Understanding Three Dimensions of Heterogeneity: The “Why”, “How” and “What” of Heterogeneous Database Systems

In a previous JDM issue, one of the Associate Editors issued a challenge to database researchers to “...encourage us to think about innovation and point out areas that ... are not only important but also open to innovation.” (Pirkul, 1995, p.33). In this editorial, I address one area that Pirkul explicitly identified as a topic for innovation, namely, *Heterogeneous Databases*. Below I present a novel framework for understanding Heterogeneous Databases and Database Systems. This framework identifies three dimensions of heterogeneity: the “Why”, “How”, and “What” of heterogeneous database systems. I claim that exploring answers to these three questions will help practitioners (1) identify heterogeneous application requirements and map them to appropriate database capabilities, (2) evaluate heterogeneous application techniques that fulfill these necessary requirements, and (3) select products that support the required techniques.

A heterogeneous database system is one where “unlike” components of the system can operate together as “like” components for the purposes of reusing, combining, and sharing data. At the heart of heterogeneity lies the issue of conflicts, inconsistencies, and incompatibilities that result from an attempt to reuse, combine, and share these unlike components. To achieve seamless heterogeneity, these differences must be reconciled at many component levels, such as hardware platform, data model, database management system (DBMS), schema, and data. Open systems architectures (such as client/server models) and database standards (such as SQL) have begun to alleviate problems of heterogeneity at the platform, model, and DBMS levels. Heterogeneity at the schema and data levels (semantic heterogeneity) is more difficult to achieve; this topic is the focus of the remainder of my comments. My hope is that the three dimensions I discuss below will serve as a framework for helping designers and developers of heterogeneous database applications identify their needs and fulfill their requirements. In the figure below, I graphically illustrate these three dimensions and the choices that I discuss for each dimension.

1st Dimension — the Why of Heterogeneity. This dimension addresses the question: What do I want to do with my data that requires a heterogeneous database system? The answer to this question can also be viewed as the “objective” of the resulting heterogeneous system. If my goal is to support the reuse of data in a dynamically changing database environment, then this implies that evolution of the database system is an important requirement. However, if a major need is to combine data from many sources by joining tables across different databases, then integration is the main objective. Finally, if the application demands data sharing by exchanging actual data values, then my main objective is interoperability. Understanding how the data will be used and manipulated sets the stage for addressing the next dimension.

2nd Dimension — the How of Heterogeneity. This second dimension asks the question: How can I achieve my objective to reuse data (through evolution) or to combine data (through integration) or to share data (through interoperability)? This question focuses on the “approach” taken to accomplish the desired objective. Two complementary approaches to heterogeneity are conflict detection and conflict resolution. Both of these processes are prerequisites