IJCSI International Journal of Computer Science Issues, Volume 11, Issue 6, No 1, November 2014
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org

A Context-based approach for troubleshooting database problems

Hassane TAHIR[1, 2]
[1] STERIA
Energy & Utilities Department, 11 avenue du Maréchal Juin,
92366 Meudon-la-Forêt Cedex France

[2] Laboratory of Computer Science-Paris 6, University Pierre & Marie Curie (UPMC),
4 Place Jussieu, 75005, Paris, France

Abstract

In the area of database administration, when a hot critical problem suddenly occurs, an organization may lose large amounts of money for each hour of downtime. In such situations, the life of a database administrator (DBA) can become stressful because of the excessive pressure to solve problems quickly. Most of the DBAs often work in a fire-fighting mode and have little opportunity to be proactive. They are continually readjusting standard procedures and developing practices to manage a multitude of specific situations that differ from the generic situation by some few contextual elements. This paper proposes to use "Contextual Graphs" approach to deal with context change in database administration. This research work presents a new version of the contextual graph platform as a basis for designing and implementing a context-based intelligent assistant system for supporting DBAs.

Keywords: *Contextual Graph, Database administration, Procedure, Practice, Intelligent Assistant System.*

1. Introduction

Today, with the fast evolution and advances in information technology and internet applications (i.e. e-commerce and social networking), the decision-making in most organizations has become increasingly complex. As a consequence, decision makers have been obliged to make the best decisions in the shortest possible time. In the area of database administration, support is needed for experts to make decisions regarding complex activities such as tuning problems and managing the continuous changes in databases. For Mullins [17], the DBA (database administrator) is the person responsible for carrying out policies set by data designers and to ensure the ongoing operational functionality and efficiency of an organization's databases and the applications that access those databases. The DBA carries out different tasks such as database design, performance monitoring and tuning, database availability, security, backup and recovery, data integrity, release migration.

Database administrators must be constantly available to deal with the variety of failures and to analyze and correct serious incidents using a large set of standard procedures. They are continually readjusting these procedures to deal with the specific situations that differ from the generic situation by some contextual elements. Contextual elements are relevant at a given time (e.g. memory size, hard drives), and the values taken by these contextual elements at that moment: (memory size: 70%, full, hard drives: HP-1, IBM-23). The DBAs often developed practices to manage these contextual elements in order to solve the problem at hand. Sometimes when one critical problem, suddenly appears, companies may lose large amounts of money for each hour of downtime. In such situations, the life of database administrators may become stressful because of the excessive pressure to solve problems quickly.

Two categories of problems can be pointed: technical and social. Technical problems can impact the performance of the entire information system of the company. This includes problems due to the database, the server, the network and/or the application. For instance, one of the most important database problems is when users are unable to connect to the database because of a locked account, slow time response or bad performance, and sometimes because the database is down. Social problems are mainly due to bad communications and collaborations with other users. One of the mysterious messages that users often see on their terminal "A database is going down". This is frequently due to some DBA procedures programmed to run automatically and to reboot the database (or a database server) in order to perform upgrades, critical patches, or any other task on a database server. How about if the boss is using the application at that moment? Other situations and contexts may be much more critical like medical applications treating a patient (collecting sensitive data from database in real-time). We cannot state all situations and contexts, the list may be long. Another example that we can give concerns some collaboration problems due to the bad collaboration

IJCSI
www.IJCSI.org

between DBA and other actors. In some cases, developers do not cooperate with a DBA to solve database errors due to a bad application coding. The reason for this is that developers may not feel comfortable while their code is being reviewed if their managers are invited.

This work relies on the Contextual-Graphs formalism [5] for implementing the different DBA activities and actions according to the different contextual elements. The main advantage of Contextual Graphs is the possibility to enrich incrementally the system with new knowledge and practice learning capability when needed. Moreover, a contextual graph is a good communication tool for helping the DBAs and actors of the organization to exchange their experiences and viewpoints.

The paper begins by the description of a case study illustrating the process of database recovery and troubleshooting in order to give the reason why a context-based intelligent assistant system is needed to support database administrators in such stressful and similar situations and contexts. After, we present related works in the literature. Then we present the main features of the used approach followed by a presentation of contextual graph platform. Finally we conclude and evaluate our work.

2. A DESCRIPTION OF THE CASE STUDY

Many types of failures can affect the database: Network Failure, Instance Failure, User Error, Media Failure, etc. Some of them can be processed and database can be recovered using tools like RMAN (Recovery Manager) provided by Oracle Corporation. There are two types of failures where RMAN can really support users: Media failure and User errors due to mistakes that lead to damaged databases. Media failure is defined as the loss of a file required for the database to function properly. The files that a DBA might need to recover if media loss occurs are: data files, control files, parameter files, and archive logs files. Being able to put these files back into action quickly is the key to recovery.

For this case study, a part of the manual procedure for database recovery is shown in the Fig. 1. This procedure shows different cases during the recovery process. Briefly, it indicates that the two main reasons why the database is not able to startup are due to missing control files or damaged data files or redo logs files. For each of these cases, actions are suggested to DBA to repair the database. For example, when a DBA starts the Oracle database while the database is still open, he gets an error message similar to this one:

ORA-01157: cannot identify/lock data file 4 - see DBWR trace file ORA-01110: data file 4: '/u01/app/oracle/oradata/dev11g/users01.dbf'

Finding out what is wrong isn't always an easy task. Therefore, a great need for an intelligent tool not to replace the DBA but to help him in such stressful situations to solve expected new critical incidents as is the case of tuning problems. The following section discusses some of the commonly used approaches to intelligent assistance for database management.

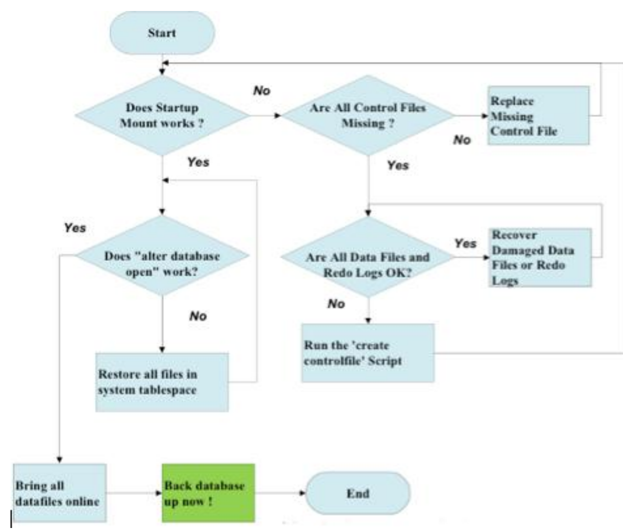

Fig. 1 Example of a database recovery procedure.

3. RELATED WORK

Intelligent assistance is one of the important active research fields within Artificial Intelligence (AI). The machine should assist humans to make decision, to search for information, to control complex objects, and finally to understand the meaning of words. Many solutions have been proposed to implement the notion of intelligent assistance (in different domains) over the years. The following lists some examples in the domain of database management:
- Expert systems: Generalized Expert System for Database Design (GESDD) by Dogac et al. [8];
- Decision Support Systems: An interactive DSS tool to support the database designer by Palvia [18];
- Case-Based Reasoning Systems: CABSYDD (Case-Based System for Database Design) by Choobineh & Lo [7].
- Intelligent Tutoring Systems (ITS): (1) Mitrovic et al. [14] and Mitrovic et al. [15] proposed DB-suite which consists of three web-based intelligent tutoring systems (SQL Tutor, NORMIT for data normalization, KERMIT

for teaching conceptual database modeling using the ER model); and (2) Risco and Reye [21] presented a Personal Access Tutor (PAT), an Intelligent Tutoring System (ITS) for Learning Rapid Application Development (RAD) in a database environment.
- Intelligent Agents and multi-agents systems: (1) Carneiro et al. [6] proposed DBSitter, a tool for monitoring database environment; (2) Moraes et al. [16] proposed a software tool called AutonomousDB that supports the task of schema evolution in heterogeneous multi-database environments where there are replicated schemas. Other similar work can be found in [10]; (3) Elfayoumy and Patel [9] proposed an intelligent agent assistant (IAA) to aid DBAs in performance monitoring tasks and the automation of resolution actions.; and (4) Oracle [20] provides a database "Grid Control Agent" which can help DBAs to monitor and maintain Oracle databases.

The above solutions cannot always successfully handle all the DBA tasks and problems encountered in multitude of specific new situations and contexts that differ from the procedures set for performing the same tasks and problems happened in other situations and contexts. Up to now, the improvement of the existing procedures is achieved through DBA practices that adapt these procedures to the context in which the incidents appear and where tasks should be performed. Another important problem is that IT tools do not provide proper support for the collaborative tasks performed by system administrators as seen in the research in ethnographic studies of system administrators carried out by Barrett et al. [1] and Haber and Bailey [12]. Kandogan et al. [11] and Haber et al. [13] concluded that improved tools for system administrator collaboration have great potential to significantly impact system administration work.

For these reasons, we are interested to take context into consideration and incorporate it in the database administration procedures. The following section presents features of the proposed approach for representing DBA practices as contextualization of procedures and the requirements of context-based an intelligent assistant system.

4. MAIN FEATURES OF THE PROPOSED APPROACH

The notion of context can be used to address dynamic change and requirements in database administration procedures. Brézillon and Pomerol [5] consider that context is "what constrains something without intervening in it explicitly." An important consequence is that we must speak of context in relationships with a focus [5] and thus distinguish three types of context, namely, external knowledge, contextual knowledge, and proceduralized context. The external knowledge is the knowledge that has nothing to do with the current focus. The contextual knowledge is the knowledge that is more or less related to the current focus. The actor proceduralizes a part of the contextual knowledge for addressing the current focus (the proceduralized context). Our study focuses both on technical and user (or human) contexts. The technical context is related to the knowledge about changes in environment, upgrades of the database, upgrades of applications, incidents related to database recovery, etc. The user context is about human knowledge and specific conditions to consider when performing database management tasks.

In the area of incident management for subway lines, Pomerol and Brézillon [19] identified two parts in a context-based reasoning, namely diagnosis and action. The diagnosis part analyzes the situation at hand and its context in order to extract the essential facts for the actions. The actions are undertaken in a predictable order to realize the desired task. Sometimes, actions are undertaken even if the situation is not completely analyzed (or even not analyzed at all). For example, a driver puts a vehicle into gear before any action or situation analysis. Diagnosis and actions constitute a continuous twofold process, not two successive phases in context-based reasoning. Moreover, actions introduce changes in the situation or in knowledge about the situation, and imply a revision of the diagnosis, and thus of the decision making process itself. As a consequence, context must be considered explicitly with knowledge and reasoning. This is the role of the Contextual-Graphs formalism on which intelligent assistant systems (IASs) rely. According to Brézillon [2], an IAS must present different properties like:

- Providing users with a first approximation of environmental trends and events;
- Pointing out useful information implicit in large volumes of data to alert users to sudden changes;
- Developing multiple scenarios and perspectives on a given line of action;
- Attracting user attention to existing and emerging strategic issues;
- Supporting users in sharing and communicating their views and perspectives;
- Guiding user attention to specific data and their interpretation in particular issues.

An Intelligent Assistant system must be designed and developed in a formalism providing a uniform representation of knowledge, reasoning, and contextual elements. The contextual graph formalism can provide the incremental knowledge acquisition and practice learning. Context is the key factor of intelligent assistant systems. Making context explicit allows us to use knowledge in its context of use, to capture variants in the reasoning (e.g. recording practices effectively developed by operators), to generate relevant explanations. The following section presents the adopted conceptual framework.

5. CONTEXTUAL GRAPHS PLATFORM

5.1 Brief Description of Contextual graphs

A contextual graph (CxG) 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 [3]. Each path in a CxG corresponds to a practice, a way to fix the problem. Fig. 2 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 [3].

A contextual graph is composed of the following elements: actions, contextual elements, activities and temporal branching.

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.

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 [22].

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. The following section describes the Contextual Graphs Platform on which the intelligent assistant system for DBAs will be based.

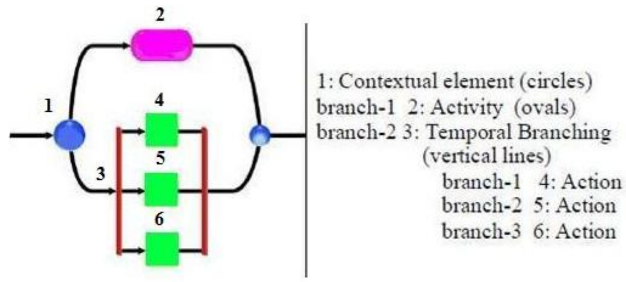

Fig. 2 Elements of a contextual graph.

5.2 Proceduralized and shared contexts

A proceduralized context (PC) is an ordered series of instantiated contextual elements (CEs). It explains how the different items along a practice were introduced. The difference between two practices is explained through the divergence between their proceduralized contexts. Two PCs have at least a different CE or a same CE with different instantiations.

We distinguish the collaborative and the individual proceduralized contexts. The collaborative proceduralized context emerges from the interaction between actors, the introduction of each CE in the PC by one actor is the result of other actors' agreement. This constitutes the shared context associated with the current focus at hand. The individual proceduralized context corresponds to an actor's interpretation of the cooperative PC and contains the collaborative way in which the focus is addressed.

Sharing context means that actors' contexts have a non empty intersection. In a collaborative-design process, the shared context corresponds to the validity context of the design focus. It is built from contextual elements coming from the different experts' contexts. The shared-context building results from an incremental enrichment of contextual elements coming from individual contexts of experts. Thus, a contextual element proposed by an expert will enter the shared context if accepted (validated) by other experts. Individual contexts are mental representations of the design focus and of its validity context (the shared context). A contextual element provided by an expert must be integrated in other experts' mental representation, i.e. each expert must find a translation of this shared contextual element in his mental representation. Thus, the collaborative-design process results by making the different views among experts compatible, not necessarily identical because all mental representations are different.

5.3 Contextual Graph Platform Architecture

The Contextual graphs Platform (or CxG Platfom) contains the building blocks of an experience base on which the context-based intelligent assistant system can

reason and accompany a user in the realization of his tasks. At the implementation level, it is built using Java Software and XML database.
The architecture of the CxG platform is shown in Fig. 3. It is composed of the following components:
CxG Editor: This component enables authorized users to manage their corresponding contextual graphs representing the main procedures and the significant changes added by them (i.e. practices). All operations such as creating, updating or deleting contextual graphs objects are allowed by the CxG Editor.
CxG Reader: This component allows only reading a desired contextual graph to execute one or more practices already created by different experts to performs a given task or activity. He can run only one practice a time. The reading and analysis process is described in the following.
CxG Manager: The CxG Manager controls and communicates with the different components of the CxG Platform and with the user.
Experience database: The CxG Manager uses this component to record and store users' practices. This database stores all information about the contextual graphs objects.
Archive database: This component manages copies of executed contextual graphs.

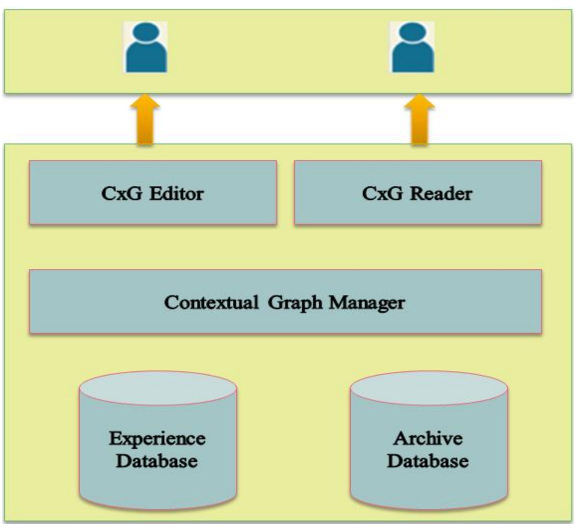

Fig. 3 Architecture of the Contextual Graphs Platform

5.4 Example of Contextual Graphs for DBA procedures

Using Contextual Graphs as shown in Fig. 4 can easily represent the database recovery procedure presented in Fig. 1. It is composed of four contextual elements (CE1, CE2, CE3 and CE4) and six actions A5, A6, A7, A8, A9 and A10 (corresponding to square boxes numbered respectively 5, 6, 7, 8, 9 and 10).

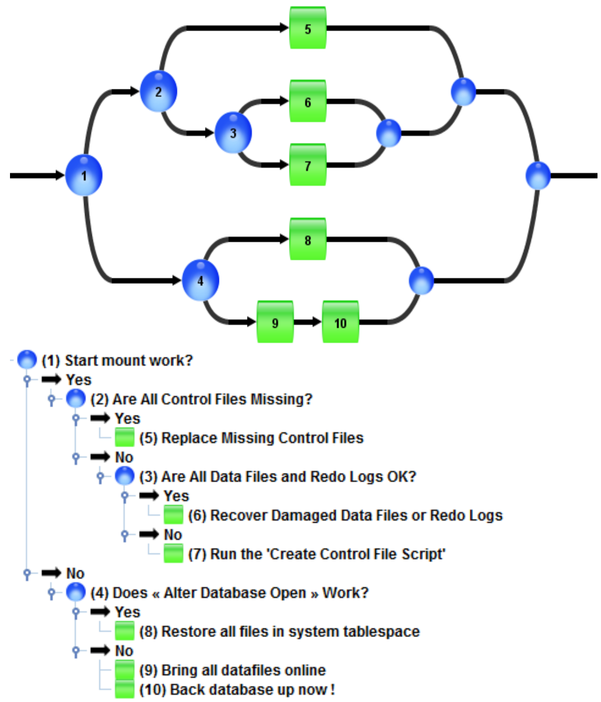

Fig. 4 Contextual Graph representation of the database recovery procedure.

During the recovery process, the DBA may be faced to a new situation and context as shown in Table 1. The initial procedure (Fig. 4) can be improved by adding new practices including new actions, contextual elements and values.

Table 1: Illustrating the change in contextual element instance values

CE Number	*Contextual Elements (CE)*	*CE Values*
CE1	Start Mount Work?	Yes
		No
CE2	Are All Control Files Missing?	Yes
		No
		I don't know
CE3	Are All Data Files and Redo Logs OK?	Yes
		No
CE4	Does « Alter Database Open » Work?	Yes
		No
CE12	Are All Control Files encrypted?	Yes
		No

The above DBA procedure for manual recovery of an Oracle database can be improved by adding new practices including new contextual elements and values as shown in Fig. 5. Other issues can be found in [23] and [24]. In this example, if the DBA doesn't know if all control files are missing or not (Val(CE2)='I don't know'), he may ask the system administrator for help (Action A11). After checking control files (CE2) as well as data and redo log files (CE3), the DBA may also check whether or not these files are encrypted or not (CE12). If the control files are encrypted, he has to enter the required password (Action A13) and then generate (i.e. unzip) the required files (Action A15).

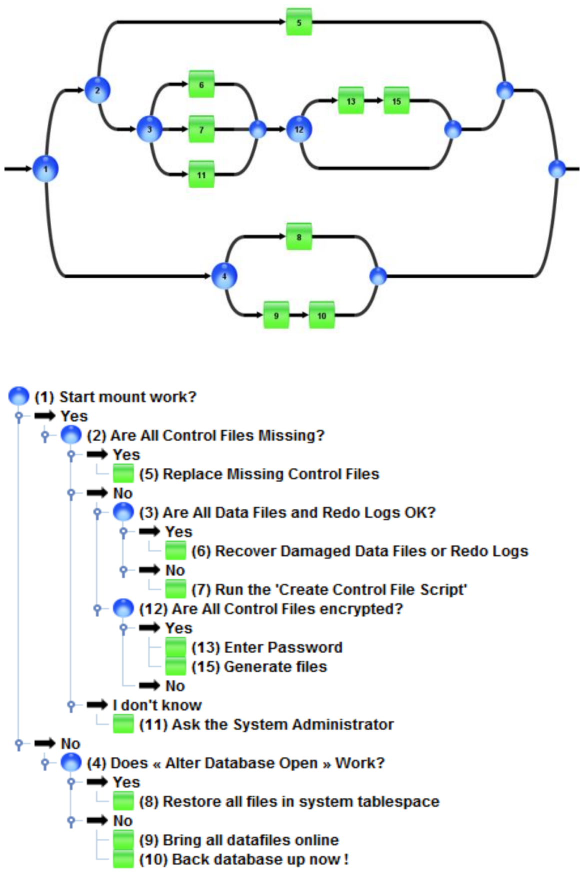

Fig. 5 Contextual Graph representation of the improved database recovery procedure.

6. Conclusion

The paper presents how to contextualize database administration procedures to perform DBA complex tasks (i.e. database troubleshooting). We have illustrated how it is easy to represent different DBA activities by using contextual graphs. The architecture of the contextual graphs platform has also been presented. Our study is in the framework of building an experience base that can be used to design a support system for DBAs. It can also be extended to several other computing areas such as monitoring systems, computer security and network management.

References

[1] R. Barrett, E. Kandogan, et al., Field Studies of Computer System Administrators: Analysis of System Management Tools and Practices. Proceedings of CSCW, 2004.

[2] P. Brézillon, From expert systems to context-based intelligent assistant systems: a testimony. The Knowledge Engineering Review, 2011, 26(1), pp. 19-24.

[3] 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, 2005, pp. 55-68.

[4] P. Brézillon, L. Pasquier, and J.-C. Pomerol, Reasoning with contextual graphs. European Journal of Operational Research, 2002, 136(2): 290-298.

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

[6] A. Carneiro, R. Passos, et al., DBSitter: An Intelligent Tool for Database Administration. Berlin-Heidelberg, Springer-Verlag, 2004.

[7] J. Choobineh, and A. W. Lo, CABSYDD: Case-based system for database design. Journal of Management Information Systems, 2004, 21(3), pp 281-314.

[8] A. Dogac, B. Yürüten, and S. Spaccapietra, A Generalized Expert System for Database Design. IEEE Transactions on Software Engineering, 1989, Volume 15, Issue 4, April, pp. 479-491.

[9] S. Elfayoumy, and J. Patel, Database Performance Monitoring and Tuning Using Intelligent Agent Assistants. In: H.R. Arabnia, L. Deligiannidis, R.R. Hashemi (eds). Proceedings of the 2012 International Conference on Information & Knowledge Engineering, IKE 2012, WORLDCOMP'12. (CSREA Press, 2012). July 16-19, Las Vegas Nevada, USA.

[10] A. J. Fougères, Agent-Based -Tools Integrated into a Co-Design Platform", Int. J. of Computer Science Issues, Vol. 7, Issue 3-8, 2010, pp. 1-10.

[11] E. Kandogan, P. P. Maglio, E. M. Haber, and J. Bailey, On the roles of policies in computer systems management. International Journal of Human-Computer Studies, 2011, 69(6), 351-361.

[12] E. M. Haber, and J. Bailey, Design Guidelines for System Administration Tools Developed through Ethnographic Field Studies. ACM, 2007.

[13] E. M. Haber, E. Kandogan, and P. P. Maglio, Collaboration in system administration. Communications of the ACM, 2011, 54(1), January, 46-53.

[14] A. Mitrovic, P. Suraweera, and B. Martin, Db-suite: Experiences with three intelligent, web-based database tutors. Journal of Interactive Learning Research, 2004, 15, pp 409-432.

[15] A. Mitrovic, and T. I. Team, Constraint based tutors. In B. P. Woolf, E. Aimeur, R. Nkambou &S. Lajoie, eds, 9th International Conference on Intelligent Tutoring Systems. LNCS 5091, Springer-Verlag, Montreal, Canada, 2008, pp. 29-32.

[16] A. C. Moraes, A. C. Salgado, P. A. Tedesco, AutonomousDB: a Tool for Autonomic Propagation of Schema Updates in Heterogeneous Multi-Database Environments. IEEE. Fifth International Conference on Autonomic and Autonomous Systems, April, 2009, 20-25, pp. 251-256.

[17] C. S. Mullins, Database Administration. The Complete Guide to Practices and Procedures. Addison Wesley, 2012.

[18] P. Palvia, An Interactive DSS Tool for Physical Database Design. Information Sciences, 1991, 54(3), pp. 239-262.

[19] J.-C. Pomerol, P. Brézillon, Context proceduralization in decision making. In: Modeling and Using Context (CONTEXT-03), P. Blackburn, C. Ghidini, R.M. Turner and F. Giunchiglia (Eds.). LNAI 2680, Springer Verlag, 491-498, 2003.

[20] Oracle, Enterprise Manager Agent Download. Available at: http://www.oracle.com/technetwork/oem/grid-control/downloads/agentsoft-090381.html, 2013.

[21] S. Risco, and J. Reye, Evaluation of an Intelligent Tutoring System used for teaching RAD in a Database Environment. Proceedings of the Fourteenth Australasian Computing Education Conference, Melbourne, Australia, 2012.

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

[23] H. Tahir, and P. Brézillon, Procedure contextualization for collaborative database administration. Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design, Lausanne, Switzerland, 2011, June 8-10.

[24] H. Tahir, and P. Brézillon, Shared context for improving collaboration in database administration. International Journal of Database Management Systems, 2013, IJDMS 5(2).

Hassane TAHIR is working at STERIA IT Company (France) as an Expert consultant in Databases and Business Intelligence. He is a Certified Professional in Computer Security from Stanford University (2009). He received a PhD in Computer Science from the University of Paris 6 (UPMC), France, in 2013. His research interests include context modeling and management for decision making, database management and administration, computer security and business intelligence techniques particularly data warehousing and big data.