Context-Awareness and Uncertainty: Current Scenario and Challenges for the Future

Leandro O. Freitas, Pedro R. Henriques and Paulo Novais

ALGORITMI Centre, Department of Informatics, University of Minho
Braga, Portugal
leanfrts@gmail.com, {prh, pjon}@di.uminho.pt

Abstract. One of the main aspects of Ambient Intelligence (AmI) refers to its capacity of act autonomously in benefit of human beings. This implies in a hard challenge to overcome and of enormous responsibility. AmI is directly related to other fields of knowledge such as Smart environments, which aim to improve user experience through the development of context-aware applications. In this paper we present the current scenario of context-aware systems with some conceptual metrics to be followed. We highlight the problem of dealing with uncertain context information, e.g. incomplete, out-dated or nebulous data, seen as one of the main obstacle in this area.

Keywords: Intelligent Environments, Context-Awareness, Uncertainty.

1 Introduction

Intelligent environments should be sensible, allowing the system to identify the current state of entities and react under different situations, considering specially the user [5]. In such environments the user is the main focus for the development of services [2]. Considering this, Ambient Intelligence (AmI) is directly related to the People-Centric Computing paradigm, which aims to increase the participation of the user inside environments full of smart devices. According [6], this paradigm supports the idea of users acting not only as final client, but also as contributors. Their actions are monitored and used as source of information to improve the system functioning.

Intelligent environments should have the ability of learning and adapting not only from users' needs or requirements, but also, for the natural evolution of their preferences or desires. This kind of domain learns by observing the user and their apprenticeship allows the development of dynamic environments [9].

Context-aware systems use events to evidence changes in the environment. When the system identifies new states of entities, new events are created. Actions are defined as response to events and are used to characterize the behaviour of applications or services. To develop context-aware systems we have to adapt the concept of event, allowing it to be programmed or manipulated by applications [10].

Context-aware systems use information provided by different types of sensors. The role of these sensors is to monitor the environment and detect relevant data. Due to the dynamicity of smart environments, there are times where the data collected is not enough for the system to deal with it properly. In other words, the set of collected data does not reflect, precisely, what is happening in the environment. In cases like this, the system may build a computational model with incorrect services that instead of assist users, it might disturb them.

Considering this scenario, there are, in literature, several approaches trying to minimize the uncertainty of information. In this paper we discuss some of them in order to contribute with the consolidation of smart environments and Ambient Intelligence.

The paper is structured as follows Section 2 we discuss concepts of context-awareness pointing some metrics of design, detection and modelling information. In section 3 we highlight some issues about dealing with uncertain data of smart environments. At last in section 4 we present our final considerations and future directions of research.

2 Context awareness

Context awareness represents a connection between the environment and the computational structure. It refers to identification of current state of users and other entities and how they can influence the behaviour of applications. Such identification can be achieved through the use of different types of sensors like location monitoring, vital signs, level of stress or fatigue, among others.

According to [11], context-awareness has three main features that must be taken into account. The first one refers to the presentation of specific information to the user based on the context. The personal device of the user should be able to present information based on his location. For example, if he is at work, it could show his tasks for that day. Another critical feature of context-aware systems is the automatic execution of services. For this to happen, the system should allow a machine-to-machine communication and exchange of information. It is imperative that information used as basis to execute a service should be completely correct to avoid miss interpretations and execution of wrong services. It is known that a context-aware system should have different types of sensors monitoring and collecting data from the environment. However, if the system does not know what to do with the data, they are useless. Thus, the last main feature is the context needs, which are the goals of the services. The context needs are essential for the collection, fusion and analysis of right information.

The popularization of the use of context data in information systems brings challenges that must be overcome. There will be scenarios where the system will have to process much more data than used to do with only direct inputs. Besides that, smart environments should be full of sensors monitoring objects, users and other entities. Probably, more than one sensor would be capable of collect the same information. Considering that sensors have specific objectives, the system must be able to decide from which sensor it should use the data collect, according to the current context.

2.1 Detection of context information

According to [11], there are several techniques for the detection of relevant information of smart environments. The first one refers to the responsibility [12], where the context-aware system makes a request to the sensor. In other case the sensor sends information detected to the system every time it detects something. In both cases the periodicity of the requests or detection may vary between instant detection or periodically. The former characterizes events that occur only once and the acquisition of information must be at the time it occurs. The latter characterizes events that tend to repeat within a period of time. In order to acquire this information, the sensor needs to detect more than once and analyse it, in order to find a pattern.

Nowadays there are several types of sensors available that can be applied to smart environments to monitor and detect context information. According to [7], they can be categorized in physical, virtual and logical sensors. The first is capable of detecting data without any external intervention. There are plenty of devices equipped with this kind of sensor. Information generated is considered as low-level context, i.e., they are more trivial [11]. On the other hand, the virtual sensors do not detect context information without intervention. They search information from other sources, through web services, and process it as sensor data. Logical sensors use physical and virtual sensors to produce, through web services, richer context information.

2.2 Context modelling

Context-aware systems must use context models that reflect the knowledge of a smart environment. According to [15], these models can be classified as static, with limited and previously defined set of information, or dynamic, discovering and manipulating new information over time. The modelling process needs to consider aspects like heterogeneity of entities, mobility of users, relations among entities, periodic events, uncertainty, reasoning, usability and efficiency of the context [3][11].

The context modelling starts with the definition of entities with their characteristics (attributes, restrictions and relations). This should be performed using a previous known context as basis. It is important to highlight that one of the main issues on context modelling refers to the subjectivity. The same set of information detected in an environment may be valid in one situation and not in another. So, the more complete is the information about a context, the more accurate will be the model built based on that. After modelling a new context, it should be integrated with rest of context information in a repository.

The definition of services of context-aware system is directly related to requirements about the current state of entities. Although, in many times the signs collected by sensors do not provide enough data to fill such requirements. That happens because, under the computational perspective, the context information is naturally dynamic, uncertain or incomplete. The result is an inexact connection between the service and what is provided by the environment [14]. The description of the context and services requirements uses similarity comparison, which analyses the necessary and provided information individually, and also global comparison, which analyses the context as a whole [14].

Considering this, we can verify that one of the main challenges of modelling context is to handle with uncertainty. A context-aware system should be able of obtaining enough data to build complete contexts, i.e., to identify clearly all information present in the environment. This is crucial for a correct orchestration of events with context-aware services. If the sensors are not able to identify a situation or if the context models do not represent real scenario of the environment, the services that would be executed may not be the more appropriate. In the next section the problem of uncertainty will be discussed.

3 Handling uncertainty

Due to the dynamic nature of AmI, we need to find ways to correctly analyse and handle incomplete information, once it could result in the construction of uncertain contexts. Uncertainty, in smart environments, can be defined as any incomplete, contradictory, vague or outdated information [16].

Information collected could be partially correct or even absent, but still should be considered for reasoning. This way, Multiagent Systems provide a generic model with the necessary flexibility, with different levels of autonomy and dependency for all components of smart environments.

The dynamism of a domain centred in the user is a complex subject, under different perspectives. We should consider aspects related to their state of mind, including level of stress, fatigue and personal preferences, and also external factors like temperature and time of the day. Such factors can influence the user causing changes in their acting patterns within a short space of time. The result is that the knowledge base that is used by the computational system starts to work with a high level of uncertainty, without the assurance that the mapped context is correct and up-to-date.

The uncertainty originated in a smart environment could be result of variability phenomenon, i.e., due to changes in the user behaviour, making them act unexpectedly [8]. Another cause refers to the lack of complete information about specific situations, increasing the complexity of the system's learning process. Besides the subjectivity generated from these two sources, the quality of the computational service must remain the same. The uncertainty identified in dynamics environments should not be ignored. We should perform an analysis of user behaviour seeking alternatives to fill information gaps evidenced during the process of building contexts [13].

Context information is naturally incomplete and uncertain, and could be result of different causes, e.g., interpretation problems of the signs sent by sensors [1]. Different manufacturers may use specific concepts and identifiers to name the same entity or relation. Also according to [1], we still have challenges to overcome regarding the uncertainty in smart environments. We should guarantee a correct modelling of the available services with formal description of their goals and requirements for execution. Besides that, we should guarantee that the service invocation is capable of verifying the reliability of the output information. At last, we should extend the conventional composition models verifying levels of probability for each service related to the detected context. The correct definition of each of these steps contributes to a more appropriate service orchestration, even if working with uncertain context information.

According to [4], the information is partial when only some of the queries in the reasoning layer can be answered. However, to build a context model aiming to assist users, these answers should be as complete as possible. Besides that, the information could also be not as trustable as it should or generate a conflicting result. In this case we have different types of sensors capable of generate the same kind of information from an entity, but, to be used in different scenarios. The data collected from them could produce conflicted information to the context model, where the same data represents more than one kind of information, depending on the situation [4]. Uncertain information can be represented in numeric models through approximation values, within an established error margin [4]. Independent of the chosen methodology, usually, a relation between two instances has external aspects that must be taken into account, once they may be relevant for the right understanding of the situation.

A good alternative to face situations of incomplete set of data is to speculate information about the future. According to [17], Speculative Computation refers to the use of defaults values as input for processing and generation of an output with previsions of future situations. These default values can be known data from a set or, in case of this data is not available, it can be defined by the user, as being a valid value for that attribute. In case of context-aware systems, by using these default beliefs as input, speculative computation helps to improve the reasoning over a context model, reducing the time of decision making and execution of applications.

Information used in any context-aware system must be represented to be computed. According to [4], this representation must have characteristics of similarity with what it represents in the real world and also, should be as simple as possible, avoiding any ambiguous meaning. To make a decision based on a specific context model, the system must analyse all the possible scenarios that match with it, to decide and orchestrate the execution of services. To ensure that the services will be the most appropriate, the information brought to the system must be complete.

4 Identification of problems and approach

The main focus of the proposed research is to tackle the problem of uncertainty handling in context-aware computing. Thus, the following problems were identified:

- **Lack of accurate data**: due to different sources of problems, many times a context-aware system cannot build a computational model that represents the knowledge of a real-world domain. Thus, the probability of mistaken analysis tends to increase considerably and, consequently, the decision of what action to take will not be the most appropriated front that specific situation;

- **Misinterpretation of contexts**: the analysis of situations performed by software agents is based on identification of patterns of behaviour of users. However, unexpected behaviour can happen and even if the system has enough context data to build a model, it will not understand why the user and his/her surroundings is behaving like that. This happens because the system learned that, in that moment of time, the user used to do specific things, but his/her behaviour changed.

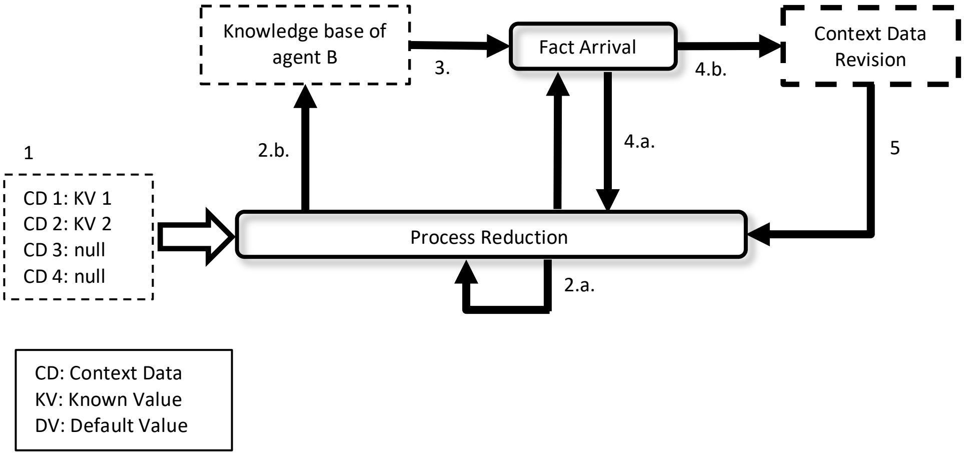

Fig 1. Speculative Computation

A good alternative to deal with these problems is to speculate information about the context. According to ~\cite{Tiago}, Speculative Computation refers to the use of defaults values as input for reasoning filling gaps of data. These default values refer to valid data attributes in previous similar situations. Thus, if what happened in the past is true, the probability of the same situation to be true in the future tend to be high. These default values help software agents to speed the reasoning over a context once they do not need to wait for all context data to start the processing. Speculative Computation is composed by two phases, Process Reduction and Fact Arrival. The former, represents the normal processes of reasoning over context data, using default values, if necessary. The latter, represents the arrival of new context data to be added to the processes. Figure 1 presents a generic architecture for Speculative Computation.

The figure above is composed by steps to help the understanding the flow of it. In *step 1*, a set of data is received by the software agent A that will process the context. There are gaps of information within this set, which represents missing context values. In other words, for some reason the data was not able to be sent to the system. In *step 2*, the process of reasoning over the context data starts, referring to the first phase of Speculative Computation. For this, the software agent A fills the gaps of data with default values (*step 2.a.*). The agent uses the known values that were valid in previous situations as basis. At the same time, agent A queries other agents about the missing data (*step 2.b.*). If the reasoning finishes before the agent A receives the answers, the default values are taken as being true. However, if the other agents answer the queries before the reasoning is finished, the flow of the Speculative Computation passes to the second phase, *Fact Arrival*, (*step 3*). In this case, the active processes are suspended, and agent A has to analyse the context data comparing them to the default values that were used until this point. If the context data from the answers are consistent with the default values, the suspended processes are resumed, and the reasoning continues until it is finished (*step 4.a.*). However, if they are not consistent, the default values must be ignored and replaced by the new context data from the answers of the queries (step 4.b.). In this case, the suspended processes are ignored, and a *Context Data Revision* is performed, replacing the missing values with the arrived context data. After that, a new phase of process reduction is started (*step 5*), to reason over the set of context data. This happens every time that software agents answer queries from agent A with new context data.

Nowadays, several approaches of context-aware system discard the missing values making the system to ignore them during the reasoning. The problem of these approaches is that this makes the system to use less data to build models to be processed. Therefore, the system may not clearly identify what happens in the environment and the probability of orchestrate a wrong execution of applications tend to grow considerably. Using these default beliefs as input, Speculative Computation helps to improve the reasoning over a context model, reducing the time of decision making and execution of applications.

5 Conclusion

Context-aware systems are, gradually, becoming more common. Smartphones that naturally interact with the user are already available on market. These new devices will help the consolidation of Ambient Intelligence, through Smart Environments. To build such kind of domains we need to develop context-aware systems, which have the premise of sense what is happening in the environment and act based on that. The system should use information of context to execute services aiming to assist users. However, this reasoning is not a trivial task, once the data collected by sensors sometimes is not complete as should be. Due to the dynamicity of the domain and its complexity, there will be times where the system will not be able to build a complete scenario due to the incompleteness of information or its reliability. This uncertainty generated must be handled appropriately, once the main idea of context-aware systems is for them to work autonomously. Considering this, we can conclude that one of the main of issues of context-aware systems is the study of approaches of handling uncertainty and what to do in situations where the system is not able to build complete context model due to information vague, partial or out-dated.

Acknowledgements

This work has been supported by COMPETE: POCI-01-0145-FE\-DER-0070

43 and FCT – Fundação para a Ciência e Tecnologia within the Project Scope UID/CEC/ 00319/2013.

References

1. Amdouni, S. et al.: Handling Uncertainty in Data Services Composition. In: International Conference On Services Computing. 11 ed.; Anchorage. IEEExplore, p. 653-660. (2014)
2. Aztiria, A. et al.: Learning Frequent Behaviors of the Users in Intelligent Environments. IEEE Transactions On Systems, Man, And Cybernetics: Systems, v. 43, n. 6, p. 1265-1278, November. (2013)
3. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive Mob. Comput., vol. 6, pp. 161–180. (2010)
4. Bhatnagar, R. K. and Kanal, L., N.: Handling Uncertain Information: A Review of Numeric and Non-Numeric Methods. 1986. In: Uncertainty in Artificial Intelligence. Ed. Kanal, L., N. and Lemmer, J. F. Elsevier Science Publishers B.V. (2014)
5. Cook, D. J., Augusto, J. C., Jakkula, V. R.: Ambient intelligence: Technologies, applications, and opportunities. Pervasive and Mobile Computing, v. 5, n. 4, p. 277-298. (2009)
6. Delmastro, F., Arnaboldi, V., Conti, M.: People-Centric Computing and Communications in Smart Cities. IEEE Communications Magazine, v. 54, n. 7, p. 122-128, July. (2016)
7. Indulska, J., Sutton, P.: Location management in pervasive systems. In: Proc. Australasian information security 2003 - Volume 21, pp. 143–151. (2003)
8. Mokhtar, S. B. et al.: Efficient Semantic Service Discovery in Pervasive Computing Environments. In: International Middleware Conference (MIDDLEWARE'06), 7 ed.; Bordeaux, France: Proceedings of the ACM/IFIP/USENIX. p. 240-259. (2006)
9. Novais, P., Carneiro, D.: The role of non-intrusive approaches in the development of people-aware systems. Prog Artif Intell, cidade, Berlin, v. 5, n. 3, p. 215–220, August. (2016)
10. Nugroho, L. E.: Context-Awareness: Connecting Computing with Its Environment. In: International Conference On Information Technology, Computer, And Electrical Engineering (ICITACEE), 2 ed.; Semarang, Indonesia. (2015)
11. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulous, D.: Context Aware Computing for the Internet of Things: A Survey. IEEE Communications Surveys & Tutorials, vol. 16, no. 1, First Quarter. (2014)
12. Pietschmann, S., Mitschick, A., Winkler, R., Meissner, K.: Croco: Ontology-based, cross-application context management," in Semantic Media Adaptation and Personalization,. SMAP '08. Third International Workshop on, dec. 2008, pp. 88 –93. (2008)
13. Samia, B., Allel, H., Aicha, A. M.: Handling Preferences Under Uncertainty In Recommender Systems. In: IEEE International Conference On Fuzzy Systems (FUZZ-IEEE). 1 ed.; Pequim, China: p. 2262-2269. (2014)
14. Vanrompay, Y., Pinheiro, M. K., Berbers, Y.: Context-Aware Service Selection with Uncertain Context Information. In: International Discotec Workshop On Context-Aware Adaptation Mechanisms For Pervasive And Ubiquitous Services (CAMPUS 2009), 2 ed. Lisboa, Portugal: Proceedings of the Second International DisCoTec Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services. v. 19. (2009)
15. Yanwei, S., Guangzhou, Z., Haitao, P.: Research on the context model of intelligent interaction system in the internet of things," in IT in Medicine and Education (ITME), 2011 International Symposium on, vol. 2, dec., pp. 379 –382. (2011)
16. Yaghlane, A. B., Denoeux, T., Mellouli. K.: Elicitation of expert opinions for

constructing belief functions. In Uncertainty and Intelligent Information System, pages 75–89. (2008)

17. Oliveira, T. J. M. Clinical Decision Support: Knowledge Representation and Uncertainty Management. PhD thesis, University of Minho - Doctoral Programme of Biomedical Engineering, R. da Universidade, 4710-057, Braga - Campus de Gualtar, 4. (20

18. 17)