Artificial Intelligence and Multi-Agent Systems Laboratory

University "Politehnica" of Bucharest

Université Pierre et Marie Curie Paris

Experiments with a MAS for Ambient Intelligence

2nd PhD Research Report

Andrei Olaru

AI-MAS Laboratory, Computer Science Department, University "Politehnica" of Bucharest
in co-tutelle with
Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie

PhD Thesis Title: A Context-Aware Multi-Agent System for AmI Environments

Supervisors:

Prof. Adina Magda Florea
AI-MAS Laboratory, Computer Science Department, University "Politehnica" of Bucharest

Prof. Amal El Fallah Seghrouchni
Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie

January 2011

Contents

1 Introduction

Ambient Intelligence is supposed to be the next wave in computing: a technology that is pervasive, integrated in objects of every-day use, silently and non-intrusively – but in the same time pro-actively – assisting people in all or most of their day-to-day activities [Wei93, Wei95, DBS+01]. Reaching this state means that all of the layers of Ambient Intelligence – hardware, network, interoperability, context-aware services, interface – must be fully developed, and all of the components that are integrated in the AmI system must work together, as a whole, to take intelligent decisions and have an intelligent behaviour.

In many previous research papers, the features that AmI systems should have, and the types of assistance that they should offer, are presented by means of scenarios. After creating the scenarios, the features are implemented, the infrastructure is built and tested, by using the scenarios for experiments. Scenarios can be considered as part of the specification of AmI systems. During our research, we have come across many scenarios devised by researchers to describe Ambient Intelligence. Some of the most relevant are presented in Section 2.

However, one can easily observe that there is a considerable difference between the scenarios that describe Ambient Intelligence and the state of actual implementations in the field. One of the reasons for this may be that many research teams attempt to build system that include features from all the layers of Ambient Intelligence, but as AmI is a very vast domain, it is hard to build a complete system. THis leads to the implementations to be very specific, and showing many times little flexibility and scalability.

In our approach [Ola10a, Ola10b], we are trying to focus the research effort on the context-aware services layer of AmI systems. This approach is oriented less on specific features, and more on obtaining a flexible, scalable infrastructure for this layer of AmI, based on software agents (and implemented as a multi-agent system). As part of this ongoing research, a scalable MAS for context-aware sharing of information has been built (but using a simple approach to context-awareness) and an architecture that links context and MAS-structure has been devised [Ola10b, EFSONS10, OGF10].

As part of the focus on the context-aware services layer, several scenarios have been designed. By comparison to the scenarios presented in Section 2, our scenarios deal very little with what kind of interfaces and devices are used (or consider using existing devices), and are focus on how these could work together in order to provide a context-aware behaviour. These scenarios are presented in Section 3.

Another part of this research, that is presented in this report, deals with a more profound look into the idea of context-awareness, dealing with the representation of context and with a generic manner in which context-aware behaviour should take place, based on patterns and pattern-matching. This approach, together with a state of the art of the field, examples and an algorithm for context matching, is presented in Section 4.

2 Existing Scenarios and Testing Platforms

Scenarios are used in the domain of Ambient Intelligence to visualize the future state of AmI system, and to create a description of the features that need to be implemented in order to realize the vision of AmI. Let us examine a few such scenarios in the following sections [OEFSF10].

2.1 Weiser's "Sal" Scenario

We will begin with Weiser's scenario featuring Sal [Wei95], one of the first scenarios for Ambient Intelligence. The first thing that one should notice in the scenario is that when the alarm clock – an intelligent appliance – asks "coffee" and Sal answers, the only responses that the clock can interpret are "yes" and "no". That is, the appliance can only understand events (in this case, the utterance of a positive or negative response) that are relevant to its function.

Intelligent windows that can display traces of neighbours that passed on the street (as well as public information like weather). Their function may be the output of a centralized service, but it may be completely local: when the neighbours pass, the window senses their presence.

While having breakfast, Sal marks some news (from a printed paper) with a smart pen, having the associated text sent to her office. The text comes from a service related to the paper (perhaps a website), that may be contacted by the pen itself or by an unspecified intermediary agent. If the paper sends the data directly to Sal's office, that may be a privacy concern (why should the paper know Sal's contact data?). A solution would be that the request is anonymous and the response comes back to the agent, which in turn sends it to the office.

Other services are mentioned, relating to the most common points in AmI scenarios: localization and information on points of interest. Then, Weiser presents more advanced uses for tabs – small displays with two buttons and wireless connectivity: gestures, storage of small pieces of information, and information-related layout. What is relevant here is that each tab relates to one piece of information, and that individual tabs can be used to point out various events. In the end, a context-aware application: finding a person (Mary) that shared a certain context, in terms of space, time, and event, and, by this association, finding the person's contact details. Finding Mary may not be special if the meeting exists in an online calendar's events and invitations (like the Google Calendar today), but it may be particular if the search is done by associations between context data related to the time of the meeting, the place, the number of people, and also to the fact that Sal did not previously know Mary.

2.2 The ISTAG Scenarios for Ambient Intelligence in 2010

The scenarios created by the ISTAG group envision AmI of 2010 [DBS+01]. While now, in 2010, most interfaces presented have not yet been developed, it is surprising how many of the services exist in one way or another. In the scenarios and in the annexed analysis, the authors emphasize the hardware and human interface features. In the following paragraphs, we will also make observations on possibilities for the internal functionality of the system.

The "Maria" scenario emphasizes two directions: first the movement of the user's electronic identity (keys), preferences and data with her, seamlessly, as well as the capacity to easily use the local resources: vehicles, utilities, computing and communication capabilities; second, the easy interaction with payment services and with the trusted storage for her presentation. We will observe that, apart for some details, almost all technologies and many of the services already exist at the present time. What lacks is the facility in their use and, although the scenario does not necessarily imply that, their interoperability and unification under a common framework.

The "Dimitrios" scenario is based on two elements: first, the very advanced digital avatars (D-Me) that can speak multiple languages and, more importantly, that can take decisions and interact naturally with other persons; second, the ease with which information transits the system: the senior person's D-Me contacts Dimitrios because he has the same heart condition, and Dimitrios's D-Me finds a child of the same age and situation with his own, for socializing and educational purposes. It is relevant here the way in which information is available to users that are interested in it, or that may take action as a response. The information is made available based on common context: similar heart condition, similar age and educational/financial situations. It is not specified how these services work internally, but, like in a sort of social networking, a distributed model may be applied.

The "Carmen" scenario includes a range of services that already exist to a certain extent: car pooling, internet shopping, smart fridges, traffic information, vehicle-to-vehicle communication. Again, it is the element of uniformity and facility that lacks in the present. There is no unified system that does all that. Moreover, all systems that she uses seem to be centralized. It is worth noting that, in order for all people to use such a great number of centralized services, a powerful infrastructure is needed; or, a smart way to bring decentralized services closer to the users, and more related to their context.

Finally, the "Anette and Solomon" scenario is placed again in a farther future, as AmI features natural communication and advanced semantic processing capabilities, being able to "converse", "suggest", and even help the users with decisions and with a part of their work. More interesting features for a present implementation of AmI are identity checking, scheduling, and selecting information that is appropriate for the current context to make public for other users.

What we find interesting in the ISTAG scenarios, apart from the advanced human-machine interfaces, is the capability of the system to provide information and services just in time; also, a range of services that already exist now on the Internet, but that should be easier to access, in a more uniform way. The design of such an AmI system is not detailed and leaves us with many questions: how will the system support offering just-in-time services, in a very continuous and frequent manner, to the great majority of people on the planet, as the quantity of data will greatly surpass the requirements of today's Internet?; the services presented do not seem to have much in common – aggregating information coming from different services may lead to more and better capabilities; dependability of the system is also important: how can one assure a dependable service, that it is unlikely to fail and that will be truly ubiquitous, using all possibilities to offer the services to the user.

2.3 Other Relevant Scenarios

Satyanarayanan mentions two short scenarios [Sat01] that are more locality-oriented: in the first one, the AmI system (named Aura) notifies the user, while surfing the web at the airport, that the quality of the wireless network is much better at another gate, as at her current location there are many users accessing the Internet. One could argue that, if AmI is available for everyone, all will receive such a notification, so the system should consider the event in which all users will start moving toward the other gate. In the second scenario, the files of a user move with him, automatically, from his computer to his PDA, and from the PDA to the projection computer. It also allows voice editing of the user's slide presentation. During the presentation, information related to the attendees' emotional states is sensed and the system suggests the presenter to not present a certain "sensitive" slide. In these scenarios, context is important: the proximity between the user and available resources, the user's current activity that involves certain files and certain devices, the relationship between the user, his / her activity and the states of nearby people.

Banavar and Bernstein present a scenario [BB02] that is focused on the seamless transfer of network connection and use of local peripheral devices (keyboard, screen) in order to assure a continuous video conversation across several situations, as well as comfortably working with personal files, using an interface in the car, on a PDA, or in a plane. This is made possible by intelligent usage of available computing and communication resources and by just-in-time decisions.

Kindberg et al [KBM+02] emphasize the necessity for web-present objects, places and devices and the need to establish relations between these, according to the current context, usually in function of the user's activity. When the user Veronica arrives in a new city, her PDA automatically proposes links to interesting places to see. When she desires to communicate with her friend Harry, depending on his availability, a telephone call or an e-mail message are proposed. In an office building, she can easily connect to an available printer by pointing her PDA at it, and her PDA can also retrieve information about objects nearby that are tagged and have web-presence (i.e. feature a page on the web).

Vallée et al also describe a scenario [VRV05] in which screens and support for video-calling are located automatically. Here, some more about the internal functioning of the system is described: intelligent agents sense the context and decide on how to announce the phonecall when there are people around, what to tell to the caller, and how the video-call is redirected to the room in which the receiver of the call moves to.

Tracking users and providing them with useful information is also discussed in the scenario of Viterbo et al [VMC+08], as well as setting preferences in function of the current activity. One interesting aspect that is rarely discussed by other authors is the need for devices to be able, in the context of heterogeneous software and a distributed system, to align their ontologies – the semantic meaning that they assign for various terms that are used in communication.

There is another class of scenarios, that we will not discuss in detail, that concerns isolated environments: the smart home and the smart conference room are the most common examples. In the smart home [AM07, BPG+09], various appliances, sensors and devices must be able to track the user, to assure a certain degree of automation, and, especially in the case of elderly and/or disabled people, to detect health disorder or other situations in which assistance may be needed. In the smart conference room [JFW02], challenges are related to the facilitation of file, control and image transfer between personal and public devices and screens. The difference between this type of local scenarios and the scenarios discussed earlier is that in these cases we are talking about a trusted environment and in which distribution is not absolutely necessary. One challenge here is the necessity to make heterogeneous devices interoperable [Hel05, JFW02].

Finally, we will refer the work of Bohn et al [BCL+05] on Ambient Intelligence, that offers a complete perspective on the implications and concerns related to AmI, from several points of view: economy, privacy, reliability, ethics, social compatibility and acceptance. The authors also describe several interesting (albeit questionably ethical) scenarios: real-time shopping, information collection and shopping done automatically by smart products (silent commerce), perfect price discrimination and more advanced personalized schemes of payment, cross-marketing products – all in function of the user's context.

2.4 Experiments with AmI

In what is a quite different perspective from many AmI scenarios like the ones described above, the actual implementations of AmI and UbiComp, and the experiments that are carried out, are much more close to what present technology is offering.

A certain class of implemented UbiComp applications deals with the management of Smart Rooms. For instance, the Interactive Workspaces project [JFW02] started with the management of large displays, and then moved to the management of a whole smart room, using a system that allows the execution of any application on any device, with any display size, and using different types of output. While the features are interesting, the infrastructure is working on a know, small-scale environment, with no security or privacy concerns, and that is supposed to be trusted.

Hellenschmidt introduces the SodaPop model [HK04, Hel05] for the flexible interfacing of devices that offer different services, or different stages in information processing. The experiments are made with typical home devices, as well as in a smart conference room. While the model in general features self-organization and layering for more flexibility, it requires many changes to the devices and is not oriented toward scalability.

There are other implementations that are oriented toward larger environments, but feature little flexibility: the iDorm application [HCC+04], as well as Dalica [CMTT08] or ASK-IT [SM06], handle specific AmI applications, which although useful, give no indication of how could a general approach for AmI result from that.

Finally, there are a few applications that have been tested in more realistic conditions, with a larger number of devices, like MyCampus [SGK05] or AmbieAgents [LW05]. AmbieAgents is one of the few agent-based AmI platforms that have been tested in real-life – it has been tested in the Helsinki airport. However, these applications come very close to what the Internet already offers today: selection of services based on some personal preferences.

2.5 Common Features in AmI Scenarios

Many scenarios for Ambient Intelligence are a mix of elements from several layers of Ambient Intelligence. Most times, they are focus on two of the layers: intelligent interfaces and context-aware services. Sometimes, issues of interoperability are also mentioned.

Mixing these elements may be confusing to the designer of an AmI system that needs to be implemented by a research team: as Ambient Intelligence is a vast domain, it is hard for the same team to develop connectivity features, context-awareness, the interoperability layer and new interfaces at the same time. That is why most implementations, while trying to tackle all this problems, result in a system that is not flexible or scalable, and is by far not as impressive as the initial scenarios described.

The specific features that the scenarios introduce, as said in the previous paragraphs, mostly relate to two topics: intelligent interfaces and context-aware services. The interfaces are suppose to be more natural, and more intuitive for people to use. However, these interfaces are not fully specified, and it is hard to realize exactly how they would work: for instance, in Weiser's scenario, Sal circles a quote from the newspaper and it is sent to her computer at work – there is no mention how does the system know where to send the text: is there a particular pen for each destination: does the pen have buttons for favourite destinations? How easy would it be for the user to send the quote to a new destination? The same questions linked to the system specification can also be posed for some other scenarios as well.

Other questions that may arise are related to the flexibility and the scalability of the scenarios: in a scenario we see how the central character is using one specific feature in a way that seems intuitive enough. But how will the user be able to use a large number of services – as computers and the Internet allow him today – that may not be able to offer non-overlapping interfaces by means of the same, well known, devices. Also, it is very important to think about how will the services be offered to a large number of users, in a dependable and reliable way. These are questions that we are trying to discuss in the next chapter.

3 Some New Scenarios

3.1 Core Features

As Weiser puts it, "there is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk in the trees relaxing and computers frustrating". For the user, Ambient intelligence must be like a familiar corner in nature: pervasive, natural, predictable. But in order to assist people, it must also be pro-active and intelligent. The features of AmI are also its greatest challenges. Let us examine the most important features in our vision [OEFSF10].

AmI must be ubiquitous, pervasive. Its architecture must support a large number of mobile devices that incessantly share large quantities of information, without the user's knowledge (but not against the user's preferences). More than that, its model must be reliable: people will get used to it and will not be able to live normally without it; in order to be invisible, people must not notice it is there, but also they must not notice when it is not. These requirements call for a distributed, redundant system, like the Internet is today. Moreover, decentralization and locality are required by the fact that most of the generated information is not needed and does not make sense outside a certain domain of space, time and social relations (acquaintances). So AmI should be *distributed* and work at a *local* level.

AmI must be natural, by using advanced multi-modal, intuitive interfaces. That requires AmI to be adaptive and flexible: as any advanced technology, some users will choose to not use it at all, some will use it only for specific tasks, and for some it will mean an essential component of their lives. AmI must adapt to all and only require the attention that the user is willing to invest. AmI must be predictable and transparent, being able to make the user understand why that information and services are there and how the system works in principle. This means that the basic principles that make AmI work should be *simple* and easy to understand by anyone and also make AmI *generic* and *adaptive*.

But AmI must also be pro-active and smart. It must take adequate action, without intruding. The action must be taken only if the user would understand the causality, and only if the user would approve the action. This is where *context-awareness* has a leading role. If the user has communicated many times with a friend, and both will be attending the same event, it is normal to automatically provide the friend with information that the user has on the event, because there is enough common context. But there is no sense in sending sensible user information to strangers on the street, except when this may be necessary in case of an emergency, when immediate action is imperative and privacy is not prevalent.

From the point of view of the application layer, it is context-awareness that is the solution for a predictable, natural flow of information. Like in social networks and shopping sites, one can assume that the user will be interested in things that are related to what he already knows, to what he does, and to the people that he is acquainted with. It is unlikely that someone is normally interested in something that bears no relation whatsoever with any part of his or her life. Useful information is information that is related to the context that the user is in. Context has several aspects: physical space, time, activity (current, past or planned), social relations and resources. Considering a space in which the dimensions relate to these five aspects, relevance of information may be defined as *proximity* in this space.

As context is based on this sort of locality, this also solves the problem of information overload. The user can only do one thing at a time, be in only one place, only a number of past actions are still relevant and only a limited number of actions can be planned. So the context space of the user is limited and will only be related with a limited amount of information, that itself can be sorted according to its degree of relevance toward the current context.

There is one more very important issue related to context-awareness and AmI: the same AmI system will have to support more than one user at a time (as we usually see in the scenarios). In public spaces there are a great number of users that AmI has to be able to notify and to assist without them losing privacy (or important information) to other users. We will discuss these features in Section 3.4.

3.2 Scenarios for Adaptability and Scalability

This section will present three new scenarios, that give an insight on how an Ambient Intelligence system may work internally. Details are focused on the application layer of the system, on how information is managed and how decisions are taken in function of context. Having this focus, we will consider that users are using today's hardware and connectivity, but devices will be enriched with AmI agents, that will form the application layer of the system.

Scenario 1. A senior person walks on the street towards her house. In the pocket she has a mobile phone with an AmI software agent installed, featuring Bluetooth and GSM connectivity and in communication with a multipurpose sensor that monitors vital signs. As she does not like technology very much, the AmI agent has been configured to communicate the least possible, so it normally does not connect with any other agents in the surroundings. The person lives in a small basement apartment. She climbs down the stairs, she misses one of the last steps and falls. She loses consciousness for a few moments.

In this short time, by means of the vital signs sensor, the AmI agent detects that the situation is not life threatening and no major injury occurred. However, care may be needed. There is no need for an ambulance, but there is a personal medical assistant that cares for this particular person and that should be called right away. This reasoning is done by a dedicated module of the AmI agent, that is especially designed for senior or disabled people. The medical nurse is reachable by phone, but there is no GSM signal at this location. The AmI agent searches for another resource that can offer communication services. It activates Bluetooth and finds a device that also runs an AmI agent. It provides to the other agent some information about the context: there is someone that needs urgent communication by phone. No personal details are provided. The other agent detects that this context fits its activity history: it has helped with this kind of actions before and it is even configured to do that without confirmation from the owner. It accepts the task. The agent of the senior person then gives to the other agent a number and a message to be sent.

While the senior becomes conscious again, the AmI agent receives, by means of another Bluetooth phone in the area, the confirmation from the nurse. She will arrive in just a few minutes.

Scenario 2. On the largest stadium of an European capital, a concert is going to be held, by one of the most popular rock groups of the time. Hundreds of thousands of people are participating. Most of them have mobile phones or smartphones which run AmI agents. Young people are more permeable to new technologies, and the agents are configured to communicate with other agents that share the same context, while keeping personal data private. At the concert, all participants share space-time coordinates, as well as the event that they are participating in. AmI agents form a temporary, anonymous social network, communicating not by means of the Internet or by GSM, but by local connectivity like Bluetooth or WiFi ad-hoc networking. They exchange, anonymously, interesting news or links that are related to the event and to the band. The users made that information public and are not necessarily aware of these exchanges, and will view the new data after the concert. Sometimes they exchange data intentionally, sending each other interesting links.

As the concerting band will be an hour late, the organizers send this information to the agents that manage the WiFi access points in the area. In turn, these agents disseminate the information to the devices connected to WiFi. The information is of great relevance to the participants, so it spreads fast among the devices of the people on the stadium. In case other users that are not participating to the event received the information, their AmI agents will discard it because their users are not participating in the event, so the information is not relevant.

Finally, the concert begins. Towards the end, a pyrotechnic event causes a fire on the stage. For security reasons, the public must be evacuated. Panic breaks out. The GSM network soon becomes unavailable and the WiFi hotspots are overloaded. Special emergency devices connect to Bluetooth phones that are located near the exists and send them directions towards the exit. From device to device, the urgent information quickly reaches all participants. AmI agents are capable of calculating the relevance of received information according to the number of links it went through, and choose which exit would be closer.

A few days after the concert, a group of participants that shared, intentionally, a lot of images and links, but not any personal details or contact information, want to find each other again. By using the concert site and the fact that they shared so much, their AmI agents are capable of communicating again and the group can meet again.

Scenario 3. Marc is a researcher from France is on the trip towards the venue of a conference he is attending, currently during an intermediate 3-hour stop on the Athens airport. At the gate, he opens his laptop, on which several AmI agents are running. He marked on his schedule several activities connected to the conference and to this trip. One AmI agent, searching possible interesting data on the Internet, finds the following associations: from the attendants at the conference, one is from the same country – France. He has a public calendar on his website (and also a photo), that specifies a flight to the conference venue, in exactly the same time interval. There are no details on how to contact an AmI agent in relation with

> the other participant so no further details can be retrieved. Still the considerable amount of context awareness makes the AmI agent inform Marc of the findings including the name and the photo. Marc takes a look around and, indeed, spots the other researcher nearby. He goes to him and, politely, makes contact. The two researchers can now talk to pass the time to the flight and will be able to share a taxi ride from the destination airport to the conference venue.

3.3 Problem Solving

In many scenarios for AmI, the features that are presented are mostly about offering people the same kind of services that they are offered today, but in a manner that makes them easier to access. There are however some scenarios that suggest that AmI could be more than that: that Artificial Intelligence may be integrated in AmI in such a way that assistance will go beyond accessibility, and will include solving of problems [DBS+01, RAS08]. Take the following scenario [OF10]:

> Alice will go to a rock concert in the evening of the current day. The concert is located at a stadium outside the city, therefore she should find some means of transportation to get there, but she hasn't yet given thought about that. Bob, her roommate, will go to the same concert but he has not talked to Alice about that yet. However, he has already booked a taxi to get to the concert. This is a typical situation for our approach [OEFSF10]: insufficient communication between people leads to a lack of otherwise relevant information that could be easily obtained by means of an AmI system.
>
> Alice and Bob are both users of the AmIciTy Ambient Intelligence system. What we want is that the system (1) detects the need for a means of transportation for Alice, (2) based on information on Bob's agenda, suggest that a taxi may be an appropriate solution for Alice as well, and (3) based on the existing shared context, propose to Alice that she uses the same taxi that Bob has already booked.

In this scenario we hardly make any reference to the hardware that is used, or to the interface. What is important is that the users are offered an intelligent service, that solves a problem for them. And this is indeed the purpose of AmI: to assist users in they daily lives, in a pro-active – but non-intrusive – manner.

A problem of the same type (not necessarily a scenario) is the following:

> A researcher is on the last day before the deadline for an important conference. As usual, writing the article has been left for the last moment, so it is critical for the researcher to finish the article before midnight. A colleague sends him a message regarding an interesting link that he found that relates to their research field. However, the AmI system decides that, *although the link is relevant to the field in general*, but not to the article, it will make show a notification for the message only on the following they, knowing that it is *very common* that any disturbance on the last day before the deadline is badly received.

The scenario above is, again, not related in any way to hardware or interfaces. It is about the decisions that the system has to make, based on context and on previous experience.

3.4 Multiple Users and Collaboration

The great majority of scenarios for AmI have a central character that we follow through a series of instances in which he or she uses the features of the AmI system. But the users are not alone. In public spaces, all or most of the people around will also be users of the AmI system, and, for instance, a vocal advertisement that says the name of the user passing by (like in the movie Minority Report) may be heard by other users as well, producing a potential lose of privacy. However, users knowing each other may be happy to be notified if they happen to be in the same shopping mall, at the same time. Context is everything. In our work (including working together with a team from the NII institute), we have developed some potentially interesting scenarios that deal with the existence of multiple users. The scenarios also include some more classic AmI features, that use existing devices from the environment, controlled intelligently, in order to assist the users.

Scenario 1. On the floor of the laboratory two researchers Alice and Bob arrive almost simultaneously, with two different elevators situated at some walking distance one from the other. Alice and Bob are going to attend a research meeting for the Panda project, in room 42, which is somewhere between the two elevators. Both researchers have the meeting in their PDAs' agendas. They are at the laboratory for the first time and feel lost in the maze of corridors.

When Alice goes out of the elevator she waits a little time and the lights near her dim, except from one light which is further down the hall, which burns more intensely it means that is the right direction. While approaching the intense light, the light dims and a light which is further indicates now the right direction. Alice then meets a group of students that pass on the hall in the other direction and all the lights return to normal. However, it happens that to her right, on the wall, there is a small display she hears a short sound that grabs her attention and she reads on the screen "Turn right", together with an image of what she sees around her and an arrow indicating the way.

When getting close to room 42, she sees Bob, who she doesn't previously know. But near them a screen on the wall lights up and displays "Panda meeting in room 42", and the light next to the door to room 42 blinks discreetly. At the same time, Trudy reaches the same area. She is a professor that works in a different department and has just opened a document on her PDA and tries to read the small writing promptly she hears a sound that is specific to her notifications and sees a screen, different from the one Alice and Bob are using, displaying her document at an acceptable size.

In the mean time, Alice and Bob enter the room, where, even if there was no one inside, the lights are already on and the room's projector displays the welcome message.

Scenario 2. Alice is a student in Computer Science. In the afternoon, she has a lecture on project management. Usually it is held in a classical amphitheater, but today she receives a message on her PDA that it will be held in a new laboratory in her university, called the SmartRoom. The message also contains indications on how to get there.

She is a bit late and she is the last one of the 15 students to arrive. When she gets sited, all lights go down automatically, the presentation screen turns on showing the first slide of today's lecture, and the teacher starts the presentation. There are microphones for each student in the room, as well as for the teacher, but only the teacher's microphone is on for the time of the presentation. Later, during the time for questions, the students' microphone get activated as well.

The second part of the lecture is dedicated to some hands-on activity. The students are invited to choose among three activities in which to participate. Using their PDAs or their laptops, they can see their friends' preference for the activities, as well as choose the one they prefer. In the end, some groups are formed.

One activity is to read some descriptions of projects and try to present their strong and weak points. This activity only requires one or two large screens, so the SmartRoom allocates the screens for this activity, and the students move closer to them.

Another activity implies only a discussion between students, so they can sit at a table. The light above the table stays fully lit, as the lights that are closer to the screens get dimmed so that the screens are more visible.

Alice participates in the third activity, which is to design a management strategy for a project. This activity is allocated to the two large touchscreens in the room. They get activated as the students assigned to the activity come close to the screens. As they work together, adding ideas to the content on the screens, Alice starts referring concepts from some previous work of hers, for another course. The SmartRoom, in collaboration with her personal agent, detects the compatibility and asks Alice if the whole file should be made visible for the other students. Alice agrees and the file is displayed on a third screen. A notification sound is played near the screen to draw the attention of the students.

At the end of the activities, the teacher passes by each group of students and evaluates their work, using an application on his PDA. Because the teacher has poor vision, the content on the screen is automatically magnified by the application when the teacher is viewing it. After evaluation and comments, the students' work is saved, and as they go back to their seats the screens turn off automatically. Before leaving, a message on the students' PDAs asks them to give anonymous feedback on the lecture. After everybody leaves the room, the lights go out all by themselves.

Scenario 3. Al, a member of the Panda team, working on the Panda 3 project, enters office EF301 that has been assigned to the team for the time this project is taking place. The EF301 office is one of the intelligent rooms of the university. The AmI system detects the movement and, based on previous observations, determines that there is on more person in the room. As it is customary, the lights grow a little brighter (the team usually works in semi-obscurity – it is their preference, so they can focus better on their work) so that other members of the team can see who has joined them and they are able to greet each other. Al gets seated at his computer and places his phone on the desk. By identifying the phone, and also the face of the user (the computer features a webcam) the system identifies Al and shows him his desktop (synchronized from the cloud), but only some windows are initially selected for display – those related to his work with the Panda team, and those he usually need when he is in this office.

As the team works intensely, some start getting hot (although the temperature in the room remains constant) and they notify that to the system using gadgets on their screens. The system aggregates their preference and also considers the hierarchy in the team in order to calculate the new appropriate room temperature.

> At a certain point, a notification tells them it is time to have a meeting. They all gather around the interactive table display, the lights go brighter, and audio recording of the discussion begins. Relevant information for the discussion are taken from the members' files and loaded into the interface of the table display. After the meeting, the members go home and all notes and modifications that were made during the meeting will be available in their personal workspaces when they will take on work again.

In the scenarios above we see two things: first, the use of devices that exist in the present day, used (many times working together) as intelligent, intuitive interfaces; second, the scenarios consider the existing of multiple users, and the changes that the AmI system needs to perform in order to adapt to that. *Preference aggregation* is one of the central features to this approach. Also the use of multiple means of notification, that are changed when the presence of other users requires it.

One issue that has been little discussed in the scenarios of this section is anticipation. Let us give an example of simple anticipative behaviour that can make a huge difference in the life of users:

> It is dark outside, and Celia is coming through the hall toward her office. She enters the office and the lights are already on. She knows that the lights are not on when she is not there: it is the AmI system that turns them on, anticipatively, before she enters the office. When she will leave, the lights will go out right after she closes the door, but nobody will see that. People are already used to finding the lights on in all rooms they enter.

3.5 Conclusion

Ambient Intelligence is a vast domain of research, that includes issues from many fields, among which most important are intelligent interfaces, artificial intelligence, context-awareness, knowledge representation, etc.

The many scenarios in this section show that there are very many aspects that a true AmI system should cover. Moreover, as the scenarios in section 2 seem to describe it, a true AmI system should be, as the Internet is not quite today, unified, and all services should be offered by means of the same type of intuitive interfaces – and it will be unification that makes the system more intuitive to use.

Among the features that we have tried to emphasize in this section are the use of context-awareness to detect associations and thus trace problems and detect solutions; the approach to context-awareness as proximity between situations; the need to choose among various means of communication and notification, according to the situation and to the context of other users; preference aggregation when multiple users use the same feature; anticipation as intuitive pro-activity in the assistance of users; the need for the system to scale so that it can be used by a very large number of people without losing its utility.

4 Modeling Context

Context-awareness is a central issue in the field of Ambient Intelligence. Pro-active, but non-intrusive behaviour would not be possible without a proper understanding from the side of the AmI system of the user's context. Actions of the system must appear to be natural and well integrated in the current situation.

Context has been defined as: "Any information that can be used to characterize the situation of entities (i.e. a person, a place or an object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves" [Dey01]. Therefore, context-awareness is not only the ability to adapt the system's reaction to the current situation, but also to decide the action to be taken by looking at the user's context.

Many authors consider context as relating almost exclusively to location, time, and other instantaneous properties of the physical environment. But there is more to context than that. First, there are more types of context – e.g. computational, temporal, user-related [CK00]. Second, context is not only formed by the properties of said context types – like where the user is located, what time it is, what is the temperature outside and what capabilities the current network connection has – context is also defined by associations between various facts that relate to the user, facts which are not necessarily contextual information of the said types. For instance, it would probably be unwise to disturb a researcher with unimportant messages on the last day before a conference's deadline. While this decision is context-aware, it is something that does not relate almost at all with any properties of the physical, computational or social environment of the user, nor to its profile or personalization options.

In our work, it is this type of context-awareness that we are trying to implement, that is based on the detection of associations and similarity between various pieces of knowledge. For this, we have defined [OF10] notions like context graphs and context patterns. Every agent receives from other agents information that it tries to integrate in its own graph, that represents all the information it has about its user. Every agent in the multi-agent system also uses patterns to detect information that is relevant to it and also to detect problems, solution to problems, and context-appropriate action: information that does not match any pattern cannot be used, therefore is not relevant; partial matches of patterns mean that pieces of information are missing, but other matches may provide potential solutions; finally, unsolved problems lead to the agent taking action that is relevant and useful in the current context.

4.1 Approaches to Context-Awareness

Previous work discussing context-awareness in the context of Ambient Intelligence applications generally revolves around two issues: on the one hand, the infrastructure for capturing and processing context information (where context information is of physical nature); on the other hand, the modeling of and the reasoning on context information.

The proposed infrastructures [HL01, HHS$^+$02, LW05, HI06, BDR07, FAJ04] usually contain several layers: sensors capture information from the environment; there is a layer for the preprocessing of that information; a layer for its storage and management; and finally the layer of the application that uses the context information [BDR07]. This type of infrastructures is useful when the context information comes from the environment and refers to environmental conditions like location, temperature, light or weather. However, physical context is only one aspect of context [CK00]. Moreover, these infrastructures are usually centralized, using context servers that are queried to obtain relevant or useful context information [DAS99, LW05]. In our approach we attempt to build an agent-based infrastructure that is decentralized, in which each agent has knowledge about the context of its user, and the main aspect of context-awareness is based on associations between different pieces of context information. All agents have a more or less equal role in the transfer of context information.

Representing context information is done in most previous works by means of tuples, logic, case-based reasoning or ontologies [PRL09, SLP04]. These are used to determine the situation that the user is in. Henricksen et al use several types of associations as well as rule-based reasoning to take context-aware decisions [HI06, BBH+10]. While these representations are good for defining the elements that exist in the representation of context as concepts in an ontology, defining the situation by means of case-based reasoning or rules is not very flexible in the context of ever-changing, and fast-changing environments and situations. In this paper we propose a more loose, but more simple, more flexible and easy-to-adapt dynamical representation of context information, inspired from concept maps and conceptual graphs. While our representation lacks the expressive power of ontologies in terms of restrictions, a graph-based representations is very flexible and extensible, so support for restriction may be added as future work.

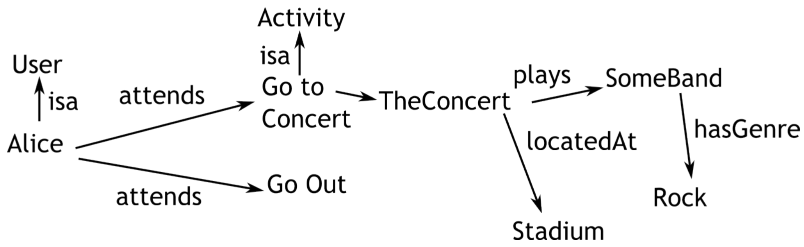
Our approach to context representation is rooted in existing knowledge representation methods like semantic networks, concept maps [NC06] and conceptual graphs [Sow00]. These structures can be used to describe situations (and context) in a more flexible manner and using less memory than ontological representations. While graph matching has been previously used, for instance for image processing [BLB+02], we attempt to use it for the matching of context graphs, also improving the graphs by means of special notation elements that allow the definition of patterns.

There has been a significant body of work in the domain of ontology alignment, which is vital for a viable implementation of Ambient Intelligence systems [VMC+08]. However, this is not the subject of this work. We assume that all agents in the system work with terms from the same ontology (where it is the case), or that ontologies have already been aligned.

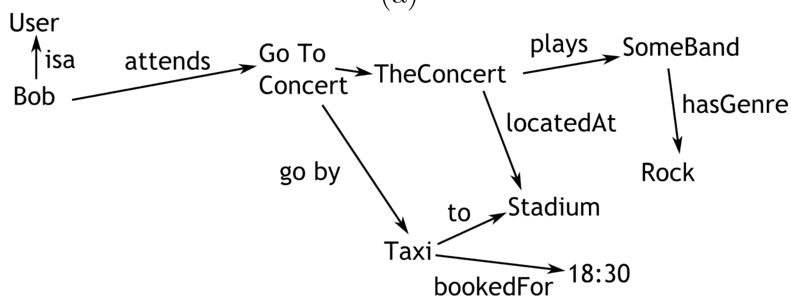
4.2 Context Graphs and Patterns

Let us work on the scenario featuring Alice and Bob, from Section 3.3. Each user of AmIciTy has an associated agent. Figure 1 (a) and (b) show the concept graph for the knowledge of the two agents (agent A for Alice and agent B or Bob) that is relevant to the scenario. Formally, the knowledge of each agent can be represented as a graph [OF10]:

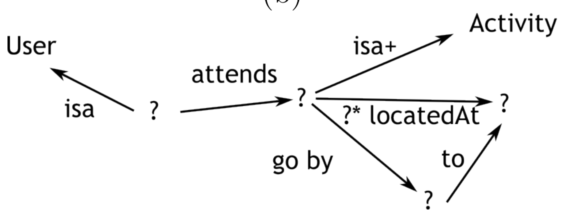
$G = (V, E)$



(a)



(b)



(c)

Figure 1: The knowledge of Alice's and Bob's agents, respectively: (a) Alice will go to a concert where a rock band is playing and which is located at the stadium, and then to go out with friends; (b) Bob will be going at the same concert, and has also booked a taxi to get there. Part of the relations and concepts that appear in the graphs may come from ontologies. (c) A pattern that says that if the user attends an activity that has a location, then the location of the activity should be reached in some way.

$V = \{v_i\}, E = \{e_k\}, e_k = (v_i, v_j, value)$
where $v_i, v_j \in V, i, j = \overline{1, n}, k = \overline{1, m}$

The values of vertices and edges can be either strings or, better, URI identifiers that designate concepts, relations, people, etc. The value of an edge may be null.

The graph that an agent has contains the knowledge that the agent has about the user and about the user's context. The graph represents the context of the user, in the measure in which the agent has perceived it (or been informed of by the user or by another agent).

First, we want the system to detect the fact that it is necessary to know how Alice will be getting at the concert. This can be done by means of the following pattern: if the user intends to attend something that is an activity, and that has a location (it's not, for instance, making a phone call), then there should also be a means for the user to reach that location. The pattern is represented in Figure 1 (c).

A *pattern* is also a graph, but there are several additional features that makes it match a wider range of situations. The graph for a pattern s is defined as:

$G_s^P = (V_s^P, E_s^P)$

$V_s^P = \{v_i\}, v_i = string \mid URI \mid ?, i = \overline{1,n}$
$E_s^P = \{e_k\}, e_k = (v_i, v_j, E\_RegExp), v_i, v_j \in V_s^P, k = \overline{1,m}$
where E_RegExp is a regular expression formed of strings or URIs.

A pattern represents a set of associations that has been observed to occur many times and that is likely to occur again. Patterns may come from past perceptions of the agent on the user's context or be extracted by means of data mining techniques from the user's history of contexts. Commonsense patterns may come from public databases, and patterns may also be exchanged between agents. However, the creation or extraction of patterns is not the subject of this paper.

The agent has a set of patterns that it matches against the current context (graph G). We will mark with the $\cdot^P$ superscript the graphs or vertex / edge sets that contain special pattern features (like ? nodes, for instance).

4.3 Context Matching

Each agent in the multi-agent system has a set of patterns that it matches against the current context (graph G). A pattern G_s^P *matches* a subgraph $G' \subseteq G$, with $G' = (V', E')$ and $G_s^P = (V_s^P, E_s^P)$, iff an injective function $f : V_s^P \rightarrow V'$ exists, so that the two conditions below are fulfilled [OF10, OSF11]:

(1) $\forall v^P \in V_s^P, value(v^P) = value(f(v^P))$ or $v^P =?$
and
(2) $\forall e^P \in E_s^P, e^P = (v_i^P, v_j^P, value)$ we have:
(2a) if $value$ is a string or an URI, $edge(f(v_i^P), f(v_j^P), value) \in E'$ or $edge(f(v_i^P), f(v_j^P))$ $\in E'$ (unlabeled edge)
(2b) if $value$ is a pair of regular expressions, then E_RegExp matches the values $value_0, value_1, ..., value_p$ of a series of edges $e_0, e_1, ..., e_p \in E'$, where
$e_0 = (f(v_i^P), v_{a_0}, value_0)$,
$e_k = (v_{a_{k-1}}, v_{a_k}, value_1), k = \overline{1, p-1}$
$e_p = (v_{a_{p-1}}, f(v_j^P), value_p)$,
where $v_{a_l} \in V'$ and $v_{a_l} \notin image(f)$ and the values of $v_{a_0}, ..., v_{a_{p-1}}$ match V_RegExp.

In other words, every non-? vertex from the pattern must match a different vertex from G'; every non-RegExp edge from the pattern must match an edge from G'; and every regular expression edge from the pattern must match a series (that can be void, if the expression allows it) of edges from G'. Subgraph G' should be *minimal*. A graph (or subgraph) G' is minimal with respect to a matching pattern G_s^P iff there is no edge in G' that is not the match (or part of the match) of an edge in G_s^P.

A pattern G_s^P *k-matches* a subgraph G' of G, if condition (2) above is fulfilled for $m - k$ edges in E_s^P, $k \in [1, m-1]$, $m = ||E_s^P||$ and G' remains connected and minimal. The relationship of *k-matching* should be interpreted as *matching except for k edges*. Non-matching vertexes inherently imply non-matching edges.

Once the agent has context and patterns represented as graphs, context matching can answer three questions: Is new information relevant to the current context? Is information we have complete, from our experience? Is information that we have able to provide

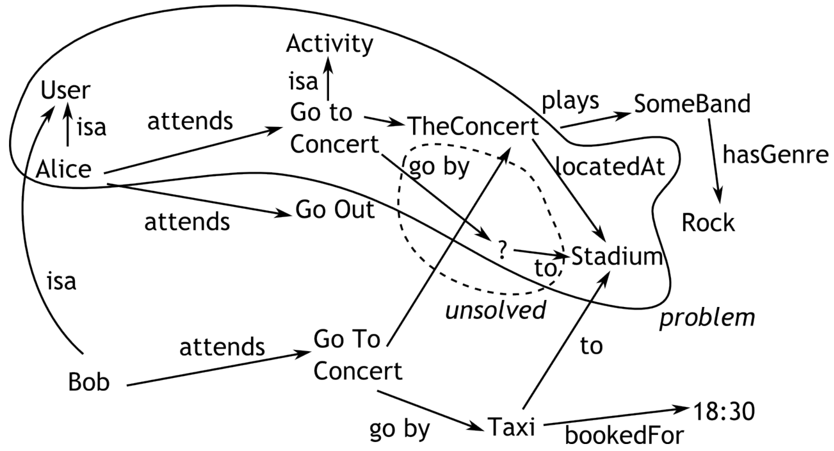


Figure 2: The knowledge base of agent A, completed with the information on Bob's agenda. Also the problem and its unsolved part are circled with a continuous and a dashed line respectively. Although the unsolved part is displayed together with the rest of the context graph, it is not a concrete or known fact so it would not be used in pattern-matching.

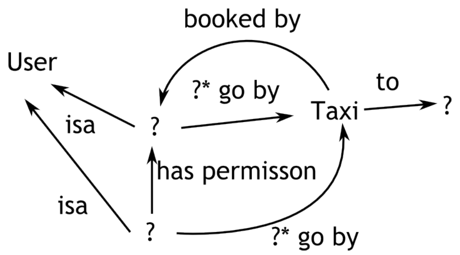


Figure 3: A second pattern, specifying that two people can use the same taxi to get to the same location if the person who has not booked the taxi has permission from the other to ride the same taxi.

answers to problems?

The first question can be answered in two ways: either match the new information against the current context, to find if there are any common points – if there are none, the agent cannot connect new information to what it knows; or it can match patterns that the agent has against the new information, to see if information is actionable, therefore relevant.

The second question is answered by matching patterns against the current context and finding parts that are missing (incomplete matches). The third question is answered by matching the patterns that do not fully fit some pieces of information, against other pieces of information, trying to find possible solutions.

4.4 Problem Solving

A *Problem* is a graph G^P that contains features that are specific for patterns (like ? nodes for instance) and that is a partial instantiation of a pattern G_s^P, according to the current context. A problem G^P is the union between the subgraph G' (of the context graph G) that k-matches pattern G_s^P and the part of G_s^P that is not matched by G'. The latter is the *unsolved* part of the problem. A problem also remains associated with the pattern that generated it. Therefore, formally, if a pattern $G_s^P = (V_s^P, E_s^P)$ k-matches the subgraph $G' = (V', E')$ of G, we can define a problem p as a tuple (G_s^P, G_p^P), where G_p^P is the problem's graph:

$$G_p^P = G' \cup G_x^P$$
$$G_x^P = (V_x^P, E_x^P)$$
$$V_x^P = \{v \in V_s^P, v \notin dom(f)\}$$
$$E_x^P = \{e \in E_s^P for\ which\ condition\ (2)\ is\ not\ fulfilled\}$$

Note that G_x^P (the unsolved part of the problem) is a subgraph of G_s^P. Also note that the unsolved part may contain edges whose vertices are not both in the unsolved part. The *problem* from our example is circled in Figure 2 with a continuous line, and its unsolved part is circled with a dashed line.

The agents in AmIciTy are not single agents. They are part of a multi-agent system. Agents A and B communicate frequently due to the fact that Alice and Bob live in the same place and exchange a lot of data. At some point in this communication, they exchange data about Alice's and Bob's agendas, which is normal for two people that share an apartment. Agent B will send the subgraph $agenda \to Concert$ and A will send $agenda \to Concert/ \to Go\ Out$ (agent A will only send the *GoOut* activity if Alice has not designated it as private).

Agent A receives the subgraph $agenda \to Concert$ and matches it against Alice's context, detecting the compatibility (a full match). So it responds by building upon this common context: it sends a larger subgraph, containing the band playing at the concert, as well as the location of the concert. Agent B does the same operations as A (they share the same context regarding the concert, so each one's context matches the other one's), just that it also sends to A the associations $Concert - go\ by \to Taxi - to \to Stadium$.

The communication between agents as described above is done based on shared context. Starting from sharing their agendas, at each step agents detect matches between the two contexts and respond with a subgraph that is larger with one level (breadth-first).

All the data that agent A has about other agents (here, agent B) is stored in the agent's knowledge base as its model of the other users. The model for the other users is not necessarily separate though: if the same concept appears in both models (provided the concept ha the same URI, or the agent is able to detect by means of common sense knowledge that it is the same concept), both subgraphs will contain the corresponding node. Figure 2 shows the knowledge of agent A regarding users Alice and Bob. The model for Bob's agenda contains the same *Concert* node that is contained in the graph for Alice. When matching patterns from its pattern set, A detects that the pattern mentioned above fully matches the model for Bob. Agent A also has a *problem* that is linked to this pattern. Since Bob's context fully matches the pattern, it means it may be a *solution* to Alice's problem: Alice may also use a taxi to reach the concert. But that would mean booking a different taxi (use a different instance of the concept).

Another pattern may be used in this context: agent A may know that two people may share the same taxi to get to the same destination, if one has permission from the person that booked the taxi (we have somewhat simplified the problem and we do not mention that the two people must leave from the same location and need to reach the destination at the same time). Matching this pattern against the knowledge of agent A about Alice's context in Figure 2 (remember that unsolved parts are not matched), a 2-match is obtained (missing relations are *has permission* and Alice's *go by*). Not only that, but adding those relation would solve the problem that Alice has. Therefore, the agent can suggest to Alice to ask permission from Bob to use the same taxi.

In this particular case, with the given knowledge and patterns, there is only one solution to the problem that arose. But in a more realistic case, where context is more complex and there are more patterns, more solutions to the same problem may be found. In case they fit equally well in the current context, then the agent must prompt the user with all of them and the user must be given the choice.

It can be argued that context-matching is a very difficult problem in the case of large graphs and complex situations. However, resource-constrained devices will work only with smaller pieces of context information (i.e. smaller graphs), that are relevant to their function. We will examine an algorithm for context matching in section

Another problem that may appear in realistic situations (as opposed to our simple example) is the abundance of simultaneous matching context patterns, possibly describing contradictory situations. This is where more refined measures must be found that will allow calculating the relevance of each match. This too will be part of our future work.

4.5 A Context Matching Algorithm

In order to perform the operation of context matching, we have devised a matching algorithm. The algorithm returns, for a graph G and a pattern G_s^P, the subgraph(s) G' (G'_i) of G that fully match(es) the pattern G_s^P or, should no such subgraph exist, the subgraph(s) G' (G'_i) of G that k-match(es) G_s^P, for the minimal existing k (the full match has $k = 0$).

The matching algorithm is described in the following paragraphs (see also Figure 4). We consider that there is known a node $v_M^P \in V_s^P$ which is one of the nodes with the maximum difference between its out-degree and its in-degree.

First, create a queue $MatchQueue$. $MatchQueue$ will contain parts of the pattern that partially match the G graph, as *matches* – tuples (G'_i, G_{xi}^P) formed of the k-matching subgraph of G and the not matching part of the pattern: $G'_i \subseteq G$ and $G_{xi}^P = (V_{xi}^P, E_{xi}^P)$:

$V_{xi}^P = \{v \in V_s^P, v \notin dom(f)\}$ for f the matching function;
$E_{xi}^P = \{e \in E_s^P \textit{ for which condition (2) in the matching of pattern } G_i^P \textit{ is not fulfilled}\}$.

Next, for each edge e_k^P in E_s^P that is a non-*RegExp* edge and that has one or more matches in graph G, add one or more *matches* $m = (e_r, G_s^P \setminus \{e_k^P\})$ in $MatchQueue$ (for easier understanding, in the figure a match also contains the matching part of the pattern), where e_r is the edge (or one of the edges) in G that match e_k^P, i.e. $value(e_k^P) = value(e_r)$. Edges in $MatchQueue$ are primarily sorted according to distance from v_M^P (ascending) and also according to distance to the closest leaf (also ascending).

The next step in matching is to try to grow the existing single-element matches to cover

```
Match(G, G^P) (G = (V, E), G^P = (V^P, E^P)):                      // matching takes a graph and a pattern
  MatchQueue ← ∅
  v_M^P ← v_i^P with max(outdegree(v_i^P) − indegree(v_i^P))
  for each e_k^P = (v_i^P, v_j^P, val) ∈ E^P
    if val = string or val = URI                                    // the value is fixed
      for each e_r = (v_s, v_t, val') ∈ E
        if (val = val' or val' = ∅) and (v_i^P and v_j^P match v_s and v_t)
          MatchQueue ← MatchQueue ∪
              ({e_r}, {e_k^P, v_i^P, v_j^P}, G^P \ {e_k^P, v_i^P, v_j^P})   // add single-edge match to queue
  sort MatchQueue by distance from v_M^P and by distance to leafs  // best is closest to v_M^P and closest to a leaf

  while MatchQueue ≠ ∅
    m = (G', G_m^P, G_x^P) ← extract(MatchQueue)                    // the matching subgraph, the matching part
                                                                    of pattern, the unmatched part of pattern
    for each e_k^P = (v_i^P, v_j^P, val) ∈ G_x^P
      if v_i^P ∈ G_m^P                                              // edge is outgoing from the matched part
        if e_k^P ∈ m^1, with m^1 ∈ MatchQueue ∪ Matches             // the edge has already been matched
          if G_m^P ∪ e_k^P matches G'', with G' ⊆ G'' ⊆ G           // full match required
              and m can merge with m^1                              // matches are compatible
            m' ← merge(m, m^1)                                      // merge the two matches
            MatchQueue ← MatchQueue ∪ m'                            // (keep MatchQueue primarily sorted by k)
        else
          find a match m^1 for e_k^P, starting from v_i^P           // v_i^P is already matched to a vertex in G
          if m can merge with m^1                                   // matches are compatible
            m' ← merge(m, m^1)                                      // merge the two matches
            MatchQueue ← MatchQueue ∪ m'
    for each e_k^P = (v_i^P, v_j^P, val) ∈ G_x^P
      if v_j^P ∈ G_m^P                                              // edge is incoming in the matched part
        same as for outgoing edges
    remove m from MatchQueue
    Matches ← Matches ∪ m                                           // keep sorted by k
  return Matches
```

Figure 4: The matching algorithm for a graph and a pattern.

most of graph G: for each match $m = (G'_i, G^P_{xi})$ in $MatchQueue$ all edges that are in G^P_{xi} and that are outgoing from $G^P_s \setminus G^P_{xi}$ (i.e. outgoing from the part that already matches the graph) are explored, and matches are attempted with the outgoing edges of G'_i. Matching edges and nodes are added to m. Edges already in $MatchQueue$ are tried first and, if matching, their match is merged with m. The same is done for incoming edges. If no more edges or nodes can be added to the match, the match is removed from $MatchQueue$ and added to the $Matches$ list.

The algorithm ends when there are no more elements in $MatchQueue$. It returns the list $Matches$, sorted ascending according to the k of the matches.

Let us present a very simple example. While this is not an AmI scenario, it emphasizes the idea of context-matching, which will be used in future work as part of an AmI system, and that will help agents select and share relevant information. Suppose that an Ambient Intelligence system, implemented by means of a multi-agent system, is helping researcher Alex in organizing his information on the computer. Among that information, he has received, by e-mail, a call for papers for the Artificial Intelligence Conference, or AI-Conf. The CFP contains the time when the conference will take place, the deadline for articles, and the date the CFP was issued. Although Alex read the e-mail, he did not take the time to organize the information inside it. What the AmI system was capable of extracting so far (possibly by means of pattern-matching) are the fact that the CFP is a document, that it contains 2 dates and an interval of time, and that it is about something called AI-Conf (a name that appears throughout the document); it also checked automatically the page

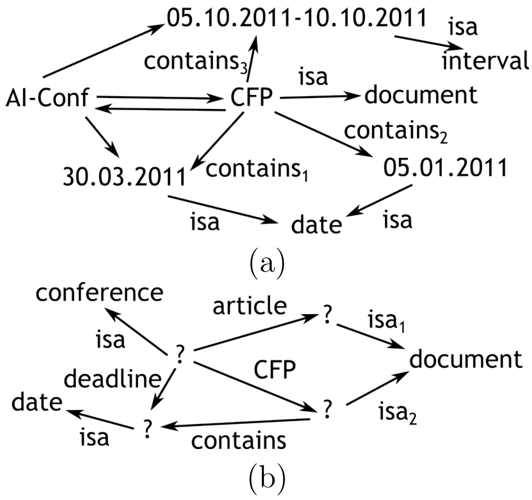


Figure 5: An example of context graph (a) and context pattern (b). Some relations have been annotated with subscripts to be able to differentiate them in the text.

linked in the document and the page too contained one of the dates (the deadline) and the interval in which the conference will take place (see also Figure 5 (a)).

Alex also owns a very useful device (similar to a tab [Wei93]) that has a very simple function: to show just the most relevant information about the next potentially interesting conference in the field of AI – let's call the device ConfCompanion. We will not focus on how to choose the relevant conference and we will presume the AI-Conf is the most relevant conference at the time. However, the job of the ConfCompanion is still not very simple: when it receives the information about the conference, it needs to be able to parse it and find the deadline of the conference and to detect the fact that Alex needs to send an article. This is difficult because while the pattern is defined by relationships, the context information only contains names for its nodes – but the nodes are not named in the pattern.

The elements of the scenario have been simplified in order to only illustrate very briefly the idea of the paper. As a side note, while this scenario does not imply multiple users, context matching is an essential part of the communication between agents in the system, in general.

In order for ConfCompanion to perform its duty, it will be using a pattern. How this pattern was extracted is not the focus of this paper, but it is safe to assume it has been mined form Alex's previous experience. This pattern is presented in Figure 5 (b). We will describe how the algorithm presented in Section 4.5 will work.

First, v_M^P will be set to the ? node that is the placeholder for the conference. Next, $MatchQueue$ will be initialized with the following matches, in their respective order: $?{-}isa_1 \rightarrow document$ matches $CFP{-}isa \rightarrow document$; $?{-}isa_2 \rightarrow document$ also matches $CFP - isa \rightarrow document$; $? - isa \rightarrow date$ matches $05.01.2011 - isa \rightarrow date$, and also $30.03.2011{-}isa \rightarrow date$; finally, $?{-}contains \rightarrow ?$ matches $CFP{-}contains \rightarrow 05.01.2011$, $CFP - contains \rightarrow 30.03.2011$ and $CFP - contains \rightarrow 05.10.2011 - 10.10.2011$.

First the algorithm will try to expand the first match: isa_1 with $CFP\ isa\ document$. There are no outgoing edges, but it is possible to match $? - article \rightarrow ?$ to $AI\text{-}Conf \rightarrow$

CFP. From there, it tries to expand with $? - deadline \rightarrow ?$ that matches *AI-Conf* $\rightarrow$ 30.03.2011 (expanding $? - CFP \rightarrow ?$ will not yield good results: in the context there are no more documents); next, the match is expanded with $? - isa \rightarrow date$ and $30.03.2011 - isa \rightarrow date$. No more expansions are possible, and the obtained match (containing nodes *AI-Conf*, CFP and 30.03.2011) has 4 matching edges, leading to a 4-match (there are 4 edges missing from the pattern).

Next, another possibility is to match isa_2 with CFP *isa document*. This time there is an outgoing edge: $? - contains \rightarrow ?$ matches, in a first case, $CFP - contains \rightarrow$ 05.01.2011; next, the $isa \rightarrow date$ matches. Then going to incoming edges, $? - CFP \rightarrow ?$ matches *AI-Conf* $\rightarrow CFP$. The algorithm would try to match $? - deadline \rightarrow ?$ with *AI-Conf* $\rightarrow$ 30.03.2011, but unfortunately the pattern has already been matched to a different date. Once again, a 4-match is obtained.

But another case is when, continuing after the first match in the previous case, $? - contains \rightarrow ?$ is matched to $CFP - contains \rightarrow$ 30.03.2011. In this case, after the $isa \rightarrow date$ and CFP matches are done, it is possible to match $? - deadline \rightarrow ?$ with *AI-Conf* $\rightarrow$ 30.03.2011. No further extensions are possible, but a 3-match has been obtained, which is better than the other two. Not only that, but an external observer can see it is the best match.

Further on, ConfCompanion can identify that indeed *AI-Conf* is a conference, and also that the *article* element is missing, so it decides to notify the user about that.

5 Conclusion

Many existing Ambient Intelligence prototypes, like the AmI vision in general, are based on scenarios that describe the features and requirements of the system. Most times the scenarios mix features from all layers of ambient intelligence: hardware, network, context-aware services and interfaces. Similarly, many implementations of AmI environments deal with features from all of the layers, not being focused on one single layer, and unfortunately losing generality and scalability in the process.

This research focuses on only one layer of the system – the context-aware services layer – and attempts to build a MAS-based infrastructure that uses agents to exchange information and transport relevant information to the interested users, as well as acting upon this information to assist users and solve problems. As past contributions of this research, a multi-agent system was built for the context-aware sharing of information, based on local behaviour and communication, and an multi-agent architecture that links system structure and context has been proposed, implemented and demonstrated [Ola10b].

As a first contribution of this report, we have devised several scenarios that emphasize the requirements of the context-aware services layer. These scenarios emphasize features like the need for flexibility and scalability in the context of different users and considering a large number of users; adaptation of the means communication and notification to the requirements of the environment and to the context and needs of the user; problem solving by means of information sharing; the support of multiple users with different preferences; the support for collaborative work. These scenarios will be the base of the test scenarios that will be used for validation of this research.

A second contribution of this research relates to the definition of context graphs and context patterns, as well as to the algorithm for context matching. The need for these notions has been noticed in our previous research [Ola10b]. Context graphs are a flexible and generic means of representation of knowledge related to the user – the user's context. By matching context patterns against context graphs, agents in an Ambient Intelligence system can answer three questions: is new information relevant for the agent? Are there any missing pieces of information that may be relevant to the agent? If there are, can these pieces of information be found by looking into the knowledge related to similar situations?

6 Future Work

Research in the immediate future relates, primarily, to the integration of the solutions that were developed so far: large numbers of locality-oriented agents sharing information, structuring of the multi-agent system according to context, and defining context as a graph and matching context patterns to solve problems. The result of this integration will be tested by using scenarios based on the ones presented in section 3.

Also as part of the future work is the more concrete definition of the interaction between the multi-agent system and the rest of the AmI infrastructure, and potentially a more formal definition of an AmI environment in general.

While the features that were developed during this research are designed bearing in mind the heterogeneity of devices in terms of processing and storage capabilities, actually tuning the agent's behaviour and stored knowledge in function of these capabilities has yet to be done.

Acknowledgments

This work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/6/1.5/S/16. Parts of this work have also been supported by Laboratoire d'Informatique de Paris 6 (LIP6), UPMC, Paris, France.

References

[AM07] Juan Carlos Augusto and Paul J. McCullagh. Ambient intelligence: Concepts and applications. *Computer Science and Information Systems (ComSIS)*, 4(1):1–27, 2007.

[BB02] G. Banavar and A. Bernstein. Software infrastructure and design challenges for ubiquitous computing applications. *Communications of the ACM*, 45(12):92–96, 2002.

[BBH+10] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.

[BCL+05] J. Bohn, V. Coroama, M. Langheinrich, F. Mattern, and M. Rohs. Social, economic, and ethical implications of ambient intelligence and ubiquitous computing. *Ambient intelligence*, pages 5–29, 2005.

[BDR07] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[BLB+02] Endika Bengoetxea, Pedro Larrañaga, Isabelle Bloch, Aymeric Perchant, and Claudia Boeres. Inexact graph matching by means of estimation of distribution algorithms. *Pattern Recognition*, 35(12):2867–2880, 2002.

[BPG+09] J. Bauchet, H. Pigot, S. Giroux, D. Lussier-Desrochers, Y. Lachapelle, and M. Mokhtari. Designing judicious interactions for cognitive assistance: the acts of assistance approach. *Proceeding of the eleventh international ACM SIGACCESS conference on Computers and accessibility*, pages 11–18, 2009.

[CK00] Guanling Chen and David Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dartmouth College, November 2000.

[CMTT08] S. Costantini, L. Mostarda, A. Tocchio, and P. Tsintza. DALICA: Agent-based ambient intelligence for cultural-heritage scenarios. *IEEE Intelligent Systems*, 23(2):34–41, 2008.

[DAS99] A.K. Dey, G.D. Abowd, and D. Salber. A context-based infrastructure for smart environments. *Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99)*, pages 114–128, 1999.

[DBS+01] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.C. Burgelman. Scenarios for ambient intelligence in 2010. Technical report, Office for Official Publications of the European Communities, February 2001.

[Dey01] A.K. Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.

[EFSONS10] Amal El Fallah Seghrouchni, Andrei Olaru, Thi Thuy Nga Nguyen, and Diego Salomone. Ao dai: Agent oriented design for ambient intelligence. In *Proceedings of PRIMA 2010, the 13th International Conference on Principles and Practice of Multi-Agent Systems*, 2010.

[FAJ04] Ling Feng, Peter M. G. Apers, and Willem Jonker. Towards context-aware data management for ambient intelligence. In Fernando Galindo, Makoto Takizawa, and Roland Traunmüller, editors, *Proceedings of DEXA 2004, 15th International Conference on Database and Expert Systems Applications, Zaragoza, Spain, August 30 - September 3*, volume 3180 of *Lecture Notes in Computer Science*, pages 422–431. Springer, 2004.

[HCC+04] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman. Creating an ambient-intelligence environment using embedded agents. *IEEE Intelligent Systems*, pages 12–20, 2004.

[Hel05] Michael Hellenschmidt. Distributed implementation of a self-organizing appliance middleware. In Nigel Davies, Thomas Kirste, and Heidrun Schumann, editors, *Mobile Computing and Ambient Intelligence*, volume 05181 of *Dagstuhl Seminar Proceedings*, pages 201–206. ACM, IBFI, Schloss Dagstuhl, Germany, 2005.

[HHS+02] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. *Wireless Networks*, 8(2):187–197, 2002.

[HI06] K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing*, 2(1):37–64, 2006.

[HK04] M. Hellenschmidt and T. Kirste. A generic topology for ambient intelligence. *Lecture notes in computer science*, pages 112–123, 2004.

[HL01] J.I. Hong and J.A. Landay. An infrastructure approach to context-aware computing. *Human-Computer Interaction*, 16(2):287–303, 2001.

[JFW02] B. Johanson, A. Fox, and T. Winograd. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE pervasive computing*, pages 67–74, 2002.

[KBM+02] T. Kindberg, J. Barton, J. Morgan, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frid, V. Krishnan, H. Morris, J. Schettino, and B. Serra. People, places, things: Web presence for the real world. *Mobile Networks and Applications*, 7(5):365–376, 2002.

[LW05] Till Christopher Lech and Leendert W. M. Wienhofen. AmbieAgents: a scalable infrastructure for mobile and context-aware information services. *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 625–631, 2005.

[NC06] Joseph D. Novak and Alberto J. Cañas. The origins of the concept mapping tool and the continuing evolution of the tool. *Information Visualization*, 5(3):175–184, 2006.

[OEFSF10] Andrei Olaru, Amal El Fallah Seghrouchni, and Adina Magda Florea. Ambient intelligence: From scenario analysis towards a bottom-up design. In Mohammad Essaaidi, Michele Malgeri, and Costin Badica, editors, *Proceedings of IDC'2010, the 4th International Symposium on Intelligent Distributed Computing*, volume 315 of *Studies in Computational Intelligence*, pages 165–170. Springer, 2010.

[OF10] Andrei Olaru and Adina Magda Florea. A graph-based approach to context matching. *Scalable Computing: Practice and Experience*, 11(4):393–399, December 2010. ISSN 1895-1767.

[OGF10] Andrei Olaru, Cristian Gratie, and Adina Magda Florea. Context-aware emergent behaviour in a MAS for information exchange. *Scalable Computing: Practice and Experience - Scientific International Journal for Parallel and Distributed Computing*, 11(1):33–42, March 2010. ISSN 1895-1767.

[Ola10a] Andrei Olaru. A context-aware multi-agent system for AmI environments. Technical report, University Politehnica of Bucharest, University Pierre et Marie Curie Paris, 2010. February.

[Ola10b] Andrei Olaru. Towards a MAS-based model for ambient intelligence. Technical report, University Politehnica of Bucharest, University Pierre et Marie Curie Paris, 2010. October.

[OSF11] Andrei Olaru, Amal El Fallah Seghrouchni, and Adina Magda Florea. Graphs and patterns for context-awareness. In *Proceedings of International Symposium on Ambient Intelligence, University of Salamanca (Spain), 6-8th April*, 2011. accepted for publication.

[PRL09] M. Perttunen, J. Riekki, and O. Lassila. Context representation and reasoning in pervasive computing: a review. *International Journal of Multimedia and Ubiquitous Engineering*, 4(4):1–28, October 2009.

[RAS08] C. Ramos, J.C. Augusto, and D. Shapiro. Ambient intelligence - the next step for artificial intelligence. *IEEE Intelligent Systems*, 23(2):15–18, 2008.

[Sat01] Mahadev Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal communications*, 8(4):10–17, 2001.

[SGK05] N.M. Sadeh, F.L. Gandon, and O.B. Kwon. Ambient intelligence: The MyCampus experience. Technical Report CMU-ISRI-05-123, School of Computer Science, Carnagie Mellon University, July 2005.

[SLP04] T. Strang and C. Linnhoff-Popien. A context modeling survey. *Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp*, pages 1–8, 2004.

[SM06] N.I. Spanoudakis and P. Moraitis. Agent based architecture in an ambient intelligence context. *Proceedings of the 4th European Workshop on Multi-Agent Systems (EUMAS'06), Lisbon, Portugal*, pages 1–12, 2006.

[Sow00] J.F. Sowa. *Knowledge representation: logical, philosophical, and computational foundations*. MIT Press, 2000.

[VMC+08] J. Viterbo, L. Mazuel, Y. Charif, M. Endler, N. Sabouret, K. Breitman, A. El Fallah Seghrouchni, and J.P. Briot. Ambient intelligence: Management of distributed and heterogeneous context knowledge. *CRC Studies in Informatics Series. Chapman & Hall*, pages 1–44, 2008.

[VRV05] M. Vallée, F. Ramparany, and L. Vercouter. A multi-agent system for dynamic service composition in ambient intelligence environments. *Advances in Pervasive Computing, Adjunct Proceedings of the Third International Conference on Pervasive Computing (Pervasive 2005)*, pages 1–8, 2005.

[Wei93] M. Weiser. Some computer science issues in ubiquitous computing. *Communications - ACM*, pages 74–87, 1993.

[Wei95] M. Weiser. The computer for the 21st century. *Scientific American*, 272(3):78–89, 1995.