Context-based techniques for improving database management

Hassane TAHIR
Links Consultants France
24 rue de Prony, 75017, Paris, France

Abstract
The business computer applications abound by a wide variety of "incidents", some of them are minor and some of them are complex requiring high expert skills and efficient collaboration between experts in order to solve them. These practices reflect the building of operational knowledge, step by step, by Database Administrators (DBAs). They are collected and compared by organizations in order to establish secure procedures for encountered incidents. This kind of procedures can be seen as internal procedures as they have been established within an organization and not delivered by DBMS vendors. In this sense, procedures are collections of safety action sequences allowing solving a given incident in any case. These procedures are based on practices, but eliminate most of contextual information and specific features of each incident. This paper presents techniques for contextualizing database management with a focus on "Contextual Graphs" formalism.

Keywords: *Context, Contextual Graphs, DBA, Database Management, Practices, Procedures.*

1. Introduction

In the area of database administration, DBMS vendors provide standard procedures for solving most of the incidents that have been well known for a long time (bad memory configuration, buffer caches, database cashes, security bugs, etc.). In addition, organizations have also established, from their perspectives, their own internal procedures for incident solving on the basis of their experience. However, each DBA develops his own practice to solve an incident, and one observes almost as many practices as DBAs in order to take into account the current proceduralized context, which is particular and specific. In many working processes human beings can be observed to develop genuine procedures to reach the efficiency that decision makers intended when designing the task. Some parts of this practice are not coded [17]. Such know-how is generally built up case by case and is complemented by non-written rules that allow DBAs to reach the required efficiency. This is a way of getting the result whatever the path followed. The validation of those unwritten rules is linked more to the result than to the procedure to reach it.

In parallel, DBAs prefer to plan again their actions in real time rather than to rely on these procedures based on company's experience, this is due to two main reasons. Firstly, the selected procedure is not always perfectly adapted to the situation at hand and can lead to improper actions or sub-optimal incident resolution strategies. Secondly, if the DBA relies on a procedure, he can miss some important facts and notice them too late to adequately solve the incident. DBAs choose generally to plan again their action continuously according to the situation. Procedures are then used as frames to construct a genuine strategy tailored to the specificity of a given situation. Such practices are based on operational knowledge and are shared by operators.

The modeling of DBAs' reasoning is a difficult task because they use several contextual elements, and also because procedures for solving complex incidents do not always offer a great flexibility and degree of freedom. Their reasoning stems from some chunks of implicit knowledge, which are imposed on the DBA because they correspond to mandatory procedures. Procedures are established from DBA's experience during similar incidents and fixed by the company. As such, practices are what we call hereafter proceduralized contexts. Brézillon in [6] gives a good example about incidents in the Paris subway line where the way the traffic agents react to incidents is similar to the way the DBA reacts to unexpected problems. The example is about the implicit piece of knowledge in that travelers are safer in a station than in a tunnel. At a deeper level, the driver has to avoid stopping the train a long time in a tunnel. The reason is that some travelers may have behavioral troubles such as claustrophobia and could leave the train to wander about on the railway (and thus may generate another type of incident such as "Traveler on the railway"). Similar situations can be found in database administration area. For instance, in the context of project development, some developers have an additional grants and authorization similar to that of the DBA. When a developer considers that one of the processes he has created is slow, he may shut down the database while his colleagues are connected, this will add additional incident, which is the non-availability of the database, waste of actor time, the loss of some data and some social conflicts. These pieces of knowledge, which are not necessarily expressed, result in more or less proceduralized actions that are compiled as a proceduralized context in comprehensive knowledge about actions.

Degani and Wiener [10] distinguish procedures, practices and techniques. Procedures are specified before hand by developers to save time during critical situations. Practices encompass what the users do with procedures. Ideally, procedures and practices should be the same, but the users either conform to procedure or deviate from it, even if the procedure is mandatory. Techniques are defined as personal methods for carrying out specific tasks without violating procedural constraints. Users develop techniques over years of experience [3][4]. Knowledge acquisition focuses on procedures and, eventually, practices, but rarely on techniques. In many different domain areas (i.e. medicine, technical process regulation, nuclear power, etc.…), the distinction between procedure and practice in the one hand, and the notion of context in the other hand is very important. Practices can appear as a contextualized expression of procedures.

Initially, general procedures have been designed to provide operators with a secure reference for incident solving. However, these general procedures forget the contextual dimension of the case at hand. Nowadays, companies are diversifying these procedures by introducing increasingly contextual considerations. This operation increases by specialization the available procedures for each type of incident type. This discussion points out that if it is relatively easy to model procedures, the modeling of the corresponding practices is not an easy task because they are as many practices as contexts of occurrence. For complex incident solving, it is not possible to establish a global procedure, but only a set of sub-procedures for solving parts of the complex incidents. Moreover, procedures cannot catch the high interaction between the solving of the incident itself and the number of related tasks that are generated by the complex incident. As a consequence, there are as many strategies for solving an incident as operators. The next section presents the adopted conceptual framework enabling database administrators to represent their procedures and easily contextualize them (i.e. representing practices).

2. Context modeling techniques

The goal of many of context modeling approaches is to provide a high level abstraction of context information in different domains. The research on the different context models was accompanied by development of context management systems that were able to gather, manage, evaluate and disseminate context information. The following presents the state-of-the-art of some of the main context models that can meet most of the requirements set for the context modeling, management and reasoning.

2.1 McCarthy Formalism

McCarthy [26] defined a context as the generalization of a collection of assumptions. The basic relation is ist(c,p) that asserts that the proposition p is true in the context c, where c is meant to capture all that is not explicit in p that is required to make p a meaningful statement representing what it is intended to state. Formulas ist(c,p) are always asserted within a context, i.e., something like ist(c', ist(c,p)): c': ist (c, p). In other words, context is always relative to another context.

McCarthy formalism is considered as the first step in the integration of context in formal reasoning systems. Reasoning based on the transfer of proposals from one context to another can help to reach a non-monotonic reasoning nearest the human reasoning. However, the transition from one context to another is hard to implement in practice.

2.2 Key-value and markup models

Key-value models rely on a simple key-value pairs to define and specify the list of attributes and their values describing context information used by context-aware applications. Theimer and Welsh [30] proposed using dynamic environment servers in order to manage contextual information for an environment (person, place or community). Contextual information is modeled as key-value pairs called environment variables and is used for mobile application customization.

Attribute-value pairs could not satisfy context modeling requirements particularly as the types of context information used by applications grew more sophisticated.

Markup-based context information models rely on a variety of markup languages including XML. Composite Capabilities/Preference Profile (CC/PP) [24] can be considered a representative both class of key-value models and of markup models. CC/PP as well as other key-value and markup-based context information models have been already discussed and evaluated in many surveys and their limitations have been shown in [25], [23], [33] and [2]. The main critics of these approaches concern their limited capabilities in: (i) capturing different types of contexts, (ii) capturing relationships, dependencies, timeliness, and quality of context information, (iii) allowing consistency checking, and (iv) supporting reasoning on context, on context uncertainty and on higher context abstractions.

2.3 W4 Context Model

The W4 context model support infrastructure developed for context-aware browsing [8]. It allows the representation of context as (Who, What, Where, When) Linda-like tuples and provides an interface to store and query such tuples. This and similar approaches are domain-focused modelling (i.e. important only for particular application domains).

2.4 Object-role based models of context information

Object-role based models of context information aims to create formal models of context to support query processing and reasoning, as well as to provide modeling constructs suitable for use in software engineering tasks. One of these models is Context Modelling Language (CML). CML was described in a preliminary form by Henricksen et al. [18] and refined in later publications [20] and [21]. CML relies on Object-Role Modeling (ORM) [16], which was developed for conceptual modeling of databases. CML provides a graphical notation in order to support the software engineer in analyzing and formally specifying the context requirements of a context-aware application. It extends ORM with modeling constructs for:

- capturing the different classes and sources of context facts;
- capturing imperfect information;
- capturing dependencies between context fact types;
- capturing histories for certain fact types and constraints on those histories;

One interesting advantage of ORM and the CML extensions is that their formality makes it possible to support a straightforward mapping from a CML-based context model to a runtime context management system that can be populated with context facts and queried by context-aware applications. The Rmap procedure for transforming a conceptual schema to a relational schema was described in [15]. In addition, an extension of Rmap that can be used to map a CML-based context model to a relational database was also developed by Henricksen [19]. For example, context model in [2] was designed for use by context-aware communication applications such as the one described in [21].

2.5 Spatial models of context information

Most context definitions, as in [31] and [11], mention the great importance of spatial context models to organize their context information by physical location such as the location of the real world entities described in the context information (e.g., the boundaries of a room), the location of the sensor that measures the context information, or for non-physical context information, an associated location as metaphor (e.g., Stick-E-Notes or virtual information towers). Two kinds of coordinate systems are supported by positioning systems. The first one is Geometric coordinates that represent points or areas in a metric space, such as the WGS84 coordinates of GPS (latitude, longitude, and elevation above sea level). The second coordinate is Symbolic coordinates which are represented by an identifier, such as a room number or the ID of a cell or access point in wireless telephone or local area networks. Nexus project (called Augmented World Model) used a spatial context model [27] and [14]. It is an object-based class hierarchy of context information that supports multi-inheritance, multi-attributes, and both a geometric coordinate system and a simple symbolic location system (based on spatial relationships). Most of the object classes inherit from the class SpatialObject, which makes the Augmented World model inherently spatial. Almost all real and virtual objects are modeled with a location, either by their physical location or by a meaningful association metaphor (i.e. the location of a virtual information tower for web sites). Furthermore, spatial existence was used as an intuitive metaphor for non-physical context information like in the Stick-e Note that makes it possible to easily allow applications to adapt their user interfaces based on the current context. The Stick-e Note architecture is based ConteXtML [29] in order to model contextual information. ConteXtML is an XML-based protocol used as a standard format to exchange contextual information between a server and a mobile user.
The disadvantage of spatial context models is the effort they take to gather the location data of the context information and to keep it up to date. Therefore, if the spatial dimension is of no importance, this effort could be saved.

2.6 Contextual schemas

Contextual schemas are frame-like knowledge structures representing kinds of problem-solving situations and are acquired from domain experts. A contextual schema (c-schema) contains a description about a particular context and a prescription about how the agent should behave in that context. Since the context space is very huge, Turner's model [17] follows the subsequent principle:

"A context should be represented as a c-schema only if: (1) it cannot be represented by merging the knowledge contained in existing c-schemas or (2) if such a merger fails to prescribe the correct behavior for the context."
A c-schema is composed of a set of five parts.
1. Situation description: It allows specifying a name for the context (e.g. "in harbor") and provides information about the actors, objects and state of the world.
2. Standing orders: Correspond to the set of actions to activate and parameters to set when entering or exiting a context. They refer to the piece of contextual knowledge that does not directly infer problem-solving activities in the current context but rather concerns other aspects of agent's behavior. Standing orders are contained in a slot of the same name.
3. Events handling: Contained in the events slot, it provides information about goals to activate in response to events, and knowledge useful for evaluating the event importance in order to decide whether to ignore or to respond to it.
4. Attention focusing information: Contain descriptions of those goals that are predicted (to some extent) to occur in the situation. Attention focusing information is contained in the goals slot.
5. Action-selection information: The actions slot contains information about appropriate actions to perform in order to achieve goals arising in the context.
One of the examples using contextual schemas is Orca project [28]. The main objective of Orca is to create an intelligent, robust, adaptive controller for real-world agents such as autonomous underwater vehicles (AUVs). Orca program represents all of its knowledge as schemas that represent patterns in the world or that are useful for dealing with the world, then brings to bear and merges those schemas at run time in order to carry out its tasks.

2.7 Context-based reasoning

Context-based reasoning is intended to representing tactical behavior of opponents and teammates in simulation-based tactical training systems. This knowledge representation paradigm is an improvement over the rule-based approach which is a common technique used to represent human behavior. It relies on the following hypothesis: tactical knowledge is highly dependent upon the context (i.e. the situation being faced) and context-based reasoning is used to encapsulate the appropriate actions and/or procedures to perform as well as possible new situations, into contexts. The research work carried out by Gonzalez and Ahlers [12] is based on the idea that by associating the possible situations and corresponding actions to specific contexts, the identification of a situation is simplified because only a subset of all possible situations is applicable under the active context. Context-based reasoning approach represents knowledge as a hierarchy of three types (levels) of contexts [12] in order to manage efficiently tactical knowledge, which is voluminous by its nature.
Mission contexts: This type refers to the set of objectives to attain, the things to avoid during the mission being undertaken as well as the constraints of the operation.
Major contexts: This type represents tactical operation undertaken as part of the Mission. The goal of these tactical operations is to assist in achieving the objectives set forth in the Mission. Gonzalez et al. [13] refers to this type of context as Main contexts.
Sub-contexts: This type represents low-level actions that are typically associated with one or more major contexts. This contributes reusability of sub-contexts by different major contexts [13].

2.8 Ontology-based models of context information

Ontologies are mainly descriptions of concepts and their relationships. Ontology-based models of context information uses the representation and reasoning power of OWL-DL [22] formalism or some of its variations for multiple objectives: (a) Using the expressiveness of the language to describe complex context data that cannot be represented by simple languages (i.e. CC/PP) [24]; (b)Possibility to share and/or integrate context among different sources by providing a formal semantics to context data; (c) Using the available reasoning tools both to check for consistency of the set of relationships describing a context scenario and to recognize that a particular set of instances of basic context data and their relationships (e.g., the user's activity can be automatically recognized).
OWL-DL formalism helps to model a particular domain by defining classes, individuals, characteristics of individuals (data type properties), and relations between individuals (object properties). By composing elementary descriptions through specific operators provided by the language, it is possible to build Complex descriptions of classes and properties.
According to [2], since ontologies provide a formal specification of the semantics of context data, they are well suited for knowledge sharing. Different OWL ontologies have been proposed to represent shared descriptions of context data. Two of the proposals are the SOUPA [9] ontology for modeling context in pervasive environments, and the CONON [38] ontology for smart home environments.
Some user-friendly graphical tools exist (e.g., Protégé) to make the design of ontological context models viable to developers that are not particularly familiar with description logics.
OWL-DL ontological context models have been adopted in several architectures for context-awareness. Nevertheless, one of the serious problems with this

approach is that ontological reasoning with OWL-DL also poses many performance issues [38] and [1].

3. Contextual Graphs Formalism

3.1 Brief Description of Contextual graphs

A contextual graph (CxG) Formalism allows the representation of the different ways to solve a problem. It is a directed graph, acyclic with one input and one output and a general structure of spindle [7]. Each path in a CxG corresponds to a practice, a way to fix the problem. Fig. 1 provides the definition of the four elements in a contextual graph. A more detailed presentation of this formalism and its implementation can be found in [7].
A contextual graph is composed of the following elements: actions, contextual elements, activities and temporal branching as shown in Fig. 1.
An **action** is the building block of contextual graphs at the chosen granularity. An action can appear on several paths but it will be in different contexts.
A **contextual element** is a couple of nodes, a contextual node and a recombination node. A contextual node has one input and N branches [1, N] corresponding to the N instantiations of the contextual element already encountered. The recombination node is [N, 1] and shows that, once items on the branch between the contextual and recombination nodes has been processed, it does not matter to know which branch was followed. Contextual elements are used to represent and implement context about the different events occurring in a given situation.
An **activity** is a contextual graph by itself that is identified by participants because it appears on different paths and/or in several contextual graphs. This recurring sub-structure is generally considered as a complex action. An activity is a kind a contextualized task that can be aggregated in a unit or expanded in a sub graph according to the needs [32].
A **temporal branching** expresses the fact (and reduces the complexity of the representation) that several groups of actions must be accomplished but that the order in which action groups must be considered is not important, or even could be done in parallel, but all actions must be accomplished before continuing the practice development. The temporal branching is the expression of a complex contextual element at a lower granularity of the representation.
Contextual graphs represent the set of known practices (strategies) in order to solve a given problem. They also allow incremental acquisition of practices and provide an understandable way to model context-based reasoning. A practice is the path from input to the output of a contextual graph. The problem-solving process is guided throw a specific path by the evolution of context over time.

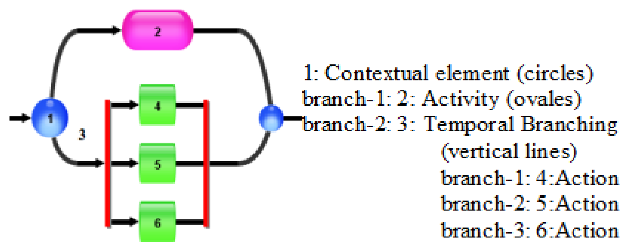

Fig. 1 Elements of a contextual graph.

Adopting a given practice or strategy among the others is dictated by the values of the different contextual elements forming the situation. However, it is not always obvious for a user to select one of these values. For example, in the area of database administration, to solve a serious performance problem within a given critical situation and context, a DBA (Database Administrator) may have different options when asking this question: what causes the slow response time of the system? Is it a network problem? Is it a bad database configuration? Is it a bad query in the application programs? Etc.
User practices are added and stored in an experience database. They may differ from each other because of their contexts that are slightly different where users used different actions at a step of the problem solving. The process of practice acquisition by the CxG system concerns the new action to integrate and the contextual element that discriminates that action with the previous one. The integration of the new practice requires either adding a new branch on an existing contextual node, or introducing of a new contextual node to distinguish the alternatives. The phase of incremental acquisition of practices relies on interaction between the CxG system and the users in order to acquire their expertise, which consists of a context-based strategy and its evolution along the process of the problem solving.

3.2 A case study

The Database Expert (i.e. DBA, Database Manager) can use contextual graphs (Fig. 2.1) to represent the initial patch procedure. He can first represent the first action A1: Install the patch on the new Home Directory, then actions A2, A3 and A4 and so one. These actions are respectively noted 1, 2, 3 and 4 in Figure. Then after completing Action A4 corresponding to “Start the database”, he can use a contextual element 5 noted C5 to test whether or not there is any problem with the patch application (patch succeeded or not). C5 can have branches C5.0 and C5.1 having respectively the two values “Yes” and “No”.

IJCSI International Journal of Computer Science Issues, Volume 15, Issue 6, November 2018
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.5281/zenodo.2544623

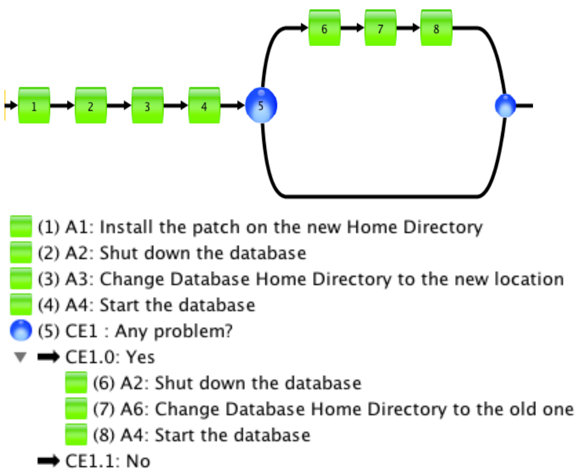

Fig. 2.1: Contextual Graph representation for DBA procedure.

The above procedure should be adapted to take into account the different contexts related to the given situation. For instance the step 1 to create a new Database Home Directory cannot be performed if the DBA doesn't have a permission to create a directory of a specified size. A manager's approval is generally required for applying such important database change management procedure because of its impacts on the running applications accessing patched database and how the users can accept or not any additional delays and costs for performing their usual tasks. Therefore, the DBA can add as many new actions or practices to adapt the initial procedure and the corresponding contextual graph to the situation he faced as shown in Fig. 2.2. Other examples can be found in [34], [35] and [36]

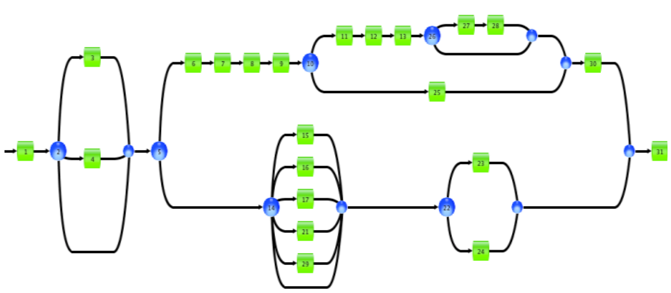

Fig. 2.2: Contextual Graph representation for DBA procedure including new practices.

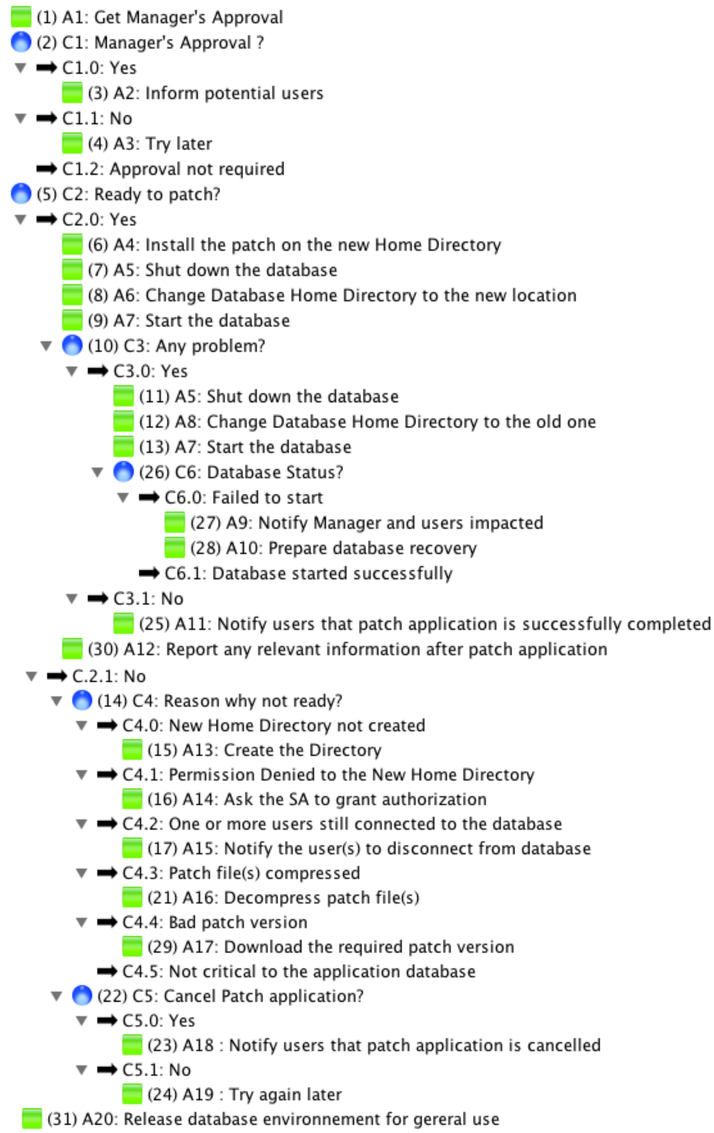

Fig. 2.3: Contextual Graph representation for DBA procedure including new practices (continued).

4. Conclusion

This paper has presented some of the techniques to contextualize the database management and related area. Contextual graphs formalism has been discussed with a case study to illustrate how it is easy to represent different DBA tasks and practices to resolve complex incidents. Our study is in the framework of designing context-based systems for improving database administration and management. It can also be extended to several other computing areas such as monitoring systems, network management, computer security and big data.

References

[1] A. Agostini, C. Bettini, D. Riboni, Hybrid Reasoning in the CARE Middleware for Context-Awareness, International Journal of Web Engineering and Technology 5 (1) pp. 3-23., 2009

[2] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathanf, and D. Riboni. A survey of context modelling and reasoning techniques, doi: 10.1016/j.pmcj.2009.06.002. Elsevier B.V., 2009.

[3] P. Brézillon, Context in problem solving. Technical Report 96/29, LAFORIA, University Paris 6, October, 37 pages (available at ftp://ftp.ibp.fr/ ibp/reports/ laforia.96/laforia.96.29.ps), 1996

[4] P. Brézillon, Context in human-machine problem solving: A survey. Knowledge Engineering Review, 14,1-34, 1999.

[5] P. Brézillon and J.-C. Pomerol, Contextual knowledge and proceduralized context. In Proceedings of the AAAI-99 Workshop on Modeling Context in AI Applications, Orlando, Florida, USA, pages 16–20, 1999.

[6] P. Brézillon, Representation of procedures and practices in Contextual graphs. The Knowledge Engineering Review, 18(2): 147-174, 2003.

[7] P. Brézillon, Task-realization models in Contextual Graphs. Modeling and Using Context (CONTEXT-05), A.Dey, B.Kokinov, D.Leake, R.Turner (Eds.), Springer Verlag, LNAI 3554, pp. 55-68., 2005.

[8] G. Castelli, A. Rossi, M. Mamei, F. Zambonelli, A simple model and infrastructure for context-aware browsing of the world, in: Proceedings of the 5th IEEE Conference on Pervasive Computing and Communications, IEEE Computer Society, 2007.

[9] H. Chen, F. Perich, T. Finin, and A. Joshi, Soupa:Standard ontology for ubiquitous and pervasive applications. In International Conference on Mobile and Ubiquitous Systems: Networking and Services, pages 258–267, 2004.

[10] A. Degani, E. L. Wiener, Procedures in complex systems: The airline cockpit. IEEE Trans. on Systems, Man, and Cybernetics-Part A: Systems and Humans 27(3) pp 302-312, 1997.

[11] A. Dey, Understanding and Using Context. Personal and Ubiquitous Computing, Springer-Verlag London, UK, Volume 5 Issue 1, February 2001, Pages 4-7, 2001.

[12] A. J. Gonzalez, and R. Ahlers, Context-based representation of intelligent behavior in training simulations. Transactions of the Society for Computer Simulation International, pages 153–166., 1999.

[13] A. J. Gonzalez, W. J. Gerber, and J. Castro, Automated acquisition of tactical knowledge through contextualization. Proceedings BRIMS2002, 2002.

[14] M. Grossmann, M. Bauer, N. Honle, U. Kappeler, D. Nicklas, T. Schwarz, Efficiently Managing Context Information for Large-Scale Scenarios,in: Proceedings of Pervasive Computing and Communications, IEEE Computer Society, 2005.

[15] T. A. Halpin, Conceptual Schema and Relational Database Design, 2nd ed., Prentice Hall Australia, Sydney, 1995.

[16] T. Halpin, Information Modeling and Relational Databases. San Francisco: Morgan Kaufmann Publishers, 2001

[17] A. Hatchuel, and B. Weil, L'expert et le système. Paris: Economica, 1992.

[18] K. Henricksen, J. Indulska, and A. Rakotonirainy, Modeling context information in pervasive computing systems, in: 1st International Conference on Pervasive Computing (Pervasive), in: Lecture Notes in Computer Science, vol. 2414, Springer, 2002.

[19] K. Henricksen, A framework for context-aware pervasive computing applications, Ph.D. Thesis, School of Information Technology and Electrical Engineering, The University of Queensland, September 2003.

[20] K. Henricksen, J. Indulska, Modelling and using imperfect context information, in: 1st Workshop on Context Modeling and Reasoning (CoMoRea), PerCom'04 Workshop Proceedings, IEEE Computer Society, 2004.

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

[22] I. Horrocks, P.F. Patel-Schneider, F. Van Harmelen, From SHIQ and RDF to OWL: The making of a web ontology language, Journal of Web Semantics 1 (1), pp. 7-26, 2003.

[23] J. Indulska, R. Robinson, A. Rakotonirainy, K. Henricksen, Experiences in Using CC/PP in Context-Aware Systems, in: M.-S. Chen, P.K. Chrysanthis, M. Sloman, A.B. Zaslavsky (Eds.), Mobile Data Management, in: Lecture Notes in Computer Science, vol. 2574, Springer, 2003.

[24] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M.H. Butler, L. Tran, Composite capability/preference profiles (CC/PP): Structure and vocabularies 1.0, W3C Recommendation, Tech. Rep., W3C, January 2004.

[25] W.Y. Lum, F.C.M. Lau, A context-aware decision engine for content adaptation, IEEE Pervasive Computing 1 (3) pp. 41-49, 2002.

[26] J. McCarthy, « Notes on formalizing context ». Proceedings of the 13th IJCAI, Vol.1, pp 555-560., 1993.

[27] D. Nicklas, and B. Mitschang, The nexus augmented world model: An extensible approach for mobile, spatially aware applications, in: Proceedings of the 7th International Conference on Object-Oriented Information Systems, Springer, 2001.

[28] ORCA, The orca project homepage, 2013.

[29] N. Ryan, Contextml: Exchanging contextual information between a mobile client and the fieldnote server. web site: http://www.cs.kent.ac.uk/projects/mobicomp/fnc/ConteXtML .html, 2004.

[30] B. Schilit, M. Theimer and B. Welch, Customizing mobile application. In Association, U., editor, Proceedings USENIX Symposium on Mobile Location Independent Computing, pages 129–138, 1993.

[31] B. Schilit, N. Adams, and R. Want, Context-aware computing applications. In Society, I. C., editor, Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, pages 85–90, 1994.

[32] J.F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

[33] T. Strang, C. Linnhoff-Popien, A Context Modeling Survey, in: J. Indulska, D.D. Roure (Eds.), Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning and Management, in conjunction with UbiComp, University of Southampton, 2004.

[34] H. Tahir and P. Brezillon, Improvement of database administration by procedure contextualization, CEUR

Proceedings 4th Workshop HCP Human Centered Processes, February 10-11, 2011.
[35] H. Tahir, A Context-based approach for troubleshooting database problems, IJCSI International Journal of Computer Science Issues, Volume 11, Issue 6, No 1, November 2014.
[36] H. Tahir and P. Brezillon, Improving Defect Tracking Using Shared Context, 10th International and Interdisciplinary Conference on Modelling and Using Context (CONTEXT-17), At Paris, France, Volume: LNCS 10257, 2017
[37] R. M. Turner, Context-mediated behavior for intelligent agents. International Journal of Human–Computer Studies special issue on Using Context in Applications, 48(3):307–330, 1998.
[38] X.H. Wang, T. Gu, D.Q. Zhang, H.K. Pung, Ontology based context modeling and reasoning using OWL, in: Proceedings of Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, IEEE Computer Society, 2004.