

SHARED CONTEXT FOR IMPROVING COLLABORATION IN DATABASE ADMINISTRATION

Hassane Tahir^{1,2} and Patrick Brézillon¹

¹Laboratory of Computer Science-Paris 6, University Pierre & Marie Curie (UPMC),
4 Place Jussieu, 75005, Paris, France

Hassane.Tahir@lip6.fr, Patrick.Brezillon@lip6.fr

²Steria France, Energy & Utilities Department, 11 avenue du Maréchal Juin 92366
Meudon-la-Forêt Cedex France

hassan.tahir@steria.com

ABSTRACT

Database administrators (DBAs) and experts face a large spectrum of procedures in order to ensure the ongoing operational functionality and efficiency of their organization's databases and the applications that access those databases. Unfortunately, these procedures cannot be used directly in a multitude of specific situations and contexts. To deal with situation specificity and contexts at hand, DBAs often cooperate with different actors such as developers, system administrators, data administrators, end users, etc. However, communication processes are often complex because (1) actors come from different domains different from DBA's domain (e.g. Business user) (2) the context in which a database activity (e.g. incident solving) occurs may not be shared and understood in the same way by all actors. The paper presents how to make shared context explicit in a cooperative work and an analysis of the cooperative work from the viewpoint of one the actors. Making context explicit is possible through a formalism allowing a uniform representation of elements of knowledge, reasoning and contexts, like Contextual-Graphs formalism. Currently, we are developing an experience base that will be used by a system to support DBAs.

KEYWORDS

Actor's viewpoints, Contextual Graphs, database administration, practices, procedures, shared context, support system, collaboration.

1. INTRODUCTION

Database administration is a continuous activity that often involves several actors within the organization. The main actor is the database administrator (DBA) whose responsibility is to ensure the ongoing operational functionality and efficiency of an organization's databases and the applications that access those databases. The DBA performs a variety of tasks including database design, performance monitoring and tuning, database availability, security, backup and recovery, data integrity, release migration. In addition, the DBA must be constantly ready to analyze and correct failures based on a large set of procedures for database administration tasks particularly incident solving [17]. However, the DBA works under temporal and financial pressures and is

continually readjusting these procedures to manage a multitude of specific situations that differ from the standard situation by some contextual elements.

Exceptions are rather the norms. Thus, a DBA often developed practices that deal with these contextual elements in order to solve the problem at hand. For each context and new situation, he may collaborate with other actors to provide him with the requested information to help in efficiently performing his main tasks. Hence, the concept of context plays an important role in cooperation, particularly on what is concerned to help in communication, interaction and knowledge sharing. This knowledge depends on each member's skills, experiences and background, and also on the common activities, conditions, facts, and situations faced during the administration of the database and the setting of the environment. Exchanges of knowledge between the DBA and the other actors can only be effective if there is a common interpretive focus and a shared context where they can understand each other and communicate.

One of the problems in cooperative database administration is that actors interacting with the DBA have different viewpoints about a database incident. Actors may not cooperate efficiently because they think that the DBA problem is not within their scope of work and the consequences of that problem will not affect them. Furthermore, even if a problem met by the DBA is understood by all actors, messages exchanged between them to solve the problem are sometimes not understood and not clear to all of them. This is mainly due to the fact that actors' viewpoint is generally left implicit in their discussion.

Another problem is about optimizing the performance of a database used by an application to provide a benefit to a business. Before entering the details of the application database, the DBA needs to ask business users to present their viewpoints about business problems and requirements and whether or not they accept the performance and the availability of their applications. In addition, the DBA may exchange messages with application developers about this problem. Unfortunately, often developers' viewpoint is that database errors are not due to a bad application coding even if it is the case. Therefore, it is necessary to coordinate all viewpoints and make them compatible to avoid bad database failures. In other words, each actor has to determine the most relevant contextual elements linked to the message and communicate them to other actors with the message. 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 DBA and all the members must share both contextual elements and values taken at the moment of the problem (the instantiations). Our objective is to develop the shared context, i.e. to make explicit contextual elements needed for the cooperation between the DBA (main actor) and other actors.

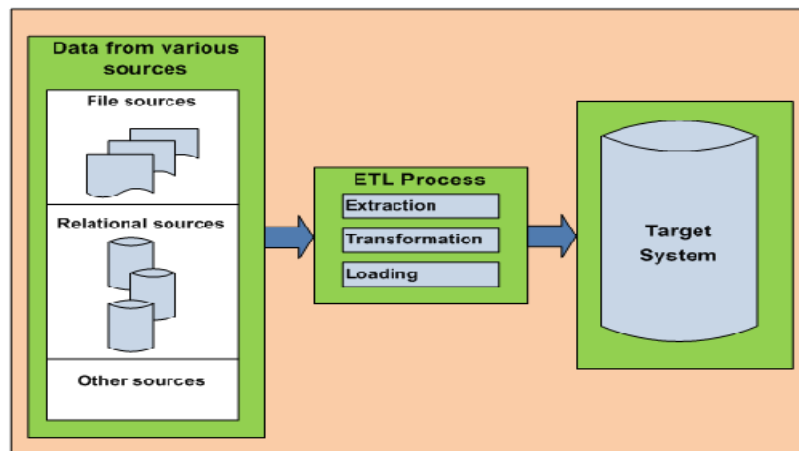
This work relies on the Contextual-Graphs formalism [6] for implementing the different actors' activities and actions according to the different contextual elements. The main advantage of Contextual Graphs is the possibility to enrich incrementally the system with new knowledge and practice learning 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 problems of collaboration in database administration during the process of data migration project. After, we present the usual problem solving in the literature. The next section describes context modeling and management. In this section, we present the conceptual framework, the implementation in Contextual Graphs of actors' cooperation in a problem solving, the notion of proceduralized context (PC) and shared

context. Then the case study will be revisited before the conclusion of the paper to illustrate our view.

2. A CASE STUDY

We consider two concrete situations met by one of the authors in his company. The project is about data migration from several existing source systems to a new target application. The application is a billing system that will be operated by an Energy supplier to satisfy new authority regulations and standards (see Figure 1). The application also should ensure customers to pay the right amount of energy and protect them from large unexpected bills and give the Energy supplier the incentive to get billing right every time.



The author plays two different roles: Developer and Application DBA. In the two situations, the author often interacts and cooperates with different stakeholders: Business users, managers, DBA, System Administrators (SAs) and other people. We now describe each situation.

2.1 Situation 1

This situation occurred in the first phase of the project (about 1 year). The author is a senior Developer whose responsibility is to design, implement the mappings and workflows required in an ETL process for migrating data. For instance, the Developer creates transformations for cleaning data, joining data from tables belonging to different sources and building the required target data according to the business rules. The Developer often communicates with the DBA and system administrator. The following is an example of the dialogs about this situation:

DBA: Hi. Today, we will stop all databases in the Development Environment between 1 pm and 2pm.

Developer: Is there any problem with installed databases?

DBA: Some databases suffer from performance problems and they should be patched ASAP. We have planned to patch all the databases on the environment.

Developer: Well, how long is the patch installation?

DBA: About 20 minutes. But please note 30 minutes of unavailability.

Developer: Thank you for calling. Please let me know the end of this operation.

DBA: OK.

The above situation is an example of collaboration between a DBA and some stakeholders. Unfortunately, this is not enough: few minutes after shutting down all the databases on the development environment, one of the test engineers called the main Developer:

Test Engineer: Hi. I have a mysterious message on the screen of my PC.
Developer: Please give me more details about the message, maybe I can help you.
Test Engineer: The message said, “A database is going down” while I am running an Excel Macro that connects to database to simulate an important test scenario...
Developer: Sorry, nobody knows you were using a Development database for testing your scenario. We alert the DBA to add you to his mailing list.

This second dialog shows that all stakeholders do not share the context about applying patch. The DBA and the Developer did not alert the Test Engineer about stopping databases because they did not know he was using the ETL database. On the other hand, the Test Engineer used the database without informing his colleagues. This has led to wasting time because the test scenario was long.

2.2 Situation 2

This situation occurred at the beginning of the second phase of the project (2 years). The author is an application DBA whose responsibility is (1) to build and maintain database schemas (Development, Test and Qualification), (2) to deploy database scripts and packages, and (3) tune the database. However, the DBA is not familiar with database backups and recoveries. Unfortunately, he faced a case where he had to recover a database with an important size (about 200 GB) on the server used by the migration team. After starting the procedure to restore the database, many error messages were displayed on the screen and generated in the log files. The DBA did not find what to do on the recovery procedure. He then decided to call other experts and the support team to help him achieve the database recovery.

2.3 Limits of database procedures

The above situations have illustrated the limits of database administration procedures. In situation 1, a typical procedure for applying a patch as that in Figure 2 cannot always be efficiently performed when new context occur (context not completely shared between the DBA and other actors). Other examples can be found in [23].

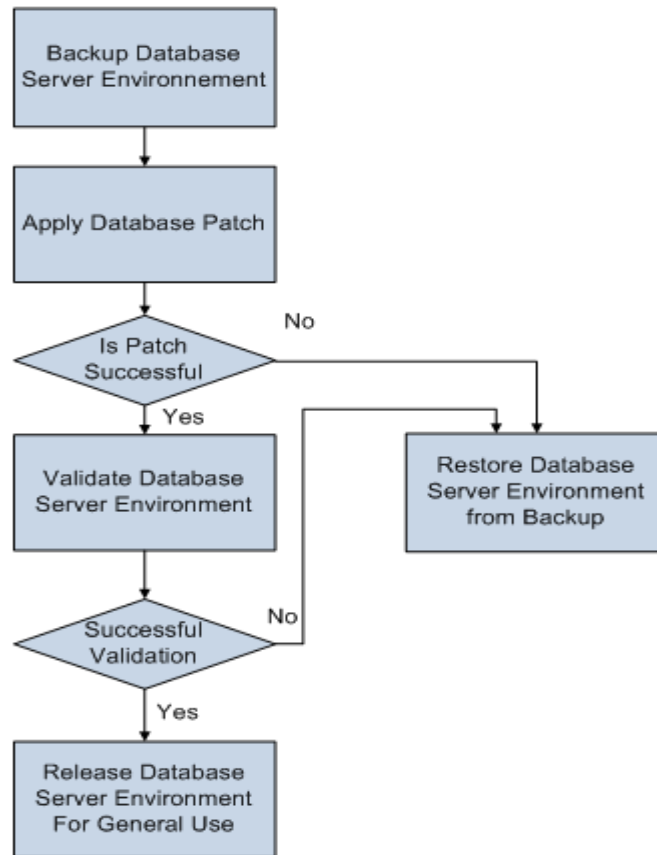


Figure 2. Procedure for applying a database patch.

For instance, the technical or business manager may disapprove the request of the DBA for applying the patch if he thinks that the patch may causes new bugs for applications (application context) or impacts the planning of some high priority activities (planning context). In situation 2, although the procedure says how to recover a database, it does not tell what to do in the context of the new error messages that appear. The DBA may also add new step(s) to the procedure to resolve the problem. In both situations, DBA interacts with actors from different domains (management, application development, system administration, etc.). In other situations, he may first diagnose the problem with other colleagues before choosing to apply or not a patch (during patch assessment). Therefore, in order to collaborate efficiently with a DBA in performing a database administration task, each actor has to understand the shared context about situations in which a problem or activity occurs. We then talk about contextualizing procedures. In the next section, we discuss some of the related research work before presenting the conceptual framework for context modeling.

3. RELATED WORK

The case study is often considered in research works especially in system administration, monitoring and software design processes. Barrett et al. [2] and Haber and Bailey [12] used ethnographic methods to analyze system administrators in context. Their findings show that IT administrators collaborate to manage risk, system complexity and system scale. They conclude that IT tools do not provide proper support for the collaborative tasks performed by IT

professionals. Haber et al. [13] concluded that improved tools for system administrator collaboration have great potential to significantly impact system administration work. On the other hand, Kandogan et al. [11] discussed real-world constraints that IT workers must deal with in their work and how many of them are implicit or require human judgment to evaluate. Another important factor for improving collaboration is the degree of shared context between actors when they collaborate to perform a given task. Werlinger et al. [22] presented an approach to investigate in detail the interactions that security practitioners have with other stakeholders within the context of the security activities. They show that the tools used by security practitioners to perform their security tasks provide insufficient support for the complex collaborative interactions that they need to perform. Ontologies enable also sharing precise and machine-understandable semantics among different applications and parties like in concerning conceptual modeling paradigms and their impact on the quality of the developed ontologies [1]. Monticolo & Gomes [15] proposed to use a Semantic Wiki to allow knowledge evolution and sharing information between actors.

Other research solutions uses Policy-based computing to implement autonomic administration capabilities into a database for enforcing policies to control and decide which changes are allowed and which ones are not. Qiao et al. [19] present a framework to define, manage, and enforce policies that are used to isolate a problem into a more specific context, upon which either general or customized solutions are derived. Another approach is to explore system-level provenance to improve the mental models, and troubleshooting process for system administrators [9]. Multi-agent system is another approach that has been used to build tools for database administrators (i.e. DBSitter as in [8]) and to support the design process involving many actors [14]. 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. The objective is to avoid that the DBA have to perform the same manual task repeatedly on different platforms to ensure the changes. Elfayoumy and Patel [10] proposed an intelligent agent assistant (IAA) to aid DBAs in performance monitoring tasks and the automation of resolution actions. The assistant notifies DBAs when performance problems are detected and resolved. DBAs are expected to provide the agent assistant with definitions for the performance problem conditions and the possible resolution actions. In information retrieval systems, Bouramoul et al. [3] proposed an approach based on the context to evaluate the performance of the search tool and the relevance of results compared to an executed query and the user's judgments.

In our approach, we consider shared context and collaboration from the viewpoint of one actors (i.e. DBA) instead of observing actors collaborating together to accomplish a given task. The present work contributes to contextualize DBA procedures by considering interactions and collaboration between the DBA and other actors sharing a number of contextual elements. To improve the usability of these procedures, contextual graphs can be used. The following section describes context modeling and management.

4. CONTEXT MODELING AND MANAGEMENT

4.1 The conceptual framework

Context is used to describe knowledge shared on physical, social, historical and other circumstances where actions or events happen. All this knowledge does not appear in the actions to execute or the events that occur, but constrains action execution and event interpretation.

Brézillon and Pomerol [7] 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 [7] and thus distinguish three types of context (see Figure 3), 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).

At the implementation view, it is important to distinguish data, information and knowledge that we put under the term of contextual element (CE). Data is the basic, atomic part of the context that can be acquired directly through virtual or physical sensors, such as location coordinates, people identification or weather temperature. Information is the CE that can be derived from data association. While the information is something that once inferred can be easily instantiated and shared between human and software agents, the contextual knowledge is personal and it is inside people's head as mental schemas that help them to interpret external events.

In the area of incident management for subway lines, Pomerol and Brézillon [18] 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.

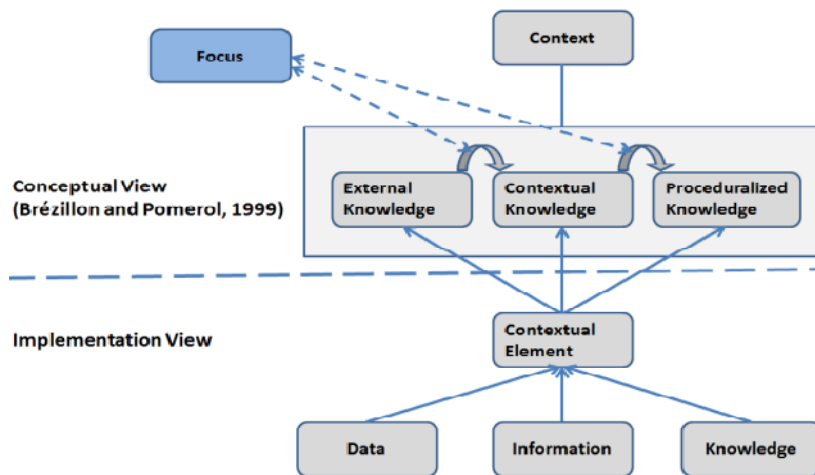


Figure 3. Context, data, information and knowledge

4.2 Implementation as Contextual Graphs

The development of our conceptual framework leads to the implementation in Contextual Graphs of our problem of actors' cooperation in a problem solving because Contextual Graphs are a user-centred formalism [5].

A contextual graph represents the different ways to solve a problem, and each path corresponds to a practice, a way to fix the problem. It is a directed graph, acyclic with one input and one output and a general structure of spindle [4]. Figure 4 provides the definition of the elements in a contextual graph. A more complete presentation of this formalism and its implementation can be found in [4]. Elements of a contextual graph are: actions, contextual elements, activities and temporal branching.

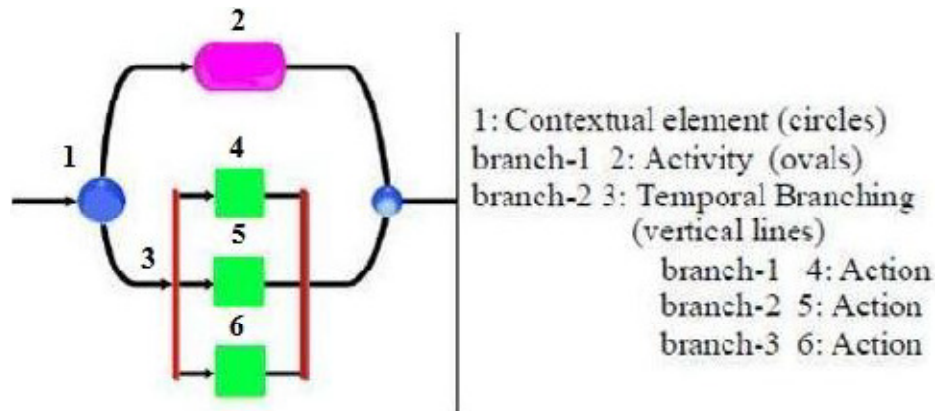


Figure 4. Elements of a contextual graph

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 context.

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 the part of the practice on the branch between the contextual and recombination nodes has been executed, 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 [20].

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.

4.3 Contextual Graph Platform

Within the data migration project, the DBA uses a Contextual Graph (CxG) platform (Figure 5) to help him to represent and to execute his daily and new activities and new tasks. The main components of this platform are: (1) The CxG Manager is used to control and communicate with the Editor and the Reader of contextual graphs. (2) CxG Editor enables users to create and edit their graphs. (3) The CxG Reader allows reading graphs. (4) The experience base is used by the CxG Manager to record user practices. (5) An archive database is also used to save copies of executed contextual graphs. The CxG platform is built using Java Software and XML database.

After this brief presentation of the contextual graph platform, it is now time to present examples of contextual graphs to illustrate the notion of proceduralized and shared context.

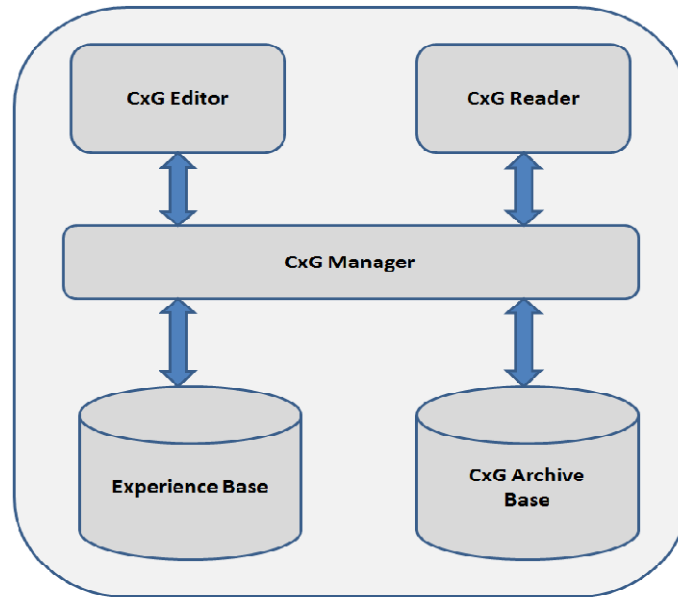


Figure 5. Contextual graph Platform.

4.4 Notion of Proceduralized context

A proceduralized context (PC) explains how the different items along a practice were introduced. A PC is an ordered series of instantiated contextual elements (CEs). 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, each CE in the PC is the result of an 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.

4.4.1. Example 1

Figure 6 gives a contextual-graph representation of the procedure for applying a patch to a database presented in the case study. The database administrator can still perform the steps of patch application during the allowed database change time windows when the users are not connected. Generally, night is the preferred period of time for generating backups, and also the ideal moment for applying database patch updates. However, it is necessary to immediately apply protection patches when the database is exposed to security problems.

This procedure should be adapted to take into account the different contexts related to the given situation. Details of some of the activities (i.e. 3) have been already discussed in our previous paper [21].

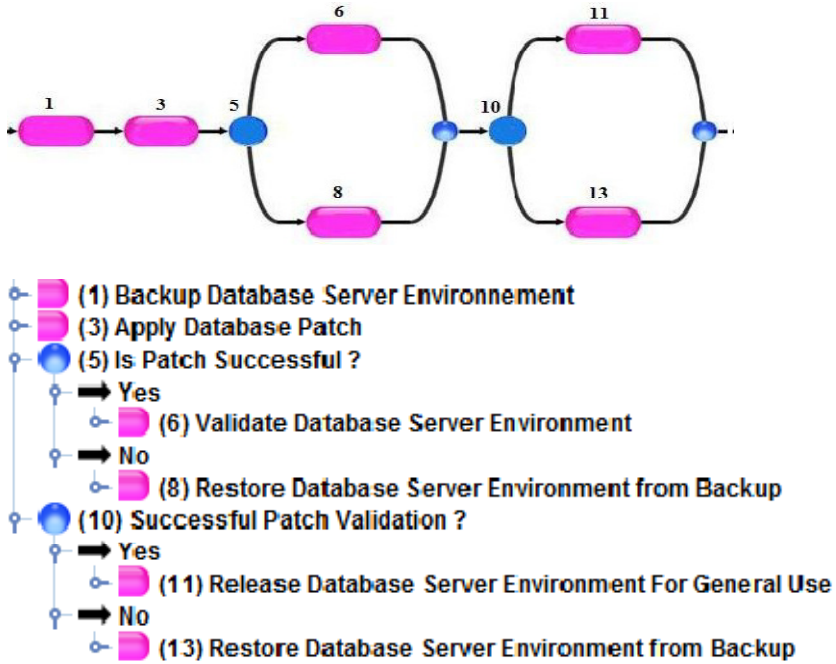
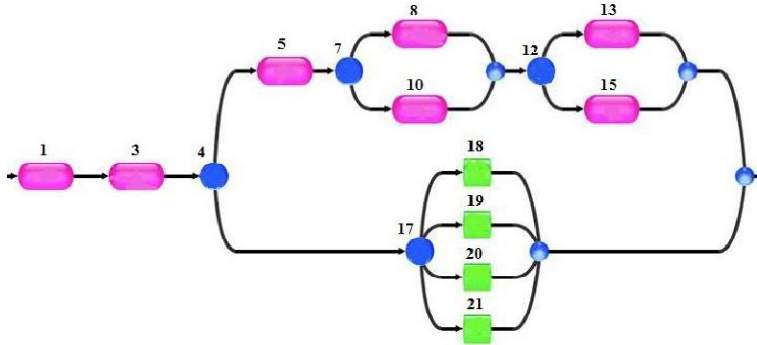


Figure 6. Contextual graph for a database patch procedure.

4.4.2. Example 2

By examining the original patch procedure in example 1 (Figure 6), we observe that it is not possible to proceed in patching the database if activity 1 is not accomplished (bad backup) without any failure, such as mysterious error messages not indicated in backup and recovery of the existing procedures. Figure 7 illustrate the new steps added for contextualizing the procedure.

After performing a backup of the database server environment (Activity 1), the DBA must validate the backup (Activity 3). One way to do this is to check if it is possible to restore the database server environment (CE-4). If the database is successfully restored, the DBA can follow the remaining patch steps (activity 5 and the remaining other activities). Otherwise, the DBA may correct failures if they are about database data or control files (action 20). He may also need to collaborate with other experts for support (actions 18, 19).



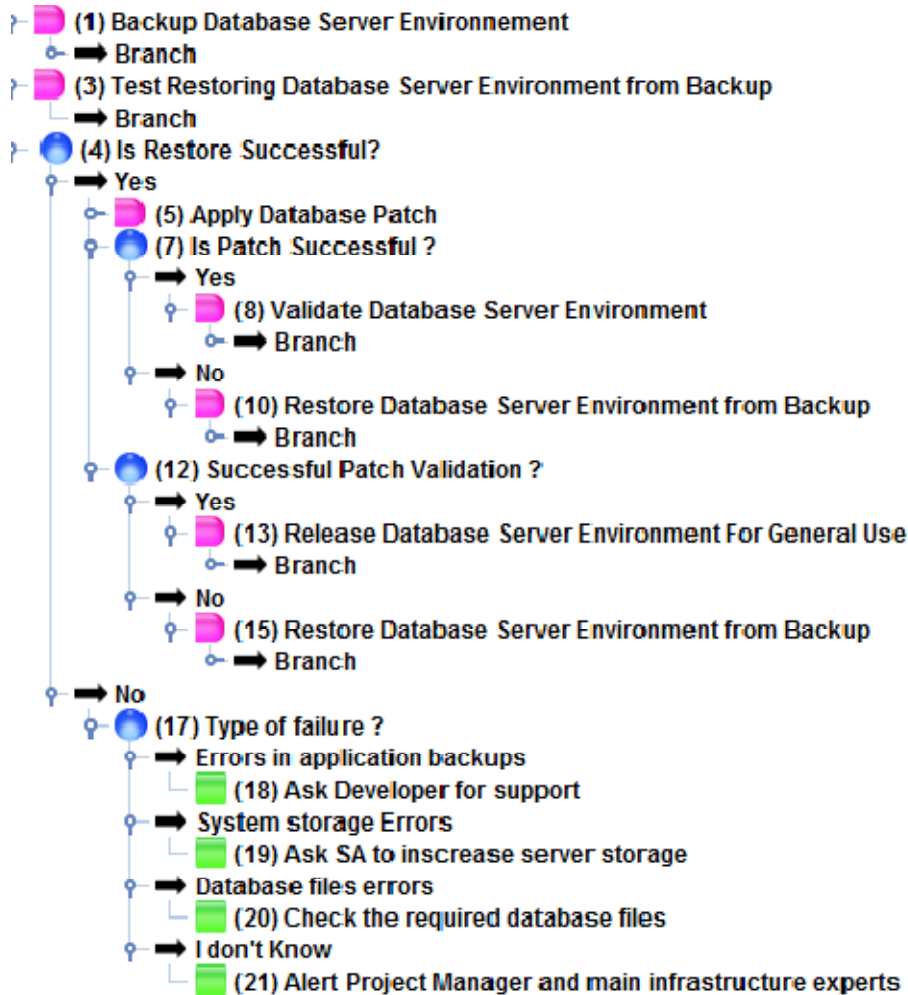


Figure 7. Contextual graph illustrating new added practices in the database patch procedure

4.5 Notion of Shared context

The PC construction from CEs is often a process of communication in a work group. The proceduralized context contains all the pieces of knowledge that have been discussed and accepted (at least made compatible) by all the actors. The shared context is build by interaction between the DBA and project team members. It contains proceduralized knowledge in the focus of attention of the DBA and another actor. These knowledge pieces are extracted from the contextual knowledge of each actor and jointly structured by the actors, and result in a shared knowledge. The more the shared context will be developed, the more efficient will be the decision making of the DBA.

4.5.1. Example 3

Most of performance problems do not need a patch in order to be solved. Rather, a DBA must develop rigorous practices with key team project members. This third example shows how contextual graphs represent the cooperation between two actors to fix an application performance problem. Figure 8 shows contextual graph representing the DBA viewpoint. The DBA interacts with a Developer to help him to process his request about optimizing system performance. In

particular, the DBA must give a feedback to the Developer about whether or not the optimization is performed successfully or not and if a solution is found to the bad response time.

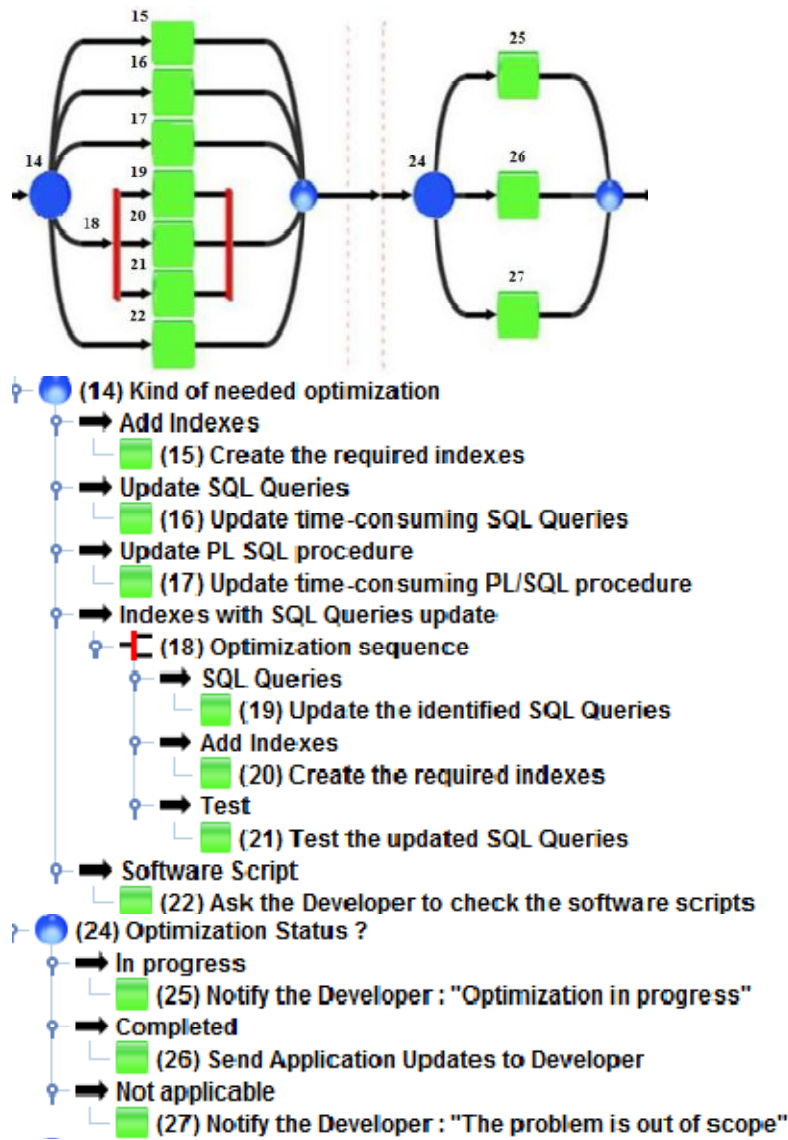


Figure 8. Contextual Graph of DBA-Developer interaction according to DBA's viewpoint.

Figure 9 shows the Developer's viewpoint. After requesting the DBA support, he must also cooperate efficiently to help him to understand parts of the ETL application code causing the performance problem. When the DBA finishes the optimization process, the Developer must take into account all the DBA recommendations about the required updates for SQL queries and PL/SQL procedures in the ETL transformations he created. Contextual graphs in Figures 8 and 9 show that actors share a set of CEs to make understandable their viewpoints by the other.

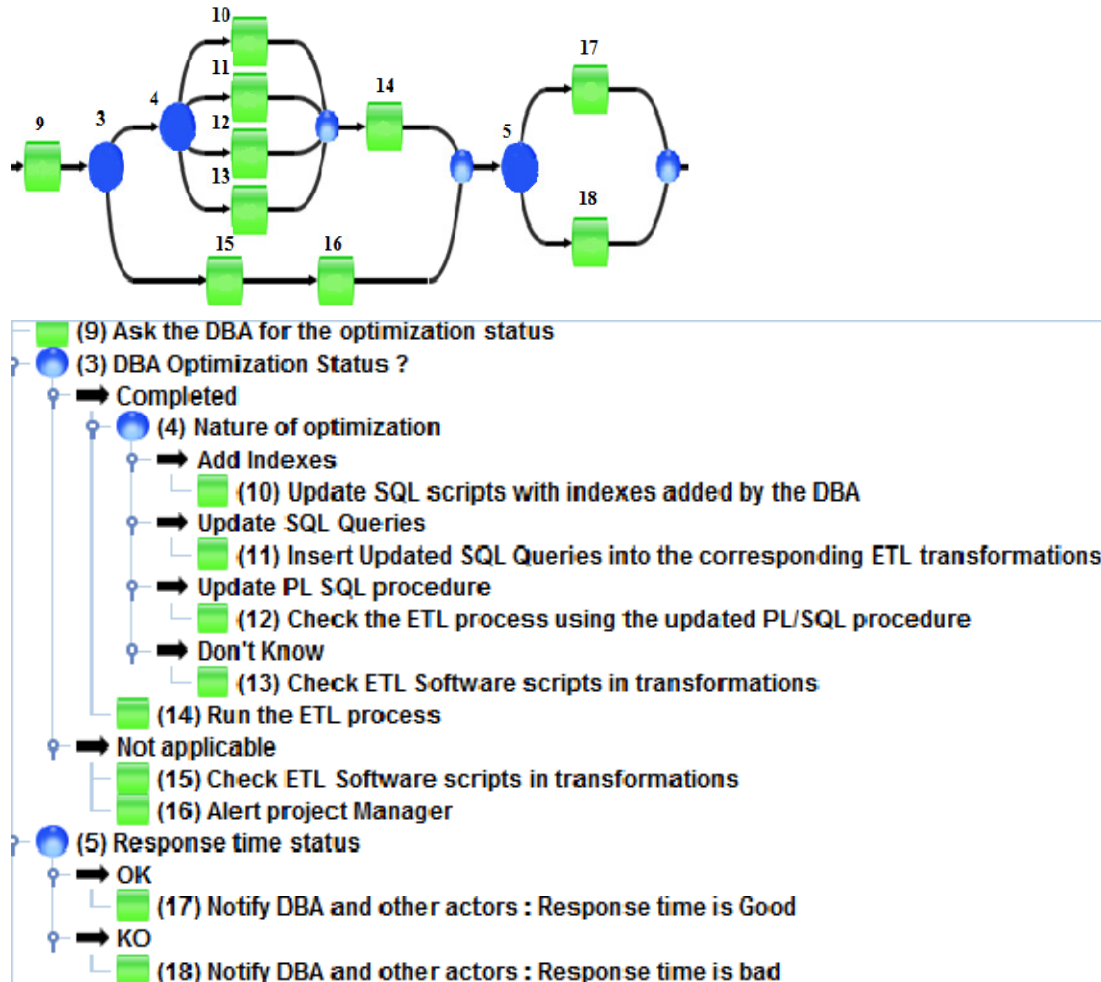


Figure 9. Contextual Graph of DBA-Developer interaction according to Developer's viewpoint.

Figure 10 represents how the shared context is built during the interaction between the DBA and the Developer. It contains proceduralized pieces of knowledge used in the focus of attention of the two actors. Each shared CE may have different values from one actor to the other. For example, the response time status is "KO" for the Developer whereas it is "Good" for the DBA. When the DBA understands that the response time is KO from the developer viewpoint he will try other techniques to process the performance problem.

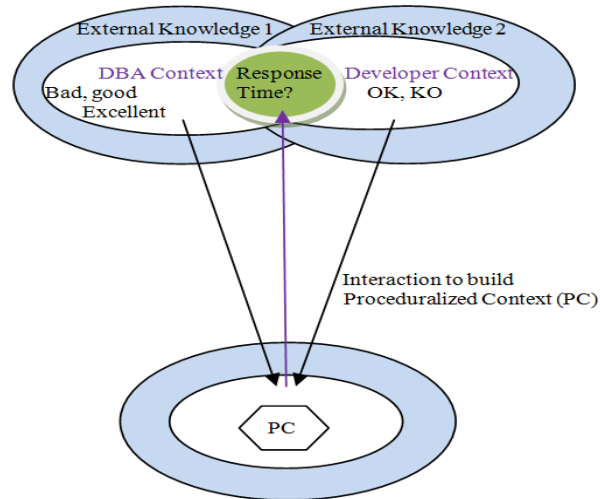


Figure 10. Building proceduralized and shared context

The DBA may also ask the developer for other contextual elements (i.e. environment and configuration parameters) that may be determinant for finding the failure causes. On the other hand, the Developer needs to know the nature of optimization. For instance, the DBA must communicate the SQL queries he has optimized in order for the Developer to take them into account in the different parts of the ETL application code especially in future releases. Therefore the values of the CEs of the shared context are very important and may be determinant for actor (i.e. DBA) to take the best decision for solving a critical problem.

5. BENEFITS OF USING CONTEXTUAL GRAPHS FOR DBA

In the above examples, we have illustrated that contextual graphs are easy to use for representing both the initial patch procedure and the new collaboration practices that have been developed by the DBA during his interaction with other actors (i.e. Developer, system administrator, manager). These practices take into account the various shared contextual elements about the situations faced by the DBA. This has been illustrated when solving a performance problem.

The main interest of contextual graphs relies on the possibility to introduce incrementally new practices based on cooperation between actors. Now we are working to complete the architecture of the CxG Platform by designing an experience base that will be used by the CxG Manager to develop a Context-Based Intelligent System (CBIAS) for the DBA. For the moment, the CxG manager is used by the DBA to:

- 1) Represent the steps performed to manage the different changes in database schemas for the different platforms (Development, Test and Qualification). This has helped the DBA in different situations and avoiding some of the failures. Thanks to CxG tool for making explicit actor's context.
- 2) Represent the steps for resolving some database issues like the performance problem in this paper.
- 3) Use Graph Reader to assist the DBA in the execution of actions of a given activity.

6. CONCLUSIONS

The paper presents how to contextualize database administration procedures based on sharing context and viewpoints between a DBA and actors collaborating with him to perform complex tasks. We have illustrated how it is easy to represent different actors' viewpoints by using contextual graphs. Our work is based on the notion of shared context that has been applied in many applications particularly in collaborative work in software design. We have shown how making shared context explicit can help to improve procedures and to avoid conflicts between actors. 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] M. Al-Debei, M. M. Al Asswad, S. de Cesare and M. Lycett. Conceptual Modelling and The Quality of Ontologies: Endurantism Vs. Perdurantism. *International Journal of Database Management Systems (IJDMS)* Vol.4, No.3, June 2012
- [2] R. Barrett, E. Kandogan et al., Field Studies of Computer System Administrators: Analysis of System Management Tools and Practices. *Proc. CSCW 2004*.
- [3] A. Bouramoul, M.-K. Kholadi and B.-L. Doan, Using Context to Improve the Evaluation of Information Retrieval Systems. *International Journal of Database Management Systems (IJDMS)*, Vol.3, No.2, May 2011.
- [4] 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
- [5] P. Brézillon, Representation of procedures and practices in Contextual graphs. *The Knowledge Engineering Review*, 18(2): 147-174, 2003.
- [6] P. Brézillon, L. Pasquier and J.-C. Pomerol, Reasoning with contextual graphs. *European Journal of Operational Research*, 136(2): 290-298, 2002.
- [7] 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.
- [8] A. Carneiro, R. Passos et al., *DBSitter: An Intelligent Tool for Database Administration*, Berlin-Heidelberg, Springer-Verlag, 2004.
- [9] M. Chiarini: Provenance for System Troubleshooting. http://static.usenix.org/event/tapp11/tech/final_files/Chiarini.pdf, Workshop on the Theory and Practice of Provenance (TaPP), Heraklion, Greece, June, 2011.
- [10] S. Elfayoumy and J. Patel. Database Performance Monitoring and Tuning Using Intelligent Agent Assistants. *Proceedings of the 2012 International Conference on Information and Knowledge Engineering, IKE 2012*, in Hamid R. Arabnia, Leonidas Deligiannidis, Ray R. Hashemi Editors, *WORLDCOMP'12*, July 16-19, Las Vegas Nevada, USA, CSREA Press, 2012.
- [11] E. Kandogan, P. P. Maglio, E. Haber, J. Bailey, On the roles of policies in computer systems management, *International Journal of Human-Computer Studies*, Volume 69 Issue 6, pages 351-361, June, 2011
- [12] E M. Haber and J. Bailey, Design Guidelines for System Administration Tools Developed through Ethnographic Field Studies. *ACM*, 2007.
- [13] E. M. Haber, Eser Kandogan, and Paul P. Maglio, Collaboration in system administration, *Communications of the ACM*, Volume 54 Issue 1, Pages 46-53, January 2011.
- [14] D. Monticolov, V. Hilaire, S. Gomes and A. Koukam, "A Multi-agent System for Building Project Memories to Facilitate Design Process," *Integrated Computer-Aided Engineering*, 15(1), pp. 3-20, 2008.
- [15] D. Monticolov, S. Gomes, Collaborative Knowledge Evaluation with a Semantic Wiki: WikiDesign, in the *International Journal of e-Collaboration*, Volume 7, Issue 3, pages 31-42, July 2011.

- [16] A. C. Moraes, A. C. Salgado and 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, 20-25, pp. 251-256. 2009
- [17] C. S. Mullins, Database Administration: The Complete Guide to Practices and Procedures, Addison Wesley, 2012.
- [18] J.-Ch. 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, pp. 491-498, 2003.
- [19] L. Qiao, B. Soetarman et al., A framework for enforcing application policies in database systems, ACM, 2007.
- [20] J.F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, 2000.
- [21] H. Tahir, P. Brézillon, Procedure contextualization for collaborative database administration. Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design, Lausanne, Switzerland, June 8-10, 2011.
- [22] R. Werlinger, K. Hawkey, D. Botta and K. Beznosov, Security Practitioners in Context: Their Activities and Interactions with Other Stakeholders within Organizations. Int. Journal of Human-Computer Studies. 67(7): 584-606, 2009.
- [23] . Wessler, Oracle DBA on Unix and Linux, SAMS,2002.

AUTHORS

Prof. Patrick Brezillon defended his Thèse d'Etat (6-year duration) in 1983 on "Mathematical Modeling of Self-Oscillating Nonlinear Systems. Application in Biology to Serotonin and Calcium Metabolisms" at University Paris 6. He belongs now to the Laboratoire d'Informatique de Paris 6 since 15 years.



The research of Prof. P. Brezillon focuses since 1992 on the notion of context, and he now is one of the main leader in the community interested by context, organizing the series of international and interdisciplinary conference on context (CONTEXT), giving invited talks, etc. Prof. Patrick Brezillon proposes a coherent conceptual framework for context management with a context-based formalism—called Contextual Graphs—for representing in uniform way elements of knowledge, reasoning and contexts (see the software at <http://cxg.fr>). This formalism is used in several real-world applications (subway, component assembling, medicine and army) concerned by the human reasoning in context. He is now applying context in management science. Since 2005, Prof. Patrick Brezillon studies the links between context management and different forms of applied management.

Prof. Patrick Brezillon also is the Head of the Knowledge and Innovation Management (KIM) Master at the University Pierre and Marie Curie and responsible of the Context-based Management of Knowledge and Competences Specialization and co-responsible of a new Specialization called Management of Situation of Crisis.

Prof. P. Brézillon published about 420 papers, alone or with a grand total of 130 co-authors in international conferences and international journals (e.g., IEEE Expert, AI Magazine, The Knowledge Engineering Review, the International Journal on Human-Computer Studies, etc.).