International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 3 No. 1 May 2013, pp. 290-309

http://www.issr-journals.org/ijias/

Study of a proactive agent in a multichannel environment: The X-CAMPUS project

Hajer Sassi and José Rouillard

LIFL Laboratory,
University of Lille 1,
59655, Villeneuve d'Ascq Cedex, France

ABSTRACT: The main characteristic of intelligent devices that compose our environment is their capability to perceive and collect relevant information (context awareness) in order to assist users in their daily tasks. However, these tasks evolve frequently and require dynamic and evolutionary systems (context-aware systems) to improve intelligent devices skills according to user's context. Some context-aware systems are described in the literature, but most of them have extremely tight coupling between the semantic used in the application and sensors used to obtained the data for this semantic interpretation. The objective of our research is to study and implement a proactive approach able to use existing sensors and to create dynamically human-machine conversational situations when needed. The system presented in this paper is named X-CAMPUS (eXtensible Conversational Agent for Multichannel Proactive Ubiquitous Services). It aims to assist user in his/her daily tasks thanks to its ability to perceive the state of the environment and interact effectively according to the user's needs. In this paper we describe our approach for proactive intelligent assistance and we illustrate it through some scenarios showing that according to a given multi-parameters context, our X-CAMPUS agent notifies the user via personalized messages (e.g., suggestion of restaurants according to menus and users' preferences) across the most appropriate channel (instant messaging, e-mail or SMS) and the most appropriate modality (text, gesture or voice). Then, we discuss our quantitative results, based on four principal hypotheses in order to evaluate our system's capability to manage many users simultaneously with different contextual information. We argue and we show that the proactive assistance is very relevant in complex situations with various criteria to take into account (user's profile, location, task, etc.).

KEYWORDS: Intelligent Interfaces, Ubiquitous Computing, Human-Computer Interaction, Proactive Assistance, Multimodal Interfaces, Multi-Channel Interfaces.

1 INTRODUCTION

Ambient Intelligence (AmI) aims at increase the comfort of users in theirs daily tasks based on context information. In our life, we often repeat usually the same tasks, for instance, consulting the weather forecast before going outside, checking appointments, controlling children's tasks, etc.).

We believe that users will appreciate hypothetical capability of intelligent systems to perceive their personal environment in order to manage some 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 [15].

One research domain that requires the computer- everywhere model of ubiquitous computing is that of the "intelligent environment" [13]. 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.

Corresponding Author: José Rouillard (jose.rouillard@univ-lille1.fr)

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 "[1]:

- 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 [18], floor-embedded presence sensors [19] 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 [14, 8]; context history can be used to recognize user's activities and to fully exploit the richness of context information.
- 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 some illustrative examples. Lastly, we state our future work and conclusion.

2 RELATED WORK

Weiser's vision in his article "The Computer for the 21st Century" [16] 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 [3] 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". In our work, to use context effectively we have decided to define context as any information that let our system initiate a conversation or change the way of interaction in order to ensure the continuity with the user.

Many projects were developed around the notion of context aware. In 1998, Coen created the Intelligent Room MIT [13]. 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 [12] 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 [5], which calculates user's position and propose service depending on his position.

In 1997 a project named Cyberguide has for goal to help a tourist in her/his visit by providing according to her/his current position the interesting sites to visit, paths to follow and other useful information. In the same domain and in the 2000 another project named Guide was created with some differences in the hardware used and web access. Each of these projects illustrates convincing results from different use cases. These architectures are *ad hoc* and very dependent to the sensors used. They do not permit the reuse of the software architecture which let the domain of use very limited. After 2000 and thanks to the emergence of new technologies able to describe more in details user's environment, new architectures appeared in order to facilitate the development of applications able to assist users in their daily tasks. In 2004 a middleware for supporting the development of context-aware applications named CASS was created [17]. It uses an object oriented model for context details and provides an important abstraction of contextual information. Context toolkit [4] is another tool which searches also to facilitate the development of context-aware systems. It is based on context widgets which offer a good abstraction and reuse of context like graphical user interface widgets in order to hide the complexity of sensors.

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, we can see that human computer interactions are drastically increasing. Consequently, it should change the way we interact with the ambient environment by providing new intelligent interfaces able to adapt 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. We also argue that implementing and maintaining interfaces, which should be at the same time proactive and intelligent, is still far from easy.

3 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.

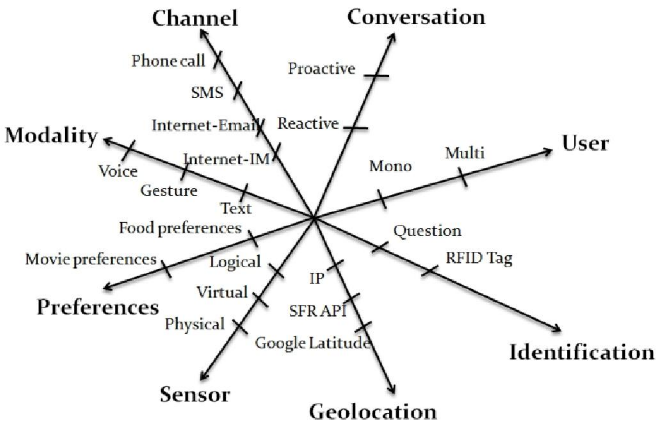

Fig. 1. X-CAMPUS's capabilities

As we can see in Figure 1, X-CAMPUS has many different skills which are represented by the different available axes. Thanks to the use of physical (e.g., RFID tags), virtual (e.g., Google calendar) and logical sensors (e.g., Google calendar combined with RFID tags), our system can interact in both reactive and proactive mode, according to user's needs.

X-CAMPUS is also capable to manage some social aspects of the context. For example, events like "Eating at restaurant" registered in the user's agenda will lead to take into account the possible guests (friends, colleagues, etc.) involved in this particular event. Thus, some relatively complex and time consuming tasks can be done directly by a software agent that is in charge to find the better choice (example: the best restaurant) according to various criteria (alimentary preferences, distances and geolocation of each person, etc.).

In order to ensure an adaptive interaction, we have decided to work on multi-channel and multi-modal interfaces and we have chosen to use two types of channels which are Internet and phone [9] and three type of modality which are text, gesture and voice.

4 PROPOSED ARCHITECTURE

To respond to our research questions, we have chosen to implement an architecture based on three principals layers (see Figure 2), 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.

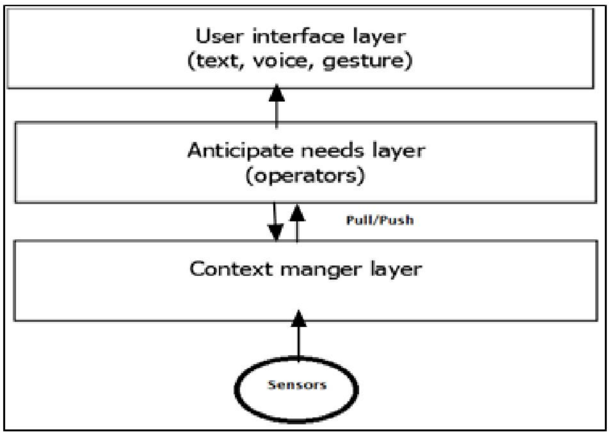

Fig. 2. Context model's architecture

Our architecture is currently deployed on a server containing a database of contextual information (based on an object oriented model for context description), supporting various constraints and rules developed in an *ad-hoc* way.

4.1 CONTEXT MANAGE 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 [2, 6, 7]. 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 3 and Figure 4, left), it firstly determines the user's name in order to salute him/her (see Figure 3 and Figure 4, right) and secondly calculates the number of persons at home.

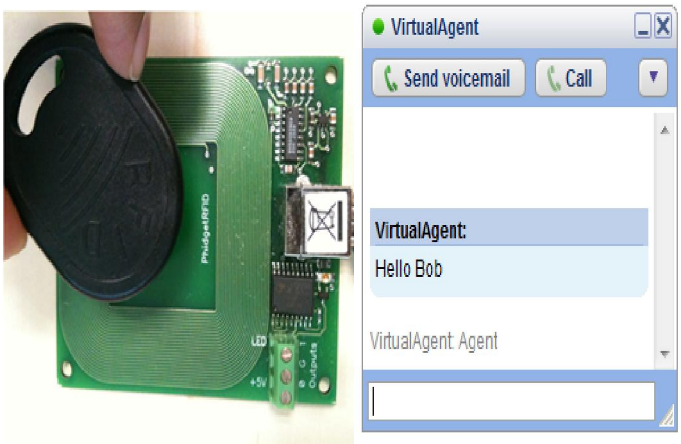

Fig. 3. (Left) Bob's RFID tag. (Right) X-CAMPUS detects Bob's RFID tag and welcomes him on GTalk.

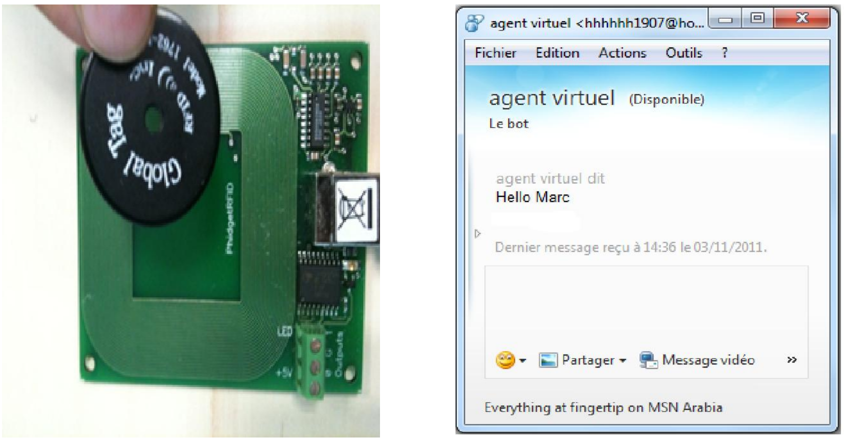

Fig. 4. (Left) Marc's RFID tag. (Right) X-CAMPUS detects Marc's RFID tag and welcomes him on MSN

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.

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

```xml
<?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="BestSerie" dataType="xsd:string">
    <ennonce>What is your favorite television series (put "0" if you do not have)?
</ennonce>
  </question>
  <question format="input" nom="BestBroadcastCategory" dataType="xsd:string">
    <ennonce>Which typifies of emission you like looking (put "0" if you do not
have)?</ennonce>
  </question>
  <question format="input" nom="BestBroadcast" dataType="xsd:string">
    <ennonce>What is your favorite broadcast (emission,issue) (put "0" if you do not
have)?</ennonce>
  </question>
 </questionnaire>
```

Fig. 5. TV questionnaire

According to context, system tries to collect user's knowledge. It triggers a questionnaire (see Figure 5) 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.

- 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.

4.2 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:

- 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.
- Extract operators: 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.
- 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.

4.3 User Interface Layer

For ubiquitous environment, the behavior of services does not just depend on explicit user interaction but also on the environment's perception. 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.

5 Illustration and Scenarios

As we see in figure 6, X-CAMPUS matches all axes by integrating one or more values for each capability. We will give more information about this figure in the section below.

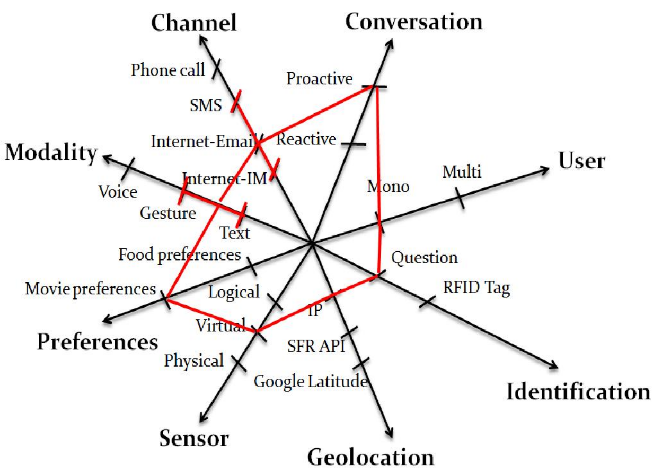

Fig. 6. X-CAMPUS's capabilities used in the favorite show service

The idea, for each scenario, is to illustrate several parts of our X-CAMPUS system, according to different axes (channel, modality, kind of preferences, geolocation, etc.). In the following lines, we will talk about the Internet and the Phone channels.

5.1.1 INTERNET CHANNEL

To demonstrate the identified requirements, a scenario is given in the following. It 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 (IP), time, number of persons, movie preferences (see Figure 7). When our system detects that Marc is connected, it salutes him ("Hello, Marc") and starts to dialog and interact with him (see Figure 8).

Fig. 7. User is logged on

Then, the system checks the timing, our TV service and the user's preferences concerning TV shows. If the TV program contains user's favorite shows, our agent calculates the remaining time from the start time 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 a message which contains the title of the show, the broadcasting time, the remaining time and the following question: "Would you like to switch on TV in X minutes. Thank you for answering by 'yes' or 'no' " (see Figure 8).

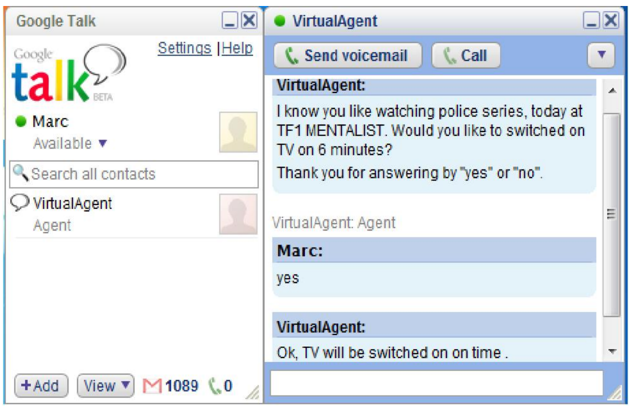

Fig. 8. X-CAMPUS notifies user about her best show

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 by using a more natural modality such as the vocal one (speech synthesis). 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 analyses 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.

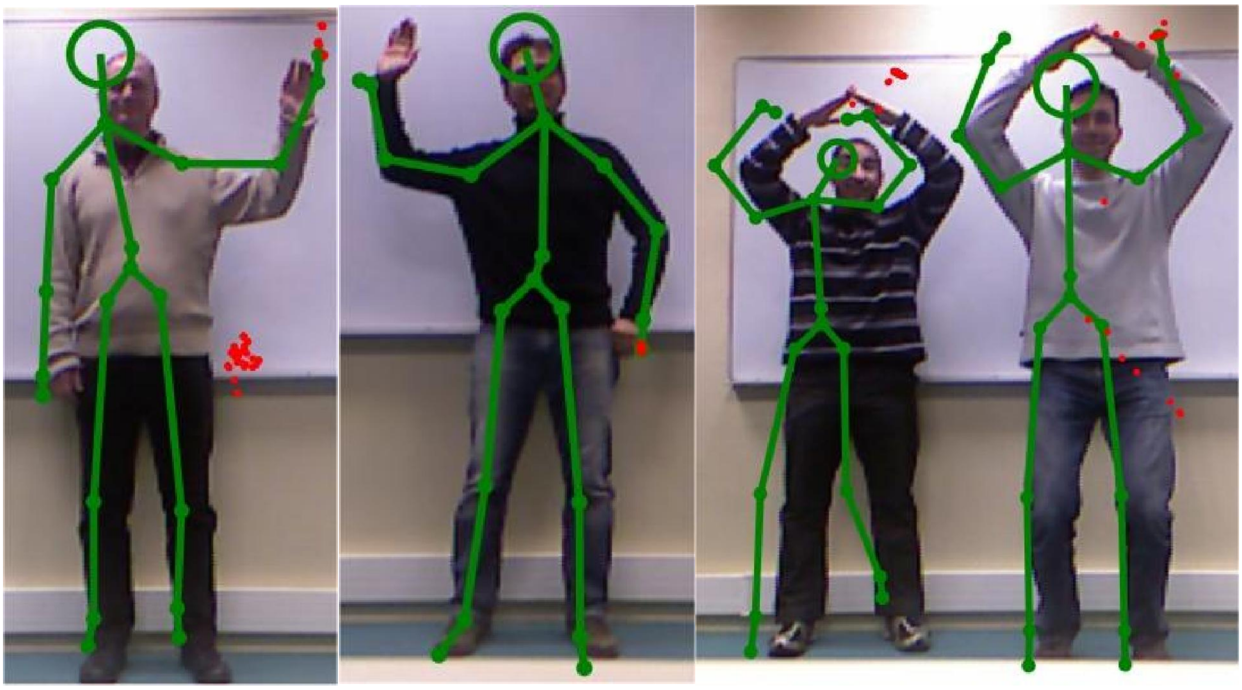

Fig. 9. Some User’s gesture responses (“yes”, “No”, “Home”).

The Figure 10 shows how X-CAMPUS notifies the user that the TV is switched on (left part) and proposes the appropriate show (NCIS, in this scenario, right part).

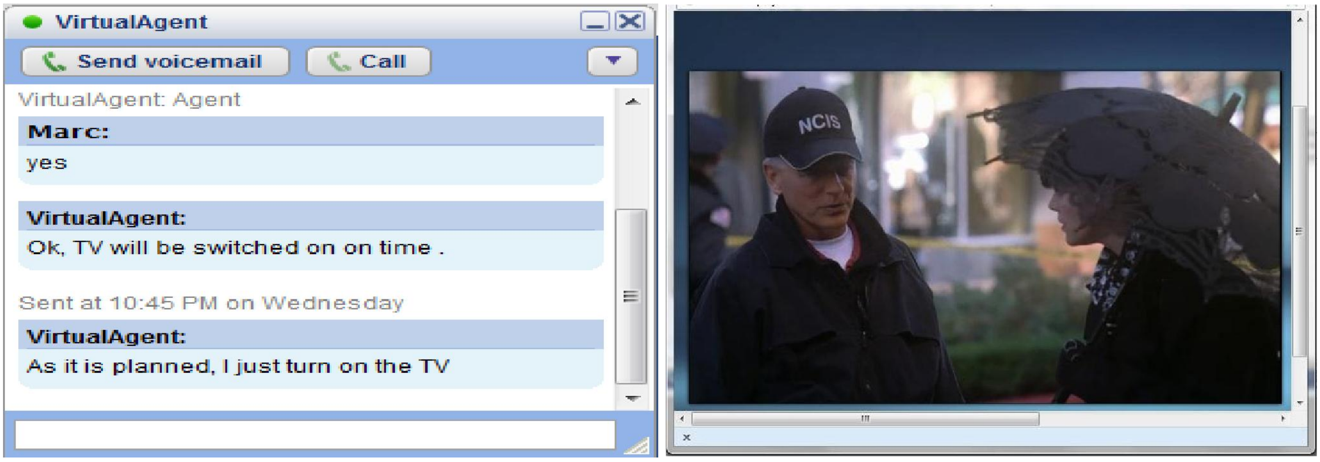

Fig. 10. X-CAMPUS notifies the user that the TV is switched on

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 why, and asks the following kind of questions “Are you still interested by this category of show” in order to understand the user 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).

5.1.2 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.

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

5.2 BASIC EVENTS

This other scenario is about Mr. Marc's favorite dishes. The system knows that Mr. Marc likes "potatoes, beef and pizza" and wants to be notified at 11 o'clock AM, thanks to data collected during previous conversations. The context used to satisfy user's favorite dish is: user's food preferences, user's favorite notification period, user's phone number, user's e-mail and restaurant menus. Every day, our system checks user's favorite notification period, menus proposed by restaurants and user's preferences dishes. If X-CAMPUS finds a minimum of one restaurant that contains a minimum of one of user's favorite dishes, it calculates the remaining time from the start period of notification and decides to send this information to the "User Interface Layer". Afterward, this latter layer sends a request to the "Context layer manager" in order to determine user's situation. For example, at the office and when user is connected, the system will provide this service using a classical text modality by sending information which contains the name of the restaurant, its menu, and on upper case user's favorite dishes through the Internet channel (see Figure 12). We can see that X-CAMPUS justifies its decision in order to let users understand easily its behaviors.

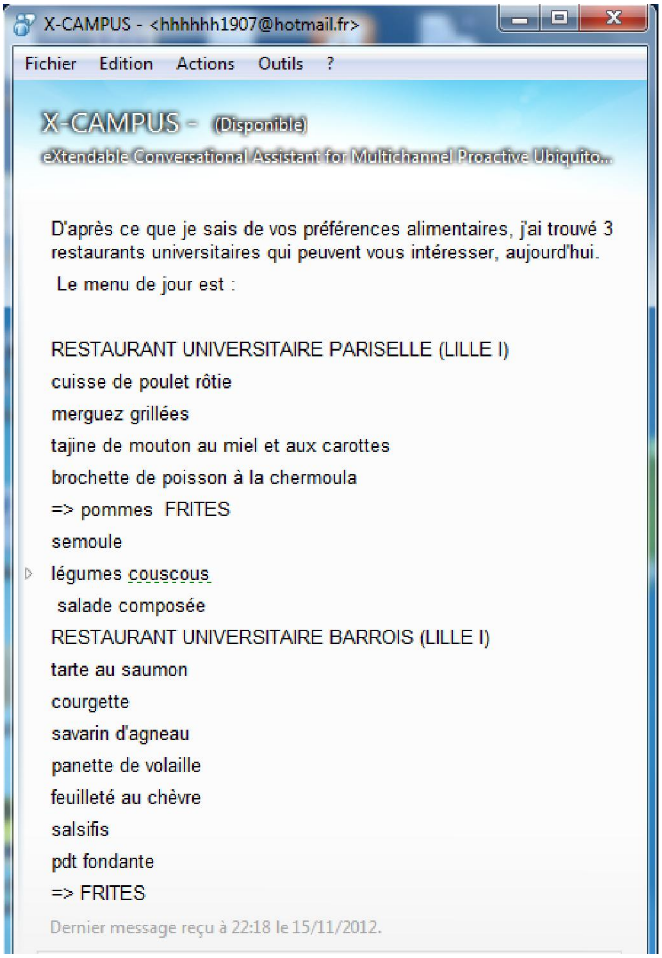

Fig. 12. X-CAMPUS's capabilities used in favorite social dish service

User can be disconnected from the Internet and in this case our system should find another way to communicate at the appropriate period/time. So thanks to a second channel that we use in our approach, messages can be delivered at the right time through the phone channel.

5.3 COMPLEX EVENTS

People frequently organize social events in order to meet, work, or discuss together. This kind of events are complex to managed (time and energy consuming) and to organized correctly. For example, going to the cinema, eating together, etc. can be some very difficult tasks to manage (for a human) as various and multiple criteria can be involved (time, weather, movie category, restaurant style, preferences of each person...).

Obviously, most of the time, the task in more and more complex as the number of invited is increasing. And sometimes, it becomes just impossible to find a convenient solution for everyone. In ordinary case, the organizer should call all participants or invite them through an electronic application such as Google Calendar or Doodle. As a consequence organizer should control by himself guests' responses and find the best solution which satisfies all participants.

In order to help users in these situations, X-CAMPUS has the capability to manage complex situations (multi-user, multi-modality, multi-channel...), according to user's situations, we have decided to integrate a new virtual sensor named "Google Calendar Sensor". Its vocation is to read user's Google Calendar and to send new events to our system. Once the system receives new events, it updates guests' profiles.

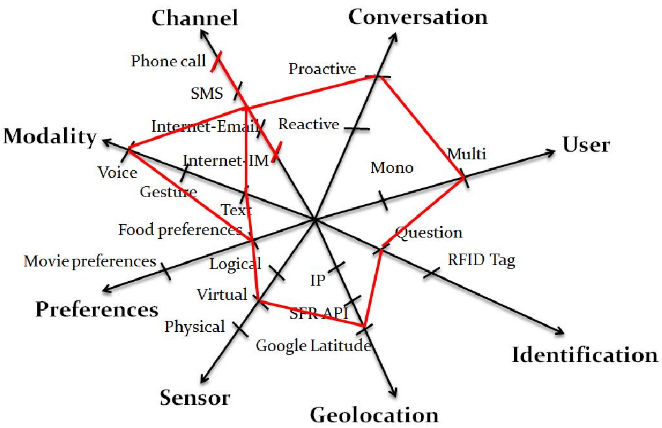

Fig. 13. X-CAMPUS's capabilities used in favorite social dish service

As we can see in figure 13, X-CAMPUS will initiate a proactive conversation by considering as contextual information: user's mode (multi-user), user's location, user's food preferences, user's state (connected or disconnected), and also the possibility to change the used modality (text, voice) and channel (phone or Internet). For example, a user (say Marc) decides to use Google Calendar to invite several colleagues and to organize a meeting in a restaurant (criteria when and where). In this kind of situation, X-CAMPUS is switching from a non-social mode to a social mode. The difference between them is located at the "User Interface Layer".

When the system determines that is time to communicate, it first checks if Marc has a social event named "Eating at university restaurant" recorded in his electronic agenda. Then it decides if it should behave in social or non-social mode. In the first case, it's a complex event management case (multiple users with multiple favorite dishes) whereas in the second case, it's a simple event management case (only one user with multiple favorite dishes). On this basic, X-CAMPUS is able to determine the best available restaurants and more precisely, the name of each restaurant with a menu containing at least one of guest's favorite dishes. When the system knows each guest's available restaurants, it starts its analysis phase in order to find common restaurants. As we can see in Table 1, our system can find the following situations: "no common restaurants", "only one restaurant", or "many restaurants" which satisfy guests' needs.

Table 1. X-CAMPUS's social behavior for two different days

	First day				Second day			
User / Restaurant	User #1	User #2	User #3	User #4	User #1	User #2	User #3	User #4
Restaurant #1	Pasta	Beef	Pizza	chips				
Restaurant #2	Pizza	Potatoes	Beef, pizza	Hamburger	Potatoes	Beef	Pizza	Pasta
Restaurant #3	Pizza, potatoes	Beef	Pizza	Pasta				

Three common restaurants (First day) — One common restaurant (Second day)

The ideal situation is the case where only one restaurant satisfies all the users' preferences food (see Table 1). In this case, it communicates by using a classical text modality through Internet channel or phone channel according to user's state. The message sent by the system contains the name of the chosen restaurant, the menu, the names of the guests and the user's favorite dishes (in uppercase).

In a quite simple situation, X-CAMPUS is able to check, each day, for each user, the best suitable (university) restaurant to propose to the users according to various personal criteria stored in the user profile, such as alimentary preferences (Pizza, Beef, Pasta...), status (student, teacher...), etc. X-CAMPUS users are daily notified about the best match between their preferences and the different menus proposed by all the restaurants available in their geographical area. This notification is launched at a moment previously chosen. It can be done across an instant messaging tool (MSN, GTalk...), or by E-mail, or by SMS, according to the context of use.

In a more complex situation, X-CAMPUS is also able to manage situations where people have planned to eat together: it calculates the more relevant restaurant(s) based upon the preferences of all the users involved in a particular meeting. If a unique solution is available, then X-CAMPUS notifies directly all the users by indicating the chosen place. But if there is not only one solution, our system is able to trigger a communication with the organizer in order to choose across a conversation, the best contextual criteria to deal with this complex situation.

Technically, when X-CAMPUS finds multiple solutions to a given situation (restaurants, movies, etc.) a vocal conversation is initiated with the organizer in order to take a decision. We are using Ippi Messenger [11] and Tropo [10] for this purpose. Ippi Messenger is a Voice Over IP (VOIP) tool, compatible with the Session Initiation Protocol (SIP). Tropo is a powerful yet simple API that adds Voice and SMS support to the programming languages that programmers already know (JavaScipt, PHP, Ruby, Python, Groovy...).

In our example, when our system finds more than one restaurants, it decides to place a vocal call to the organizer. During this conversation, some other criteria are proposed to enlarge the contextual information set (geolocation of the participants, for instance).

6 QUANTITATIVE RESULTS

In order to evaluate our work we have realised an evaluation of our system by inviting some users to try, on several days, some services proposed by the X-CAMPUS system. The evaluation consists to subscribe to two different services which are "weather-broadcast service" and "restaurant service". The first one should inform user about the temperature and some other information of her/his city, and the second one should inform user about the best suitable (university) restaurant, according to various parameters. Both of them should interact in appropriate time through the most relevant modality and channel according to the user's situation.

Our evaluation was launched over four weeks from the 4th February to the 1st of March 2013. We have sent an e-mail to approximately 200 users which are principally teacher, researcher and students of our university and 27 users decided to subscribe to our X-CAMPUS agent.

In order to communicate adequately and to ensure continuity with the users, we have decided to use two different channels which are Internet Channel and Phone Channel. So when we use Internet Channel, our agent X-CAMPUS uses the user's state (Online, Busy and Away) as contextual information given by XMPP protocol (logical sensor). However when a user is disconnected from the Internet, our agent tries to find another way (second channel) to reach him/her where ever she/he is. In this evaluation, the second channel that we used is the Phone Channel.

We have contact users by sending an email which illustrates briefly how to use our X-CAMPUS agent. We have just mentioned the e-mail addresses of our agent and how to subscribe to the two mentioned above services. The ultimate goal of our evaluation is to study the effect of the interaction with a conversational proactive agent. So, based on the following four hypotheses we will try to determine the felt of participants.

- Hypothesis H1: We suppose that most of users (more than 80 %) will be disconnected early in the morning (5.00 to 8.00 A.M.).

- Hypothesis H2: We suppose that most of users (more than 80 %) will be disconnected the week-end.

- Hypothesis H3: We suppose that most of users (more than 80 %) will want to be notified about the weather-forecast early in the morning (before 8.00 A.M.). So behind this hypothesis, we are supposing that the users will not be necessarily online when they will receive this weather-forecast notification.

- Hypothesis H4: We suppose that for the restaurant service, more than 50% of the users will be connected (instant messaging) to receive the notification.

- Hypothesis H5: We suppose that for the restaurant service, most of users (more than 80%) will choose to be notified between 10.00 and 12.30 A.M.

We have chosen two different services which should occurred according to our first hypothesis during two different periods. The first one is about the weather forecast and which should occur earlier in the morning however the second one is about restaurant services and which should occur later in the morning.

The two types of services were chosen to be communicated at two different times, in order to be able to study our agent's capability to manage many users simultaneously and to change its behavior according to each user's profile.

As mentioned in figure 14, the results related to the weather forecast service show that **our first hypothesis (H1) is not confirmed**. During 26 days, 217 SMS were sent to disconnected users (68%), 74 instant messages (MI) were sent to connected users with the status "available" or "Away" (23%) and 27 e-mails were sent to users connected with the status "busy" (9%). These results indicate that users are mostly disconnected early in the morning (but not more than 80%, as we supposed with our H1 hypothesis) and those who are connected are mostly connected with status as "available" which justifies the number of e-mails sent during the four weeks of evaluation.

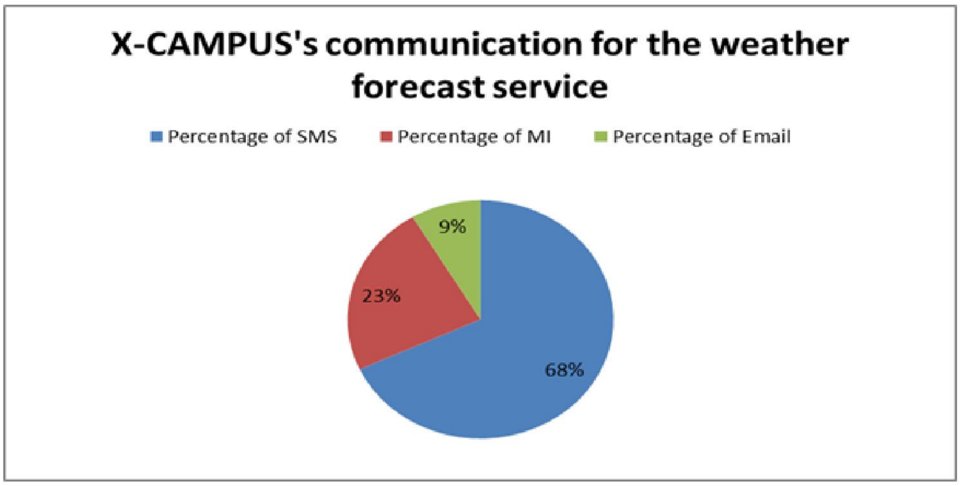

Fig. 14. X-CAMPUS's notifications for the weather forecast service

As we can see in figure 15 the distribution of SMS, MI and Email is different from one day to another, but we remark that usually the number of SMS for each day is greater than the number of MI and the number of Email sent in the same day, with some rare exception, such as the 13rd of February in which the number of SMS is equal to the number of MI sent.

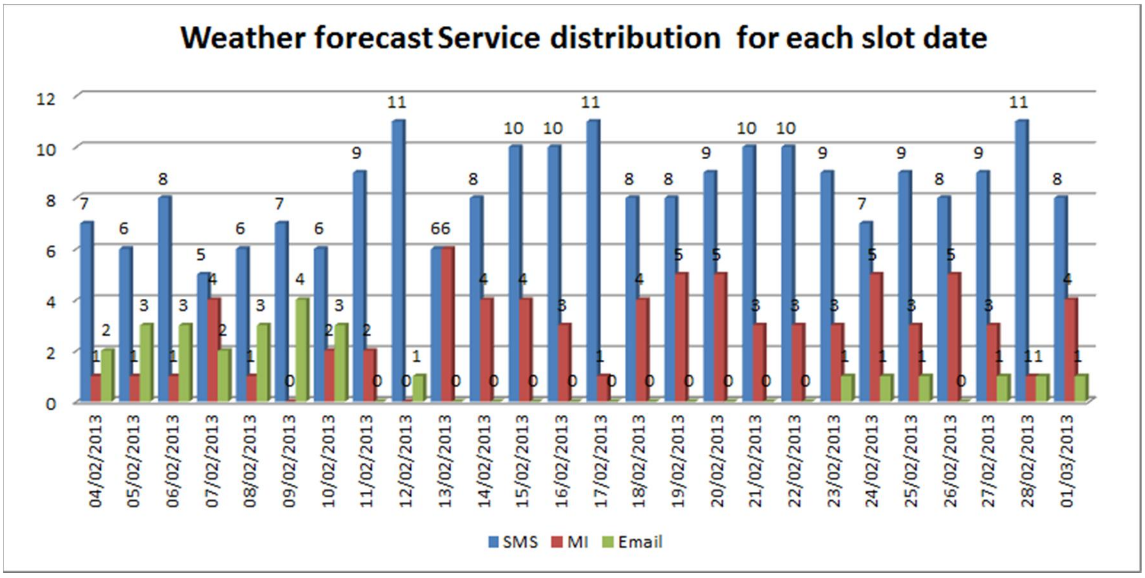

Fig. 15. Weather forecast service distribution for each slot date

We can also see that the use of MI is not negligible during the four weeks of our evaluation. Each day our agent has sent a minimum of one MI except for two days (9th and 12th of February), which let us infer that some users are connected as "available" or "away" early in the morning. However the number of e-mails sent during the period of our evaluation is not very high (negligible); in figure 15 we have 14 days with 0 emails sent, among 26 days (only 9% of the communication in figure 14).

Table 2. Weather forecast service distribution for each week-end

	First weekend			Second weekend			Third weekend		
	Saturday	Sunday		Saturday	Sunday		Sunday	Sunday	
SMS	7	6	59,09%	10	11	84%	9	7	61,53%
MI	0	2	9,09%	3	1	16%	3	5	30,76%
Email	4	3	31,81%	0	0	0%	1	1	7,69%
Number of users	22			25			26		
Result	50/73 = 0.68 (68%)								

We supposed that 80% of the users will be disconnected during the week-ends however results show that this **hypothesis (H2) is not globally confirmed** with only 68,49% of users disconnected during this period (see table 2). However when we detailed week-end by week-end we remark that for the second one our hypothesis is confirmed with 84% of users which are disconnected whereas for the two others week-ends (the first and the third week-end) our hypothesis is infirmed with respectively 59,09% and 61,53%.

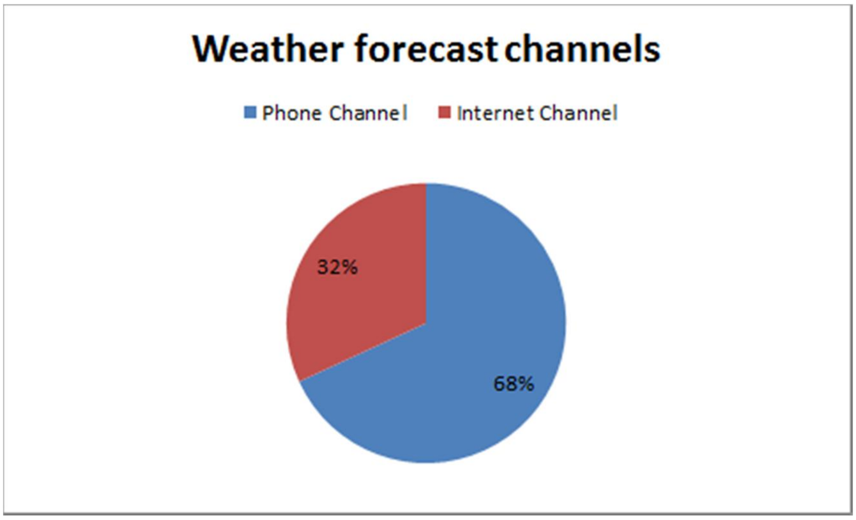

Fig. 16. Weather forecast channels

As we can see in figure 16 the global distribution of the weather forecast channels reveals that globally the use of the Phone Channel is more dominant than the use of the Internet Channel. Despite the fact that thanks to the second channels our agent can send information by MI and/or Email this one is still less used (32%) than the Phone channel, used for distributing SMS (68%), for the four consecutive weeks of the evaluation.

More details on the daily distribution of channels are given in figure 17. We can see that practically the number of SMS sent daily by X-CAMPUS is greater than the number of MI and Email with only the exception of one day which is the 13th of February, in which the number of SMS is equal to the number of MI and email.

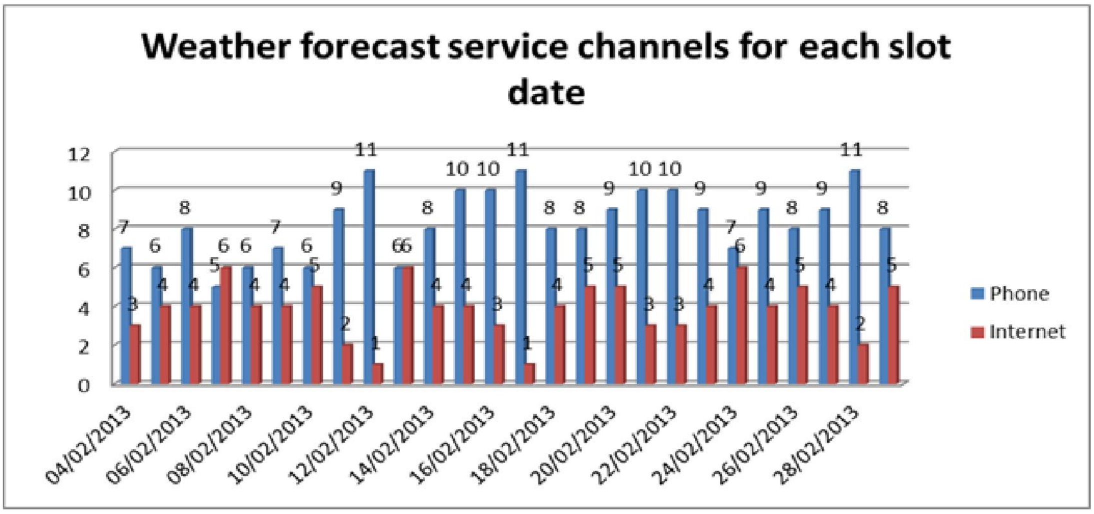

Fig. 17. Weather forecast service channels for each slot date

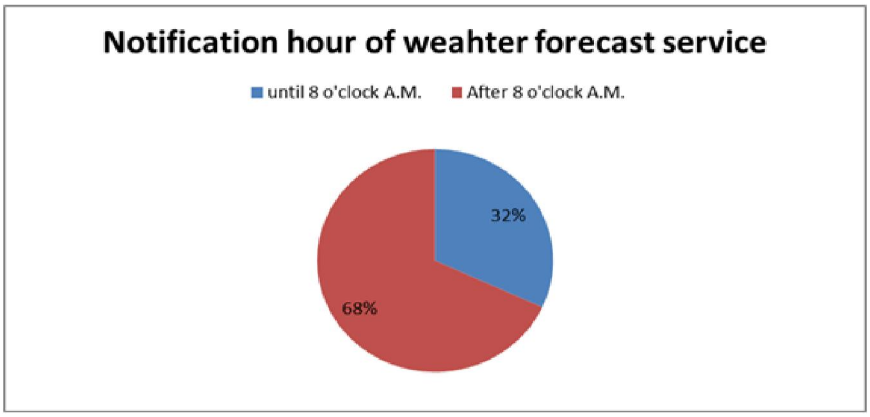

Fig. 18. Notification hour of weather forecast service

During the evaluation, our agent has sent more information about weather forecast after 8 o'clock. **This fact leads to the reject of our third hypothesis (H3)**, because, as we can see on figure 18, 68% of information was sent after 8 o'clock however only 32 % of information was sent before 8.00 A.M. (8 o'clock included), during 26 days.

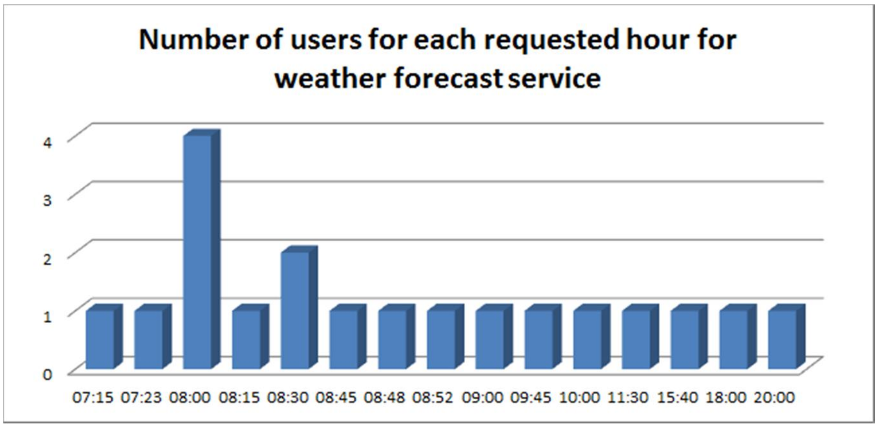

Fig. 19. Number of users for each requested hour for weather forecast service

As we can see in figure 19, the range of slot of times in which users want to be notified for the weather forecast service is from 7.15 A.M. to 8.00 P.M. We supposed that most of users (i.e more than 80 %) will want to be notified about the weather-forecast early in the morning (i.e. 5.00 to 8.00 A.M.) however, only three slots of time are less or equal to 8.00 A.M. (7.15 A.M., 7.23 A.M. and 8.00 A.M. ; this latter is the most requested slot (four users).

Consequently, the hypothesis H3 is effectively rejected with only 31.57 % of users which have choose to be notified before 8.00 A.M. (8 o'clock included), during 26 days.

We have 13 users which are subscribed to the X-CAMPUS's weather forecast service and we have detected 15 requested slots (see figure 19); this means that some users have change their hour of notification during the evaluation period, probably in order to test the X-CAMPUS's adaptation capabilities.

Let now detailed our second scenario and as we see in figure 20 for the X-CAMPUS's restaurant service, the number of MI sent during our evaluation is more important (47%) than the number of SMS (44%) and the number of Email (9%). **This confirms our hypothesis H4** with 56% of users subscribed to X-CAMPUS's restaurant service are connected as "available", "away" or "absent" late on the morning (or early in the afternoon).

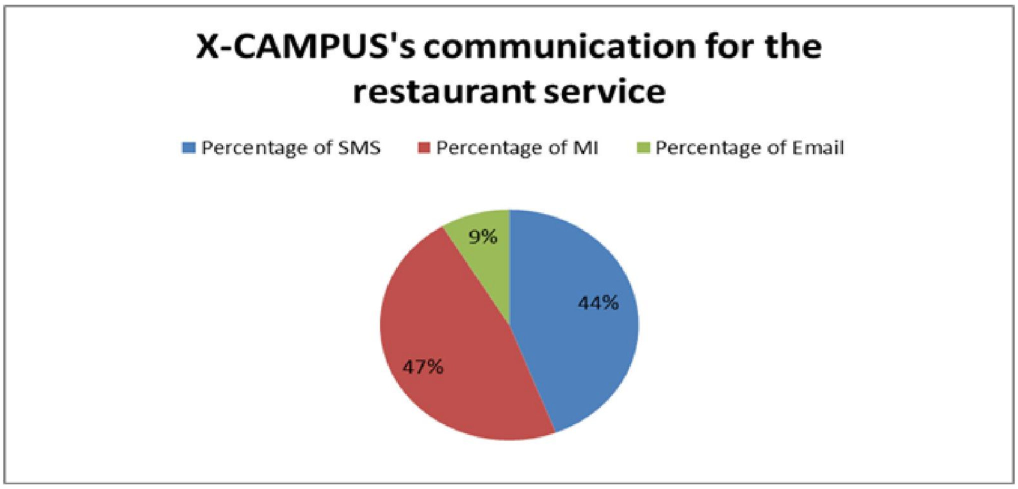

Fig. 20. X-CAMPUS's communication for the restaurant service

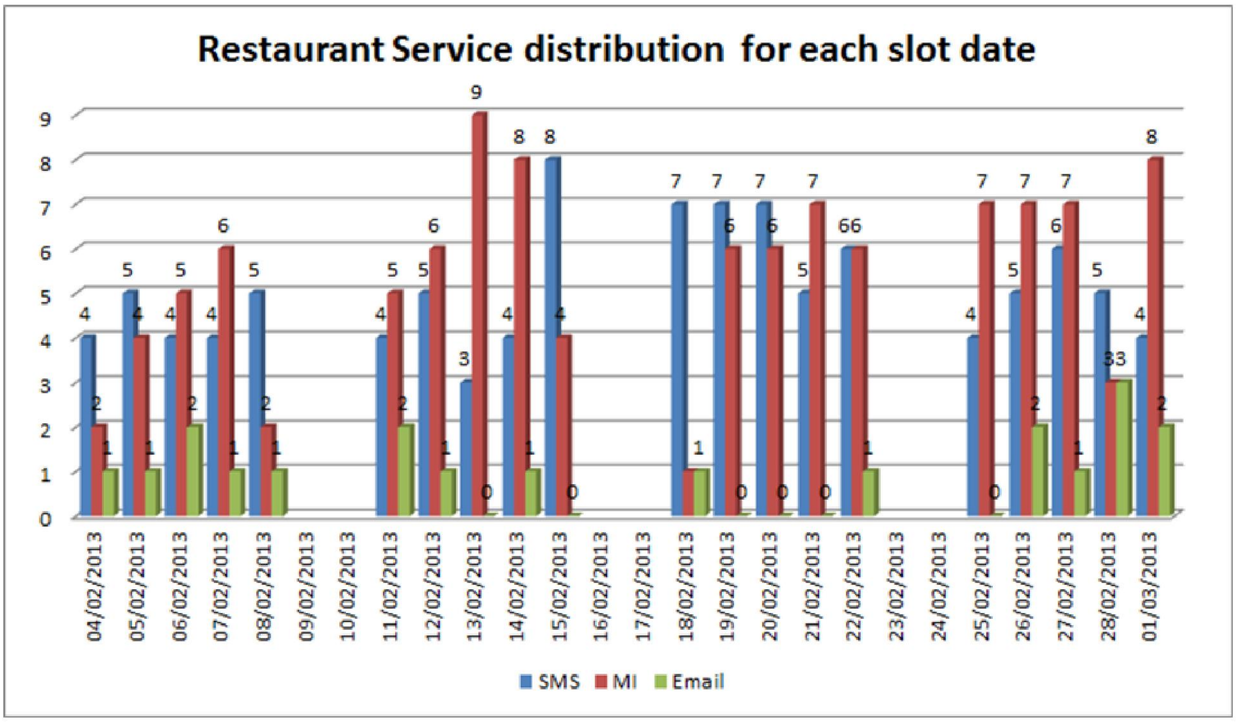

Fig. 21. Restaurant service distribution for each slot date

As we can see in figure 21, the number of SMS, MI and e-mails sent for each slot of date shows more en details the dominance of MI (47%) versus the SMS (44%) and the Email (9%).

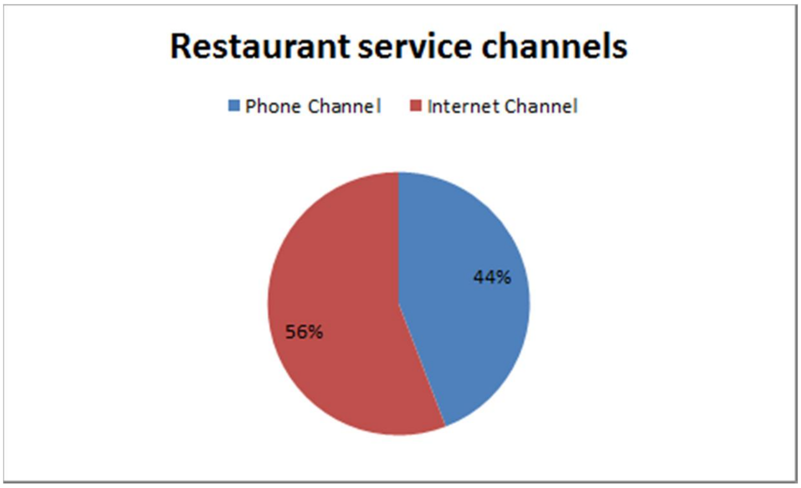

Fig. 22. Restaurant service channels

As we can see in figure 22, the use of Internet channel with the restaurant service appears more important than the Phone channel: our interpretation is that users are probably more connected late in the morning than early (compare to the forecast service, for example). **This confirms again our hypothesis H4**. The use of Internet channel for the restaurant service (56%) is greater than the use of Internet channel for the weather forecast service (32%). So, we observe that services supposed to be used early in the morning, use more the phone channel than the Internet channel, and it's the opposite for the services supposed to be used later in the morning.

We remark that the notion of multi-channel is important in our work because users are not necessary connected according to their requested hour of notification. For instance, during 20 days of evaluation, 44% of communication between the user and the system, for the restaurant service, is made via SMS, that is to say, via the phone channel (see figure 23). Thanks to the capability of our agent to switch from one channel to another, users seem to receive the appropriate information according to the appropriate channel.

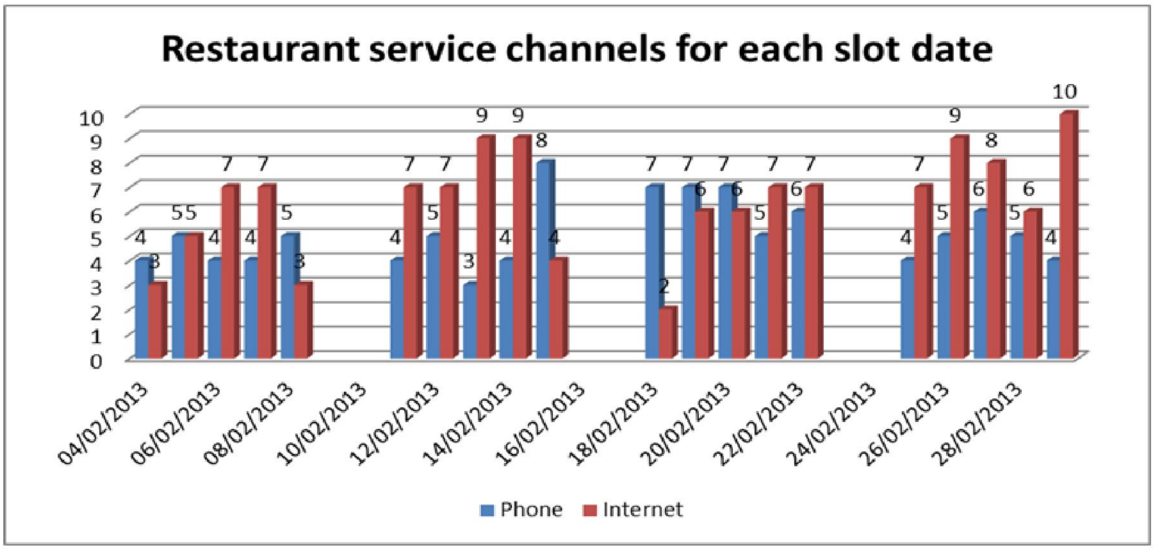

Fig. 23. Restaurant service channels for each slot date

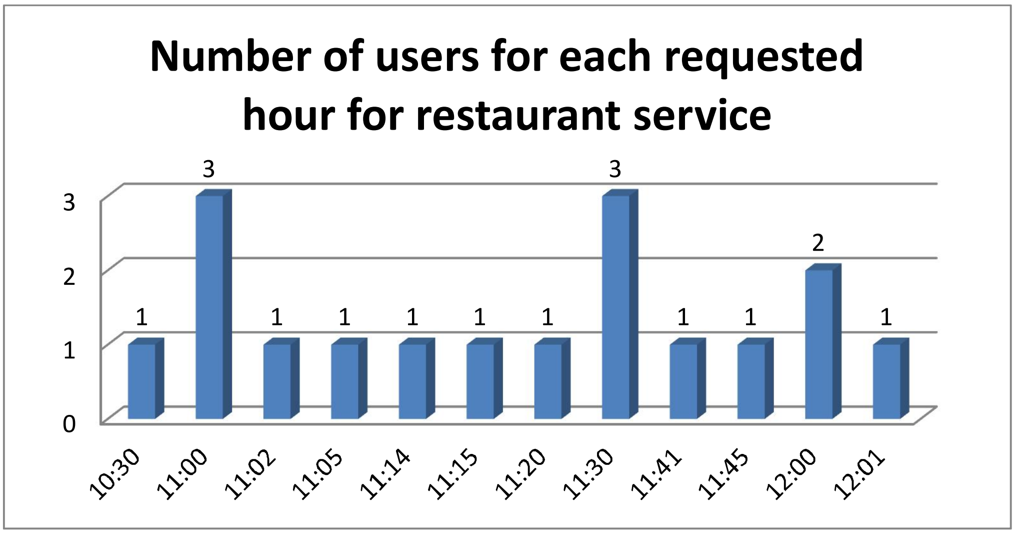

Fig. 24. Number of users for each requested hour for restaurant service

As we can see in figure 24, the range of times notification chosen by users for the restaurant service is from 10.30 A.M. to 12.01 A.M. **The hypothesis H5 is validated by this observation**, because we supposed that most of users (i.e more than 80 %) will choose to be notified about the restaurant service later in the morning (between 10.00 and 12:30 A.M.) As we can see in figure 24, the more requested slots are 11:00 and 11:30, with three users for each. As we know that we have 14 registered users for this service, and we observed 17 notifications, we can conclude that some users have changed their notification time during the experiment.

7 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. We have presented an approach 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. We have showed, with a quantitative evaluation of our work, that the X-CAMPUS system is really able to choose the appropriate channel in order to deliver relevant information (weather forecast, restaurant menu) to the users, connected or not.

Finally, the results concerning the overall experiments are the following:

- Hypothesis H1: We suppose that most of users (more than 80 %) will be disconnected early in the morning (5.00 to 8.00 A.M.), is rejected.

- Hypothesis H2: We suppose that most of users (more than 80 %) will be disconnected the week-end, is partially rejected.

- Hypothesis H3: We suppose that most of users (more than 80 %) will want to be notified about the weather-forecast early in the morning (before 8.00 A.M.), is rejected.

- Hypothesis H4: We suppose that for the restaurant service, more than 50% of the users will be connected (instant messaging) to receive the notification, is validated.

- Hypothesis H5: We suppose that for the restaurant service, most of users (more than 80%) will choose to be notified between 10.00 and 12.30 A.M., is validated.

In the very close future, we envisage an evaluation with users by proposing a set of proactive services in order to study the users' behavior and the X-CAMPUS capabilities to manage multiple users simultaneously. We will also focus on the way to manage multiple needs at the same time (with possible incompatibility between them), and how to detect that several triggers are semantically related to the same user's need. We have already a theoretical solution for the first problem: we will add a priority ponderation to user's desires. The second problem is being currently studied and we should obtain quickly some relevant solutions in order to respond to users' expectations.

References

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

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

[3] 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.

[4] A. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications", Human- Computer Interaction, 16:97–166, 2001.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] https://www.tropo.com

[11] http://www.ippi.fr/index.php?page=home&lang=44&lang=44

[12] 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.

[13] 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.

[14] 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.

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

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

[17] P. Fahy, S. Clarke, "CASS – a middleware for mobile context-aware applications", Workshop on Context-Awareness,MobiSys 2004.

[18] 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.

[19] 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.