This paper focuses on a portion of the design of a user interface to multiple heterogeneous databases. The design of such an interface involves independently mapping the schema of each database to a common representation or data model. These representations are then combined to form an integrated schema which provides the user with a consolidated view of the overall information space. More specifically, this paper addresses the merging of object-oriented database descriptions into a unified object model. Four tests are provided to identify relevant differences between object classes. A hypermedia interface, based on the unique class views and their relationships, is described.

Recent technical advances that have been made, including the availability of optical storage and the introduction of the graphical user interface (GUI) in the PC, have spawned interest in non-record oriented databases. Some of these databases store information that has a time-varying quality (audio or video), while others store images or drawings. A host of multimedia DBMS’s and front-end products, such as dVoice, MediaBase, Concept, and Spase, are now on the market (Barry, 1992). One of the primary information technology shifts of the 90’s, noted in Tapscott’s Paradigm Shift, is this shift to multimedia (Tapscott & Caston, 1993).

Furthermore, evolving communications standards and broadband optical networks are facilitating the development of remote links to and between databases (Cannata, 1991). Large corporations want to rapidly integrate information from diverse internal sources for strategic reasons. The use of public data banks compounds the problem of relating information from physically remote and incompatible systems (Bertino, 1991). Multi-database interfaces are now being designed which involve virtually autonomous databases, which have more and more heterogeneous data types. While each database can function independently, the shared functionality involves communication with a user who is unaware of the details of this heterogeneity. Ideally, this user queries data in the multiple databases using a single language and obtains integrated results. In 1991, IEEE sponsored the first conference of international scope to focus on these systems and the specific problems involved in database interoperability (Kambayashi et al., 1991).

The integration of autonomous and already existing databases involves resolution of syntactic and semantic heterogeneities. The syntactic differences have to do with how specific database products organize and access stored information. One protocol might involve storing items that are needed together (eg. part inventory and costs). Another might store things of the same type together (eg. each of the various inventories). The variety of storage and access protocols, embedded in specific hardware and software, makes integration of different databases non-trivial. There are several approaches to creating a single platform for access, depending somewhat, on if the user is meant to perceive that there are, in fact, multiple databases (Rose, 1992). Regardless of the approach, an integrating data model is required. Frameworks have been developed based on Codd’s relational data model (Litwin & Abdellatif, 1986; Deen et al., 1987), Su’s semantic resolution in multi-database environments.
data model (Krishnamurthy et al., 1988), and Shipman’s functional data model (Motro, 1987; Conners & Lyngback, 1988). Several researchers have taken an object-oriented approach to integration (Kaul et al., 1990; Czejdo & Taylor, 1991; Li & McLeod, 1991; Towell, 1993). The object-oriented paradigm is considered especially appropriate for systems with dynamic constituency or with significantly heterogeneous data types such as multimedia systems (Banerjee et al., 1987).

The semantic differences stem from the fact that designers name and partition things somewhat differently (e.g. a part may be in a bin in one inventory system, whereas a unit, which could be a part, may be on a shelf in another inventory system). Naming standards, as well as the issue of granularity (i.e. what is the smallest unit described) must be understood before database information can be meaningfully combined. It is also non-trivial to develop general strategies for solving semantic conflicts.

This paper focuses on the semantic issues in object oriented environments. Database schemas that have been independently mapped to object representations are combined into a unified object model of the information space. To facilitate this process, tests of overlap, cardinality, assignment, and level are provided to evaluate object class equivalence. Identification of key object relationships is also explored. The purpose of such exercises is to present to the user a single integrated “map” of the available information in the various databases. For each unique object class, in the consolidated set, a class view is then created. This view serves as an expanded picture of the class and provides direct links to views of related classes. The steps in developing an object-oriented interface to multiple databases are represented in Figure 1. The contribution in this work is a formalization of step 3 in this process.

This work is organized in the following manner. Section 2 discusses a set of tests to establish object equivalence for merging object class sets from multiple databases. Unique object classes in the unified object schema are represented with individual class views. Section 3 explores key class relationships so that the interface can incorporate direct

---

**Figure 1. Design of a Multi-Database Interface**

1) Multiple databases with related domains but significantly different data types are identified.

2) The schema for each database is mapped to a common object representation.  

3) The individual object sets are combined and then presented as a set of object class views in a hypermedia interface.  

4) A user accesses the multiple databases through the single interface which facilitates browsing and retrieval of object instances.  

5) After retrieval, general-purpose and application-specific utilities can be provided to structure and reduce the information retrieved.
Related Content

Metaschemas for ER, ORM and UML Data Models: A Comparison
www.igi-global.com/article/metaschemas-orm-uml-data-models/3277?camid=4v1a

The Schema Mapper: An Expert System that Determines the Least Cost Physical File Structure in a Database Management System
www.igi-global.com/article/schema-mapper-expert-system-determines/51177?camid=4v1a

Electronic Usage Statistics
www.igi-global.com/chapter/electronic-usage-statistics/7919?camid=4v1a

Future Directions for Public Service Delivery in the Digital Era
www.igi-global.com/chapter/future-directions-for-public-service-delivery-in-the-digital-era/233413?camid=4v1a