Improvement of database administration by procedure contextualization

H. TAHIR and P. BREZILLON
Laboratory of Computer Science-Paris 6 (LIP6)
University Pierre and Marie Curie (UPMC), Paris, France
{Hassane.Tahir, Patrick.Brezillon}@lip6.fr

Abstract

Database Administrators (DBAs) relieve on a large set of procedures for incident solving in database. However, in the one hand, they have to work under temporal and financial pressures, and, in the other hand, DBAs are continually readjusting these procedures to manage a multitude of specific situations that differ from the generic situation by some few contextual elements. The exceptions are rather the norms. Thus DBAs developed practices that deal with these contextual elements in order to solve the problem at hand.

Capturing and managing practices is far more difficult than procedures. If a procedure-based support system is easy to implement (procedures are well established), a practice-based support system is difficult to design because there are almost as many practices as contextual variants. However, if the context has an infinite dimension, the number of practices is finite. The key elements are the incremental acquisition of knowledge and the learning of new practices, thanks to a software called Contextual Graphs.

The goal of our work is to support DBAs by collecting all the practices developed by the DBAs and proposing them to DBAs to benefit of the experience of the other DBAs and to provide a support system acting as a real context-based intelligent assistant system.

Introduction

The Internet era has enabled organizations to accelerate their changes in order to be ready to the new challenges particularly those related to the management and the development of their information systems. Most of organizations use a database management system (DBMS) to manage their data, and the presence of a database administrator (DBA) today is mandatory. A DBA is the information engineer responsible for ensuring the ongoing operational functionality and efficiency of an organization's databases and the applications that access those databases (Mullins, 2002). This requires him to perform a variety of tasks in different areas including database design, performance monitoring and tuning, database availability, security, backup and recovery, data integrity, release migration. All tasks involve the company's databases.

Database administration is a continuous activity, and DBA spend long days with lots of overtime, especially when there is complex performance problems to resolve. In addition DBAs frequently has to work on weekends and holidays to perform database maintenance and reorganizations during off peak hours. He must be constantly available to analyze and correct failures. Most DBAs carry cell phones and other contact tools so that they can be joined at any time to solve problems encountered to get the applications back up and running, and to avoid database downtime that can completely shut down business processes. Of course, the DBA master all database management fundamentals, but he must be expert in the specific DBMS used in the organization: Oracle, SQL Server, Sybase, DB2, etc… In addition, he needs to acquire and develop exceptional communication skills because he frequently interfaces with many different types of users: technicians, programmers, end users, customers, and managers.

A good DBA participate in all phases of the application development life cycle. For example, in the initiation and requirements gathering phase, he identifies data components of the project to check if the required data already exists elsewhere in the organization or if the data is new. He must also help to determine the final status of the data used by the application which is not longer useful and he is responsible for managing the overall database environment that includes installing the DBMS and setting up the IT infrastructure to allow applications to access databases. Two modes of actions can be distinguished: reactive and proactive. The reactive DBA attempts to resolve problems only after problems occur and he is focused on resolving the main problem confronting him. In the other hand, the proactive DBA implements and develop practices and procedures to avoid problems before they occur.

As far as procedures are concerned, those issued by the database administration manuals cannot all explain because they cannot take into account all possible contexts. Therefore, they are often applied with caution to solve a given problem. In some cases, the DBA prefers to plan his actions in real time (and thus transform the procedure into a practice) to take into account the specific context of the current situation and the problem that he faces.

For example, for a reported anomaly with an error message in log file (or trace file), a procedure may recommend changes in the parameters list, then stop and subsequent restart of the database to make effective the new setting. This recommendation may not always be applied by the DBA because another method may be preferable to apply, method that will not disrupt end users and to affect the data availability during the resolution of the detected failure. Thus, the DBA undertakes a contextualization of the procedure that will become a practice. However, if it is practically automatic and unconscious for DBAs, capturing and managing practices is far more difficult than procedures.

The above motivation shows that an intelligent assistant system for DBAs must, on the one hand, use explicitly contextual information, and, on the other hand, works on an experience base that capitalizes past experience. This supposes that the experience base is developed in a uniform representation of knowledge, reasoning and context. Our study is in the realm of research on context modeling and management particularly that initiated in Brézillon and Pomerol (1999). This paper describes the improvements obtained by contextualizing users' procedures in applications like database administration.

The paper starts by presenting database administration problems followed by a discussion of the context approach and the related work. Then it presents the different kinds of Knowledge in Database Administration. After this, it discusses DBA procedures and practices and how they are modeled using a contextual graph based on example of a simple procedure for applying a database patch. Finally, the paper lists the remaining future work and concludes what has been reached.

Database Administration problems

DBA activities include many tasks. Some of them must be performed on a regular basis, others in response to emergencies or specific user needs. Every day, the DBA follows the main established administration procedures within the company like in the following list:

- Verify that the database is up and accessible to the end users ;
- Verify that backup was successful ;
- Check performance statistics and if there are enough resources for acceptable performance ;
- Check the status of previous night's processes ;
- Look for any new alert log entries to check whether or not new database errors have appeared since the last time.

In addition, he must be constantly available and ready to deal with problems and prevent them before they occur.

One or many 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. Many of the database problems fall into the following categories:

1) Unable to connect to the database: Users cannot connect to the database because of locked account or bad configuration file, and sometimes because the database is down. The system administrator and DBA must interact and work together to set shared server and database configuration. The DBA often writes scripts to allow one or more critical databases to restart automatically if for any reason the server reboots. These scripts can include commands to start databases and connection processes (i.e. oracle listener process). If the DBA forgets to include a command to start one connection process, users will not be able to connect to the database even if the database is started.

2) Slow time response and bad performance: Users cannot get the results of their queries in an acceptable time frame.

3) Privileges to access the database: This category of problems can occur in two cases. The first one is when DBA do not grant sufficient rights to the users. The second one is when DBA grants excessive privileges to some users or applications causing serious security vulnerability.

4) Change in database structure: The application can be impacted if any an unanticipated change occurred at the schema level of a database such as accidentally invalidating and dropping objects (table, column, etc …).

5) Bad database deployment: The DBAs may accidentally propagate the changes made to the database in the testing environment to the production system. One of the other reasons of this category of problems is due to the bugs in the DBA's deployment scripts required to move from one environment to another.

6) Database maintenance: Some of problems occur when the required maintenance tasks are not performed by the database administrator. An example of this is when database parameters like memory, size and path of data files need to be modified. Other DBA maintenance problems are due to inattentive space management or erroneous backup scripts resulting in an incorrect data recovery or bad patch updates as it will be discussed later on in this paper.

7) Other problems: This category is about the least frequent problems. Examples of problems include DBMS bugs, data loss and hardware failure, etc ….

Details, descriptions and real examples about database administration problems can be found in (Mullins, 2002), (Oliveira & Nagaraja et al., 2006) and (Wessler, 2002).

Sometimes when of the above problems suddenly appear, companies may lose large amounts of money for each hour of downtime. In such situation, A DBA life can become stressful because of the excessive pressure to solve problems quickly. In this framework, our work will contribute to find ways to minimize those DBA stressful situations. One way to do this is by designing and implementing a context-based intelligent assistant system (CBIAS) to help the DBA by providing him with a comprehensive list of information to resolve critical problems he faced. This system will be developed using a methodology that should take into consideration contextual information about DBA procedures and practices as well as the different situations where he performs his tasks to ensure the availability and the safety of the database.

Context approach

Brézillon and Pomerol (1999) show that context plays an important role in a number of domains since a long time. This is especially true for activities as predicting context changes, explaining unanticipated events and helping to handle them, and helping to focus attention. Context is used to describe knowledge shared on physical, social, historical and other circumstances where actions or events happen. All this knowledge is not part of the actions to execute or the events that occur, but will constrain the execution of an action and event interpretation.

Context has an infinite dimension and it is not clearly defined. To deal with a large number of contextual information, Brézillon and Pomerol (1999) distinguish between three types of context (Figure 1) for a given focus of attention, namely, external knowledge, contextual knowledge, and proceduralized context. The external knowledge is the knowledge that has nothing to do with the current focus. Conversely, the contextual knowledge is the knowledge that is more or less relevant for the current focus of attention. Always at a given focus, part of the contextual knowledge is proceduralized. The proceduralized context is a part of the contextual knowledge, which is invoked, organized, structured and situated according to this focus.

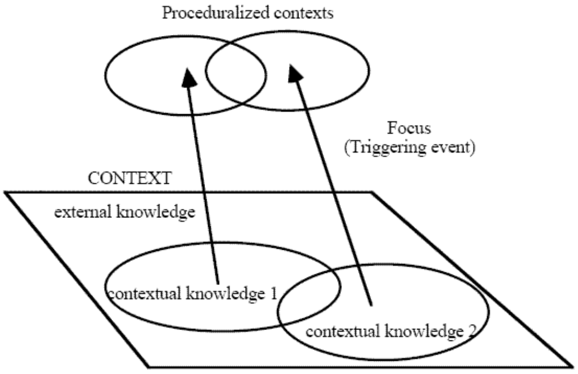


Figure 1: Different types of context.

Context can be modeled using an approach based on contextual graphs (Brézillon, Pasquier and Pomerol, 2002; Brézillon, 2002) where the contextual elements are acquired incrementally when needed. Later on in this paper we show to apply this approach to represent contextual information in database administration procedures.

Related works

Nowadays there are many automated database administration tools that helps the DBA in his decision-making. Traditional expert systems like Oracle Expert offer much functionality such as assisting the DBA by automating routine database maintenance and tuning tasks (ORACLE, 2001). The main principle of these systems is to suggest solutions to correct detected failures based on information already stored in a Knowledge Base. There exists also solutions with a learning mechanism to suggest the most adequate solution to each detected problem like in DBSitter which is based on the combination of two methods: (1) Case Based Reasoning (CBR) to adapt known solutions and enrich the Knowledge Base (2) Mutli-Agent where a set of intelligent agents is used to monitor the Database environment and actuate on it (Carneiro & Passos et al., 2004).

Other recent research works 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 & Soetarman et al., 2007) 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. Jabbour and Menasee (2008) introduce the notion of embedding policies into the database itself and enabling these policies to block every attempt to compromise the state of the database, or to alter its configuration in a way that contradicts what has been established and fed into the policy by the system owner. The Common observations concerning all these mentioned works and some similar works not discussed in this paper are the full considerations to technical aspects like using sensors to collect the contextual information about the administration environment like in DBSitter. Most of them seldom consider the contextual information about the preferences of DBA (experience matters, social and cultural, etc. ...) and the dynamics aspects of contexts. Nevertheless, some other research works have performed ethnographic field studies to develop guidelines for tools to better support how administrators actually work as in (Barrett & Kandogan et al., 2004) and (Haber & Bailey, 2007). The present work contributes to contextualize DBA procedures based on user-centered approach in database administration. To improve the usability of these procedures, the contextual graphs can be used. The following section will help understanding the different kinds of knowledge when dealing with database administration procedures.

A Knowledge Categorization in Database Administration

Effective database administration requires mix of three types of knowledge (Wessler, 2002): technical knowledge, business knowledge, and human knowledge as shown in Figure 2. These three main knowledge categories in database administration are themselves divided into sub-categories.

Technical knowledge

Technical knowledge corresponds to the technical skills the DBA needs to possess. It can be broken down into three main categories: database knowledge, application knowledge, and system knowledge.

Database knowledge

Database knowledge is about the RDBMS the DBA is responsible for. He should know everything about the database he monitors on a daily basis and provides all the maintenance for it.

Application Knowledge

Application Knowledge concerns program code affecting the database. The DBA needs to understand what it does and how it connects to the database. He also needs to support the packages and procedures stored inside the database. However tuning the code is a shared responsibility between the DBA and the application developers.

System Knowledge

System Knowledge corresponds to the different aspects of systems administration and networking that affect the database. It is interesting that the DBA should be able to serve as a backup System Administrator if needed.

Business knowledge

Business knowledge corresponds to the business skills the DBA needs to possess. It consists of Organization's Processes and industry trends. So the DBA have to consider and understand the business process as whole to add value to the process and to the organization. On the other hand, he should follow the industry the organization is in and to keep up with the IT industry. This will help him to be in a good position to come up with new ideas.

Human knowledge

Human knowledge corresponds to the people skills the DBA needs to master. It can be broken down into four main categories: Communication, Management, and Problem solving and Education.

Communication

Communication corresponds to the interaction between the DBA and other people, both inside and outside the organization. This will help in avoiding misunderstandings and accidents such as destroying each other's work or imposed system downtime.

Management

Knowledge of the entire system (not just the database) and how it supports the business makes the DBA a very knowledgeable person and often qualifies him to take on project management responsibilities. The people he manages may include not only DBAs but also System Administrators and programmers in support of a project.

Problem Solving

Much of what a DBA does is about gathering information and making judgments to solve both technical and non-technical problems. Often the real source of an error is hidden and will be found only by an experienced DBA who understands well how the database, operating system, and application interact with each other. Some other problems require skills in negotiation and compromise.

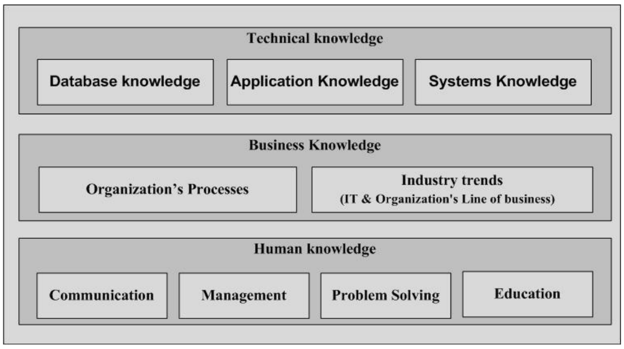


Figure 2: Kinds of Knowledge in Database Administration.

Education

Education concerns the DBA professional certification and training both inside and outside his organization.

DBA procedures and practices

Database administration procedures cannot always be successfully applied by database administrators (DBAs) to perform their tasks and solve the different problems they face. This is why assisted tools based on these procedures cannot always be efficiently used because they cannot take into account all possible contexts. In such situations, DBAs develop their own practices like to correct a failure by considering the context in which that failure occurs.

Degani and Wiener (1997) distinguish procedures, practices and techniques. Procedures are specified beforehand 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. About the finite dimension of practices, the number of practices is of a reasonable size. Later in the paper, we will discuss a software tool called Contextual Graphs which is used to support incremental acquisition of new practices.

Techniques are defined as personal methods for carrying out specific tasks without violating procedural constraints.

The following is an example of a procedure of applying a patch to a database:

1. Shut down the database
2. Apply patch to the Database Home
3. Start the database
4. In case of patch application not succeeded perform steps 5, 6, 7:
5. Shut down the database
6. Roll back the patch
7. Start the database

Of course if the patch is ready, a DBA can perform the above steps during the allowed database change time windows when the users are not always connected to the database. Generally, night is the preferred period of time for generating backups, and also the ideal moment for applying database patch updates. However the DBA may not apply the procedure during the day when the database is running. He should take into account the different contexts related to the given situation. For instance, theses contexts concern the type of database environment (Development, Test, Pre-Production, Production) and whether end users and customers are connected or not. For each context, he may request information or authorization from other colleagues or his manager to perform the required patch actions. Hence, the step 1 to shut down the database cannot be performed if users are connected especially if any user is connected to the database on the production environment. In this case the database administrator may request an authorization from a manager before applying the patch. If the authorization to shut down the database is given, he can start to execute the actions 1 to 3. So, the DBA has adapted the above procedure to the contextual knowledge he has acquired over time by developing practices to reach the goal fixed by this procedure.

As explained by Brézillon (2003), the construction of the proceduralized context from contextual knowledge is often a process of communication in a community of practice, even if members of this community come from different domains. Figure 3 represents how the different proceduralized contexts built from contextual knowledge during the interactions between the DBA and each category of users.

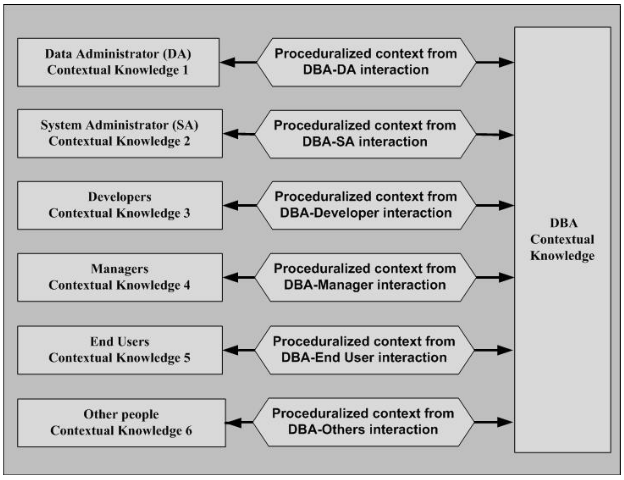

Data Administrator (DA) Contextual Knowledge 1
Proceduralized context from DBA-DA interaction
System Administrator (SA) Contextual Knowledge 2
Proceduralized context from DBA-SA interaction
Developers Contextual Knowledge 3
Proceduralized context from DBA-Developer interaction
Managers Contextual Knowledge 4
Proceduralized context from DBA-Manager interaction
End Users Contextual Knowledge 5
Proceduralized context from DBA-End User interaction
Other people Contextual Knowledge 6
Proceduralized context from DBA-Others interaction
DBA Contextual Knowledge

Figure 3: Context in DBA-User's interactions

The different interaction contexts contain elements of the contextual knowledge for the building of the proceduralized context in the focus of attention of the DBA and the corresponding user. These elements of knowledge in the interaction context are extracted from the contextual knowledge of each category of user. Then they are proceduralized by the two persons, and result in a shared chunk of knowledge. The resulting proceduralized context contains all the pieces of knowledge that have been discussed, accepted and assembled by the DBA and each category of user. For example, in the DBA-System Administrator (SA) Interaction, the SA should provide relevant information to the DBA when he requests information about the server such as the disk space and the needed permissions.

As shown in the Figure 3, in most IT organizations, a DBA will interact with five types of users: data administrators, system administrators, developers, managers, customers and end users, other people outside the organization.

DBA-DA Interaction

The role of data administrator (DA) is managing data from a theoretical standpoint to build conceptual data models. The DBA-DA Interaction is about how logical models are actually implemented as tables. The DA functions mainly in larger companies. However, when there is no DA role in an organization (as often in small companies), the DBA must assume the role of Data modeler.

DBA-SA Interaction

The main role of the system administrator (SA) is to keep all the system boxes running efficiently and in a secure manner. If no system administration group exists, the DBA assumes responsibility for DBMS-related system administration and programming. The DBA-SA Interaction is needed where the SA should provide relevant information to the DBA and resources to support the Database especially in the following cases:

- At the moment of the creation of a new database or server ;
- When a DBA requests information about the server's backup strategy, the disk layout, RAID level, machine memory;
- When a DBA needs to know if there are any major non-database applications planned ;
- When DBA needs more disk space ;

The interaction between the SA and the DBA is not always a very close and cordial relationship. There is often contention between the two administrators on some issues such as the moment to reboot the machine, use of system resources (disk space and memory), backup and recovery procedures, and user policy.

DBA-Developer Interaction

The interaction between the DBA and Developers is needed mainly in the following cases:

- When Developers writes and maintain programs, the DBA should implement any required database or environment changes;
- Supporting developers in tuning their database queries used by their programs ;
- Database errors are the result of bad program code, the DBA asks the developers if they have changed anything to help him in troubleshooting.

DBA-Manager Interaction

The DBA-Manager interaction concerns the following:

- The DBA must report to his manager who may be technical or may be from a non-technical background ;
- The DBA explains to the manager the different parts of a system (at a high level) and explaining how they are dependent on each other. He can also explain what it takes to provide high system uptime, both in terms of hardware and in terms of trained IT professionals.
- The DBA may have to provide an explanation to his manager if problems still occur even if they might not even be related to the database.

DBA-End User Interaction

The DBA-End User Interaction concerns the communication between the DBA and a population of users with a different background and needs inside the organization, outside the organization, or even outside the company. Often it is the end user or a customer who is the first to detect a failure in the system, even before the IT staff does. The administrator needs to listen to what is being said and provide support and answers that are understandable.

DBA-Other people Interaction

The DBA-Other people interaction concerns the different communications and contacts between the DBA and RDBMS vendors (Oracle, Microsoft, IBM … etc), and other outside companies. The main objectives of these interactions are the following:
1) To choose products and negotiate support contracts;
2) To ensure that any new purchased software, related to both database and non-database uses, does not adversely affect data protection or availability of the existing databases. It is up to the DBA to do the needed testing to guarantee that no unexpected problems or bugs occur because of a new software addition;
3) To Sign up for the highly recommended training classes, pursuing technical support issues.

The following section presents contextual-graphs and how they can be used in any situation in which database administrators developed practices from the existing procedures adopted by the organization.

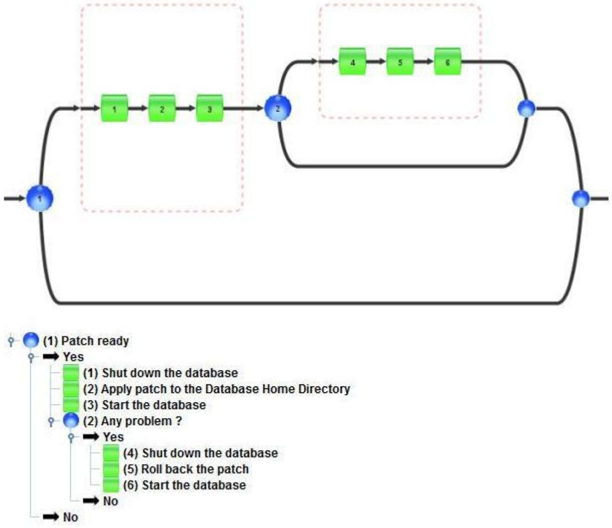
Contextual graphs for database administration

Contextual graphs have been initially designed for an application for incident solving on a subway line (Brezillon & Pomerol, 2000). A contextual graph is an acyclic directed graph with a one input, one output, and a serial-parallel organization of nodes connected by oriented arcs. It is an acyclic graph because user's tasks are generally performed in ordered sequences. For example, the activity “Make sure users are disconnected from the database” is always considered at the beginning of an incident solving, never at the end of the incident solving. There are different types of nodes in a contextual graph: actions, contextual and recombination nodes, sub-graphs and parallel grouping. A sub-graph allows the modeling of DBA activities, and thus contextual graphs give a representation of the reasoning directly understandable by database administrators. A path is an ordered sequence of elements (contextual and recombination nodes and actions) of the contextual graph from the input source to the output. Each path represents, by its sequence of actions, a practice.

In a contextual graph (CxG), the dimension of context is finite, limited by the number of practices learnt by the system. The initial structure of a Contextual graph (its skeleton) is defined by the procedure that is established by the DBA company for the problem solving. Different practices can be developped for a given incident depending on the context in which the incident must be solved. CxGs support incremental acquisition of new practices.

Example of a DBA procedure for applying a database patch:

As an example, we use contextual graphs to model the context-based DBA procedure for applying a database patch. The previous procedure can be represented using the contextual graph in Figure 4.

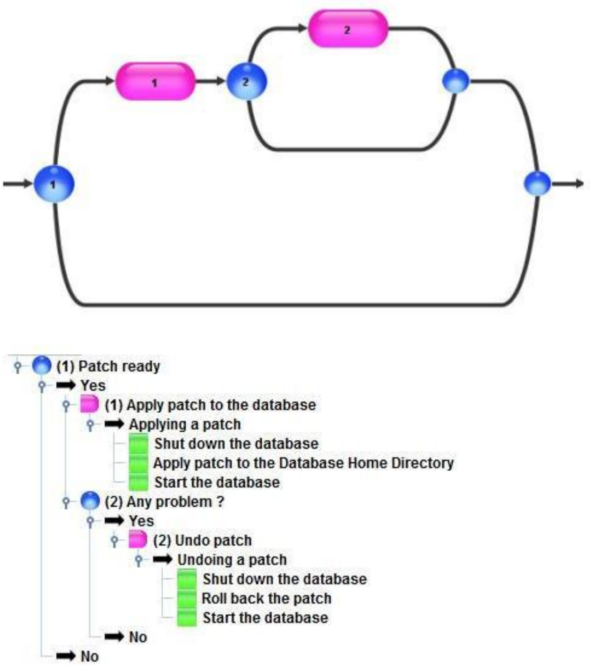


- (1) Patch ready
  - Yes
    - (1) Shut down the database
    - (2) Apply patch to the Database Home Directory
    - (3) Start the database
    - (2) Any problem ?
      - Yes
        - (4) Shut down the database
        - (5) Roll back the patch
        - (6) Start the database
      - No
  - No

Figure 4: Contextual graph of a database Administration procedure

The graph contains two contextual elements 1 and 2 that can be noted as CE1 and CE2. CE1 is about if the patch is ready or not. In other words CE1can have two values (Yes, No). When the patch is ready (Value (CE1) = Yes), the sequence of actions 1, 2, 3 is executed by the DBA. These actions can be noted A1, A2 and A3. On the other hand CE2 allows checking if any problem occurs after the end of action A3. In the case of problem (or patch fails), the sequence of actions 4, 5 and 6 (noted A4, A5 and A6) is executed. Notice that A1 and A4 are instances of the action “Shut down the database” and that A3 and A6 are instances of the action “Start the database”. The two dotted frames in Figure 5 correspond to the two main DBA activities in the procedure for applying the database patch. Actions A1, A2 and A3 form the “Apply database patch” activity whereas actions A4, A5 and A6 form the “Undo patch” activity. The Figure 5 shows another representation of the DBA procedures using contextual elements and activities.

The procedure represented in Figure 5 can be applied only in the ideal conditions where the non availability of database for a period of time will not affect users and the business of the company which is not always the case. In practice, the DBA adapt administration procedure to the different contexts without losing the main purpose of the procedure. In our example, he should consider contexts such as whether or not users connected to database, and what should be done in the case that the patch is not available. So, in this case for example he should take into account other contexts like if he uses or not web support to download the patch and if the web site is down or not, etc.



- (1) Patch ready
  - Yes
    - (1) Apply patch to the database
      - Applying a patch
        - Shut down the database
        - Apply patch to the Database Home Directory
        - Start the database
    - (2) Any problem ?
      - Yes
        - (2) Undo patch
          - Undoing a patch
            - Shut down the database
            - Roll back the patch
            - Start the database
      - No
  - No

Figure 5: Using DBA activities in contextual graphs

The Figure 6(a) details the different practices developed by the DBA to consider the different contextual elements. The context of the contextual graph presented in Figure 6(a) is given by the elements {CE1, CE2, CE3, CE4, CE5, CE6, CE7, CE8, CE9}. Figure 6(b) and Figure 6(c) explain numbers on Figure 6(a). The contextual graph shows that DBA has developed many practices around the original procedure for applying the database which is not sufficient in the multi-user interactions. We showed that it is important to consider not only the technical contexts affecting the database like the type of environment and the availability of website to download the patch, but also the contexts shared in the different interactions between the DBA and other group of users (Developers, managers, system administrators, etc).

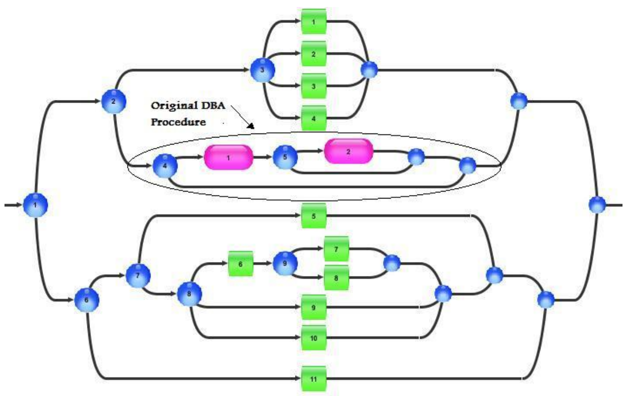
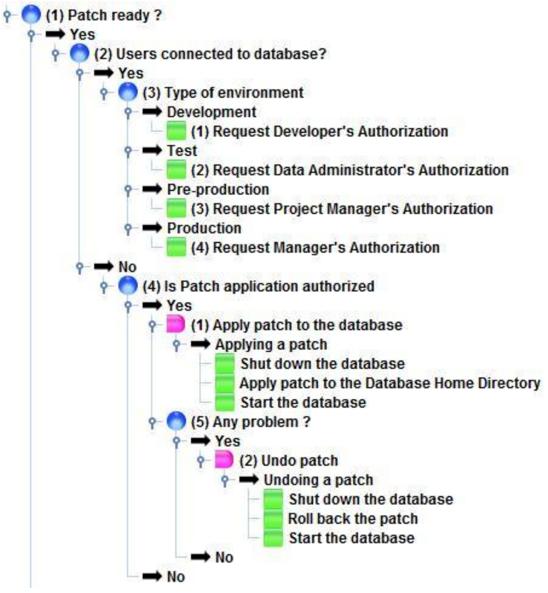
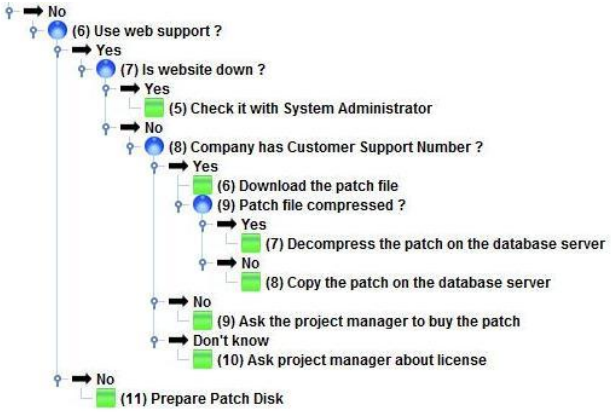


Figure 6(a): Adding DBA practices in contextual graphs



- (1) Patch ready ?
  - Yes
    - (2) Users connected to database?
      - Yes
        - (3) Type of environment
          - Development
            - (1) Request Developer's Authorization
          - Test
            - (2) Request Data Administrator's Authorization
          - Pre-production
            - (3) Request Project Manager's Authorization
          - Production
            - (4) Request Manager's Authorization
      - No
        - (4) Is Patch application authorized
          - Yes
            - (1) Apply patch to the database
              - Applying a patch
                - Shut down the database
                - Apply patch to the Database Home Directory
                - Start the database
            - (5) Any problem ?
              - Yes
                - (2) Undo patch
                  - Undoing a patch
                    - Shut down the database
                    - Roll back the patch
                    - Start the database
              - No
          - No

Figure 6(b): Description of practices



- No
  - (6) Use web support ?
    - Yes
      - (7) Is website down ?
        - Yes
          - (5) Check it with System Administrator
        - No
          - (8) Company has Customer Support Number ?
            - Yes
              - (6) Download the patch file
              - (9) Patch file compressed ?
                - Yes
                  - (7) Decompress the patch on the database server
                - No
                  - (8) Copy the patch on the database server
            - No
              - (9) Ask the project manager to buy the patch
            - Don't know
              - (10) Ask project manager about license
    - No
      - (11) Prepare Patch Disk

Figure 6(c): Description of practices (continued)

The context of an action (e.g. A7) is composed of two parts: the contextual elements used on the path from the input and the other elements. On the path, some of the contextual elements has a value that intervenes in the practice (CE1, CE6, CE7, CE8 and CE9) and other not (CE2, CE3, CE4 and CE5). The first ones intervene in an ordered way and are called the proceduralized context. The second one is the set of elements called the contextual knowledge.

Thus, the context of the action A6 is defined by:

- Its proceduralized context: {CE1, CE6, CE7, CE8, CE9}, supposing that the actions A5, A6 are realized.
- The contextual knowledge: {CE2, CE3, CE4, CE5}

The main interest of contextual graphs relies on the possibility to introduce easily new practices in contextual graphs. A new practice generally corresponds at a known practice with few changes introduced by contextual nodes. Thus, a contextual graph based system either knows a practice used by a DBA or acquires it when needed.

Future work

The paper has shown that it is possible to use contextual graphs to model and represent database administration procedure. This has been illustrated using an example of a patch to be applied by a DBA to update the database in order to resolve incidents encountered by users or detected by the monitoring tools in the log files (or trace). In the case studied, we have pointed not only the technical contexts related to the patch application like the availability of a web site to download the patch but also the contexts about the different interactions between the DBA and other actors. The database patch procedure that has been used is based on a conventional or traditional approach of applying a patch. In our future work, we will continue our research by considering the other existing approaches for that procedure and the following aspects:

1) Complete the contextual graph by considering other contextual elements about:

- DBA-user's interactions;
- Contexts related to the technical details of the database patch (i.e. version number, release number, upgrade number, files copied, and bug) as well as database objects affected by the patch, etc.... Contextual graphs will also be expanded to other administration procedures such as database recovery after a failure causing loss of data. This will help to find out the new practices that will be developed by the DBA;

2) Now by using contextual graphs, we are able to represent a DBA task for applying a patch to the database. This can be extended to represent all the DBA tasks in order to build a real experience base.

3) Explore the possible interactions between Contextual graphs representing different tasks and their consequences.

4) Design and implement a context-based intelligent assistant system (CBIAS) that uses an experience base to help the DBA. The experience base should be developed in a uniform representation of knowledge, reasoning and context.

5) Evaluate the CBIAS by DBA for a set of procedures. The feedback of the database administrator is important for us to evaluate the practical efficiency of the system and to list the set of cases where difficulties may be encountered.

6) To generalize and extend the context-based intelligent assistant system to other domains of applications.

Conclusion

The paper has shown through an example that it is possible to improve the database administration by the contextualization of procedures used by the DBAs. The example concerns a procedure for applying a patch to update a database. It has illustrated that it is so easy to use contextual graphs to model the procedures and practices that have been developed by the DBA to consider the various contexts about situations he faced.

Contextual graphs have been used to represent both the initial administration procedure and the new DBA practices. These practices concern both technical and human contexts, with an emphasis on the various multi-actor interactions (DBA-Developer, DBA-Manager, DBA-End User, etc. ...).

Finally it is important to note that our research work is in the framework of the design of context-based intelligent assistant system not only for database administrators but also to be used in many other areas such as software design support, medicine, energy,…etc.

References

Barrett, R., & Kandogan E., et al. (2004). *Field Studies of Computer System Administrators: Analysis of System Management Tools and Practices.* Proc. CSCW 2004, November 6–10, Chicago, Illinois, USA. ACM

Brézillon, P. (1996). *Context in Human-Machine problem Solving: A Survey*. K. E. Review: 42.

Brézillon, P. (2006). *Some Characteristics of Context.* M. Ali and R. Dapoigny (Eds.): IEA/AIE 2006, LNAI 4031, Springer-Verlag Berlin Heidelberg, pp. 146 – 154.

Brézillon, P., & Pomerol J.-C. (1999). *Contextual knowledge and proceduralized context.* Proceedings of the AAAI-99 Workshop on Modeling Context in AI Applications, Orlando, Florida, USA, July.

Brézillon, P., & Pasquier, L., & Pomerol J.-Ch. (2002). *Reasoning with contextual graphs.* European Journal of Operational Research, 136(2): 290-298.

Brézillon, P. (2003). *Context-based modeling of operators' practices by contextual graphs.* Proceedings of HCP-2003, the 14th Mini-Euro Conference on Human Centered Processes, R. Bisdorf (Ed.), Fonds National de la Recherche, Luxembourg.

Carneiro, A., & Passos R., et al. (2004). *DBSitter: An Intelligent Tool for Database Administration,* Lecture Notes in Computer Science, Volume 3180/2004, 171-180, Berlin-Heidelberg, Springer-Verlag.

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

Haber, E. M., & Bailey, J. (2007). *Design Guidelines for System Administration Tools Developed through Ethnographic Field Studies*, Vol6, pp. 388-395, ACM.

Jabbour, G. G., & Menasee, D. A. (2008). *Policy-Based Enforcement of Database Security Configuration through Autonomic Capabilities.* Fourth International Conference on Autonomic and Autonomous Systems ICAS 2008. Gosier ICAS.

Mullins, C. S. (2002). *Database Administration: The Complete Guide to Practices and Procedures,* Addison Wesley.

Oliveira, F., & Nagaraja K., et al. (2006). *Understanding and Validating Database System Administration* http://www.cs.rutgers.edu/~ricardob/papers/usenix06.pdf

ORACLE (2001). *Introduction to Oracle Expert.* http://download.oracle.com/docs/html/A86647_01/intro.htm.

Qiao, L., & Soetarman B., & et al. (2007). *A framework. for enforcing application policies in database systems.* ACM.

Wessler, M. (2002). *Oracle DBA on Unix and Linux*, *for* SAMS.