IDEA GROUP PUBLISHING **ITB13080**

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

This paper appears in the publication, *Advanced Topics in Database Research, vol. 5*
edited by Keng Siau © 2006, Idea Group Inc.

Chapter XXI

Semantic Integration in Multidatabase Systems: How Much Can We Integrate?

Te-Wei Wang, University of Illinois, USA
Kenneth E. Murphy, Willamette University, USA

ABSTRACT

This chapter reviews briefly the semantic integration issues in multidatabase development and provides a standardized representation for classifying semantic conflicts. We begin by summarizing the methods and issues in multidatabase design. From the perspective of database integration, we identify that semantic conflict is the main issue. We explore the idea further by examining semantic conflicts and propose taxonomy to classify semantic conflicts in different groups. This taxonomy is then evaluated by two different methods. Finally, we conclude by discussing the limits of database integration and how this challenge may be addressed.

INTRODUCTION

Semantic heterogeneity or semantic conflict is the main source of problems in multidatabase design. In this chapter, a brief review of previous work in semantic conflict identification is presented, which leads to the creation of a taxonomy for resolving conflicts in multidatabase design that is more inclusive when compared to existing frameworks, for example, that of Batini, Lenzerini, and Navathe (1986). A metadata

structure, based on this taxonomy, will be proposed that can be used as a point of reference (a common protocol) for semantic conflict resolution.

For the last three decades, multidatabase research has focused on resolving the problem of semantic heterogeneity or semantic conflicts. Semantic heterogeneity is often present in multidatabase systems because of the lack of global schema definition. The situation is similar to common misunderstandings that occur in everyday interpersonal communication. Misunderstandings can result between two people who speak different languages. They cannot understand one another unless interpreters are present. Even when interpreters are used, concepts that cannot be precisely translated remain. In fact, the level of shared understanding between the parties after communication depends heavily on the knowledge of the interpreters. Even if persons participating in the conversation are speaking the same language, misunderstandings can persist due to the ambiguity of language or the quality of the original information. Based on this analogy, it is apparent that not all semantic conflicts can be systematically resolved. We argue that a good conflict resolution system should have a data structure to separate resolvable and irresolvable conflicts. Given such a data structure, corresponding procedures can be created to integrate results from different data sources and to report inconsistencies that cannot be resolved.

To create multidatabases without semantic conflicts, a significant amount of research has focused on schema integration at the conceptual level (Lim & Chiang, 2000). Real-world examples such as the Cyc knowledge base (Collet, Huhns, & Shen, 1991; Deaton et al., 2005; Masters, 2002; Reed & Lenat, 2002; Singh et al., 1997) and the CORDS multidatabase (Barrowman & Martin., 1998; Martin & Powley, 1997) all use similar schema-integration concepts to provide multidatabase systems with an integrated view at the logical level. However, in practice, semantic conflicts exist not only at the logical or conceptual level, but also at the instance or run-time level. Therefore, in practice, many conflict resolutions may need to be performed at query run-time (Lonski, 1997). To facilitate run-time semantic conflict resolution, the integration engine should have the ability to construct consistent metadata at run-time. In this chapter, a data structure for the purpose of capturing the metadata generated by such a run-time integration problem is proposed. We organized this chapter by addressing the following questions in sequence:

- What is a multidatabase system?
- What are the methods currently used in multidatabase systems to resolve semantic heterogeneity?
- What are different types of semantic heterogeneity?
- Can a "better" taxonomy for classifying semantic heterogeneity be found resulting in a meta-data structure to assist in addressing semantic conflicts?
- Is the meta-data structure proposed sufficient for practice?

WHAT IS A MULTIDATABASE SYSTEM?

Typically, a database system is designed to address an organization's needs at a fixed point in time. Organizations, however, have information needs that are dynamic. The original design of a particular database system can soon be and often is quickly outdated.