Knowledge and Object-Oriented Approach for Interoperability of Heterogeneous Information Management Systems

Chin-Wan Chung and Chang-Ryong Kim
Korea Advanced Institute of Science and Technology, Korea

Son Dao
Hughes Research Laboratories, USA

For the interoperability of heterogeneous information management systems, schema mapping approaches have been used to build a unified view. The schema mapping approach offers full transparencies and is very powerful from the user’s point of view. However, the traditional mapping approach needs to be strengthened for information management systems that have non-traditional data types, no schema, or incompatible schemas. We have incorporated numerous concepts and constructs associated with the knowledge and object-oriented paradigm such as abstract views with a set of procedures, encapsulation, inheritance and class composition hierarchies to resolve the above problem. This extension also accommodates the ability to determine and explicitly represent the semantics in the schema. Additionally, we have outlined a query processing method using the unified view. We are currently developing a prototype to support seamless access to structured data and unstructured data managed by different information management systems.

The information of a large organization is distributed across diverse information management systems. The diversity is mainly caused by the difference of requirements of applications, the advances in information management technologies, the ad hoc historical development of systems, and evolution of systems in organizations. Therefore, it is unlikely that the diversity will diminish. The interoperability of heterogeneous information management systems is necessary to provide the sharing of an organization’s information.

In past years, several projects have been developed to address the interoperability. Their approaches, federated or schema mapping approach, centered around the ability to define a unified view and to support translation to/from local schemas (Chung, 1990; Dao et al., 1987; Garcia-Solaco et al., 1995; Thomas, 1990). The schema integration process for building this unified view is a very critical process in the federated approach.

The federated approach using the relational data model as a common data model offers full data distribution transparency and is very powerful from the user’s point of view. But this approach is limited with respect to information management systems that have no schema or incompatible schema models (e.g., file systems, text, spatial, geographical information systems).

In the federated approach, a common data model is needed to represent the unified view. Several research projects suggest the use of relational, semantic, entity-relationship (ER)/extended ER, and object-oriented data models, etc, for the common data model.

Databases offer facilities for managing large amounts of data, but are limited in their expression and structuring facilities; while object-oriented programming languages provide features for expressing and structuring complex entities (through data abstraction, encapsulation, and inheritance).
The object-oriented data model incorporates ideas from the semantic data model and the object-oriented programming language. Semantic data modeling offers richer types of relationships (i.e. aggregation and groupings), whereas the object-oriented language encapsulates behavioral aspects of objects. The object-oriented data model is currently implemented in several object-oriented database management systems (OODBMS) (Butterworth, 1991; Deux, 1991; Kim et al., 1990; Lamb et al., 1991; Soloviev, 1992).

Our approach is to use advanced modeling concepts (i.e. a semantic data model, knowledge representation) in an object-oriented paradigm to form a common data model. The goal of the model is to provide flexible features to resolve data structure incompatibilities of underlying data models, interrelationships of objects at different locations during schema integration, and the semantics required by different applications.

Using the common model, a unified view can be built from local schemas through schema integration. Once a unified view is provided, users can develop applications which access data objects in the unified view. This access will be translated to an access to local data.

The use of object-oriented techniques has already been presented in other papers (Ahmed et al., 1991; Bertino et al., 1989). While we use a federated approach, (Ahmed et al., 1991; Bertino et al., 1989) used a multidatabase approach. In a multidatabase approach, a common data model is selected, then local schemas for local databases are imported to the location(s) which uses the common data model. The imported schemas are translated to schemas in the common data model to provide a uniform interface; however, the schemas are not integrated. Therefore, the multidatabase approach does not require a unified view. On the other hand, the approach does not support the data distribution transparency. The approach in (Bertino et al., 1989) is based on the operational mapping which consists of defining the correspondence between operations at different levels. Our approach is based mainly on the structural mapping and somewhat on the operational mapping. The formation of a query utilizes the structure and the message sending to an object utilizes the operation.

Non-object-oriented approaches (traditional approaches) use data models other than the OO data model as a common data model. As mentioned earlier, the object-oriented data model combines the features of the semantic data model and the object-oriented programming language. The OO data model is a superset of other data models. Therefore, other data models cannot effectively represent some of the features of the OO data model such as multimedia data types and methods. Consequently, it is difficult to interface the OODBMS using a non-object-oriented approach. We use a federated and object-oriented approach to support the transparency of the locations of diverse databases including object-oriented databases.

From the above discussions, we observe that there are needs for a new approach for an effective interoperability, and they are the motivations of our research as follows:

- It is necessary to provide the interoperability among information management systems that have non-traditional data types or incompatible schemas.
- In any case, the users should not be responsible for finding the locations of necessary data. The system should support the data distribution transparency.
- In the past, only the feasibility of the interoperability was considered. However, an environment to achieve interoperability without much difficulty must be provided.

The initial local DBMSs to be interfaced are the relational DBMS, the object-oriented DBMS, and the hierarchical DBMS. Recently, use of the relational DBMS has been widespread, while the object-oriented DBMS is suitable for engineering and manufacturing applications. An interface to the hierarchical DBMS IMS in a heterogeneous database environment is important because IMS has been the most heavily used mainframe DBMS in large organizations. The interoperability of these three major types of DBMSs covers many important issues and it will be the basis for future expansion.

The remainder of this paper is organized as follows: The common data model is presented in the next section. The following section describes the interface between the object-oriented database and the relational database, and the subsequent section briefly explains the interface to the hierarchical database. The architecture of a prototype is described in the following section. A comprehensive example is given in the next section.

**Integrated Knowledge-object-oriented Data Model**

A common data model is required to support the integration of multiple databases. Extensibility and flexibility features are the major advantages of the OO data model as a common data model compared with other data models. Although the OO data model provides basic features for general requirements, specific features need to be added to effectively support different areas of applications. In particular, since we are concerned with the integration of heterogeneous databases, our approach is to consider the structures and representations of objects tightly coupled with the operations that one may want to use on them. Thus, we propose to start with the OO data model to support general concepts. We then extend the concepts to support more specific issues such as query decomposition, data distribution, and translation of queries and responses. The result is a model that merges OO concepts with special extensions to access heterogeneous databases. This model is called an Integrated Knowledge-Object-Oriented Data Model (IKOODM) and it can be implemented using an OODBMS.

In other words, IKOODM is an extension of the OO data model.