Knowledge map for distributed information system re-engineering

Jorge Pardellas

University of Geneva, CUI, 24 rue General Dufour,
CH-1211 Geneva 4, Switzerland
jorge.pardellas@cui.unige.ch

Abstract. The distributed information system design decision has always been taken in the implementation step. The contribution of designer for this decision can decrease the evolutionary and the corrective maintenance. This designer's contribution can be realised through the specification of integrity constraints at distribution step. In this paper, we illustrate the contribution of such an approach for the design of distributed information system and we explain how this approach will be taken into account for the evolutionary re-engineering of such systems. This approach relies on the knowledge maps established between designer and developer that encapsulate the set of information and thus contribute to the distribution implementation.

1 Introduction

Increasingly today, organisations and their environment, due to their dynamic behaviour, generate new informational and structural needs that should be satisfied for their survival. The unification of organisations and the effect of the globalisation imply an integration of multiple data coming from various structures to coordinate them and thus ensure their integrity. This integration is done through their different correspondent information systems. The information systems have a major role intra and inter organisational by ensuring the collection, storage, sharing and the integrity of the data. In addition, an operational and robust information system must provide the right information to the right person at the good moment and with the lowest cost [7]. Information system has to be conceived as the principal actor in the decision-making process of any organisation [5].

These information systems, therefore, must imperatively be able to evolve with the structural and informational dynamism in order to satisfy the updated organisational and technical requirements. Because of their conceptual nature, the traditional information systems, generally centralized, are not easily maintainable and consequently cannot be adapted to these changes in order to reflect adequately the new requirements [3]. Hence, the traditional information systems do not support the evolution of the organisation's conceptual diagram, i.e. the internal data and consequently the internal database diagram, the integrity constraints and the specifications of the dynamic aspects. To contribute to the maintenance of coherence in the database, it is necessary to specify constraints, which the database must respect to preserve the data integrity. These constraints change and should be changed in order to meet the new needs. In fact, the evolution of the organisational environment or/and the informational environment often implies a radical change in the thinking of these constraints, which control these environments.

The maturation of database management system (DBMS) technology has coincided with significant developments in distributed computing and parallel processing technologies. The result is the emergence of distributed information systems based on distributed database management systems and distributed applications system. A distributed information system is a collection of data, transactions, integrity constraints and sites [2]. Data is concretized by a set of multiple, logically interrelated databases distributed over a computer network. This set is managed by a distributed database management system, which makes the distribution transparent to the users. Some researches on integrity constraint checking in distributed database environments addresses the problem of checking the global integrity constraints locally. [1] proposed the demarcation protocol which relies on storing some extra information about data on remote sites and the solution applies only to a limited class of integrity constraint. [2] presents ways to produce sufficient integrity constraint verification tests given a modification to the database and the entire original database. [6] describes techniques for distributing global constraints between sites in a way that reduces communication time.

Constraints of distributed information system are distributed on various levels. They can be implemented on the different existing levels of the system. In architecture such n-tiers levels, constraints can be implemented in graphical interfaces, application level or the data level. It is important to make the evolution of these integrity constraints easier to ensure the survival of information system. The information system evolution based on its integrity constraints evolution is difficult to manage because this evolution is complex and still controlled neither by the distributed database management system nor by the design and development tools of information system. The control of this evolution is required since it is the guarantor of the data's quality and the information system's performances [7].

The specification of integrity constraints in a distributed information system can cause problems at three different levels: transaction design, development and maintenance of integrity constraints. Firstly, the transaction design step guarantees the coherence of each transaction with respect to the integrity constraints. It has to meet the performance requirements ensuring a minimal data flow during execution of an integrity constraint using shared data between different sites. Secondly, the development of integrity constraints has to respect the design step and has to be executed in order to ease the future maintenance and the reliability of the constraints. Finally, the general problem is to check that the integrity constraints based on distributed data among different sites, are verified after any performed transaction.

2 Distribution framework

In order to answer the distribution problems, we needed to develop a framework and an associated suite of methods that enable designers to have assistance for the database schema specification while considering the different parameters involved.

2.1 Overlap knowledge pattern

We define an overlap knowledge zone between designers and developers in which they share common information and more important common distribution decisions. The distributed information system design knowledge pattern treats the knowledge surrounding the design step. This knowledge represents information obtained from the analysis step and expresses the main directions of the system. The distributed information system implementation knowledge pattern contains different generic knowledge about the constraints imposed by the different technical platforms. It specifies also the different performance and quality of service objectives. The distributed information system overlap knowledge pattern is essentially composed of shared decisions that must be taken both by designers and developers. It contains information about the integrity constraint design and the way of implementation, data locations and the different site relations. The designer's knowledge can contribute to the specification of the distributed information system through the different information extracted from the analysis at design step, i.e. triggering site, end-user queries, design justification, critical data, constraint weighting, validation site, etc. The developer's knowledge, i.e. the technical constraints he faces, has to be taken into consideration as well. This knowledge from both actors has to be shared in order to design and implement a distributed information system and its integrity constraints. Considering this common knowledge will help building evolutionary and easily maintainable systems.

2.2 Distribution algorithms

Several algorithms are proposed FKDO: Foreign Key Dependencies Optimisation, ICDO: Integrity Constraint Dependencies Optimisation, GIDO: Global Integrity Dependencies Optimisation are the main component of the Overlap Knowledge Pattern. The first algorithm is based on the foreign key distribution optimisation. Its aim is to decrease the number of the different site relations resulting from foreign key matching. The second algorithm treats the other different integrity constraints. Its result is a set of distribution solutions. The different generated solutions minimise the number of site connections and thus their dependencies. They also reduce the data flow and deliver better system performance and better answering time. Finally these algorithms structured in a global framework, that takes in consideration the various constraints' importance, recalculate the different solutions while matching every constraint with a weighting value. This value evolves continuously according to the frequency of the constraint call. Our different algorithms are the principal component of our purposed methodology for the design of distributed information system based on integrity constraints dependencies optimisation. The first goal of this approach is to give the designer a real view of the distribution system. This is done by generating the different distribution sets and comparing them according to their respective transaction costs. The second goal is to underline the most advantageous solution while giving the possibility to the designer to interact and influence this solution. This flexibility is concretized by the ability of the designer to fix needed data on predefined site and to order the different integrity constraints by determining the respective weighting. The third goal is to obtain the lowest distribution cost while maximizing the system integrity and availability.

3 Future directions

From the different knowledge patterns, the focus has been brought to the overlap knowledge pattern by defining different algorithms to allow the implementation of the distributed database diagrams while relying on the various components of the pattern. We will now focus on the other different components of the design and implementation knowledge pattern. We need to specify the implementation knowledge pattern to be able to re-engineer the conceptual diagram through the use of the integrity constraints. Indeed, the proposed distribution approach weight the transaction cost in an arbitrary way. During runtime, the system's data flow created by an integrity constraint could lead to rethink the conceptual diagram. By this way, the weighting criteria will be adjusted according to the actual call of every constraints and the regulation of the distribution combination will satisfy the reel system requirements.

The development of a CASE tool integrating the different distribution approaches is a very important issue. The FKDO algorithm is now supported by our implemented tool M7Tool. The extension of this tool to ICDO and GIDO need to be done to provide a complete tool to assist designers in the specification step of a distributed information system.

References

1. Barbara, D., Garcia-Molina, H.: The Demarcation Protocol: A Technique for Maintaining Arithmetic Constraints in Distributed Database Systems. In Extending Database Technology Conference, LNCS 580, pages 373--397, Vienna, March, 1992.
2. Bry, F., Manthey, R., Martens, B.: Integrity Verification in Knowledge Bases. In Logic Programming, LNAI 592 (subseries of LNCS), pages 114--139, 1992.

3. De Weger, K. M., A. Vissers, C.:Issues in design methodologies for distributed information systems, Center for Telematics and Information Technology, University of Twente, The Netherlands,1999.
4. Leitzelman, M., Dou, H.: Essai de typologie des Systèmes d'Information,. International Journal of Information Sciences for Decision Making N°2, April 1998.
5. Leonard, M: Database design theory, Macmilan Publication Ltd, Macmilan Computer science Series, 1992.
6. Qian, X:. Distributed design of integrity constraints. In Proceedings of the Second International Conference on Expert Database Systems, pages 205-226. The Benjamin/Cummings Publishing Company, 1988.
7. Snene, M.: Knowledge Patterns of Distributed Information Systems – The Case of Distribution Design and Implementation Based on Integrity Constraints Optimisation, PhD Thesis n°578, Computer Science University Center, Geneva, 2004.