A Service Oriented Definition of Context for Pervasive Computing

Moeiz Miraoui, Chakib Tadj

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

Abstract

In pervasive computing, 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. To adapt provided services, devices must be context-aware, which requires a good understanding and use of context. This term is still general, vague and doesn't help developer of context-aware applications in spite of the large number of proposed definitions. In this paper, we provide a service based definition of context that is abstract enough to be used in pervasive computing and limit the set of information required to make service adaptation. We use some scenarios to make clear our definition. We present also a new definition of context-awareness and a categorization of context based on the concept of service, which seems, to us, a fruitful approach in pervasive computing. Finally, we discuss our approach and contribution.

1. Introduction

Pervasive computing is a new field of computing resulting from the technological evolution in both distributed computing and mobile computing. Nowadays the user is surrounded by many smart devices (of small sizes and low costs) ranging from laptop, cellular phone, and GPS to objects of everyday life. The task of using and managing these devices becomes so difficult to the user and takes an important time from him. In order to help the user in his everyday life, these devices must communicate and collaborate together and provide adapted services to the user without his explicit intervention.

An important characteristic of devices in a pervasive system is their awareness to the global context, because they have to provide adapted services to both the user and applications according to this global context. From this appears the importance of the concept of context in developing a pervasive computing application. In their interactions, humans use and take into account context in an implicit manner. For example two people discussing in a noisy café have to raise their voice to hear each other and have to reduce their voice in a collective studying room in order not to bother other people. This shows that humans react according to their context. Computers and smart devices don't have this ability of changing their behavior (provided services) according to their global context (including user's and task's context), which limits the bandwidth of communication between users and devices on one hand, and between devices on the other hand, and limits also the usability of them. Making these devices aware to context will provide more adequate services to the user and make these devices more useful.

The first task of designing pervasive computing applications will consist of understanding context and establish its components. This will make the development phase easier and the application architecture more clear. Many attempts were made in order to give a concise definition of context. However, the term is still a general and vague word which does not help designers and developers of pervasive computing applications. In this paper we will propose a new definition of both context and context-awareness and make a categorization of contextual information based on the concept of service. Service is a key word in pervasive systems, because the latter provide adapted services to the user according to the context, so the approach based on services seems to us a fruitful (suitable) one.

In this paper we will review previous definitions of context, discuss them and give our definition of context in pervasive computing. We will do the same task for both context awareness and context categorization. Before concluding this paper and presenting our further work, we will do (present) a discussion to show the originality of our approach and to show our contribution.

2. Context

In general, the word context refers to the set of information surrounding an activity and some supplementary information about the environment of it. In computing domain, context is an old word used in particular in operating systems theory to characterize the minimum set of information required about a task in execution (process). This set allows the operating system to return to the execution of the process after the occurrence of an interruption and the running of the processing program of the interruption. In a survey of context-aware systems, Chen and Kotz [1] concluded that existent definitions of context are very general, vague and not clear enough to help its understanding in computing systems. In the following we will present, in chronological order of appearance, a non exhaustive list of definitions of context in computing.

2.1. Previous definitions

In 1994, Schilit and al. [2] considered that context has three important aspects consisting of responses to the questions: where are you? Who are with you? And what resources are nearby you? They extended their definition [3] and defined context as the location, identification of nearby people and objects as well as changes to those objects over time. Brown [4] defined context as the elements of the user's environment that the computer knows about. Brown and al. [5] proposed a set of extensible elements for context based on: location set of objects needed by the user, time and direction (spatial orientation). Ryan and al. [6] considered context as user's location, environment, identity and time. Word and al. [7] defined context as the set of possible environmental states of the application. Pascoe [8] defined context as a subset of physical and conceptual states having an interest to a particular entity. Brézillon and Pomerol [10] defined context as all the knowledge that constrains a problem solving at a given step without intervening in it explicitly. Chen and Kotz [1] considered context as the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user. Dey [11] proposed the following definition: context is any information that can be used to characterize the situation of an entity. An entity is a person, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves. Salber [12] described context as any information required by the interaction between the user and the application which can be sensed by the application. Henrickson and al. [13] defined context as the circumstance or the situation in which a computing task occurs.

In spite of the great (large) number of proposed definitions for context and the similarities between most of them (several definitions refer to the user's location and the environment), the word context is still general, vague and needs more clarity. Two techniques are used in most definitions, the first one consists of enumerating examples of context like location, identity, etc. and the second one attempts to formalize the word and give it a high level of abstraction. The first one is not practical in all situations because it is so difficult to enumerate all aspects of context, and it depends on the situation. Also, some aspects are important in an application but not in others. The second one gives a more abstract definition of context. However, it doesn't help developers limit the set of elements that constitute the context, and the context will include several useless information. This will lead to a great set of information that requires a complementary (an additional) effort of storage and management. Most of these definitions were given for either mobile computing (including localization applications) or human-computer interaction applications, which make them less generic, because pervasive computing include other kind of applications.

A survey conducted by Brézillon and al. [14] concluded that most of proposed definitions are responses to the following questions: why? What? Where? When? and how? That will generate an enormous quantity of useful and useless information.

2.2. Service based approach

The principal objective of a pervasive computing system is to provide proactively adapted services to the user and applications according to the global context (without an explicit intervention of the user). This service adaptation can be realized in two ways: automatic triggering of a service according to the context or changing the quality of a provided service according to the context (the service will be provided in another format) because one or more information of context has or have changed its value or their values (figure 1). This will lead us to make an abstraction of the concept of context and to view it from a service point of view.

We define context in pervasive computing as follows:

"*Any information that trigger a service or change the quality of a service if its value changes*"

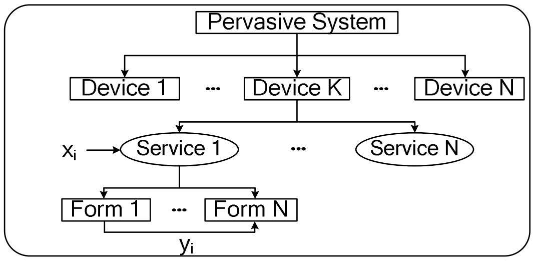

Figure 1. Components of a pervasive system: The change in value of x_i will trigger service1. The change in value of y_j will change the form of service.

This definition is a compromise between the abstraction of the term context and the limitation of the set of contextual information related to services provided (goal of pervasive computing). It does not enumerate examples of context which makes it more generic and independent of the application. It does not take into account information that might characterize context but do not play a relevant part in service adaptation. Instead of viewing context according to the user or to the system, we will view it according to the principal relation between them: a system provides services to the user. That means a vision oriented service. To make clear the consistency of our definition, we are going to give three scenarios and show how it is easy to characterize contextual information using the previous definition.

Scenario 1

A cellular phone screen must be adapted according to the context. In dark space it becomes brighter and in a bright space it becomes darker.

In this scenario:

- Service: display
- Qualities:{higher luminance (bright) and lower luminance (dark)}
- Contextual information: brightness of the cellular phone space (if the value of this information changes then the quality (form) of the provided service (display) will change too without user's intervention (fig. 2 and table 1)).

Scenario 2

A desktop computer is equipped with two modes of connection to the internet. Normally, to connect to the internet, the system uses a network cable, but if the speed of the connection is too low, the system changes automatically the mode of connection and uses the wireless one (Wi-Fi) with the hypothesis that this latter provides a better speed.

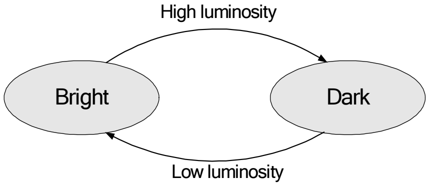

Figure 2. **Transition diagram of scenario 1.**

Table 1. Contextual information of scenario 1

Device	Service	Form	Contextual information	
			Trigger	Changing form
	Display	Dark	——	Luminosity of Surrounding environment
		Bright		

In this scenario:

- Service: internet connection
- Qualities: {cable connection, wireless connection}
- Contextual information: speed of internet connection (if the value of this information changes then the quality of provided service will change too (figure 3 and table 2)).

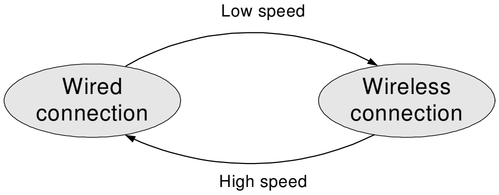

Figure 3. Transition diagram of scenario 2.

Table 2. Contextual information of scenario 2

Device	Service	Form	Contextual information	
			Trigger	Changing form
	Internet connection	Wired	——	Speed of wired connection
		Wireless		

Scenario 3

A user has written in the activities calendar of his cellular phone a task to do on the date X at the time Y. At these temporal coordinates, the cellular phone was set by the user to silent mode (ring tone volume = 0) because he is in a studying room with other people. The cellular phone communicates with the user's laptop (using Bluetooth or Wi-Fi mode) to know if the user is using it (his cellular phone). If it is the case, it sends a message to the laptop containing the task to do that appears in the screen of the laptop.

In this scenario:

- Service: electronic reminder
- Qualities: {reminder by a cellular ring tone, reminder by display on laptop screen}
- Contextual information: date, hour (if the date is X and the hour is Y then reminding service is triggered) and the volume of the cellular phone ring tone (if the volume = 0 then the service will be provided in other format (figure 4 and table 3)).

These scenarios show that using the approach based on service will characterize contextual information in an easier manner and limit contextual information to those required to services adaptation.

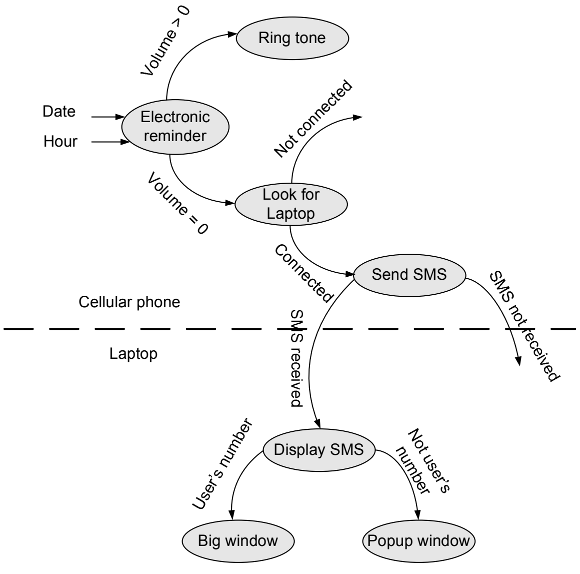

Figure 4. Transition diagram of scenario 3.

3. Context-awareness

The principal characteristic of a pervasive computing environment is the high dynamic changes. In order to better help the user in his everyday tasks, pervasive computing systems must be more autonomic, demanding a minimum user intervention, and must be context-aware, to adapt provided services according to the global context. The capability of adapting services according to the global context is called context-awareness.

The term context-awareness was introduced for the first time by Schilit and al. [2] in their work on a localization system. They defined context-awareness as the ability of an application to adapt to the context of its execution according to: location set of nearby people, machines, accessible devices, and also the modification of these objects over time.

Table 3. Contextual information of scenario 3

Device	Service	Form	Contextual information	
			Trigger	Changing form
📱	Electronic reminder	Ring tone	Date, hour	Ring tone volume
		others		
	Connect to laptop	Wireless	Ring tone volume	──
	Send SMS to laptop	Wireless	Laptop connected	──
💻	Display SMS	Big window	SMS received	Sender's number
		Popup window		

Brown and al. [5] defined context-awareness in their works on a touristy guide as any application that takes into account the user context. Schilit and al. [9] defined context-awareness as knowledge about the user's and IT devices state, including surroundings, situation, and, to a lesser extent, location. Pascoe and al. [8] defined it as the ability of any computing device to detect, interpret and respond to aspects of user's local environment and computing devices themselves. Dey [11] considered a system to be context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task. Chen and Kotz [1] gave two definitions of context-awareness: active context-awareness in which an application automatically adapts to discovered context, by changing the application's behavior and passive context-awareness in which an application presents the new or adapted context to an interested user or makes the context persistent for the user to retrieve later.

These definitions and others are indeed very similar and define context-awareness as the ability of a computer device to change or adapt its behavior according to the user's environments and application's context. They are not generic enough and depend on the application domain (human-computer interaction, mobile computing), and not abstract enough to help designers and developers of context-aware applications.

In the same way and based on the service approach, we give 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*"

This definition describes better a context-aware system because it explains awareness as a reaction of the system to modifications of information's values in term of changing the qualities of services or triggering a service whatever the kind of application (more generic).

4. Context categorization

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 do a classification or a categorization of them in order to make the adaptation operation easier.

Many researchers in this field proposed some categorizations using different approaches. Schilit and al. [3] and Dey [11] did a categorization in two classes, the primary context containing information about location, identity, time and the activity (status) and the secondary context can be deduced from the primary (for instance from the location of the user we can deduce the nearby people). Chen and Kotz [1] proposed two categories: the active context which influences the behavior of an application and the passive context which is necessary but not critical to the application. Peterelli and al. [15] considered two classes: the material context (location, device, existent platform) and the social context (social aspects like relations between people). Gwidska [16] proposed two categories: internal context containing user's state and external context containing environment's state. Hofer et al. [17] made a categorization in two classes: the physical context that can be measured by physical sensors and the logical context which contains information about the interaction (user's emotional state, his goals, etc.). Razzaque and al. [18] proposed a categorization in six classes: user's context, physical context, network context, activity context, material context and service context. This categorization has the advantage to cover the elements considered by the others categorizations, however if we refer to the six questions introduced by Brézillon [14] to list contextual information, the question why? is ignored in this categorization.

Another categorization is proposed by the same authors but it is based on values that can take contextual information: continuous context, enumerative context, state context and descriptive context.

Certainly there are other categorizations but none of them is exhaustive. Other categorizations may appear when other contextual information will be discovered. The main problem of the presented categorizations is related to context definition: not generic enough and no compromise between abstraction of the term and limitation of contextual information. Adopting the same approach (based on service), we will propose a categorization that seems more expressive to us and that helps developers and designers classify contextual information in two classes:

- Trigger information: information whose change in value causes automatic release of services provided by the pervasive system.
- Quality changing information: information whose change in value causes the change of service's format.

This categorization has two main advantages: it is simple because it has only two classes and complete because it covers all aspects of context, in particular the six questions of Brézillon [14].

5. Discussion

The use of context in a pervasive system to adapt provided services according to it requires that the word context has a consistent definition. Basically there are two methods used to define the term: the enumerative one which is not practical in all situations because some applications don't need some kind of information while others do, and the formalizing method that abstracts the word but it doesn't limit contextual information and it generates so much information that extra effort is required to store and manage them. Most of definitions were proposed by researchers of two domains: human-computer interaction and mobile computing (including localization systems), which makes these definitions related to these two domains. In other words, they are not generic enough to be used in pervasive computing that may include other kinds of applications.

In our approach, instead of defining context based on the user or the system, we defined it based on the relation between them: system provide services to the user. System and user change over time but the relation between them remains the same. The service is a key concept that can help to give a consistent definition of context. Our definition makes a compromise between the two main methods used to define context. It is abstract enough to enable the usability of context in computing and it limits the set of information required to service adaptation which leads to easier manipulation (management) of such information because there is no useless information. The proposed definition is based on measurable information (whose values change) which eases (simplifies) their implementation in a computing environment.

The categorization introduced in this paper caver all aspects of context and is composed of two classes (simple), which will give much help in designing the architecture of context-aware systems.

6. Conclusion and further work

Context plays a crucial role in pervasive computing systems. The inappropriate definition of the word will lead to a bad use of it, which will affect the application of service adaptation according to context.

This paper deals with definition of context in pervasive computing based on the concept of service. The context is viewed from the service point of view instead of user or system point of view, which makes the definition more generic and makes a compromise between abstracting the word and limiting the set of contextual information to ease their use. In such definition, contextual information are those with changing values, so these information are measurable, which eases their implementation. Most of the proposed methods are either too abstract or require gathering a lot of useless information, demanding extra effort for manipulation. The proposed definition has lead to a better definition of context-aware systems and a simple and complete categorization of context. Our further work will be the design of service based architecture for context-aware systems using the same approach.

References

[1] G. Chen, D. Koltz, “A survey of context-aware mobile computing research,” Department of Computer Science, Dartmouth College, Technical report 2000.

[2] 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.

[3] B. Schilit and M. Theimer. “Disseminating active map information to mobile hosts.” IEEE Nenvork, 8(5):22-32, 1994.

[4] P. J. Brown, “The stick-e document: A framework for creating context-aware applications.” In Proceedings of the Electronic Publishing, Palo Alto, pages 259– 272.

[5] P. J. Brown, J. D. Bovey, and X. Chen. “Context-aware applications: from the laboratory to the marketplace.” IEEE Personal Communications, 4(5):58–64, 1997.

[6] N. Ryan , J. Pascoe, D. Morse, “Enhanced Reality Fieldwork: the Context- Aware Archeological Assistant,” Computer Applications in Archeology, 1997

[7] Ward, A. Hopper, “A New Location Technique for the Active Office”, IEEE personnal Communications, 1997, p. 42-47.

[8] J. Pascoe, “Adding Generic Contextual Capabilities to Wearable Computers”, 2nd International Symposium on Wearable Computers, 1998, p. 92-99.

[9] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde. Advanced interaction in context. Proceedings of First biteniarional Symposium on Handheld and Ubiquitous Computing (HUC99), pages 89-101, 1999.

[10] P. Brézillon and J. C. Pomerol. Contextual knowledge sharing and cooperation in intelligent assistant systems. Le Travail Humain, 62(3):223-246, 1999.

[11] A. K. Dey, “Understanding and using context,” Personal and ubiquitous computing, Vol. 5, pp. 4-7, 2001.

[12] D. Salber, “Context-awarness and multimodality”, Colloque sur la multimodalité, Mai 2000, IMAG, Grenoble.

[13] K. Henricksen, J. Indulska, and A. Rakotonirainy. “Modeling context information in pervasive computing systems.” Pi-oceedings of Pervasive 2002, (Zurich, Switzerland): 167-180, 2002.

[14] Brézillon, P., Borges, M.R., Pino, J.A., Pomerol, J.Ch.: Context-Awareness in Group Work: Three Case Studies. Proc. of 2004 IFIP Int. Conf. on Decision Support Systems (DSS 2004), Prato, Italie, Juillet, 2004

[15] D. Petrelli, E. Not, C. Strapparava, O. Stock, M. Zancanaro. “Modeling Context is Like Taking Pictures.” CHI2000 Workshop, 2000.

[16] J. Gwizdka, “What’s in the context?, ” In proceedings of workshop on context-awareness (CHI’2000), 2000.

[17] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger & J. Altmann, ‘’ Context-awareness on mobile devices – the hydrogen approach. In Proceedings of the 36th Annual Hawaii International Conference on System Sciences, pages 292–302. 2002

[18] M. A. Razzaque, S. Dobson & P. Nixon, “Categorization and modeling of quality in context information, ” In proceedings of the IJCAI 2005 Workshop on AI and Automatic communications, 2005.