an understanding of how context can be used helps application designers to determine the context-aware behaviours necessary to support in their applications. Nevertheless, the context is not a static but a dynamic concept composed of entities like people, devices, locations or even computing applications which, in their turn, are characterized by attributes. Once these concepts are properly integrated into the design, the system is entitled to take the necessary actions according to different combinations of these entities using different techniques such as rules or analysis of this information provided by the context.

The process of understanding the context is not explicit and thus not trivial; it depends on previous knowledge. Experience provides means to classify and highlight specific situations and to relate them with the received stimuli of the sensory system. It is supposed that same situations are promoted by the same stimulus, and those implications resultant from those situations serve as an extension on the recall of experience [23]. Sensing of location, environmental conditions and capturing explicit interactions are the general inputs for context extraction. Nevertheless, it is desirable for smart environments to interpret the available information by perceptual means similar to those of humans [24], for this reason it is necessary to describe the situations in a human readable format. The ATML fulfil this point offering the possibility to describe all the situations where the sensory system is involved and the actions that can be interesting to trigger.

Current implementations of AmI in the field of SmartHome and ATs are focused on providing modifiable assistance according to the user context, the so-called personalized assistance. The more frequent implementations are based on machine learning algorithms and intelligence systems. Nevertheless, this is not the only possible approach: there exist systems based on the use of rule based engines that determine the actions to be triggered by the system. Some examples can be find at the literature, for instance Acampora and Loia [1] present a distributed AmI system, based on agents, communication protocols (TCP/IP) and Fuzzy Logic [31] using as a language for description of knowledge and rules, the Fuzzy Markup Language [2]. In our project, it is not necessary the use of distributed logic and agents.

On the other side, the platform DOAPAmI [17] uses a Domain Specific Language (DSL) in order to define the complete platform including services, sensors, profiles of physical platforms where executing the system. This DSL also allows the specification of rules for the different situations, but focused on the proper running of the system not on the user needs and preferences. Papamarkos et al. [21] present an Even-Condition-Action centred approach based on RuleML [19] and focused on Semantic Web, far from the focus of our objective. Other example of use of XML languages in order to define rules based on the context was presented by Schmidt [22] introducing an extension of the Standard Generalized Markup Language (SGML) defined in [7] [8] is introduced. This extension allows the language to define triggers and represent the context to improve the user interface in small devices. So the context represented by this proposal is only related to the attributes and characteristics of the device where the application shows its user interface.

4 The ATML Language

As we have already commented in the previous section, some applications use rules engines to define the actions to be triggered in several situations. Although, these rules can be defined in several ways, frequently are platform dependent and not based on standards. In order to establish a platform-independent and flexible definition of these rules, and to specify conditions and actions to be triggered (hereinafter, triggers), we propose to use a human readable XML-based language called AmI Triggering Markup Language (ATML). This approach allows exporting the rules definitions to any AmI system and reaching the same status on the configuration of the intelligence.

ATML is compliant with the RuleML. In fact, most of ATML definition relies on the RuleML definitions. RuleML is an initiative part of the research community's effort to develop the Semantic Web. RuleML is, at its heart, an XML syntax for rule knowledge representation that is interoperable among major commercial rule systems. RuleML is based on a fundamental rule knowledge representation, declarative logic programs, which expressively extend ordinary logic programs with features for prioritized conflict handling and procedural attachments to perform actions and queries.

Let us consider an example of trigger definition:

```
<Trigger>
  <name>Activate Managed Ambience</name>
  <Properties>
    <occurrence>Continuous</occurrence>
    <enabled>true</enabled>
  </Properties>
  <Implies>
    . . .
  </Implies>
</Trigger>
```

As you can see, a trigger in ATML is firstly described by its *name*, in order to distinguish it from the overall set of triggers. This is very helpful for both user and designers to perform the import/export operations as rules can be easily identifiable. After assigning the name, the set of *properties* has to be defined. Properties incorporate different attributes helping to understand how to interact with the rule. As an example, let us consider an *occurrence* property that provides the time frame in which the rule needs to be evaluated. In the code of the example reported above, we can also find the enabled property that indicates to the system if that rule is active and we have to check and react, if needed.

After these two sections the rules define the implications of the action. That section includes two main sub-sections: *head* and *body*:

```
<Implies>
    <head>
      <TriggerAction>
        <op> <rel>open</rel> </op>
        <who>trigger</who>
        <device>CUR_DD_001</device>
      </TriggerAction>
```

```
  </head>
  <body>
    <And>
      <Atom>
        <op> <rel>greater than</rel> </op>
        <var>TEMP_ENV_001</var>
        <value>28</value>
      </Atom>
      <Atom>
        <op><rel>lower or equal than</rel></op>
        <var>TEMP_HVAC_001</var>
        <value>23</value>
      </Atom>
    </And>
  </body>
</Implies>
```

Head is used to define the actions (*TriggerAction*) to be executed whenever the condition of the rule is met. Every *TriggerAction* is defined by a single action to be performed when a given condition (defined in the body section) is met. To define a *TriggerAction*, three different tags are necessary:

- *op*, which indicates the command to be performed on the target, in the example the *op* indicates that the value of the variable $TEMP_ENV$_001 must be greater than *28*;
- *who*, describing which system performs the operation (in the example the trigger engine defined by the identifier trigger);
- *device*, which defines the targeted appliance (in the example, the device with identifier *CUR_DD*_001).
- *body*, which expresses the condition of the rule by means of comparison and logic operation. The rule defined in the example contains two conditions, the first one related with the value of the variable $TEMP_ENV$_001 and the second one with the value of $TEMP_HVAC$_001.

A single comparison operation, called *atom*, includes an operand, namely *equal*, *not equal*, *greater than*, *lower than* and *contains*. To concatenate and combine different atoms, the three classical logic operands (*and*, *or*, and *not*) are allowed. It is remarkable to say that logic operations are allowed to be nested, that is, a logic operation may contain inner logic operands.

5 Real Case of ATML Use: BackHome Project

ATML is currently used in the BackHome project. The project BackHome, an European initiative funded by the FP7 program, is willing to play a role in empowering the end-users to become more autonomous and independent in their activities of daily life by means of a novel concept in ATs. BackHome is about boosting physical and social autonomy of people with disabilities taking a broad approach and is aimed at supporting the transition from institutional care to home, post rehabilitation and discharge

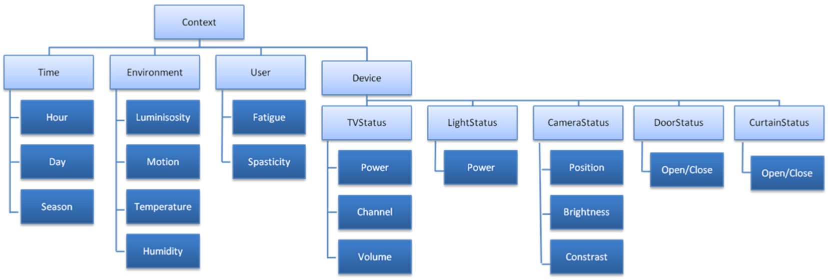


Fig. 1. Context definition in BackHome.

[12]. The project offers as an innovative AAL platform, a sensor-based Telemonotoring and Home Support System (TMHSS), devoted to help the user to be more independent by providing an AAL environment improved with the AmI principles [28] [27]. BackHome also takes care of the isolation problems often associated to disability and, therefore, incorporates eInclusion with the possibility to interact with the most popular social networks such as Facebook or Twitter, and other Internet related services, like Web browsing and e-mail. Finally, the system has added value features in the field of telemedicine: cognitive rehabilitation and quality of life automatic assessment [26]. Within the project, the achievement of these objectives strongly relies on the usage of BNCIs as principal interface but integrating other assistive technologies as well. BNCIs rely on the direct measures of brain activity complemented with other technologies. This project is a perfect environment to test the flexibility and scalability of ATML.

5.1 Ambient Intelligence in BackHome

Suitable AmI features are provided in BackHome. In particular, the sensor-based TMHSS is aimed at acquiring contextual information through data coming from sources of different nature: BNCI system that allows monitoring ElectroEncephalo-Gram (EEG), ElectroOculoGram (EOG), and ElectroMyoGram (EMG); wearable, physiological, and biometric sensors, such as ElectroCardioGram (ECG), heart-rate sensor, respiration-rate sensor, Galvanic Skin Response (GSR) sensor, EMG switches, and inertial sensors (e.g., accelerometer, gyrocompass, and magnetometer); environmental sensors (e.g., temperature and humidity sensors); SmartHome devices (e.g., wheelchairs, lights, TVs, doors, windows and shutters); devices that allow interaction activities (e.g., a desktop PC); as well as devices to perform rehabilitation tasks (e.g., a robot).

As part of the design of BackHome, it was mandatory to devise a knowledge representation of the context which needs to be captured, and stored, from the sources presented in the previous list. The outcome of the context formalization is depicted in Figure 1 in which the different values of the environment together with parameters of the user (e.g. fatigue) are presented. This definition of the context incorporates different categories taken into account when evaluating the context:

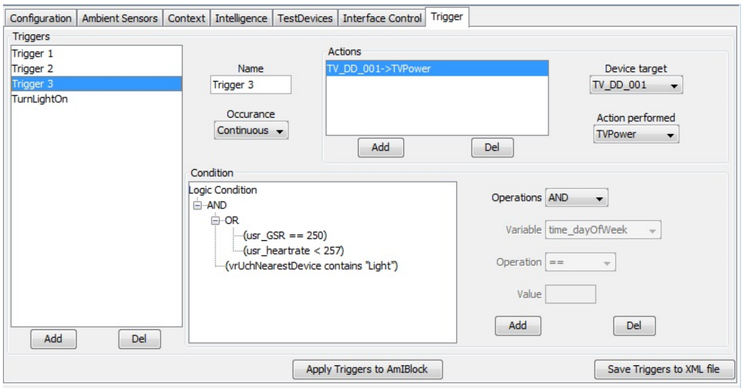
1. Time Variables: representing the current moment taking place.

2. Environmental Variables: these variables refer to those measures that give direct information of the context such as luminosity or motion.
3. User Variables: providing information about physiological measurements (e.g. fatigue and spasticity).
4. Device Variables: these are the variables referred to the status of the devices controlled by the AmI-enabled system.

BackHome takes advantage of AmI to provide advanced assistance through two main assistive services based on AmI: Personalized Adaptation and Proactive Performance. The *Personalized Adaptation* is an approach based on the user's preferences and habits. This method learns the habits from the user applying machine learning techniques in order to infer the behavioral patterns of the system. Currently, it includes the proposals of the BrainAble [16] project based on AdaBoost with C45 as a weak algorithm [9]. The *Proactive Performance* is a module that applies rules user defined which should be activated for a particular situation without an explicit request from the user. When the environment arrives to a given status, which matches with any of the pre-established rules, the system reacts consequently. Currently, the system updates the interface suggesting to the user some actions that might be interesting for him/her.

5.2 Use-Case Scenario

In the current implementation of the BackHome project, some proactive context-trigger actions have been designed and developed. Context-Trigger actions are clear examples of the proactive nature of AmI: whenever a rule condition is met the corresponding action is triggered.



Trigger 1
Trigger 2
Trigger 3
TurnLightOn

Name: Trigger 3
Occurance: Continuous
Actions: TV_DD_001->TVPower
Device target: TV_DD_001
Action performed: TVPower

```
Logic Condition
⊟-AND
   ⊟-OR
      -(usr_GSR == 250)
      -(usr_heartrate < 257)
   -(vrUchNearestDevice contains "Light")
```

Operations: AND
Variable: time_dayOfWeek
Operation: ==
Value:

Fig. 2. Trigger Definition User Interface in BackHome.

These rules, expressed in ATML, are configurable by the end-user with a dedicated interface (see Figure 2), which facilitates the creation of the rules which are stored in a separated file. As such, the rules can be defined as portable across different AmI-enabled systems. As an example, let us consider the next piece of code that presents a rule expressed in ATML:

```
<Trigger>
  <name>Activate Managed Ambience</name>
  <Properties>
    <occurrence>Continuous</occurrence>
    <enabled>true</enabled>
  </Properties>
  <Implies>
    <head>
      <TriggerAction>
        <op> <rel>open</rel> </op>
        <who>trigger</who>
        <device>CUR_DD_001</device>
      </TriggerAction>
      <TriggerAction>
        <op> <rel>PowerOn</rel> </op>
        <who>trigger</who>
        <device>HVAC_DD_001</device>
      </TriggerAction>
    </head>
    <body>
      <And>
        <Atom>
          <op> <rel>greater than</rel> </op>
          <var>TEMP_ENV_001</var>
          <value>28</value>
        </Atom>
        <Atom>
          <op><rel>lower or equal than</rel></op>
          <var>TEMP_HVAC_001</var>
          <value>23</value>
        </Atom>
        <Neg>
          <Atom>
            <op> <rel>equal</rel> </op>
            <var>MOTION_ENV_001</var>
            <value>true</value>
          </Atom>
        </Neg>
      </And>
    </body>
  </Implies>
</Trigger>
```

As discussed in Section 4, first, the name is defined and then the required properties (e.g., *occurrence* and *enabled*). The rule itself is defined under the tag *Implies* and is divided in two sections, *head* and *body*. The *head* contains the different actions (called *TriggerAction*) to be made once the condition of the body is met. Each *TriggerAction* defines who executes the action, on which device it is performed, and under the tag *op/rel* the command to execute is specified. In this case, the trigger will execute *open* on the device *CUR_DD_001*, and *PowerOn* on *HVAC_DD_001*. The condition of the trigger is defined in the body section, where different logic operands called *Atom* are linked by logic operations, *And*, *Or* and *Neg*. The *Atom* contains the variable, the value, and like with the actions, under the tag *op/rel* the operand to them.

As a result, the rule of the example opens the curtains and turns on the air conditioning when the condition of evaluating the environment is met. The specific condition evaluated for this rule is based on the measure of the environmental temperature, the target temperature of the HVAC and the presence of someone in the surroundings.

6 Conclusion

The latest progress in three domains, i.e., microelectronics, communication and networking technologies, as well as intelligent agents and user interfaces has given rise to the idea of ambient intelligence. It provides added value services by combining the features of the different appliances creating smart environments. This paper presents a novel markup language called Ambient intelligence Triggering MarkUp Language (ATML). This new language, based on RuleML, is aimed at describing in a flexible and scalable way all the possible situations where an AmI system can react to fit the needs and preferences of the users in different situations. In order to validate the design of the ATML and the robustness of the concept it was incorporated to the BackHome project, an European-funded FP7 project, aimed at creating assisting environments for people with severe impairments. It is an ongoing project where the services and devices to be included are growing and changing with the evolution of the project and it will be a real test of ATML. Also, the users tests (technicians or not) will be a valuable input that will help to improve the definition and will be presented in the next steps of the project.

Acknowledgement

The research leading to these results has received funding from the European Communitys, Seventh Framework Programme FP7/2007-2013, BackHome project grant agreement n. 288566.

References

1. Acampora, G., Loia, V.: Using fml and fuzzy technology in adaptive ambient intelligence environments. International Journal of Computational Intelligence Research **1**(1), 171–182 (2005)
2. Acampora, G., Loia, V.: Using fuzzy technology in ambient intelligence environments. In: Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on, pp. 465–470. IEEE (2005)
3. Augusto, J.: Ambient intelligence: Basic concepts and applications. In: J. Filipe, B. Shishkov, M. Helfert (eds.) Software and Data Technologies, *Communications in Computer and Information Science*, vol. 10, pp. 16–26. Springer Berlin Heidelberg (2008)
4. Augusto, J.C., Nakashima, H., Aghajan, H.: Ambient intelligence and smart environments: A state of the art. In: in Handbook of Ambient Intelligence and Smart Environments, pp. 3–31. Springer (2010)

5. Barger, T., Brown, D., Alwan, M.: Health status monitoring through analysis of behavioral patterns. In: 8th congress of the Italian Association for Artificial Intelligence (AI*IA) on Ambient Intelligence, pp. 22–27. Springer-Verlag (2003)
6. Bechtold, U., Sotoudeh, M.: Assistive technologies: Their development from a technology assessment perspective. Gerontechnology **11**(4), 521–533 (2013)
7. Brown, P., Bovey, J., Chen, X.: Context-aware applications: from the laboratory to the marketplace. Personal Communications, IEEE **4**(5), 58–64 (1997)
8. Brown, P.J.: The Stick-e Document: a Framework for Creating Context-aware Applications. In: Proceedings of EP'96, Palo Alto, pp. 259–272 (1996)
9. Casale, P., Fernández, J.M., Rafael, X., Torrellas, S., Ratsgoo, M., Miralles, F.: Enhanching user experience with brain neural computer interfaces in smart home environments. In: 8th IEEE International Conference of Intelligent Environments 2012, INTENV12 (2012)
10. Cook, D.J., Augusto, J.C., Jakkula, V.R.: Ambient intelligence: Technologies, applications, and opportunities (2007)
11. Cook, D.J., Youngblood, G.M., Jain, G.: Algorithms for smart spaces. Technology for Aging, Disability and Independence: Computer and Engineering for Design and Applications, Wiley (2008)
12. Daly, J., Armstrong, E., Miralles, F., Vargiu, E., Müller-Putz, G., Hintermller, C., Guger, C., Kuebler, A., Martin, S.: Backhome: Brain-neural-computer interfaces on track to home. In: RAatE 2012 - Recent Advances in Assistive Technology & Engineering (2012)
13. Das, S.K., Cook, D.J.: Health monitoring in an agent-based smart home. In: In Proceedings of the International Conference on Smart Homes and Health Telematics (ICOST, pp. 3–14. IOS Press (2004)
14. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. **5**(1), 4–7 (2001)
15. Farringdon, J., Nashold, S.: Continuous body monitoring. In: Y. Cai (ed.) Ambient Intelligence for Scientific Discovery, *Lecture Notes in Computer Science*, vol. 3345, pp. 202–223. Springer Berlin Heidelberg (2005)
16. Fernández, J.M., Dauwalder, S., Torrellas, S., Faller, J., Scherer, R., Omedas, P., Verschure, P., Espinosa, A., Guger, C., Carmichael, C., Costa, U., Opisso, E., Tormos, J., Miralles, F.: Connecting the disabled to their physical and social world: The BrainAble experience. In: TOBI Workshop IV Practical Brain-Computer Interfaces for End-Users: Progress and Challenges (2013)
17. Fuentes, L., Jimenez, D., Pinto, M.: An ambient intelligent language for dynamic adaptation. In: Proceedings of Object Technology for Ambient Intelligence workshop (OT4AmI), Glasgow, Uk (2005)
18. Gentry, T.: Smart home technologies for people with cognitive impairment: An affordable, rehabilitative approach. In: Handbook of Ambient Assisted Living, pp. 535–548 (2012)
19. Grosof, B.N.: Representing e-commerce rules via situated courteous logic programs in RuleML (2003)
20. Nambu, M., Nakajima, K., Noshiro, M., Tamura, T.: An algorithm for the automatic detection of health conditions. Engineering in Medicine and Biology Magazine, IEEE **24**(4), 38–42 (2005)
21. Papamarkos G., P.A., Wood, P.T.: Event-Condition-Action Rule Languages for the Semantic Web. In: Proceedings of Workshop on Semantic Web and Databases, Palo Alto, pp. 309–327 (2003)
22. Schmidt, A.: Implicit human computer interaction through context. Personal Technologies **4**(2-3), 191–199 (2000)
23. Schmidt, A.: Ubiquitous computing-computing in context. Ph.D. thesis, Lancaster University (2003)
24. Schmidt, A.: Interactive context-aware systems interacting with ambient intelligence. Ambient intelligence (Part 3), 159–178 (2005)

25. Streitz, N.: From human computer interaction to human?environment interaction: Ambient intelligence and the disappearing computer. In: C. Stephanidis, M. Pieper (eds.) Universal Access in Ambient Intelligence Environments, *Lecture Notes in Computer Science*, vol. 4397, pp. 3–13. Springer Berlin Heidelberg (2007)
26. Vargiu, E., Fernández, J.M., Miralles, F.: Context-aware based quality of life telemonitoring. In: Distributed Systems and Applications of Information Filtering and Retrieval. DART 2012: Revised and Invited Papers. C. Lai, A. Giuliani and G. Semeraro (eds.) (inpress)
27. Vargiu, E., Fernández, J.M., Torrellas, S., Dauwalder, S., Solà, M., Miralles, F.: A sensor-based telemonitoring and home support system to improve quality of life through bnci. In: 12th European AAATE Conference (2013)
28. Vargiu, E., Miralles, F., Martin, S., Markey, D.: BackHome: Assisting and telemonitoring people with disabilities. In: RAatE 2012 - Recent Advances in Assistive Technology & Engineering (2012)
29. Weiser, M.: Some computer science issues in ubiquitous computing. Commun. ACM **36**(7), 75–84 (1993)
30. Youngblood, G.M., Cook, D.J., Holder, L.B.: A learning architecture for automating the intelligent environment. In: Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3, IAAI'05, pp. 1576–1581. AAAI Press (2005)
31. Zadeh, L.A.: Fuzzy sets. Information and control **8**(3), 338–353 (1965)