Proactive Assistance Within Ambient Environment

Towards intelligent agent server that anticipate and provide users' needs

Hajer Sassi
LIFL Laboratory – University of Lille 1
xBrainLab-USILINK
59655 Villeneuve d'Ascq Cedex – France
hajer.sassi@ed.univ-lille1.fr

José Rouillard
LIFL Laboratory – University of Lille 1
59655 Villeneuve d'Ascq Cedex –France
jose.rouillard@univ-lille1.fr

***Abstract*—User needs are expanding and becoming more and more complex with the emergence of newly adopted technologies. As a result, the convergence of smart devices, having the capability to communicate as well as sharing information and ensuring user need satisfaction, leads to profoundly change the way we interact with our environment. They should provide an adaptive assistance in both reactive and proactive mode and new communication methods focused on multimodal and multichannel interfaces. However, most of existing context-aware systems have extremely tight coupling between applications' semantic and sensor's details. So, the objective of our research is to implement an approach which can support the ability to reuse sensors and to evolve existing applications to use new context types. In this paper, we illustrate our approach for proactive intelligent assistance and we describe our architecture based on three principal layers. These layers are designed in order to build applications which can increase the welfare of the user situated in intelligent environment.**

Keywords-Intelligent Interfaces; ubiquitous computing; human-computer interaction; proactive assistance; multimodal interfaces; multi-channel interfaces.

I. INTRODUCTION

Ambient Intelligence (AmI) aims at insuring the comfort of users in theirs daily tasks based on context information. In our life, we often repeat usually the same tasks. For example, seeing the weather forecast before going outside, consulting agenda to verify appointments, control children tasks, etc.

User searches to delegate a majority of these daily tasks to her intelligent environment in order to decrease her responsibilities. As a consequence, she wants to satisfy her needs without any explicit intervention through the capability of the intelligent environment to perceive user's personal environment in order to resolve her daily tasks. Therefore, AmI follows the goals of Ubiquitous Computing, a paradigm that was first suggested by Weiser in the early 1990s. His vision was to increase the welfare of a user situated in a computer everywhere environment by supporting human assistance in an intimate way [1].

One research domain that requires the computer- everywhere model of ubiquitous computing is that of the "intelligent environment" [2]. In this domain, a wide range of simple information (e.g., light sensor, audio/video sensor, temperature sensor, google calendar, information from the web, etc.) and composite information (e.g., presence sensor and preferences of users) can be collected from heterogeneous sensors in order to determine automatically users' needs based on their context's information.

In this context-aware domain, many ad hoc systems exist in order to be able to perform an adaptive assistance. However, these systems present two main limits: the difficulty to develop due to the requirements of dealing directly with sensors and the difficulty to evolve because the application semantics are not separated from the sensor details (also rules).

So, building applications, depending on context-aware which can support reuse sensors and new context types stays hard tasks, which covered many context-aware features.

As said by Dey in his thesis "context has the following properties that lead to the difficulty in use "[3]:

- Context is acquired from non-traditional devices (i.e., not mice and keyboards), with which we have limited experience. For example, tracking the location of people or detecting their presence may require Active Badge devices [4], floor-embedded presence sensors [5] and video image processing…
- Context must be abstracted to make sense to the application; Active Badges provide IDs, which must be abstracted into user names and locations.
- Context may be acquired from multiple distributed and heterogeneous sources. Detecting the presence of user in a room reliably may require combining the results of several techniques such as image processing, audio processing, floor-embedded pressure, etc.
- Context is dynamic; changes in the environment must be detected in real time and applications must change behavior to constant changes.
- Context information history, as shown by context-based retrieval applications [6, 7]; context history can be used to recognize user's activities and to fully exploit the richness of context information.

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-224-0

These difficulties prevent to build context-aware applications the ability to support reuse of sensing technologies in new applications and evolution to use new context in new ways. In this paper, we present a system which can support new context types and evolve dynamically according to user's preferences.

This document is organized as the following: First, we describe some previous context-aware applications. Second, we present our research problematic and how we proceeded to resolve it. Third, we describe our proposed architecture and an illustrative example. Lastly, we state our future works and conclusions.

II. RELATED WORK

Weiser's vision in his article "The Computer for the 21st Century" [8] is to serve people's daily tasks through an intelligent environment which should acts invisibly and unobtrusively in the background and freeing users from tedious routine tasks in order to reduce users' responsibilities.

Ubiquitous computing aims to integrate each intelligent entity that can be identified and provide information about user's context such as sensors which can provide immediate information according to user's situation. Thus, user's goals and desires can be anticipated from the interaction context which is defined by Dey [9] 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".

Many projects are developed around context aware. In 1998, Coen created the Intelligent Room MIT [2]. This is a conference room equipped with 12 cameras, 2 video projectors, display devices, microphones and loudspeakers. The goal of this room is to interact with different form of modality. In the field of home automation, Mozer created the Adaptive House [10] which is an intelligent home equipped with 75 sensors in order to provide information such as temperature, ambient light, door's and window's situations. Adaptive house has also the capability to manage energy. Microsoft has also created the project named EasyLiving [11], which calculates user's position and propose service depending on his position. Each of these projects illustrates convincing results from different uses cases which proposed. On the other hand ubiquitous computing aims to change ordinary interfaces by intelligent interfaces in order to let user feeling natural communication on many levels (complexity, size, and portability). In the 70's, the technology-driven focus on interfaces was slowly changed and in the 80's the new field of Human-Machine Interaction (HCI) appeared. With the appearance of new technologies such as data mining, machine learning, speech/voice recognition, facial recognition and omnipresent computing, the basic technology based on ordinary interfaces can difficultly use. Consequently, the interaction human-machine should change the way we interact with the ambient environment by providing new intelligent interfaces able to adapt its behavior according to user's situation. Around 1994 until 1996, intelligent agents, practical speech recognition and natural language applications appeared. However, since then intelligent user interfaces evolve slowly. On the other hand, implementing and maintaining interfaces, which should be at the same time proactive and intelligent, is still far from easy.

III. RESEARCH QUESTIONS

The inference of user's requirements or proactive assistance is a very delicate problem, which we have chosen to explore through the following question, "proactive assistance: why, when and how to use it?"

The first question "Why" has for objective to search how can proactive assistance reduces user's responsibilities. As we know, we have many boring routine tasks and we search to delegate more of them to our intelligent environment in order to have more time for other more complex tasks. Thus, by the capability of the intelligent environment to perceive environment and user's habits, system based on proactive assistance could anticipate users' needs without any explicit request. The second question "When" is devoted to determine the adequate time; when intelligent environment decide to communicate user's need. Once intelligent environment determines user's needs, it should interpret user's real situation in order to decide if service can be communicated. However, the last question "How" is interested to adapt the way we interact with our environment. Depending on context, our system should find the adequate modality (text, speech and gesture) and channel (Internet and phone channel) according to user's situation.

IV. PROPOSED ARCHITECTURE

To respond to our research questions, we have chosen to implement an architecture based on three principals layers (see Figure 1), which can communicate between them throw two different modes: the push and the pull modes, which are used, in our system, to provide reactive and proactive interactions. Each layer has for role to provide a service to the layer above in order to resolve user's needs. However, the mechanism of adaptation is shared between the second and third layer.

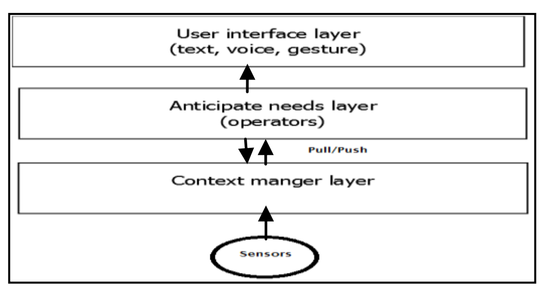

Figure 1. Context model's architecture

A. *Context Manager Layer*

To build systems able to act differently according to context awareness, intelligent environment should perceive and control sensors networks regularly through the "context manager layer". This layer should communicate with heterogeneous sources in order to collect information and register them in the database [12, 13, 14]. This layer is based on context provider and context repository. It controls the behavior of sensors and saves new issues values (static, temporary and dynamic information) in context repository. It should also communicate directly with the second layer in order to publish information even before context repository registers information in database for later use. An example of sensors that we used to collect information is a Radio Frequency Identification (RFID) reader accompanied with RFID tags. When RFID reader detects an RFID tag (see Figure 2 and Figure 4), it firstly determines the user's name in order to salute him/her (see Figure 3 and Figure 5) and secondly calculates the number of persons at home.

Figure 2. Bob's RFID tag.

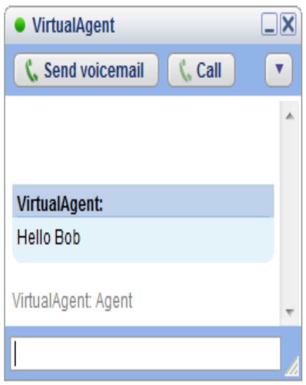

Figure 3. Agent detects Bob's RFID tag and welcomes him on Gtalk.

Figure 4. Marc's RFID tag.

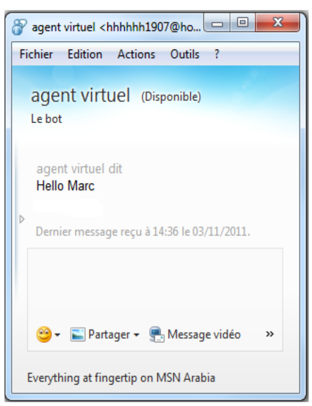

Figure 5. Agent detects Marc'sRFID tag and welcomes him on MSN.

To gather user's information (current activity and preferences), we have chosen to ask some questions according to the user's context as follows:

Firstly, our system have not any information about user, it learns user's information by asking a set of questions which are triggered depending on the context.

1) Case one: we create an xml file which contains some questions grouped by theme.

```
<?xml version="1.0" encoding="utf-8"?>
<questionnaire id="1" theme="Tv">
  <question format="input" nom="FrequenceTv" dataType="xsd:string">
    <ennonce>Do you often watch TV? (yes, no, sometimes)</ennonce>
  </question>
  <question format="input"  nom="SeriePreferee" dataType="xsd:string">
    <ennonce>What is your favorite television series? (put "0" if you do not
have)?</ennonce>
  </question>
  <question format="input"  nom="EmissionRegardee" dataType="xsd:string">
    <ennonce>Which type of emission you like watching? (put "0" if you do not
have)?</ennonce>
  </question>
  <question format="input"  nom="EmissionPreferee" dataType="xsd:string">
    <ennonce>What is your favorite broadcast? (put "0" if you do not
have)?</ennonce>
  </question>
</questionnaire>
```

Figure 6. TV questionnaire

According to context, system tries to collect user's knowledge. It triggers a questionnaire (see Figure 6) depending on user's situation (e.g., user is watching TV), and it stores responses in the database, thanks to a natural language multimodal dialog. As we can see in Figure 6, we have chosen four questions about user's frequency of watching TV, her favorite series, her favorite category of emission and its title. Based on answers given by user, system will infer new decision related on her preferences such as send notification when program TV contains user's favorite category of emission.

2) Case two: User can also enter data through a software entity (e.g., Website, Google calendar, Face- book, etc.) and provide access to system which can use this software in order to more help user. This layer distinguishes three types of information: the static information, the temporary information and the dynamic information. Static information remains unchanged during the process of learning (e.g., name, age, etc.). Temporary information can be sometimes changed (e.g., preferences, taste, etc.). However dynamic information changes frequently (e.g., location, mood). All these types of information are stored in a database in order to be used later.

B. *Anticipate Needs Layer*

In our research, we are based on "context manager layer" in order to anticipate user's services. In this layer, we try to exploit stored data context manager by associating a set of adaptive operators. Actually, we distinguish three types of operators:

1) Conversion operator: the context manager stores a data in initial format, after that "anticipate needs layer" tries to adapt this format in order to associate a meaning manageable by the system. For example: when temperature sensor sends the raw data "2", the conversion operator interprets this value as "it's cold" or "it's hot", according to the real situation of the user.

2) Extract operator: in many cases our system integrates logical sensors such as google calendar, RSS stream, etc. However these sources provide imprecise information. Therefore, the mission of this operator should extract only relevant information. Example: extract just the minute from the current time.

3) Coupling operator: in other cases, system should aggregate various and heterogeneous (logical and/or physical) data. Thus we propose a coupling operator which tries to collect many data in order to "understand" non-trivial situations. For example detecting the location of users in a living room requires gathering information from multiple sensors throughout the intelligent home. It should also, in many cases, combine the results of several techniques such as image processing, audio processing, floor-embedded pressure sensors, etc., in order to provide valid information.

C. User Interface Layer

In a ubiquitous environment, the behavior of services does not depend just on explicit user interaction but also on environment's perceptions. Combing these two sources of information, system can better respond to user's expectations. Our system has to provide an adaptive way of interaction according to the user's situations. The "user interface layer" should be able to define the context and choose the best way to interact by selecting the appropriate modalities and channels.

Our work tackles the ability of ambient computing to permit context-aware interactions between humans and machines. To do so, we rely on the use of multimodal and multi-channel interfaces in various fields of application such as coaching, learning, health care diagnosis, or home automation.

Using a multimodal approach allows users to employ different kinds of modalities (keyboard/mouse, voice, gesture, etc.) in order to interact with a system. The synergistic multimodality is quite natural for humans, but very difficult to implement, mainly because it requires some sharp synchronizations. Fusion mechanisms are used to interprets inputs (from user to machine) while fission mechanisms are used to generate outputs (from machine to user).

Using a multi-channel approach allows users to interact with several channels choosing the most appropriate one in order to exchange with an entity. Such channels could be, for instance, plain paper, e-mail, phone, web site. For the moment, our prototype supports text, speech and gesture as inputs and text and speech as outputs. Once system anticipates user's need through the second layer, "user interface layer" communicates with "context manager layer" in order to check information related to user's situation (e.g., user location, user status, etc.)

In our approach, the influence of the context appears in both second and third layers. The context is used, firstly, to anticipate user's needs and secondly to find the appropriate way of interaction depending on user's situation.

V. SCENARIO

In order to ensure the communication with user anyway she is, we decided to work on multi-channel interfaces and we have chosen to use two types of channels which are: internet channel and phone channel [15].

A. Internet Channel

To demonstrate the identified requirements, a scenario is given in the following. The scenario is about Mr. Marc's favorite TV show. The smart home of Mr. Marc is initially equipped with a standard set of context sensors: in-house location, time, number of persons, favorite show and favorite channel. When our system detects that Marc is connected, it salutes him ("Hello, Marc") and starts to dialog and interact with him (see Figure 7).

Figure 7. User is logged on

Then, the system checks time, our TV service and user's preferences concerning TV shows. If program TV contains user's preferences show, agent calculates the remaining time from the start date of the show and decides to send this information to the "User Interface Layer". Afterward, this last layer sends a request to the "Context layer manager" in order to determine user's situation. For example, at the office, the system will provide this service using a classical text modality by sending information which contains the title of the show, the time of diffusion, the remaining time and the following question: "Thank you for answering by "YES" or "NO" (see Figure 8).

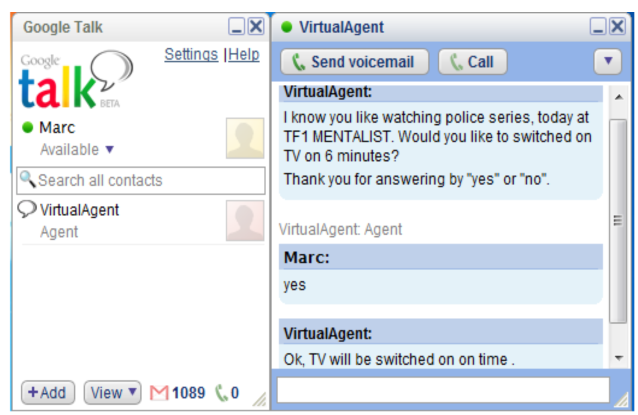

Figure 8. Agent notify user about her best show

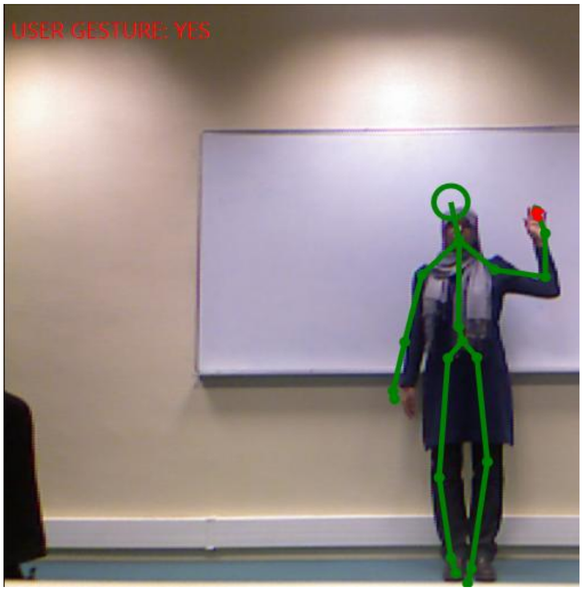

Figure 9. User's gesture response "yes"

If the user responds "YES" using either a keyboard (see Figure 8), a voice recognition or a gesture through a Kinect sensor (see Figure 9), the agent turns on TV in the appropriate time. In that scenario, by executing this action, the system sends, after six minutes, a new text message to the user, telling that the TV is switched on TF1 channel (see Figure 10), but it can also in other situations (e.g., user at home) communicate the same service using a more natural modality such as the vocal one (speech synthesis).

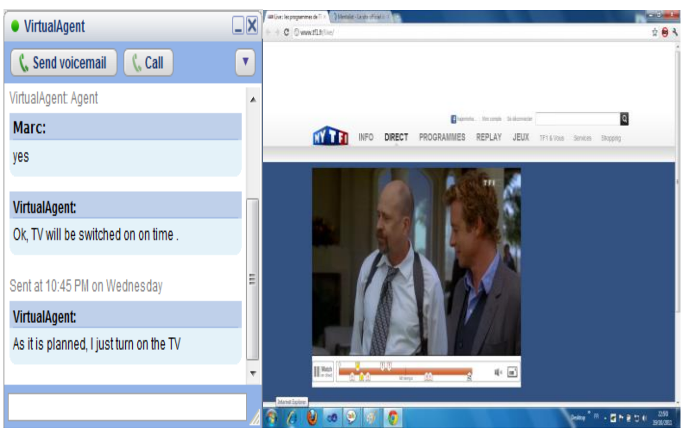

Figure 10. Agent notifies user that the TV is switched on

If the user responds "NO" (to a question such as "Would you like to watch that show now?", see Figure 8), our agent tries to understand the reason, and asks the following question "Are you still interested by this category of show" in order to understand her motivations. If the user responds also "NO", the agent updates this information in the database (the user is no more interested by this TV show).

As a motion sensing, we have chosen to use the Kinect sensor which can used to interpret specific gestures by using an infrared projector, camera and a special microchip to track the movement of individuals in three dimensions. To implement gesture recognition, we firstly define a set of constraints to describe gesture (the joint, the distance, etc.), and secondly, we associate to this gesture a specific event.

In our scenario we have chosen as joint the head, the left and the right hand. If the user raises her left hand, the system interprets this gesture as "NO" and if she raises her right hand, the system interprets it as "YES". Afterward, our system behaves as for the text modality. For the voice recognition, we also used the Kinect sensor's capabilities to recognize human voices. So, user can response by saying "YES" or "NO" vocally and system analyzes this response according to the grammar defined previously.

The goal of using many modalities such as text, voice and gesture is to let the user choose, according to her situation, the most adequately modalities.

B. Phone Channel

As we said in previous sections, we tried to provide proactive intelligent interfaces which can associate different types of modalities with different channels. However, when the user is disconnected from the internet network channel, and if the agent has important information to communicate to her, it should find a new way of communication to reach her wherever she is (home, office, outside, etc.). So, as second channel of communication that can be interesting in our work, we have chosen the phone channel, which allows our system to communicate with people when they are disconnected from the internet. This step is very important in our research; it ensures the continuity with the user by sending for example a Short Message Service message (SMS) as illustrated with Figure 11.

Figure 11. Sending SMS through phone channel to reach disconnected user

VI. CONCLUSION AND OUTLOOK

In this paper, we have proposed the notion of proactive assistance as a solution to increase the productivity and the welfare of the user situated in intelligent environments. So, we have presented an approach model based on three principal layers: “context manager layer”, “anticipate needs layer” and “user interface layer”. Each one has a specific functionality: the first one communicates with heterogeneous sensors in order to collect context’s information, in real time. The second layer tries to adapt collected information to anticipate user’s needs. Afterward and depending on person's situations, “user interface layer” chooses the appropriate way of interaction throw the capabilities of the system to support multimodal and multi-channel interfaces; it can manage text, voice and gesture modalities as inputs, and text and/or speech as outputs. We have realized a prototype based on the architecture layers described below. This prototype, about TV show preferences, illustrates our approach and implements proactive services which can adapt themselves depending on each user’s situation. We have also implemented other services (using Google Agenda, weather forecast, Phydgets sensors, etc.) which are not described in this paper.

In the very close future, we envisage an evaluation with users by proposing a set of proactive services in order to study users’ behavior and our approach capabilities to manage many users simultaneously. In the medium-term, we want to focus on how system can react when context anticipates more than one need in the same time or when several triggers are at the origin of the same need. We have already a theoretical solution for the first problem; we will add a priority ponderation to user’s desires. Moreover, the second problem is being studied and we should obtain quickly solutions in order to respond to users’ expectations.

REFERENCES

[1] M. Weiser, Some computer science issues in ubiquitous computing, Communications of the ACM (1993), Vol 36(7).

[2] M. H. Coen, Design principles for intelligent environments, Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence (1998), American Association for Artificial Intelligence, Madison, Wisconsin, United States, pp. 547-554.

[3] A. K. Dey, Providing architectural support for building context-aware applications. Thesis, 2000.

[4] R. Want, A. Hopper, V. Falcao, and J. Gibbons, The active badge location system. ACM Transactions on Information Systems 10(1): pp. 91-102. January 1992.

[5] R. J. Orr and G. D Abowd, The Smart Floor: a mechanism for natural user identification and tracking. In the Proceedings of the Conference on Human Factors in Computing Systems (CHI 2000), pp. 275-276, The Hague, Netherlands, ACM. April 1-6, 2000.

[6] M. Lamming, and M. Flynn, Forget-me-not: intimate computing in support of human memory. In the Proceedings of the FRIEND 21: International Symposium on Next Generation Human Interfaces, pp. 125-128, Meguro Gajoen, Japan. 1994.

[7] J. Pascoe, Adding generic contextual capabilities to wearable computers. In the Proceedings of the 2nd IEEE International Symposium on Wearable Computers (ISWC'98), pp. 92-99, Pittsburgh, PA, IEEE. October 19-20, 1998.

[8] M. Weiser, “The computer for the 21st century”, Scientific American, 265(3):94–104, September 1991.

[9] A. K. Dey, and G. D. Abowd, Towards a better understanding of context and context awareness. In Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, affiliated with the CHI 2000 Conference on Human Factors in Computer Systems, The Hague, Netherlands. New York, NY: ACM Press, 2000.

[10] M. C. Mozer, The neural network house: an environment that adapts to its inhabitants. Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments, AAAI Press (1998), Menlo Park , CA, pp. 110-114.

[11] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, EasyLiving: technologies for intelligent environments. Handheld and Ubiquitous Computing (2000), pp. 97-119.

[12] A. K. Dey, D. Salber, M. Futakaw and G. D. Abowd, An architecture to support context-aware applications. Submitted to UIST, 99.

[13] G. Chen, M. Li, and D. Kotz, Data-centric middleware for context-aware pervasive computing. Journal of Pervasive and Mobile Computing, Volume 4, Issue 2, pp. 216-253, 2008.

[14] J. Hong, The context fabric: an infrastructure for context-aware computing. CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA. ACM 1-58113-454-1/02/0004.

[15] J. Rouillard, J-C Tarby., X. Le Pallec R. Marvie, Facilitating the design of multi-channel interfaces for ambient computing, The Third International Conferences on Advances in Computer-Human Interactions, ACHI 2010, St. Maarten, Netherlands Antilles, pp. 95-100.