Is ‘Context-aware Reasoning = Case-based Reasoning’?

Nawaz Khan, Unai Alegre, Dean Kramer, Juan Carlos Augusto

R.G. on Development of Intelligent Environments,
Department of Computer Science, Middlesex University, London, UK
N.x.Khan,U.Alegre,D.Kramer,J.Augusto@mdx.ac.uk

Abstract. The purpose of this paper is to explore the similarities and differences and then argue for the potential synergies between two methodologies, namely Context-aware Reasoning and Case-based Reasoning, that are amongst the tools which can be used for intelligent environment (IE) system development. Through a case study supported by a review of the literature, we argue that context awareness and case based reasoning are not equal and are complementary methodologies to solve a domain specific problem, rather, the IE development paradigm must build a cooperation between these two approaches to overcome the individual drawbacks and to maximise the success of the IE systems.

1 Introduction

Todays complex systems generate vast amounts of data from the external environment in order to provide an informed decision to the user. This data is collected through sensors, cameras, computers and by many other input devices, thus, equipping us with intelligent services. However, this sheer amount of data may cause the system effectiveness to be lost in hyper-space if not classified into useful and meaningful preference based set of choices [33]. In the domain of ubiquitious computing, developers are faced with plethora of techniques [6]. Some of these are well-established, others are reasonably new. Some researchers even combined multiple approaches as an integrated process[32]. This article looks at commonly used approaches, an established technique like case-based reasoning (CBR) and a newly more ad-hoc development like context-awareness reasoning (C-AR). The question we want to examine in this paper is whether context-awareness brought something truly new to ubiquitous computing in terms of knowledge presentation and reasoning. More specifically we are exploring how the synergies between C-AR and CBR can improve the quality of users' interaction with IE system.

Ubiquitous computing may use context-awareness or a case-based reasoning to develop a personalised recommendation, may be looking at the past history of the user, or may be by understanding his attempt to search and find a product [36]. To illustrate this: we may say, a user can visit Amazon to find a book for himself, in this case the recommendation set for the user will be those books that the user has bought in the past. However, if the user is willing to purchase book as a gift for his wife, then the choice set of recommendation will be different than the books he buys for himself. So, we would see these two disjoint events as an application of case base reasoning or may be an application of context awareness. This article presents first a brief description of C-AR and CBR in section 2 and 3, we compare them conceptually in section 4 and 5 and we illustrate the conceptual comparison in section 6 with a case study.

2 Concept of Context Awareness (C-AR)

Contextual awareness stemmed from the desire of Ubiquitous Computing to increase the usability of computerised systems through the implicit interpretation of what the user wants. The concept of Context-awareness was introduced by Schilit and Theimer [34] when they proposed information mapping for a network system. Context-awareness deploys an environmental information set that is available for a given circumstances in order to perform an act that assists a user to make a decision [15]. The ambition is to embed human-like contextual awareness into systems, to make the interaction with them more natural. It has to be mentioned that there is a lack of agreement on the definition of context [3]. The theory of context varies as it has been defined in different research areas, for example, Schilit and Theimer identified a context by means of location, identities, neighbourhood objects and people and any variations on the objects[34]. It presents the multifarious dimensionality towards identifying a context and these are: physical factors dimension, human factors dimension and time dimension. Whereas, Benerecetti et al. [11] classified context into two subsets, namely, 'physical context' and 'cultural context'. The cultural context includes the users personal data, social and belief information.

Dey and Abowd [14] defined the context as 'any information that can be used to characterise the situation of an entity, where an entity can be a person, place, physical or computational object that is considered relevant to the interaction between a user and an application, including the user and the application themselves'. Dey extends his work by proposing a framework where context values will be obtained from four types of context, i.e., location, identity, time and activity. Abowd et al. [2] considered these as 'primary context'. Furthermore, they considered secondary context any context that can be found using primary context. Perera et al. [31] classified the context into operational and conceptual. In the operational view, context is considered either primary if the information is retrieved without using other existing context information or secondary, if it can be computed using existing context. In the conceptual perspective, context is classified into activity, time, identity and location [14]. The rest of the contexts in this perspective can be obtained combining the elements between themselves. He proposed the context aware application as 'a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the users task'.

Dey [15] suggested that the three main activities that context aware applications should perform are: (i) the presentation of information and services to a user, (ii) automatic execution of a service, and (iii) tagging of context to information for later retrieval. C-AR have been closely related to human interaction since its inception. Alegre et al. [4] studied the way the system can interact with the users, classifying it into the Execution and Configuration phases. The first one refers to the actions/behaviours of the system when a specific situation arises. The second one is related to the adjustment of actions/behaviours that a system will be exhibiting in the future. These two independent modalities of interaction can have a degree of active/passiveness. By 'active', they refer to the interactions in which the system changes its content autonomously. By 'passive', they mean those where the user has explicit involvement in the actions taken by the system. By observing the interaction modalities of users with context-aware systems, they extended Dey and Abowds [15] classification of the features of a C-AR as: a) Presentation of information to the stakeholders; b) Active or passive execution of a service; c) Active or passive configuration of a service; d) Tagging context to information [3].

Typically, the information of a context-aware system has four stages [31]. They are:

1. Acquisition: where the data considered as context is retrieved from the sources. These sources can be very disparate, including any type of physical, logical or virtual sensors [20].
2. Modelling: after the data is acquired, it needs to be Modelled, formatting the values from the sensors into something that can be used by an application (e.g., from coordinates to the name of a city).
3. Reasoning: this data can also be used to create higher level information by Reasoning.
4. Dissemination: Low and high level context information has to be available to the different applications, through a process known as Dissemination.

The next two sections focus on the techniques available for context modelling and reasoning.

2.1 Context Modelling

One important aspect of a C-AR, is how the information will be processed and stored. Typically, these kind of applications use a context model, which identifies a concrete subset of the context that is realistically attainable from sensors, applications and from users and then subsequently are exploited in the execution of the task [19]. The most popular context modelling techniques are [18][31][14]:

- Key-value Modelling: key-value pairs are used for context information.
- Markup Scheme Modelling: Models data using tags such as XML.
- Graphical Modelling: demonstrates relationships of context data using graphical notations

– Object Based Modelling: Object oriented concepts are used to model data using class hierarchies and relationships
– Logic Based Modelling: Facts, expressions and rules are used to represent context information.
– Ontology Based Modelling: The context is organized into ontologies, using semantic technologies.

2.2 Context Reasoning

Abowd et al. [2] suggest classifying reasoning in C-AR as: i. Pre-processing: where the aim is to make later processing easier by recognizing the relevant context attributes, handling missing ones and clearing the data, ii. Sensor data fusion: where the data is integrated from multiple resources and iii. Context inference: that creates high-level information from low-level information.

Further to this, numerous works [31][8][19], surveyed the approaches deployed for context-reasoning in Ubiquitous computing. They are summarised as:

– Supervised learning: Training examples are collected, labelled according to the expected results and then a function is derived to generate the expected results by using the training example data.
– Unsupervised learning: This category of techniques can find hidden structures in unlabelled data.
– Rules: It defines what has to be done in a simple way, typically in IF-THEN-ELSE format.
– Probabilistic logic: Allows decisions to be made based on probabilities attached to the facts related to the problem.
– Fuzzy-logic: In traditional logic acceptable truth values are as 0 and 1, in fuzzy logic partial values in between are acceptable. This method is similar to probabilistic reasoning, but membership set represents degree of confidence rather than probability.
– Ontology based: Its foundations are on description logic, a family of logic based knowledge representation formalisms.

3 Concept of Case-based Reasoning (CBR)

CBR is an artificial intelligence approach to determine a similarity amongst a set of cases Kolodners [25] CBR method of similarity indexing is foundation of many intelligent systems. This method entails, indexing cases, retrieving the best past case from memory, adapting the old solution to conform to the new situation, testing whether the proposed solution is successful, and learning to prohibit solution fails. CBR has been viewed as a technology for automated, intelligent problem solving based on cases that present a circumstance.

CBR is based on the notion that the similar cases will have similar solutions. Hence, this method looks into the previous cases, analyse the similarity of cases, analyse the previous solutions of those cases and then propose a new solution based on the similarity of the previous cases and solutions [38]. The knowledge base contains the general domain knowledge, which is the set of rules needed for reasoning in this domain and sometimes some general facts (i.e., which are known to be always true). In a rule-based system, these IF-THEN rules and general facts represent the knowledge of the system. However, one of the important advantages of CBR is its ability to evolve by accumulating cases and each time a new solution set is emerged from the evolving case base. The nearest neighbour algorithm determines the similarity or dissimilarity of a new case with the case base and follows a cyclical process of Retrieve-Reuse-Revise-Retain [1].

However, the overall objective of this case based reasoning is to build up context reasoning, in other words, representing knowledge. Therefore, the concept of context awareness underpins the context reasoning for knowledge representation. Context reasoning system development paradigm uses two types of symbolic knowledge as a context: Facts and Rules. The facts and rules are then used for reasoning within the domain of a given problem. The reasoning process is based on logic, and allows the rule-based system to search through the problem space and arrive to a conclusion or a solution of a given problem based on the initial conditions (initial facts). The initial condition can be, for example, a set of facts describing a fault in a system, and the conclusion may be another fact stating the reason of that fault. Unlike conventional programs in which the precise steps are defined in clear algorithmic terms, rule-based systems are largely declarative and parallel in nature.

Perera at al. [31] describes the knowledge reasoning as a process of giving high-level context deductions from a set of contexts. The most fundamental aspects of building up reasoning involves: i. propositions: description of facts, ii. Logical operators: negate, conjunction and disjunction and iii. Inference: deriving conclusion about one fact based on a set of rules, in other words, fact is the implication of rules which are derived from context data. However, one crucial aspect of reasoning inference is conflict resolution [7], therefore, there should be a strategy to select one fact from the conflict set. There are different strategies for conflict resolution, for example, i. Refraction: once the rule has fired it will not be used, ii. Recency: use the fact that has been used recently in such a situation, iii. Specificity: use the fact with the more specific contexts, iv. Priority: ranking the factors and selection of priority with the highest rank, and and v. Parallel: all facts are contributed as a different set of reasoning.

4 CBR and C-AR in IE Development

In the Intelligent Environment development paradigm, sensors are utilised to capture contexts to produce cases. Therefore, capturing all these contexts for reasoning is not feasible [31]. The case based system implementation relies on decision making based on three parameters: pattern recognition-conflict resolution-action. The Recognition-action cycle is repeated until the solution is reached or no applicable rule can be found in the knowledge base. The reasoning build up by the environment is very similar to those of human experts. Basically, there are two ways to carry out the reasoning process: forward chaining (or data-driven) and backward chaining (or goal-driven). In both reasoning processes, we need to use the search strategies to guide the reasoning. Because a practical rule-based system normally contains a high volume context information which creates a large searching space.

There is much evidence where we see CBR is applied to the development of context aware applications and vice-versa [32]. It is demonstrated in Leakes work [27] that CBR can benefit developing a context aware system in designing smart homes where users can customise the requirements by building a case and system can gain knowledge from the context of the environment and then match a case. Kofod and Aamondt [21][22] developed a mobile context aware system where context information is embedded into cases for a situations assessment. Kumar et al. [26] developed an interesting system for e-commerce applications where two distinctive cases are created, i.e., user cases and product cases. These two cases are built upon the context of users and products to incorporate multiple context dimensions to the cases. Lee and Lee [28] developed a music recommendation system where users behaviour, demographics and context details are used for a case base recommendation. However, it can be argued that incorporating CBR into C-AR depends on the availability of contextual knowledge of an action, if the requisite contextual data is not available CBR can support C-AR by recalling previous case. This integration can improve quality of IE system when domain knowledge is limited.

5 CBR and C-AR Distinctions and Dynamic Relationship

This section focuses on the synergies and differences between the aspects of CBR and C-AR. Some of the techniques for CBR could also be used in C-AR and vice versa. On one hand, the origins of context-awareness stemmed from the need of different areas such as Ubiquitous Computing, Ambient Intelligence or Intelligent environ-ments of having an enhanced human-computer interaction. On the other hand, the CBR method is used to solve problems that are based on similar solutions applied before. Applied to context-aware systems, this could help its configuration, enabling the evolution of the system after its implementation. At a first sight, it looks like CBR could be used to develop the configuration of context-aware systems in both its active and passive modalities.

Also, the information in CBR and C-AR is treated in a different way. C-AR need to acquire the information from distributed and heterogeneous sources. All this data needs to be modelled and translated in a meaningful way, so that it can be reasoned to obtain higher level information. Finally, the data needs to be ready for different applications on demand. On the contrary, CBR follows a different process. The first step is similar to the acquisition of context, but instead of using different distributed sources, it retrieves the information from a memory of existing cases. Then, this process is followed by the reuse stage, where the solution is mapped from the previous case to the target problem. This stage can also involve an adaptation to the solution according to the circumstantial needs. Once the solution is mapped to the target situation, it is tested and revised until it has been successfully adapted to the target problem. At this point the solution is stored in the case memory. As it can be observed, these two process are quite different.

The context modelling stage is related to the knowledge representation of the system. These techniques are currently used to assign a meaning to the information that comes from the sensors but it could also be used to store the different cases in the memory. Case based reasoning supported by a rich knowledge model could be a promising approach to assess situations by being context aware [22][23]. Markup scheme and Ontology based modelling seem the most promising ones for CBR. Reasoning techniques in context-aware systems, are used for pre-processing, sensor data fusion and context inference. On the other hand, CBR takes an alternative view. Rather than seeing reasoning as primarily a composition process, it looks at it as remembering one or concrete set of concrete instances or cases and basing decisions on comparisons between the new situation and the old instance [25]. Nevertheless, most of the reasoning techniques used in context-awareness have been also used in CBR approaches [10].

Kofod-Petersen [21] identified the main challenges for using CBR in Ambient Intelligence are: i. Acquiring the initial cases , ii. Coping with the vast number of case being constructed during run time, iii. Knowing when to initiate a case-based reasoning cycle and iv. Knowing whether a case was classified correctly.

In Table 1, we summarise the pros and cons of both the C-AR and CBR methods of problem solving. We also attempt to identify the synergies between these two methods and derived a distinction between C-AR and CBR (Table 2). Table 2 points out how these two reasoning development methodologies are complementary to each other.

6 Case Study

In an effort to understand the differences and synergies between both C-AR and CBR, let us consider a case study based on an EU Project named POSEIDON [5]. The project is aimed at increasing the social independence and integration of people with Downs Syndrome through the use of IE technologies. An approach taken in POSEIDON to help better integrate our user group in society is through the improvement of individual effectiveness at navigating their environment, for example from home to work. In Table 3, we summarize the user driven situations within our case study where C-AR and CBR can be applied as a solution.

6.1 Uses of Context-Awareness in POSEIDON

Services developed in the POSEIDON project include a mobile element. This mobile element is composed of multiple services, including an app for navigation and calendaring, and a mobile context reasoner. The context reasoning for the mobile is designed to be a centralized entity for providing context-awareness

Table 1. A comparison study of CBR and CAR in IE development

Attributes	Context aware Reasoning	Case based Reasoning
Generic Feature	Improve the HCI using implicit information from situation Process involves: Acquisition, Modelling, Reasoning and Dissemination	Solve problems based on solutions to similar problem Process involves: Retrieve, Reuse, Revise and Retain
Merging Challenge	CBR in C-AR: Acquiring the initial cases Coping with high amount of cases constructed at runtime Knowing when to initiate a case-based reasoning cycle Knowing whether a case was classified correctly	CAR in CBR: Inferring facts from cases Storing the inferences in a meaningful way in the context model
Merging Advantages in IE	Evolution: C-AR can benefit from the possibility of retrieving similar cases and adapting them according to the circumstances	pattern recognition: context validation through data semantics
		Conflict Resolution: Semantics matching from context database
	Expert system like solutions: It can enable C-AR to solve problems in contextual situations that are partially understood since it creates a better understanding of the situation for each case.	Knowledge representation: Allow edit, search and debugging knowledge
	High level context awareness can be generated from database	Knowledge base: Reasoning process accumulates new facts and add them to the searching space.
Merging challenge in IE	Evolution: CBR can adapt solutions that are better match for the current situation. Nevertheless, this introduces some loss of control over what the system is executing and why	Pattern recognition: Depends on the reliability of the context acquisition. Semantic conflicts may arise.
	Heavy-weight operations: CBR requires to compare all the possible cases at runtime, which can be computationally expensive.	C-AR typically operates with devices that do not have plentiful resources.
		Conflict resolution: validity of the resolution depends on real data sets. New set of data are generated without the context facts.

Table 2. Synergies between C-AR and CBR methods for IE development

Synergies	Configuration interaction modality of C-AR could be based on the solutions to similar cases. Internal and external contexts can be added in an intelligent environment system to derive reasoning for a case in order to represent knowledge. Context reasoning enables to build up a case for a fact and can represent knowledge. Cases are modelled by deploying context mapping based knowledge representation. Data storage for C-AR and CBR is domain specific and establishes a relationship amongst the facts. It is supported by meta logic and follows the same principles of data acquisition Context modelling involves knowledge acquisition knowledge representation and knowledge extraction; thus, build a case for a problem. Context distribution is achieved when a reasoning is generated from a set of contextual facts, thus, represent the knowledge Goal driven case based reasoning can be supported by multi-dimensional context acquisition and by supplying personalised and localised context reasoning

services to multiple mobile applications on a single device. This reasoner can infer contexts using data from mobile sensors, data from web services, and data from other applications on the device. Context information is inferred over by the use of stream reasoning, using C-SPARQL [9] for atomic level contexts, and forward reasoning [4] for context aggregations. These rules enable the rea-soner to infer contexts, which are then broadcasted to each of the requiring applications.

In POSEIDON, context-awareness is used predominantly to provide information that can benefit the user, and also detect and provide options if the system suspects the user requires assistance. Different examples of the use of context include:

- Clothing advice based on weather conditions at planned destination. This can include to wear a coat if it is particularly cold, or take an umbrella if rain is expected. Lastly, we suggest to apply sunblock if the day is especially hot and sunny.
- Offering to call their carer if the system finds they deviate from the route, or are standstill for too long. Deviations requiring the offering of help include large single deviations where rerouting is required, or when the primary user makes small deviations too often. Being standstill for too long can be sign that their connection service e.g. a bus has not arrived, and may require further assistance.
- Determining when to begin giving navigational instructions based if they are indoors or outdoors. Instructions should be given when they are required. If

the user is not outside ready to begin, it is of little benefit informing them make a particular turn.

6.2 Possible Uses of Case Based Reasoning

When considering possible uses of CBR within our project, we can foresee both isolated uses, and uses in conjunction with the context-awareness technologies already adopted. Firstly, in our project part of improving the independence of our user group is through navigation assistance. A second way that we can assist is helping our user group in daily tasks. This can include job related tasks, and leisure/home related tasks e.g. making lunch, cleaning. For our user group, they can have difficulty with abstraction, which can make learning tasks difficult, even if similar to each other. Often, minor differences in the context can render a task as a completely different one. Already, video based instructions are used for different task including making a coffee. One problem that arises is that new videos are required for different coffee machines and equipment they might use. A user friendly expert system could be useful for these situations assist the users by applying the knowledge of other similar tasks. This will be helpful in situations such as the ones described as only differing fragments of the task will be needed to be added to the system. In terms of supporting the creation/maintenance of cases, this will need to be an ongoing task of the primary user, and their carers.

Researchers have in the past used CBR in context-aware systems [22][23]. These approaches tend to have strengths on the ability to deal with new situations at runtime, where approaches based on predefined adaptation paths can struggle. In terms of context-aware development, CBR can be more useful in the adaptation/result phase, after context inference. The reason for this is that when adding additional contexts will often require developer intervention. This can include implantation of components to acquire new raw data from a sensor/data source, and specific rules on what situations that raw data represents, including aggregations with others. How the system behaves and/or adapts when in that collection of context situations can however change. For example, when considering navigational assistance, it is possible as the individual becomes more independent, other conditions in conjunction with the system determining the user needs assistance could determine a different action is taken. This could include not providing them the ability to call their carer, if they know the way home. Using CBR for adaptation strategies could also help prevent the issues with too many cases caused by arbitrary raw data including time.

7 Discussion

This article aims to initiate the further study of two different approaches that could potentially be used for strengthen each other. On one hand, C-AS aims to improve HCI by using of context, on the other hand, CBR tries to solve problems by using similar solutions used before to solve alike problems. C-AS can benefit from retrieving simi-lar cases to provide different services or information. This

Table 3. Challenges that can be met by C-AR/CBR or combination of both

Problem domain	Solution domain and reasons
Context raw data highly variable	Rule based context awareness, as does not require too many cases to be created
Dealing with evolving context adaptation	CBR based adaptation, as follows new adaptations strategies to be applied and evolve.
Knowledge based system to assist users in tasks	CBR only, as it can assist the user in new situations where a particular task guide does not currently exist

could enable this kind of systems to solve problems that are partially understood since it can enable the creation of a better understanding in each case. On the other hand CBR could benefit from the pattern recognition, conflict resolution knowledge representation and knowledge bases of C-AS. The foremost challenges we identified are: i. Lack of methodical description of why context awareness or case based rea-soning is adopted for the implementation of intelligent systems, ii. Can we perform the same task by deploying either of these two methods, and iii. If there is a set of synergy in between these two approaches where they can supplement each other to derive reasoning.

There are challenges when trying to use CBR in C-AR. The first one will be to acquire the initial cases and coping with a high amount of cases constructed. Also, knowing when to initiate a case-based reasoning cycle and if it was correctly classified. Involvement from the users could be required for this purpose, such as feedback. On the other hand, when using C-AR in CBR, the challenge will be to infer facts from cases and storing the inferences in a meaningful way. The term 'context' mainly used when contexts are derived when sensors perceive an environment and when there are complex method of acquiring data that can add more semantics to the data set in order to perceive an environment, i.e., cameras, move around sensing devices. The heterogeneous sources of information that elicit physical contexts as well as cultural contexts, an intelligent environment can build and use a contextual picture of the situation to perceive the environmental change [5]. Whereas, 'context reasoning' terms are used when high degree of meaningful information is extracted from the context data and provide a fact for an action to be taken [30]. This context awareness is middleware that supplies necessary feedback to the interface to adapt a situation for better services. Interestingly, contextual information can be used to derive a set of higher level contexts to deal with domain specific cases, which has then be termed as case based reasoning in several applications. The ap-proach of higher degree of context awareness to build up a fact are evident in many intelligent systems, i.e., CARISMA [12], SOCAM [18] etc. Context-aware systems can be configured in order to evolve for adapting the changing needs of the users. We acknowledge that the adaptation of a context-aware system can be supported by CBR.

The challenge of building IE systems is to know exactly what reasoning method to use, however, it is still not very clear if context awareness and case base reasoning performs the same services or the concepts can be used interchangeably depending on the problem solution that we sought for. However, we can reach to a consensus that context awareness deals with continuous environment adap-tation when contextual data are captured in real time, case based reasoning is the knowledge representation from a set of the facts. In our case study, context awareness predominantly assists the users by capturing atomic level context and then aggregating them to enable users to adapt specific situations. We can see that the use of CBR can help improve systems like POSEIDON by use of features in isolation, e.g., to assist task learning, but also in the current context-awareness system through use in each application on selecting the correction adaptation path for that set of context situations it is in. This solution can improve how the system deals in particular situations by allowing the adaptation strategy to change, but also helps keep the number of cases smaller than using it in the context inference itself.

8 Conclusion

In this paper we attempted to compare context-aware reasoning and case-based reasoning in order to build an argument how their dynamic relationship can assist scientists developing intelligent environment system. The applicability of these approaches is used in problem solving, but, it is not clear their range of applicability in the area of IEs. We have studied the main points where context-aware computing can benefit of case-based reasoning. We have provided a first insight on the distinction between when to use of context-awareness or case based reasoning and the key synergies for merging these approaches. In this review, with the use of literature and a case study, we made a case that expectation on intelligent system usability influence the selection of the methodologies. However, it is evident from our discussion that the fundamental development paradigm for such an intelligent system must consider multi-dimensional context data capture and then making these context data available to middleware in order to build up the context awareness and this context awareness gradually develops the cases and facts for knowledge representation. In other words, we can say the synergies between case based reasoning and context awareness are the fundamentals for the success of an intelligent system where contextual data is acquired for immediate situation adaptation.

References

1. Aamodt, A., Plaza, E. (1994) Case-based Reasoning: Fundamental Issues, Methodological Varia-tions, and System Approaches. Artificial Intelligence Communication 7, 3959.

2. Abowd, G.D, Dey, A. and Brown, P.J et al. (1999). Towards a better understanding of context and context-awareness, 1st international sym-posium on Handheld and Ubiquitous Computing, ser. HUC 99. Springer-Verlag, pp. 304307.
3. Alegre, U. , Augusto, J.C. , Aztiria, A. , (2014). Temporal reasoning for intuitive specification of context-awareness. In: Proceedings of the 2014 International Conference on Intelligent Environments (IE). IEEE, pp. 234241.
4. Alegre, U., Augusto, J.C., Clark, T. (2016). Engineering Context-Aware Systems and Applications: A survey. Journal of Systems and Software Vol. 117, pp. 55-83. ISSN 0164-1212. Elsevier.
5. Augusto, J.C., Grimstad, T., Wichert, et al. (2013). Personalized Smart Environments to Increase Inclusion of People with Downs Syndrome. 4th International Joint Conference on Ambient Intelligence. pp 223-228. 3-5th. Dublin. Springer Verlag
6. Augusto, J.C., Callaghan, V., Cook, D., Kameas, A., and Satoh, I., (2013). "Intelligent Environments: a manifesto" Human-centric Computing and Information Sciences, 3:12 doi:10.1186/2192-1962-3-12
7. Aztiria, A. , Augusto, J.C. , Basagoiti, R. , Izaguirre, A. , Cook, D.J. , (2013). Learning frequent behaviours of the users in intelligent environments. IEEE Trans. Syst. Man Cybern. Syst. 43 (6), 12651278 .
8. Baldauf, M., Dustdar, S., Rosenberg, F., (2007).A survey on context-aware systems, International Journal of Ad Hoc and Ubiquitous Computing 2 (4) 263277.
9. Barbieri, D., Braga, D., Ceri, S., Della Valle, E., and Grossniklaus, M., (2010). Querying RDF streams with C-SPARQL. SIGMOD Rec. 39, 1, pp. 20-26. DOI=http://dx.doi.org/10.1145/1860702.1860705
10. Begum, Shahina, et al. (2011) Case-based reasoning systems in the health sciences: a survey of recent trends and developments. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 41,4: 421-434.
11. Benerecetti, M., Bouquet, P., Bonifacio, M., (2000). Distributed Context-Aware System. Human-Computer Interaction 16, 213228.
12. L. Capra, W. Emmerich, and C. Mascolo, (2003). Carisma: context-aware reflective middleware system for mobile applications, IEEE Trans. Softw. Eng., vol. 29, no. 10, pp. 929 945.
13. Cuddy, S., Katchabaw, M., Lutfiyya, H., (2005). Context-Aware Service Selection based on Dynamic and Static Service Attributes. In: Wireless and Mobile Computing, Networking and Communica-tions, IEEE International Conference
14. Dey, A.K., (2001). Understanding and Using Context. Personal and Ubiquitous Computing 5, 47
15. Dey, A.K., Abowd, G.D. (2000). Towards a Better Understanding of Context and Context-Awareness. CHI 2000 Workshop on the What, Who, Where, When, Why, and How of Context-Awareness, The Hague, Nether-lands, pp. 16
16. Dey, A.K., Futakawa, M., Salber, D., Abowd,G.D. (1999). The Conference Assistant: Combining Context-Awareness with Wearable Computing. In: Proceedings Symposium on Wearable Computers, pp. 2128
17. Dey, A.K., Salber, D., Abowd, G. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction 16, 97166
18. Gu, T., Pung, H. K., and Zhang, D. Q., (2005). A service-oriented middleware for building context-aware services, J. Netw. Comput.Appl., vol. 28, no. 1, pp. 118.
19. Henricksen, K., (2003) A framework for context-aware pervasive computing applications, Computer Science, School of Information Technology and Electrical Engineering, The University of Queensland.

20. Indulska J., and Sutton, P., (2003). Location management in pervasive systems, Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21, Australian Computer Society, Inc., 2003, pp. 143151.
21. Kofod-Petersen, A, (2006). Challenges in Case-Based Reasoning for Context Awareness in Ambient Intelligent Systems, 1st Workshop on Case-based Reasoning and Context Awareness (CACOA'06).
22. Kofod-Petersen, A., Aamodt, A.: (2006) Contextualised ambient intelligence through casebased reasoning. In: Proceedings of the Eighth European Conference on Case Based Reasoning (ECCBR 2006).
23. Kofod-Petersen, A., Aamodt, A., (2003). Case-based Situation Assessment in a Mobile Context-Aware Systems. In: AIMS2003. Workshop on Artificial Intelligence for Mobile Systems, Seattle.
24. Kolodner, J.L., (1993). Case-based Reasoning. Morgan Kaufman, San Mateo, CA.
25. Kolodner, J. (2014). Case-based reasoning. Morgan Kaufmann.
26. Kumar, P., Gopalan, S., Sridhar, V. (2005). Context Enabled Multi-CBR based Recommendation Engine for E-Commerce. In: ICEBE 2005. Proceedings of IEEE International Conference on e-Business Engineering, pp. 237244. IEEE Computer Society Press, Los Alamitos.
27. Leake, D., Maguitman, A., Reichherzer, T. (2006). Cases, Context, and Comfort: Opportunities for Case-Based Reasoning in Smart Homes. In: Augusto, J.C., Nugent, C.D. (eds.) Designing Smart Homes. LNAI, vol. 4008, pp. 109131. Springer, Heidelberg.
28. Lee, J and Lee, C. (2007). Context Awareness by Case-Based Reasoning in a Music Recommenda-tion System H. Ichikawa et al. (Eds.): UCS 2007, LNCS 4836, pp. 4558, 2007. Springer-Verlag Berlin Heidelberg.
29. Ma, Tinghuai, et al. (2005) Context-aware implementation based on CBR for smart home. Wireless And Mobile Computing, Networking And Communications, (WiMob'2005), IEEE International Con-ference on. Vol. 4. IEEE.
30. Mishra, N., Petrovic, S., and Sundar, S. (2011). "A self-adaptive case-based reasoning system for dose planning in prostate cancer radiotherapy." Medical physics 38.12 (2011): 6528-6538.
31. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D., (2014). Context Aware Computing for The Internet of Things: A Survey, IEEE communications Surveys and Tutorials. VOL. 16, NO. 1.
32. Pla A., Coll J., Mordvaniuk N., Lpez B. (2014) Context-Aware Case-Based Reasoning. In: Prasath R., OReilly P., Kathirvalavakumar T. (eds) Mining Intelligence and Knowledge Exploration. Lecture Notes in Computer Science, vol 8891. Springer, Cham
33. Resnick, P., Varian, H.R. (1997). Recommender Systems. Communications of the ACM 40, 5658.
34. Schilit, B., Theimer, M. (1994). Disseminating Active Map Information to Mobile Hosts. IEEE Network 8, 2232.
35. Schmidt, A., Beigl, M., Gellersen, H.W. (1999). There is More to Context Than Location. Computers and Graphics 23, 893901.
36. Tsatsoulis, C., Cheng, Q. , Wei, H.Y., (1997). Integrating cased-based reasoning and decision theory, IEEE Expert 12 (4) 4655.
37. Ur, B., McManus, E., Ho, M.P.Y., and Littman, M.L., (2014). Practical trigger-action programming in the smart home. In Proceedings of CHI 2014, ACM, pp. 803-812.
38. Watson, I. (1997). Applying Case-based Reasoning: Techniques for Enterprise System. Morgan Kaufmann, San Francisco, CA.