International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:3, 2008

Context Modeling and Context-Aware Service Adaptation for Pervasive Computing Systems

Moeiz Miraoui, Chakib Tadj and Chokri ben Amar

***Abstract*—**Devices in a pervasive computing system (PCS) are characterized by their context-awareness. It permits them to provide proactively adapted services to the user and applications. To do so, context must be well understood and modeled in an appropriate form which enhance its sharing between devices and provide a high level of abstraction. The most interesting methods for modeling context are those based on ontology however the majority of the proposed methods fail in proposing a generic ontology for context which limit their usability and keep them specific to a particular domain. The adaptation task must be done automatically and without an explicit intervention of the user. Devices of a PCS must acquire some intelligence which permits them to sense the current context and trigger the appropriate service or provide a service in a better suitable form. In this paper we will propose a generic service ontology for context modeling and a context-aware service adaptation based on a service oriented definition of context.

***Keywords*—**Pervasive computing system, context, context-awareness, service, context modeling, ontology, adaptation, machine learning.

I. Introduction

IN a PCS a set of smart devices communicate and collaborate together in order to provide adapted services to the user and applications, and help the former in his everyday life tasks by making the use, management and manipulation of such devices more easier. In order to provide a good adapted services, devices in a PCS must be context-aware (sensitive to the global context) which requires a good understanding and use of context. From this appears the importance of the concept of context and its description (modeling or representation) in the development of a pervasive computing application. The first task of designing a pervasive computing application consists of understanding context, establishing its components and modeling it in a precise and concise manner. This will enhance a better usability, more flexible share between devices and ease the adaptation task.

Moeiz Miraoui is with the LATIS Laboratory, Université du Québec, École de technologie supérieure 1100, rue Notre-Dame Ouest, Montréal, Québec H3C 1K3 Canada. moeiz.miraoui.1@ens.etsmtl.ca

Chakib Tadj is with the LATIS Laboratory, Université du Québec, École de technologie supérieure 1100, rue Notre-Dame Ouest, Montréal, Québec H3C 1K3 Canada. ctadj@ele.etsmtl.ca

Chokri ben Amar is with the REGIM Laboratory, Université de Sfax, École Nationale d'Ingénieurs de Sfax, Route de Soukra, B.P. W, 3038 Sfax – Tunisie. Chokri.benamar@enis.rnu.tn

Many methods were proposed for context modeling having particularities related to the techniques used, and the most interesting ones are those based on ontology. But they still suffer from the problem of being specific and not generic which keep their use limited to some particular applications. The use of ontology to model contextual information has many advantages compared to other methods. It is considered as a promising tool for the future of context-aware systems in particular and PCS in general. In this paper, we propose a service ontology for context-aware devices that allows a good modeling of context in a PCS and which is enough generic to be used for several kind of devices and applications.

To adapt provided services, devices must be context-aware, which requires a good understanding and use of context. The adaptation task must be done accordingly to this context. Mainly there are three domains of adaptation: a) content adaptation, b) adaptation of behavior (services) and c) presentation (or interface) adaptation. We will restrict our adaptation approach to context-aware services adaptation. The need of services adaptation has been long recognized [1, 2, 3, 4, 5, 6, 7, 8, 9], and both manual and automatic approaches to service adaptation have been proposed. Generally the adaptation task consists of specifying for each service a set of rules for the possible contextual configurations and associates each rule to a kind of service. The developer has to predict before the operation of the system the set of rules relative to each service and associate it to each context configuration. The common problem of the proposed approach is how to determine and limit the set of possible context configurations which consists of specifying the set of context elements for each service and the set of meaningful context configurations. This permits to cover all possible running time contextual configurations and enhances the dynamic adaptation. Our aim is to propose an approach to context-aware services adaptation which takes into account these two aspects. This will be done by using a service oriented definition of context which is enough abstract and helps a lot to limit the set of contextual information and a learning machine approach to predict all possible and meaningful context configurations.

This paper is organized as follows, in section 2 we briefly introduce context and context-awareness in PCS. In section 3 we present methods used for context modeling. We focus in the ones based on ontology and show their weakness. In

section 4 we present our approach to model contextual information by using simple service ontology. In section 5 we detailed our approach for context-aware services adaptation. In section 6 we present an application scenario and a realization to make clear our approach of both modeling and service adaptation. The last section consists of a conclusion and future work.

II. Context and Context-Awareness

The context is some information needed to interpret something but this definition is very general and does not permit a good understanding and is useless for computing [10]. Many researchers have proposed several definitions of context. Some of them were based on enumerating contextual information (localization, nearby people, time, date, etc…) like those proposed in [11,12,13]. Others were based on providing more formal definitions in order to abstract the term like the one proposed by Dey [14]. Most of these definitions were specific to a particular domain such as human-computer interaction and localization systems. In our previous work [15] we have proposed a service oriented definition of context for a PCS. The definition was based on the concept of service because such one plays a crucial role in the operation of a PCS. The main objective of a PCS is to proactively provide adapted services according to the global context. Service adaptation can be done in two ways: a) automatic triggering of a service according to the context or b) changing the quality (form) of a provided service according to the context because one or more of contextual information has or have changed its or their value(s). They have defined context in a PCS environment as follows: "Any information that trigger a service or change the quality of a service if its value changes". This definition is enough abstract and limits the set of contextual information. Figure 1 shows the components of a PCS and the information that cause the adaptation which considered as part of the global context.

One important characteristic of a PCS environment is its high dynamic changes caused by the mobility. In order to better help the user in his tasks, devices must be a) more autonomic, demanding a minimum intervention from the user and b) context-aware to adapt provided services according to the global context. These capabilities are called context-awareness. In the same way and based on the concept of service they have introduced the following definition of context-awareness : "A system is said to be context-aware if it can change automatically the quality of its services or provide a service as a response to the change in the value of an information or set of information that characterize those services". Not like other definition of context-awareness [10,16,17,23] this definition explains awareness as a reaction of the system to modifications of information values in term of triggering a service or changing its form whatever the kind of application (more generic).

Context-awareness requires that contextual information be collected and presented to the adaptation application. Because of the diversity, the heterogeneity and the quality of these information, it is suggested to classify them in order to make the adaptation operation easier. Several categorizations were proposed, most of them made a classification in two categories like [14,10,18,19] others have made a classification to several categories like [11]. We have made [15] a classification in two categories that seems to us more expressive and helpful: a) trigger information whose change in value causes automatic release of a service and b) form changing information whose change in value causes the changing of a service's form. This categorization is simple because it has just two classes and complete because it covers all aspects of context.

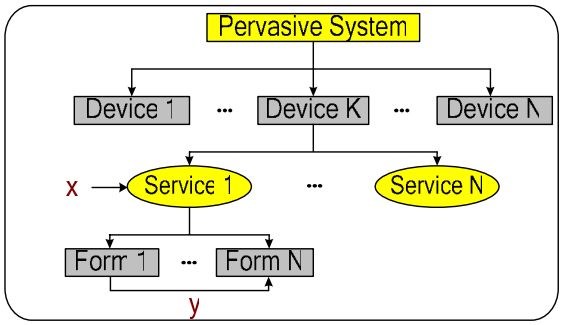

Fig. 1 Components of a pervasive system: The change in value of xi will trigger service1. The change in value of yj will change the form of service.

III. Context Modeling

Context modeling is a fundamental step for every adaptation application and the development of context-aware systems. It consists of the analyze and design of contextual information contained in the system in an abstract form on the level of data structure as well as the semantic level. Several approaches were proposed for the representation of context. A survey made by Strang et al. [21] contains an interesting comparative study of different modeling methods. They conclude that ontology makes the best description of context compared to the surveyed methods because it provides a good sharing of information with common semantics. A detailed discussion of these methods is out of the scope of this paper but the authors distinguished basically the following models for context representation: a) Key-value models, b) Mark-up scheme models, c) Graphical models, d) Object oriented models, e) Logic based models, and f) Ontology based models.

Problems: Key-value model is easy to manipulate but it is not convenient for complicated structure and does not permit a good reasoning on context. It is also specific to localization applications. Mark-up scheme models are most of the time specific to a particular field and are limited to some aspects of context (localization, environment...). Graphical models are simple but less formal than the other methods. Object oriented models require low level execution agreements between applications to ensure interoperability. They are not adapted for knowledge sharing in open and dynamic environments. Logic based models are often based on a centralized context management, a solution which is not adapted to the principle

of context distribution in a PCS. The relations between continuous data cannot be easily described. The ontology seems to be the most suitable tool for context modeling.

IV. ONTOLOGY-BASED MODELING

Ontology is a set of structured concepts that are organized in a graph where relations can be either semantic relations or composition and heritage relations. The main objective of ontology is to model the set of knowledge of a given domain. This means to choose a manner for describing domain information in a form comprehensible by computers. It provides a representative vocabulary for a given domain and permits a consistent interpretation of data. Ontology is a powerful tool for knowledge sharing. It permits the structuring of concepts and properties as well as classes and attributes did in object oriented model. For a specific domain there are many correct ontologies not just one; it depends on the understanding of such domain by the ontology designer and his point of view on the domain. OWL (Ontology Web Language) language is a recent recommendation of W3C (The World Wide Web Consortium) which permits to describe ontologies based on RDF (Resource Description Framework) scheme. The advantages of ontology are: a) enhance data sharing by eliminating sources of ambiguities (the same meaning for a given concept) and b) the possibility of making logic reasoning easily by using a related descriptive logic (deduce implicit facts of context).

A. Related work

Applications in a pervasive and ubiquitous environment require that ontology provides knowledge sharing, reasoning on context and interoperability in such environment. This was the aim of SOUPA (standard ontology for ubiquitous and pervasive applications) [22]. SOUPA is composed of two sets of ontologies : SOUPA core ontologies and SOUPA extension ontologies. Core ontologies try to define a generic vocabulary which is universal for different pervasive computing applications. Extension ontologies (extended from SOUPA core ontologies) define additional ontologies to support specific kind of applications and provide examples for extending future ontologies. Several terms used by SOUPA are mapped to stranger ontologies terms by using ontology mapping constructs OWL standard (owl: equivallentClass and owl: equivallentProperty). Ontologies referenced by SOUPA are (but not limited to) friend-of-friend (FOAF) ontology, spatial ontology (OpenCyc), COBRA-ONT, etc…

CONON [23] is a context ontology coded in OWL for modeling context in pervasive environment and support context reasoning. CONON provides an upper context ontology that sense general concepts of basic context and provides also the extension of add ontologies specific to a particular domain in a hierarchical manner. The authors have judged that the localization, user, activity and computer entities (devices) are the basic elements of context that compose the ontology (the upper one).

CoBrA (context broker agent) [24] is an architecture to support the development of context-aware systems in an intelligent space (smart house, smart car, etc…). In CoBrA where defined ontologies collections called COBRA-ONT for modeling contextual information in such intelligent spaces. COBRA-ONT are expressed with OWL which defines typical concepts associated with places, agents and events. Ontology plays a crucial role in CoBrA, it helps the context broker (server) to share contextual information with other agents and permit him to make context reasoning. The ontology is categorized in four classes: 1) physical place ontology, 2) agent (human or software) ontology, 3) agent localization agent ontology, 4) agent activity context ontology.

CoOL (context ontology language) [25] is a context modeling approach which uses ontology as formal foundation and the ASC (Aspect-Scale-Context) context model. CoOL context model permits context-awareness and context interoperability during services discovery in a distributed architecture. In CoOL contextual information that characterizes the content of other contextual information is called a meta-information or upper contextual information which expresses the quality of low contextual information.

SOCAM (Service-Oriented Context-Aware Middleware) [26] is an architecture for the building and rapid prototyping of context-aware services in intelligent environment. The context ontologies are divided into upper ontology which captures general context knowledge about the physical world and composed of: computational entity, location, user and activity. The domain-specific ontologies which are a collection of low-level ontologies that defines the details of general concepts and their properties in each sub-domain. We have presented only the methods which seem to us the most interesting ones but many other approaches of context modeling based on ontology were proposed.

Problems: most of proposed ontologies do not offer a complete description of contextual information. In order to propose more extensible and reusable ontologies, they suggest the use of two categories of ontologies. One composed of basic ontology concepts and the other is adapted according to the application domain. However, the basic ontology concepts must be used even when they are useless in the application. The core ontology (basic) differs from one method to another and its basic classes depend on the definition of context adopted by authors. The extension ontology (domain-specific) needs an extra effort from the developer to make the adaptation to a specific application. The proposed models fails in presenting a generic ontology for context which cover all aspect of context and may contain useless ontologies when used in a particular application (example: location ontology is useless in a context-aware system operating in local area), this will limit their usability and extensibility. We think that this is a common problem in all ontologies, however for context modeling the problem of genericity is strongly related to the definition of context: “more generic definition implies more generic ontology”.

Our aim is to propose a service ontology that covers all aspects of context in a PCS (enough generic) and easy to use what ever the application without an extra effort for adaptation.

B. Service ontology

In order to build a PCS, devices must be context-aware, this permit devices to provide adapted services for both user and

applications according to the global context. Context model should provide a common understanding of contextual information between devices in order to enable a good context sharing between them in a pervasive computing environment. A generic ontology must not be based on predefined aspects of context to enhance its usability and extensibility because these aspects vary from one application to another depending on the domain. Context ontology should be able to capture only the necessary contextual information needed to service adaptation and not all information which can contain useless contextual information. Every device in a PCS provides some services and the main goal is to adapt these services according to the global context, either by triggering them automatically depending on the current context or to change the form of the service knowing that every service has some forms (for example the service of indication of incoming calls for a cellular phone can be provided through different forms: ring tones, vibrator, silent, ring tones with vibrator, etc.). We have proposed [15] a service oriented definition of context (see section 2) which seems to us enough generic and limit the set of contextual information to perform the adaptation task. Based on this definition and the concept of service which plays a crucial role in a PCS, we will propose service ontology for modeling contextual information. The context ontology should contain only the basic elements which are common to any PCS. Our design approach is based on the following general idea of a PCS: a device provides some services in several forms. It contains some sensors attached to it which will gather with others sensors of the PCS a set of contextual information. These contextual information will either trigger a service or change the form of the service according to the current form. This leads us to design ontology composed of five classes: a) Device, b) Sensor, c) Service, d) Form and e) Context.

Device

The basic element of the PCS, it provides services to the user and may contain some sensors to collect contextual information. Each device has its own characteristics and capabilities depending on its usage (examples: desktop computer (CPU frequency, memory, hard disk capacity…), cellular phone (battery level, graphical resolution, size, memory …)). Devices in a PCS should be able to communicate with each other in order to exchange contextual information to make the necessary service adaptation.

Sensor

Can be attached (integrated) to a device or not. It is an important source of contextual information since it permits to gather many kinds of contextual information (light, temperature, location, time …). A device may contain many types of sensors.

Service

It is provided by a device to the user or applications. Each service has some functionality and can be provided in several forms. A service has a set of triggering information which permits to start it when such information takes some specific values. A service has also some characteristics like duration, input, output …etc.

Form

It is the manner of providing a service or the quality of a service. Each form of a specific service has its own characteristics (media used, modality used …). For every form of a service there is a set of changing information which permits to change from one form of a service to another when such information takes some specific values.

Context

A set of information gathered by sensors of the PCS that specifies the current situation which regroup information and events happened in the PCS. There are two categories of contextual information: a) service triggering information which the changes of their values trigger a service and b) form changing information which the change of their values changes the form of a service depending on the current form. Figure 2 shows the classes of our ontology and relations between them.

Depending on the device used, we can enumerate the set of services that can be provided by this device as well as the forms of each service. In the same manner we can enumerate for each service the set of information that triggers it if their values change and for each form of that service the set of information that permits to change it from one form to another if their values changes. Such information will make part of the context. Only this kind of information will be contained in the class context.

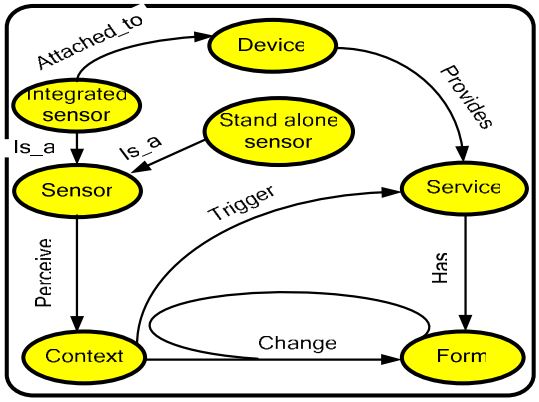

Fig. 2 Classes of the service ontology

Relations between these classes will exist whatever the type of device. This permit to affirm that this ontology is enough generic and there is no need to define basic and extension or domain-specific context concepts because this varies depending on the application. But it is very easy to determine and limit the set of contextual information by using our

previous definition of context [15].

To make clear the usability and the employability of our ontology, we will present further in this paper an application scenario.

One major advantage of ontology-based context modeling resides in the possibility to carry out logical reasoning. It permits to deduce relevant new contextual information that can not be explicitly provided by sensors of the PCS. Ontology reasoning permit to do three main tasks: a) check context inconsistency and conflicts caused by imperfect sensing of contextual information, b) check whether concepts are non contradictory and c) find subsumption relationships between classes and instances (for instance a user located in his bathroom means his located at home too, because his bathroom is a part of his home). There are many automated tools for ontology reasoning, in our work we have used the pellet [27, 28] ontology reasoner to check the consistency of our ontology and infer new implicit context information.

V. CONTEXT-AWARE SERVICE ADAPTATION

A PCS is composed of a set of devices communicating together in order to provide proactively adapted services to user and applications. Each device of the PCS provides several services and each one has several forms over which it can be provided. A service or a form of service is provided when a context configuration occurs. The context-aware adaptation consists of mapping a context configuration to a service or a form of a service. For simplification raisons we will restrict our approach to a single device with several services and several forms for each service, the same approach can be applied for all devices of the PCS. The approach consists of the following steps.

A. Determine Context Components

For each device of the PCS and using the previous service oriented definition of context [15], specify for each service the set of information which if their values change trigger the service. In the same manner, specify the set of information which if their values change, the form of service change too. These two kinds of information will compose the global context.

B. Determine Context Configurations

For each context element, specify the set of possible values (for instance light level (high, low), localization (at home, at university, outside university …)). Depending on the number of context elements (suppose equal to N), we will get N vectors of different sizes. The following task consists of enumerating all possible configurations of context. This process will provide vectors of size N containing all possible values of context elements. This task can be done automatically by using a simple algorithm which generates all possible configurations from context element vectors. Figure 3 illustrate elements of context with different values and possible context configurations.

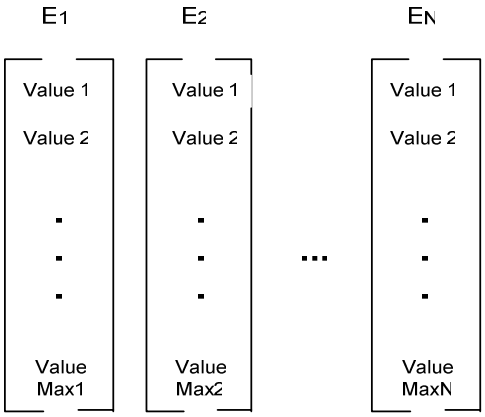

Fig. 3 Context elements with different values.

The total number of possible context configurations for context elements of figure 3 is given by this formula:

$$Max = \prod_{i=1...N} Max\ i$$

CC1 (Value 1 (E1), Value 1 (E2) … Value 1 (EN))

…

CCMax (Value i (E1), Value j (E2) … Value k (EN))

E: context element
CC: context configuration
Max: total number of possible context configurations

Each context configuration can be interpreted as a conjunction of different values of context elements.

The set of vectors of context configurations may include some meaningless context configurations. For instance suppose we have a context configuration composed of three elements: network connection (on/off), localization (place1, place2, place3, place4) and connection speed (high, low). The context configuration may contain the following configurations:

CC1 (on, place1, high)
CC2 (on, place2, low)
CC3 (off, place3, high)
CC4 (off, place4, low)

CC3 and CC4 are nonsense because if the network connection is off we can not speak about connection speed. We need some procedures to eliminate such nonsense context configuration.

C. Simplification

A context configuration is a conjunction of context elements values: Value i (E1) and Value j (E2) and … and Value k (EN). To eliminate one context configuration we need some contradictory rules composed of at least two

contradictory values of two context elements. The procedure will take these rules one by one and then browse all context configurations and eliminate each one that contains the elements of the contradictory rule. For instance in the above example of a context composed of network connection, localization and connection speed, we can choose as contradictory rules:

(Network connection, localization, connection speed)

1- (off, ? , High)
2- (off, ?, low)

? Means whatever the value

So context configurations CC3 and CC4 will be eliminated from the set of possible context configurations. This step contains two tasks, the first one must be done by the developer to determine the set of contradictory rules and the second one can be performed automatically using a simple search algorithm.

D. Learning and Mapping

This step consists of mapping each context configuration to the appropriate service form i.e. depending on the values of context elements, the service will be provided in a specific form. This task will be done for each service and for each device of the PCS.

For each service of a PCS's device, construct the set of context configurations that permit to change the form of the service when that context configuration occurs. This set of context configurations will serve as a training set to a supervised learning algorithm. The trained model will then predict for each context configuration (section 5.3) the form of service over which it will be provided.

Because of the limited resources of devices in a PCS (battery charge, processing, memory …), the training process will be performed off line and the result of training will be saved in the device's memory before its operation in the PCS.

VI. Application Scenario and Realization

In this scenario, the device is a cellular phone and we will show the steps to follow in order to model the context for a particular service as well as the service adaptation. The same steps can be followed for all other services.

In the general case the modeling process starts (step 1) by specifying for each device of the PCS the set of services that can provide. In our scenario, we will take as example of service provided by a cellular phone, the indication of incoming calls (modeling of contextual information for one device and a particular service).

The second step of the modeling process consists of specifying for each service the set of forms in which the service can be provided. In our case and in the default state, the cellular phone indicates incoming calls by using ring tones (default form) but for one or another reason it can indicates incoming calls in other forms as follows:

- Vibrator: when the user (we assume that he is a student) is at the university and either in the library (to avoid bothering other students and respect directives) or at the classroom (the hours of study are sensed from the study calendar of the student which we assume it is saved in the cellular phone).
- Ring tone with vibrator: when the user is in a noisy place, this form will be used to attract user's attention.
- Silent: when the user is sleeping (we assume that the user usually sleeps from 00 AM to 08 AM and his environment is dark with low noise).

In a PCS, devices have limited resources in particular battery charge. For this reason, if this one is low, the service (call indication) will be provided in the silent form whatever the convenient form. This is done to avoid emptying the battery and keep the cellular phone in operation as long as possible. Figure 4 shows the transition diagram of forms for call indications.

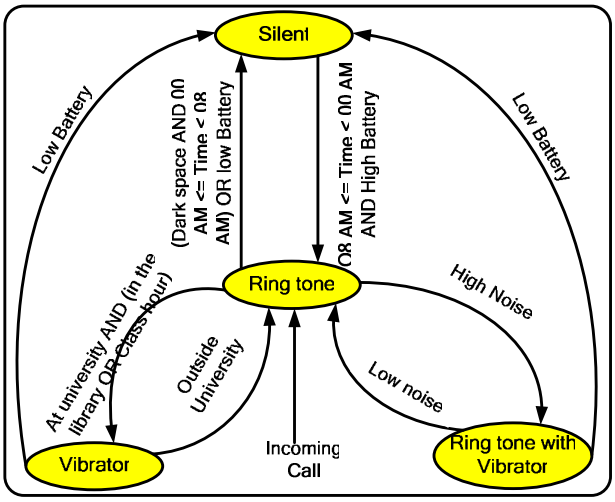

Fig. 4 Transition diagram of the application scenario

The third step of the modeling process consists of specifying for each service the set of information which their change of values will trigger the service and the set of information which their change of values will change the form of a service. This information will compose the global context of this scenario. It is clear that our method helps to limit contextual information to only those related to the application and does not contain useless information. We have done this explicitly at the time of defining the forms of the service (indication of incoming calls) in the second step. Table 1 shows the result of the three steps.

The last step consists representing the ontology graph of the

TABLE 1 RESULT OF STEP 1, 2 AND 3 OF THE MODELING PROCESS (CONTEXTUAL INFORMATION IN GRAY)

Device	Service	Triggering information	Form changing information	Form
Cellular phone	Incoming calls indication	Incoming call	Charge level = Low	Silent
			Current form = ring tone	
			Current form = vibrator	
			Current form = ring tone with vibrator	
			Location = university	Vibrat.
			Location = library	
			Current form = ring tone	
			Noise level = High	Ring tone with vibrat.
			Location = outside university	
			Current form = ring tone	
			Light level = Low	Silent
			Time = 00 AM to 08 AM	
			Location = home	
			Current form = ring tone	
			Location = university	Vibrat.
			Time = class hours	
			Current form = ring tone	
			Charge level = High	Ring tone
			Time = 08 AM to 00 Am	
			Current form = silent	
			Location = outside university	Ring tone
			Current form = vibrator	
			Noise level = Low	ring tone
			Current form = ring tone with vibrator	

service indication of incoming calls according to the service ontology presented in section 4.2 (figure 2) and then translate this graph in an OWL file coded in XML after checking its consistency.

It i s clear that our method helps to limit contextual information to only those related to the application and does not contain useless information and permit a good knowledge sharing between devices in a PCS. Figure 5 shows the ontology components of the application scenario.

We have used PROTÈGÈ 3.4 beta to implement the ontology of this scenario and in the end we have obtained an OWL file coded in XML which presents the strong characteristic of portability and enhance the knowledge sharing of contextual information between devices that compose a PCS.

The process of adaptation consists in the first step of collecting the set of information that change the forms of the service (incoming calls indication) if their values change as follow:

- Localization,
- Time,
- Noise level,
- Light level and
- Battery charge level

This information will compose the global context of this scenario.

The second step consists of specifying for each context element the set of its possible values as follows:

- *Localization (Classroom, library, home, outside university)*
- *Time (Sleeping hours, class hours, free hours)*
- *Noise level (High noise, low noise)*
- *Light level (High light, low light)*
- *Battery charge level (High Charge, low Charge)*

Of course some values of these context elements need preliminary interpretation like the time value sleeping hours which will be deduced when the hour is between 00AM and 08AM or noise, light and battery charge level if they are under a threshold they will be assumed to be low and high otherwise.

The third step consists of generating all possible context configurations based on different values of context elements. In our case there are ninety six (96) possible context configurations each of them is composed of five elements.

The next step consists of specifying the contradictory rules to eliminate meaningless context configurations. In our scenario we can put the following rules:

► At sleeping hours the localization can not be the classroom or the library, so we must eliminate all context configurations containing both of these couples (Classroom, sleeping hours) and (library, sleeping hours).

► At free hours the localization can not be the classroom, so we must eliminate all context configurations containing couple (Classroom, free hours)

After applying the simplification procedure there will remain seventy two (72) possible context configurations.

The next step consists of determining the set of context configurations that permit to change the form of the service when that context configuration occurs which will serve as a training set to a supervised learning algorithm. In our scenario this set is composed of the following context configurations:

► *?,?,?,?,LowCharge >> Silent*
► *Outside,?,HighNoise,?,HighCharge >> RingtoneWithVibrator*
► *ClassRoom,?,?,?,HighCharge >>Vibrator*
► *LearningRoom,?,?,?,HighCharge >> Vibrator*
► *Home,SleepingHousr,LowNoise,LowLight,? >>Silent*
► *Outside,?,LowNoise,?,HighCharge >>Ringtone*

? Means whatever the value

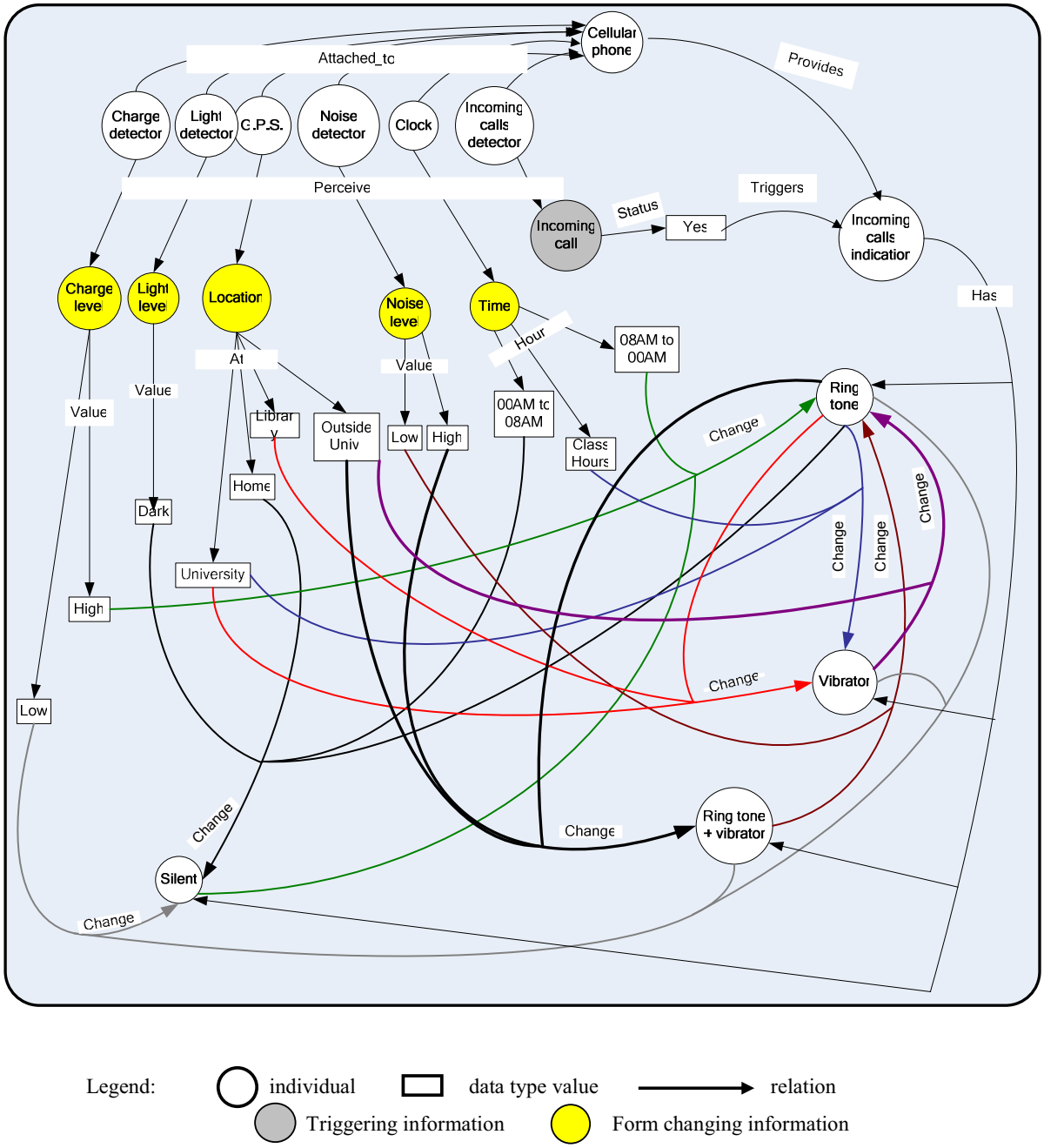

Fig. 5 Ontology components of the application scenario

The last step consists of using the trained model to predict the forms of all possible context configurations after the step of simplification. For the implementation of the application scenario, we have used the Java 1.5 platform and one of the most famous and interesting learning algorithm: the multilayer perceptions neural network with the following architecture and characteristics as shown in table 2.

TABLE II ARCHITECTURE AND CHARACTERISTICS OF THE USED NEURAL NETWORK

Legend	Characteristics
Input	Five neurons, one for each context element {localization, time, noise level, light level, battery charge level}
Output	Four neurons, one for each service form (class) {silent, ring tone, vibrator, ring tone with vibrator}
Hidden layers	3 hidden layers
Activation function	Sigmoid function
Learning algorithm	Back propagation supervised training
Learning rate	0,7
Momentum	0,3
Training set	10
Test and validation set	72
Training time	1000
Time to build model	0,61 seconds
Mean absolute error	0,0129

After running our program, we have made a manual verification of the predicted outputs and we have remarked that among the seventy two possible configurations there was only one error in the predicted form of service (class). The following context configuration (home, sleeping hours, low noise, low light, high charge) outputs ring tones as form of service instead of silent (the right form). This low rate of error permits to affirm that our approach gives good results for context-aware service adaptation.

VII. CONCLUSION AND FUTURE WORK

The main goal of pervasive computing system is to provide proactively adapted services to user and applications. The adaptation task must be context-aware which needs a good understanding and more precisely a good mechanism for context modeling that permit a consistent description and enhances context sharing between devices. Ontology-based modeling seems the most powerful tool for such task. But most of the proposed approaches fail to present a generic ontology of context for context-aware devices. Several approaches were proposed for dynamic services adaptation most of them use the same strategy which consists of specifying for each service a set of rules and associate them to each possible context. The inconvenient of such approaches is their weak specification of context elements which has a great influence on the adaptation task. In this paper we have proposed a generic ontology for context modeling based on the concept of service which plays an important part in the operation of a PCS. Our ontology is enough generic to be used for context modeling of any device to make it context-aware. We have proposed also an approach for context-aware services adaptation based on the same definition of context to better specify context elements and a learning mechanism to better mapping each context configuration to the appropriate service or form of service. Our future work consists of using fuzzy values for context elements in order to be more accurate in specifying context configurations.

REFERENCES

[1] T. Ledoux "Etat de l'art sur l'adaptabilité", Project RNTL ARCAD D.1.1 Ecole de Mines de Nantes, 4, rue Alfred Kastler, 44307 Nantes Cedex. 2001
[2] M. Aksit.and Z. Choukair "Dynamic, Adaptive and Reconfigurable Systems Overview and Prospective Vision", Proceedings of the ICDCSW'03, Providence, Rhode Island, USA, May 19-22, pp. 84-92. 2003
[3] J. Keeney and Cahill "Chisel: A Policy-Driven, Context-Aware, Dynamic Adaptation Framework", IEEE transaction of the 4th International Workshop on Policies for Distributed Systems and Networks, June 04 - 06, Lake Como, Italy, pp.3-13 2003
[4] M.T. Segara and F. André "A Framework for Dynamic Adaptation in Wireless Environments", IRISA Research Insitute, Proceedings of the Technology of Object Oriented Languages and systems (TOOLS 33), June 05 - 08, St. Malo, France. 2000
[5] D. Narayanan, J. Flinn and M. Satyanarayanan "Using history to improve mobile application adaptation", In Proceedings of the 3rd IEEE Workshop on Mobile Computing Systems and Applications, Monterey, California, Dec. 2000
[6] C. Efstratiou, K. Cheverst., N. Davies and A. Friday "An Architecture for the Effective Support of Adaptive Context- Aware Applications", Proceedings of the. 2nd Int. Conf. in Mobile Data Management (MDM'01), Hong Kong, pp. 15-26, January. 2001
[7] M. Fayad and P. Marshall Cline "Aspects of software adaptability", Commun. ACM 39 no. 10, pp. 58–59. 1996
[8] G. South., A.P. Lenaghan and R.R. Malyan "Using reflection for service adaptation in mobile clients (t4)", Tech. report, Kingston University-UK. 2000
[9] Yarvis M., Peter L. Reiher and Gerald J. Popek, (1999) "Conductor: A framework for distributed adaptation", Workshop on Hot Topics in Operating Systems, pp. 44–49.
[10] G. Chen and D. Kotz "A survey of context-aware mobile computing research", Department of computer science, Darmouth college, Technical report. 2000
[11] S. Schilit and M. Theimer "Disseminating Active Map Information to Mobile Hosts", IEEE Nenvork, 8(5):22-32. 1994
[12] P.J. Brown, J.D. Bovey and X. Chen "Context-aware Applications: From the Aboratory to the Marketplace" IEEE Personal Communications, 4(5):58–64. 1997
[13] N. Ryan., J. Pascoe and D. Morse "Enhanced Reality Fieldwork:the Context-Aware Archeological Assistant" Computer Applications in Archeology. 1997

[14] A.K. Dey "Understanding and Using Context" Journal of Personal and ubiquitous computing, Vol. 5, pp. 4-7. 2001
[15] M. M. Miraoui, C. Tadj, A service Oriented Definition of Context for Pervasive Computing, in Proceedings of the 16th International Conference on Computing, IEEE computer society press (to appear), Mexico city, Mexico, Nov. 2007.2007.
[16] B. Schilit, N. Adams and N. Want "Context-aware Computing Applications", Proceedings of the IEEE Workshop on Mobile Computing System and Application, pages 85-90, December. 1994
[17] J. Pascoe "Adding Generic Contextual Capabilities to Wearable Computers", 2nd International Symposium on Wearable Computers, pp. 92-99. 1998
[18] D. Petrelli , E. Not, C. Strapparava, O. Stock and M. Zancanaro "Modeling Context is Like Taking Pictures", CHI2000 Workshop.
[19] Gwizdka J. (2000) "What's in the Context?" In proceedings of workshop on context-awareness (CHI'2000). 2000
[20] M.A. Razzaque, S. Dobson and P. Nixon P "Categorization and Modeling of Quality in Context Information", In proceedings of the IJCAI 2005 Workshop on AI and Automatic communications. 2005
[21] T. Strang and C. Linnhoff-Popien "A Context Modeling survey", In the first International Workshop on Advanced context modeling, Reasoning and management, UbiComp 2004.
[22] H. Chen, F. Perich, T.W. Finin and A. Joshi "Soupa : Standard Ontology for Ubiquitous and Pervasive Applications", In MobiQuitous, pages 258-267. IEEE Computer Society. 2004
[23] X.H. Wang, D.Q. Zhang, T. Gu and H.K. Pung H. K. "Ontology Based Context Modeling and Reasoning Using owl", In PERCOMW '04 : Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, page 18, Washington, DC, USA, IEEE Computer Society. 2004
[24] H. Chen, T. Finin and A. Joshi ":An Ontology for Context-aware Pervasive Computing Environments", Knowledge Engineering Review, vol. 18, pp. 197-207, November. Springer Verlag. 2003
[25] D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers and K. De Bosschere "Towards an Extensible Context Ontology for ambient intelligence", EUSAI, volume 3295 of LNCS, pp. 148-159. Springer. 2004
[26] T. Gu, X. H. Wang, H.K. Pung and D.Q. Zhang "An Ontology-based Context Model in Intelligent Environments", In Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, San Diego, California, January 2004
[27] http://pellet.owldl.com/ (accessed 30 August 2008).
[28] B. Parsia. And E. Sirin "Pellet: An OWL DL Reasoner". Poster, In Third International Semantic Web Conference (ISWC2004), Hiroshima, Japan, November 2004

Moeiz Miraoui received a BS degree in Computer Science from the National School for computer studies University of Tunisia in 1999. He obtained the Master degree in computer science from the National Engineering School of Sfax (Tunisia) in 2003. He is currently a PhD student in the Laboratory of Information Processing and Signals at Ecole de Technologie Supérieure (ETS) University of Quebec at Montreal (Canada). He is a member of ACM, IEEE, and IEEE Computer Society. His research interests include pervasive computing, context-aware systems, ontology and semantic web.

LATIS Laboratory, Université du Québec, École de technologie supérieure 1100, rue Notre-Dame Ouest, Montréal, Québec H3C 1K3 Cana*da.* moeiz.miraoui.1@ens.etsmtl.ca

Chakib Tadj is a professor at ETS, University of Quebec at Montreal (Canada). He received his Ph.D. degree from ENST Paris in 1995. He is a member of Laboratory of Information Processing and Signals at ETS. His main research interests are ubiquitous computing, multimodal systems automatic recognition of speech and voice mark.

LATIS Laboratory, Université du Québec, École de technologie supérieure 1100, rue Notre-Dame Ouest, Montréal, Québec H3C 1K3 Canada. ***ctadj@ele.etsmtl.ca***

Chokri Ben Amar was born in Sfax, Tunisia, in 1964. He graduated in Electrical Engineering in 1989, obtained the Master degree, Ph.D. degree and then the HDR degree in Electrical Engineering in 1990, 1994 and 2006, respectively. He is now an Associate Professor in Electrical and Computer Engineering at the University of Sfax. His research interest includes Computer Integrated Manufacturing, Multi-Resolution Image and Video Analysis, Wavelets, Wavelet and Neural networks, with applications to data Classification and Pattern Recognition. He focuses his research on Vision-Robotic, Manufacturing, Image and video coding and Indexing. He was the chairman of the organizing committees of the International Conference on Machine Intelligence ACIDCA-ICMI'2005 and the International Conference on Signals, Circuits and Systems SCS'2004. He is an IEEE senior member.

REGIM Laboratory, Université de Sfax, École Nationale d'Ingénieurs de Sfax, Route de Soukra, B.P. W, 3038 Sfax – Tunisia. Chokri.benamar@enis.rnu.tn