한국산업정보학회논문지 제6권 제3호 (2001. 9)

Schema Integration Methodology and Toolkit for Heterogeneous and Distributed Geographic Databases

Jin-Soo Park*

요 약 스키마 통합은 이종 분산(Heterogeneous and Distributed) 지리데이타베이스 시스템 (GDS, Geographic Database Systems)에 있어서 해결해야 할 가장 과제들 중의 하나이다. 다양한 응용분야에 있어서 공간정보(spatial information)의 사용이 점차적으로 증가해 감에 따라 지리정보의 통합은 의사결정자들에게 있어 대단히 중요한 문제가 되었다. 그러나, 데이타베이스 관련 문헌에 기술되고 있는 기존의 스키마통합 기법은 시각적인 데이터, 공간정보, 임시적인 정보들을 내포하고 있는 복잡한 객체들간의 이질성(heterogeneity)의 관리라는 문제를 간과하고 있다. 스키마통합의 어려움은 의미(semantics)상의 혼돈뿐만 아니라 공간모형에 대한 상이한 표현으로부터도 초래된다. 그러므로, 지리데이타베이스 분야에서 데이터베이스간의 상호작용성(interoperability)을 실현하는 것은 생각했던 것보다 훨씬 복잡한 문제를 야기하게 되는 것이다. 본 연구에서는 이러한 문제의 해결을 시도하기 위하여 이종 분산 지리데이타베이스에 있어서 스키마통합을 지원할 수 있는 방법론과 프로토타입 도구를 소개하고자 한다.

Abstract Schema integration is one of the most difficult issues in the heterogeneous and distributed geographic database systems (GDSs). As the use of spatial information in various application areas becomes increasingly popular, the integration of geographic information has become a crucial task for decision makers. Most existing schema integration techniques described in the database literature, however, do not address the problems of managing heterogeneities among complex objects that contain visual data and/or spatial and temporal information. The difficulties arise not only from the semantic conflicts, but also from the different representations of spatial models. Consequently, it is much more complex to achieve interoperability in the area of geographic databases. This research attempts to provide a solution to such problems. The research reported in this paper describes a schema integration methodology and a prototype toolkit developed to assist in schema integration activities for GDSs.

1. Introduction

For the past three decades, traditional data processing is continually being replaced by database management systems (DBMS). During this period, organizations such as businesses, governments and colleges have heavily invested in computer-based information systems. As businesses grow, many organizations develop multiple islands of different computer and database systems. Over time, databases in these environments are developed independently to meet their specific requirements, and these heterogeneous databases are frequently accessed through organizational computer networks, corporate intranets, and the Internet. One important outcome of such independent database development is *semantic conflicts*. Semantic conflicts are mismatches encountered in information representation and structure. Semantic conflicts occur when semantically similar information is represented by, for example, different names and different data structures in different local databases. Local data access terms are developed to meet specific local requirements and are not globally consistent. In addition, most of these

* Information and Decision Sciences Department, Carlson School of Management, University of Minnesota, 조교수

database systems are not fully and accurately documented. Since each existing local database must be completely understood in order to encompass all underlying assumptions and semantics, semantic conflicts make the design of an integrated system difficult. Moreover, integrating disparate systems relies on subjective judgment (i.e., knowledge about the application domain, intended use of the integrated schema, etc.) of human beings, and cannot be generated totally automatically (Sheth et al. 1993).

The problem is even more complex in geographic databases because of the nature of complexities in geographic data (also called spatial data). These datasets are time and space specific, and come in various formats that must be integrated into a Geographic Information System (GIS) from different sources and geographic locations, which are captured by various types of devices (Medeiros and Pires 1994). As the use of geographic information in various applications becomes increasingly popular, interoperability among various geographic databases has become an important issue for decision makers. Most of the studies described in the multidatabase interoperability literature, however, do not address the problems of managing heterogeneities in various geographic database systems (GDSs). GDSs provide spatial data manipulation and query, as well as support for GIS operations such as spatial search and overlay. Since geographic data tends to be collected from various sources and archived locally before being shared with the rest of the scientific community, most of geographic databases are heterogeneous(i.e., different types, different resolutions and different spatial and temporal properties under different formats) and distributed. Some of the major problems in geographic databases are a large semantic gap between current geographic data and users (Ram and Park 1996), and semantic heterogeneity among geographic databases (Worboys and Deen 1991). This research addresses the inherent problems of semantic conflicts in geographic databases and proposes a methodology for schema integration and semi-automated tools to achieve semantic interoperability among various GDSs.

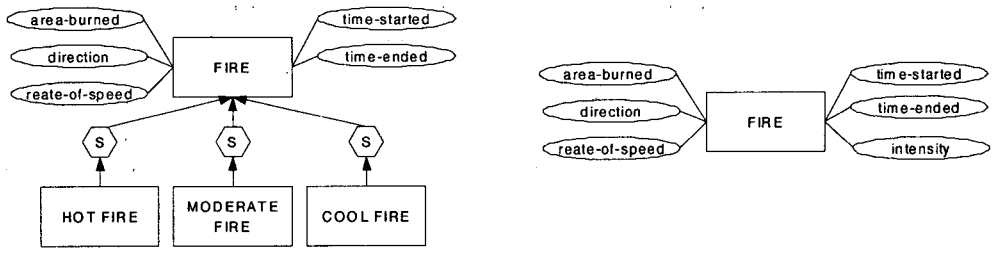
The remainder of this article is organized as follows. Section 2 reviews existing schema integration methodologies. Section 3 addresses some of the inherent heterogeneity problems encountered in the geographic database area. Different levels of heterogeneities found in GDSs are also discussed in section 3. In section 4, a schema integration methodology for heterogeneous geographic databases is proposed. Associated supporting tools for the proposed methodology are described in section 5. Finally, section 6 concludes this article with a discussion of the contributions of our research and future directions.

2. Background of the Research

In this section, we first examine the different types of semantic heterogeneities that are often found in geographic databases. We then review some existing approaches to schema integration. Ram et al. (1999b) provide a comprehensive framework for the classification of semantic heterogeneity. They state that the semantic heterogeneity can be broadly categorized into two different levels: schema-level and data-level.

Schema-level heterogeneities result from the differences in logical structures and/or inconsistencies in metadata of the same domain used in different databases. Two basic causes include (1) the use of different structures (tables and attributes) for the same information, and (2) the use of different specifications (e.g., names, data types or constraints) for the same structure. Data heterogeneities result from the differences in data domains caused by the multiple representations and interpretations of the semantically same data. According to Ram et al. (1999b), schema-level heterogeneities are further classified into six different types of conflicts: naming conflicts (homonyms and synonyms for entities and attributes), entity identifier conflicts, schema isomorphism conflicts, generalization conflicts, aggregation conflicts including spatial aggregation conflicts, and schematic discrepancies Figure 1 shows an example of schema-level heterogeneity typically found in geographic databases. The partitioning of FIRE into HOT FIRE, MODERATE FIRE, and COOL FIRE is represented by a generalization hierarchy in one schema (left side) and by an attribute intensity of the entity FIRE in the other schema (right side).

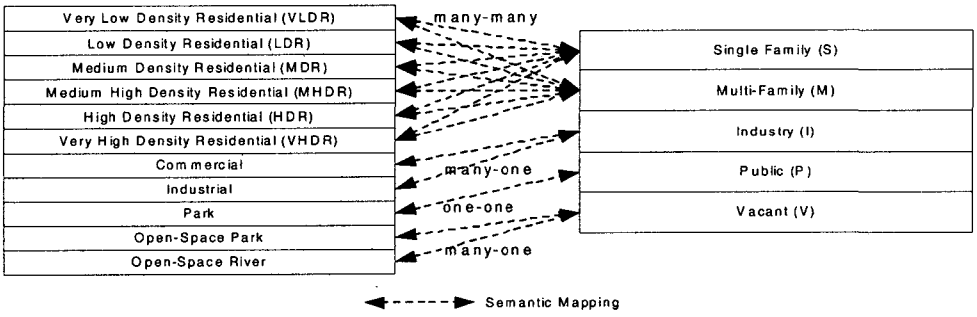
.Data-level heterogeneities can also be classified into six



<Figure 1> Equivalent Representations from Two Schemas

different types: data value conflicts, data representation conflicts, data unit conflicts, data precision conflicts including data granularity and spatial resolution, known data value reliability conflicts, and spatial domain conflicts. For example, spatial domain conflicts occur when the specifications of geographic regions or objects always be possible to specify mappings between one value and another because of many-to-many mappings in all contexts so that precise semantic transformation between the two may not be possible

To the best of our knowledge, no schema integration methodology has been proposed specifically for the



<Figure 2> Spatial Domain Conflict Example

are differently but legally defined by different people. Conflicts of this type are commonly found in land use databases. For instance, the land use information of a certain region may be measured by Residence Per Acre (e.g., very low density residential, low density residential, medium density residential, medium high density residential, high density residential, very high density residential, commercial, industrial, park, open-space park, open-space river) or dwelling uses (e.g., single family, multi family, public, industry, vacant). These conflicts may occur because of the different needs of different application domains. Their being correctly categorized may be defended as legally defined. The only difference is the adoption of measures that have been based on different land use criteria. In such a case, as illustrated in Figure 2 it may not

heterogeneous geographic database environment. However, from the existing work in schema integration techniques, we have acquired a substantial understanding of research in this area. Thus, it is worthwhile reviewing existing integration methodologies. The semantic perspective of schema integration uses schema-level information and structural conflicts to resolve the semantic heterogeneity problem (Kim and Seo 1991; Ram and Ramesh 1999). Structural conflicts arise when the real world is represented by different views using different schemas. For example, the same object of the real world might be represented as an entity type in one schema and as an attribute of an entity type in another schema (see Figure 1)

Three popular approaches to heterogeneous database integration are the global schema approach, federated

schema approach and the semi-decentralized approach. A global schema approach produces a single logical view of the integrated databases. A federated schema approach, which is based on the federated database approach, integrates multiple export schemas from each local database, and allows the database administrator to create an import schema (Sheth and Larson 1990). The import schema describes data that can be accessible in the local database. The semi-decentralized approach integrates both global and federated schema approaches (Papazoglou et al. 1990). This approach facilitates the object-oriented data model, which consists of the object definition and the object transformation layer. The object transformation layer performs the interschema transformations. Most of the recently developed multidatabase systems use a federated approach for schema integration (Breitbart 1990).

Schema integration is at the core of methodologies that use either of these approaches to provide heterogeneous database interoperability (Ram and Ramesh 1999). It has been argued by Ram and Ramesh (1999) that the term schema integration has been loosely used in the literature to refer to methodologies that facilitate integration of schemas and methodologies for view integration. They provide a clear distinction between the two terms: schema integration and view integration. They define the schema integration as the process of generating one or more integrated schemas from existing schemas and cannot breach the semantics of the existing databases. The view integration refers to the process of generating a single integrated schema from multiple user views and is typically used in the design of a new database schema. Accordingly, the schema integration is a bottom-up database design approach, and the view integration is a top-down approach. We believe that the view integration provides more flexibility in the interpretation of the semantics of abstract objects. In this paper, we regard view integration as part of schema integration.

The purpose of view integration is to build a conceptual schema, starting from an informal description of user requirements. Lavathe and Schkolnick (1978) discuss view modeling and view integration in the process of logical database design. They present a scheme for view representation that facilitates the process of view integration. View modeling is used to model the usage and information structure of the real world from the point of view of different users and/or applications. View modeling should explicitly represent each users view of the real world (*external schema*). Specification of user data and processing requirements must be analyzed extensively in the view modeling phase.

A most critical part of the database design process is the integration of different user views into a unified, non-redundant conceptual schema. Schema integration is very complex because the same portion of reality is usually modeled in different ways in each schema. The main difficulty of schema integration is to detect the differences in the schema to be merged. Differences in user views are due to the differences in user perspectives. In the conceptual database design process, users model the same objects from their own point of view. Thus, concepts may be seen at different levels of abstraction, or represented using different properties. A variety of representation structures in conceptual database design result in different equivalent representations of the same reality. Incompatible design specifications also cause conflicts in schema. For example, errors during view modeling regarding names, structures and integrity constraints may produce erroneous inputs for the integration activity. During schema integration, these errors should be captured and corrected.

The schema integration process combines different user views into a single global view. The purpose of schema integration is to find all parts of the input conceptual schemas that refer to the same portion of reality, and to unify their representation. Through the schema integration process, several conflicting user views must be merged and integrated into one or more global schemas of the required data. In case of conflicts, a concession should be established through negotiation among users. Since alternative models may exist, the model produced during the schema integration process must be analyzed and refined into an optimal structure. These processes are iterated until an integrated global conceptual schema is produced. The integrated conceptual schema finally captures the complete meaning of all the information maintained in all the database systems.

Batini et al. (1986) use a four-phase integration

process: (1) preintegration, (2) comparison of the schemas, (3) conformation of the schemas, and (4) merging and restructuring of the schemas. During the preintegration phase, database administrators and designers select schemas, decide the order of integration and set an integration policy or preference (e.g., binary or *n*-ary integration process). Then, schemas are analyzed and compared to detect possible schema and data conflicts. Interschema properties can be discovered while comparing schemas. The third phase requires close interaction between designers and users to resolve such conflicts so that the merging of various schemas can be performed. A global schema is finally created after restructuring some intermediate integrated schemas. They argue that the global schema should be tested against four qualitative criteria: completeness, correctness, minimality and understandability. The details of conflict analysis and transformation techniques are given in Batini et al.

Dayal and Hwang (1984) present an integration methodology for functional models. They examine several kinds of structural and data inconsistencies that may exist during the conceptual database design. Generalization abstraction is uniformly used as a means to combine entities and resolve different types of conflicts. They also provide a detailed algorithm for query modification. They propose suitable transformations introducing subset-generalization relationships in the integrated schema. Their methodology involves integrating databases by translating heterogeneous logical schemas into a conceptual data representation. A semantic data model with generalization abstract is used as an intermediate model to facilitate the integration. Their approach utilizes the concept of generalization. It is suggested that all objects should be given uniform treatment in models of the real world. They try to resolve schema differences between entity types using generalization. Their methodology to integrate schema differences is divided into three phases: (1) resolving conflicts among concepts in the local schema, (2) solving differences among data in existing databases, and then (3) modifying queries to make them consistent with the global schema. They categorize four schema differences (naming conflicts, scale conflicts, structural conflicts and differences in abstraction) and two data conflicts (mutually inconsistent local databases containing correct or incorrect information).

Schema integration is a difficult and complex task. An expert system approach to database design in general and schema integration in particular on the basis of the rules and heuristics of design is worth investigating (Batini et al. 1986). Hayne and Ram (1990) introduce an expert system, called Multi-User View Integration System (MUVIS). MUVIS supports a simultaneous database view entry from several users under a distributed environment. MUVIS uses a semantic data model as the underlying object-oriented model and provides graphical specification of the user views to help the designer represent user views and integrate them into a global schema. MUVIS uses existing integration methodologies, rules, and heuristics to capture schema conflicts between two objects and then determines the degree of object equivalence. MUVIS automatically compares the differences in schemas in a binary fashion. If a conflict is detected in a global schema, the system presents a recommendation to the designer to resolve the conflict. If the conflict cannot be resolved by the designer, the system provides an electronic discussion between designers to resolve conflicts. The system was developed to assist designers in expediting the view integration process. However, the system does not evaluate and provide several alternative schema transformations to allow the designers to select among alternative schemas for integration when a conflict must be solved. If the system can provide several alternatives from its knowledge base, conflicts may be more easily resolved.

Larson et al. (1989) present the concept of equivalence to integrate attributes, entity classes, and relationships between entities from different databases. They also provide formulations of different strategies for attribute, entity class, and relationship integration. They define four types of equivalence, *equal*, *contains*, *contained_in*, and *overlap*, for interschema transformation and schema integration. Their methodology uses both schema-level and data-level information for schema integration. On the other hand, Kashyap and Sheth (1996) use the concept of semantic proximity to compare the context in the domains of two objects and schema correspondences to represent structural similarities between entity classes. This work

attempts to resolve both schematic and data level conflict by relating the schema correspondence with the context of the semantic proximity among entity classes.

Ram and Ramesh (1995) propose the use of blackboard architecture for schema integration. One of the major purposes of using blackboard architecture is to facilitate cooperation among multiple knowledge agents. The architecture is composed of knowledge sources, a blackboard (a shared global database), and a scheduler that makes it possible for some tasks to be performed concurrently. There are four types of knowledge sources (schema translation engine, conflict identification engine, conflict resolution engine, and human integrator) and four levels of the blackboard (data level, assertion level, fact level, and goal level). The highest level of the blackboard architecture is the goal level. In this level, the integrated schemas are generated from the fact level information. One distinct feature of this work is that they employ blackboard architecture to support human interaction during all of the phases of schema integration. Human judgment in schema integration is very important because a completely automated schema integration process is not possible (Sheth and Larson 1990). This approach utilizes various knowledge sources using different processing paradigms to provide effective interaction between humans and other knowledge sources.

3. Heterogeneities in Geographic Databases

Interoperability among heterogeneous GDSs is one of the major challenging issues (Frank 1986; Worboys and Deen 1991). However, little research has been conducted to date regarding how different schemas of geographic databases have to be integrated and synthesized to help decision-makers. The difficulties arise not only from the semantic conflicts (schematic and data level conflicts) in spatial and temporal data, but also from the different representation of spatial models (vector format vs. raster format). It is much more complex to resolve heterogeneities in the geographic databases because both spatial and non-spatial data heterogeneities must be resolved to achieve interoperability. *Spatial data* refers to any information related to a location, and *non-spatial data* (also called a *thematic attribute* or simply an *attribute*) describes the characteristics of the spatial object (Dangermond 1993). Consequently, an integration methodology in geographic databases should include methods for managing these two different types of data. One of the most difficult problems to achieve interoperability is due to the nature of spatial data itself, such as spatial dimension, inherent inaccuracy (e.g., converting from an infinite set of continuous points in space to discrete formats), variations in the level of abstraction, discrepancies caused by different input sources (e.g., remote-sensing, digitizing, scanning, survey, etc.), standardization, different formats and scales, etc.

Based on the above observation, we propose an integration methodology that incorporates both types of data. This involves two different types of integration processes: *spatial integration* and *non-spatial integration*. The spatial integration process consists of *visual integration* and *analytical integration*. The visual integration is defined as the process of generating an integrated visual representation from various existing *layers* of raster and vector image data formats. The analytical integration refers to the process of resolving conflicts in the numerical data used to perform spatial analysis, i.e., spatial overlay, spatial simulation, geo-statistics, etc. The non-spatial integration refers to domain mismatch problems in thematic attributes and schematic conflicts. The proposed methodology is discussed in the following section.

In this section, we discuss three different levels of heterogeneities in geographic databases. They are *internal representation heterogeneity*, *model representation heterogeneity*, and *spatial reasoning heterogeneity*. Each layer focuses on different issues in schema integration. Internal representation heterogeneity arises when different geographic databases use different format types to store and represent spatial data: raster vs. vector-based systems. In the raster-based system, the space is divided into regularly sized and shaped cells. Each cell is called a grid. The spatial location of each cell is implied by its position in the cell matrix. Thus, the spatial coordinates of the cells need not be stored. In the vector-based systems, each bit of information is represented as a set of connected points, where the line segment between two points can be

considered a vector. Data of this type is stored on a mathematical topology and includes operations to determine the boundary of a given object. Since the data is not stored in the cell, the spatial location must be explicitly defined. Note that, in the case of non-geographic databases, existing schema integration methodologies are addressed at the logical/conceptual level, which is independent of its physical representation. This is not the case in geographic databases. For example, if the spatial query requires an answer from multiple databases where each database has information that contains different scales and different data formats, the operation will require proper data transformation from one format (e.g., raster) to another (e.g., vector) in addition to semantic transformation between different scales. The operation thus should handle possible errors and translation constraints (Worboys and Deen 1991) from vector-to-raster (or raster-to-vector) transformati on, and still provide local transparency between the user and the system.

While the internal heterogeneity is caused by differences in spatial data formats, a more pervasive problem lies in the semantic interpretation of the spatial objects involved in the interschema transformation (Frank 1986; Nyerges 1989). This type of problem is called model representation heterogeneity. Model representation heterogeneity occurs when semantic and syntactic conflicts exist among different geographic databases at both schema-level and data-level (spatial data and thematic attributes), such as different semantics for the conceptual schemas in the same domain, expressions, units, scales, precisions, etc. With respect to spatial units (visual and analytical data), conversions based on spatial overlay, geo-statistics, surface and areal estimates, and aggregation / disaggregation of survey samples are required. Difficulties can arise in resolving schema conflicts at this level because of the lack of appropriate conceptual models for geographic databases. Even though the importance of the conceptual database design for GIS and geographic databases are mentioned in several studies (Guptill 1990; Lee and Isdale 1991; Morehouse 1990), few attempts are made to formally define spatial and temporal data at the conceptual level of geographic databases. Note that most of the well-developed current methodologies rely on a semantic data modeling approach for schema integration, but none of them addresses issues related with internal representation and spatial reasoning heterogeneities. The absence of conceptual models for geographic databases has led us to develop a formal semantic data model, called USM* (Unifying Semantic Model*), which is an extension of USM (Ram 1995). The USM* defines several modeling constructs to capture the spatial and temporal nature of geographic data, as well as the dynamic behavior of spatial objects (e.g., fire, wind, erosion, etc). The formal definitions of the USM* constructs are described in Ram et al. (1999a).

The spatial reasoning heterogeneity can occur due to the different philosophical backgrounds and understandings of the nature and structure of space and spatial objects. The modeling of spatial phenomena may depend on different perceptions of the real world because human beings may employee different methods to conceptualize space (Frank 1992). Several studies discuss these different views (Goodchild 1992; Güting 1994; Medeiros and Pires 1994; Peuquet 1994). They generally agree that there are two different approaches: one views the world as a set of fully definable, discrete objects; the other views the world as complex continua in space. The former refers to the object-based view; the latter refers to the space-based view. Table 1 **summarizes the differences between these two views. Note that the spatial reasoning heterogeneity differs from the internal and representational schema heterogeneities, and may be impossible and undesirable to integrate. A detailed discussion of this issue can be found in Park (2000).**

<Table 1> Comparisons of Entity-Based View and Space-Based View

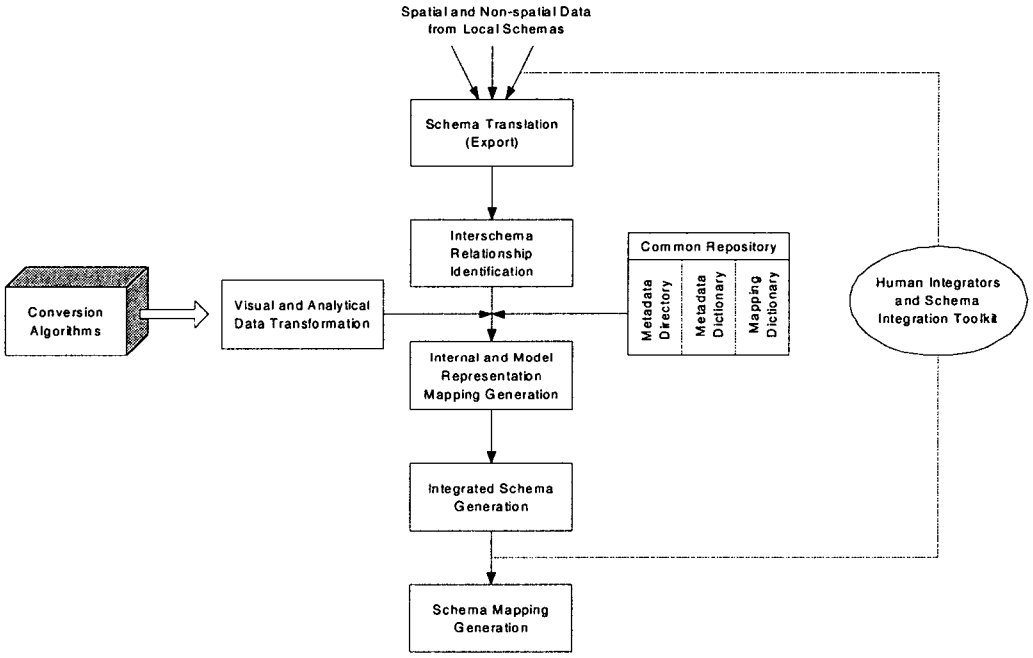
Object-Based View	Space-Based View
The world as a set of descriptive entities/objects	The world as the complex continua of space
Spatial entities are fully definable	Spatial entities are incompletely definable
Entities are described by their attributes, relationships, and rules	Entities are described by discretized surfaces (grid) or continuous mathematical functions
Represented by vector-data model structures	Represented by raster-data model structure

4. Spatial Schema Integration Methodology

The proposed methodology for spatial schema integration is based on the schema integration method from Ram and Ramesh (1999). Their integration methodology is extended to perform the spatial schema integration for the geographic databases. We assume that federated database architecture (Sheth and Larson 1990) is used for schema integration. The federated database approach provides more flexibility for spatial schema integration in a heterogeneous and distributed environment because it supports local autonomy of the underlying databases and does not need to create a single global schema for a large number of databases (Breitbart 1990; Litwin and Abdellatif 1986;). It is very important that the users of the system need not be aware of the location and the source of the data The system should hide the complex interconnections among various underlying databases so that users believe they are dealing with a single database (local transparency). This can be accomplished by providing a uniform schema in the form of an integrated view that hides the structural difference of the underlying databases. In the current implementation, the integrated schema is represented by the USM* schema.

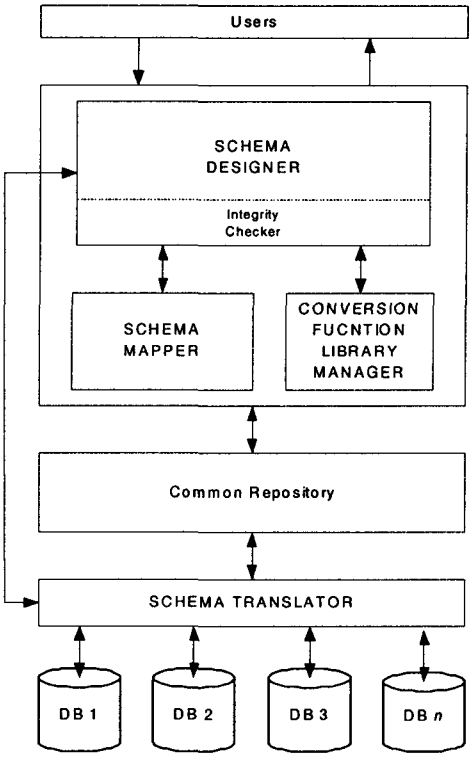
The spatial schema integration process, which consists of six major steps (see Figure 3) is described in the following paragraph.

The *schema translation* phase prepares a local schema (or a portion of a local schema) to be available to the schema integration process by translating the local schema into its corresponding representation in the common data model, USM*. The USM* schema should be able to completely represent the semantics of the underlying local database. During the *interschema relationship identification* phase, conflicts in spatial and thematic attributes, relationships and entities are identified. The spatial data transformation (*visual and analytical data transformation*) is then processed simultaneously in conjunction with the interschema relationship identification. This can be accomplished when the local schema has associated spatial and analytical data that contains mismatched domains and different data formats. A wide range of procedures are available from the conversion algorithms for calculating and estimating such data.



<Figure3> A Framework for Spatial Schema Integration Process

A common repository is used during the interschema relationship identification phase and the internal and model representation mapping generation phase in order to generate a reliable set of relationships and classification of data. The metadata dictionary stores standard semantics for attributes, entity classes, and relationships to categorize interschema relationships. The *internal and model representation mapping generation* phase uses the mapping dictionary to map the transformed schema to the corresponding local schema and the metadata directory to store the location of each data object to enable accessing such data. During the *integrated schema generation,* interschema relationships are used to create an integrated schema that represents the underlying schemas. The integrated schema resolves and hides all kinds of heterogeneities, thus providing a single unified view of the underlying heterogeneous databases. The *schema mapping generation* phase maps the integrated schema into the transformed schemas and stores information about mappings for spatial query transformation. This integration process is *iterative,* and requires interactions with human integrators during the entire process.



<Figure 4> A Toolkit for Spatial Schema Integration

5. Software Toolkit for Spatial Schema Integration

A prototype toolkit for spatial schema integration has been implemented. The toolkit consists of a schema designer, a schema translator, a schema mapper, and a conversion function library manager. Figure 4 illustrates the overall architecture of these tool components. These tools were implemented and tested on a Windows NT Server, a Sun Ultra Solaris Workstation, and a Linux server. The programming language used to develop these tools is Java (Java 2 SDK). The common repository has been implemented using an Oracle 8*i* server. The toolkit can be accessed through Java-enabled web browsers or used as a stand-alone client-server application.

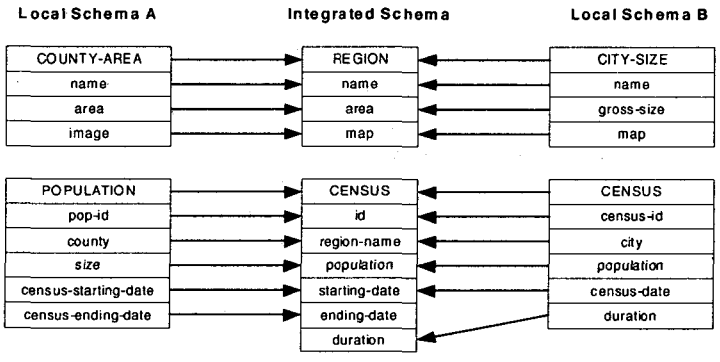
The *schema designer* allows database administrators and authorized users to create a federated schema (i.e., an integrated schema) or to translate local schemas into the USM* schema. Remember that the integrated schema is expressed in the USM* schema. Users can define various types of entity classes and their relationships using a graphical user interface with an intelligent dialog. During the schema design process, the user of the system essentially describes the data in the underlying database in terms of the USM* constructs (Ram et al. 1999a). A built-in integrity checker prevents errors during schema design. The schema designer allows users to browse through the metadata and query the underlying data.

The *schema translator* automates the translation from its own local schema to USM*. It automatically produces export table definitions that can be edited by human integrators. To expedite this process, we have implemented Semantic Metadata Extracting and Visualizing Agent (SMEVA) (Lee and Hwang 2001).

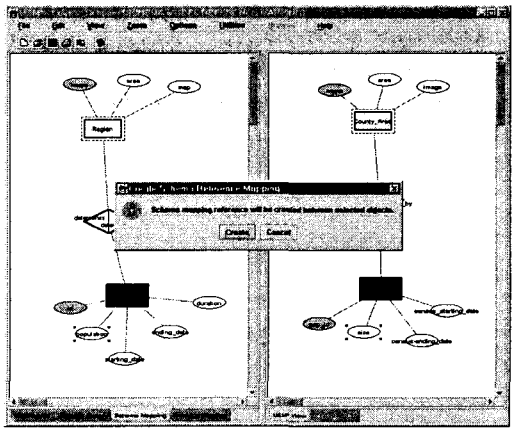
SMEVA is a reverse engineering agent that transforms relational schema to conceptual schema using USM* constructs. The schema translator is designed to help both database experts and novices. Thus, users who do not have the necessary knowledge of databases and USM* can easily transform a local schema to a USM* schema. The schema translator can significantly reduce schema design time.

The *schema mapper* allows the user to specify interschema relationships between an integrated schema and multiple local schemas at a metadata level. This schema mapping is a two-step process. First, the user identifies semantically equivalent schema components based on metadata provided. As discussed previously, this is a subjective activity relying heavily on the expertise of humans. In most cases, there is no automatic way to perform this activity. After analyzing schemas and metadata, the user determines the attribute equivalence and entity/relationship equivalence based on previous knowledge, application domain knowledge, intended use of the integrated schema, and so on. After schema analysis, the user invokes the schema mapper to establish mappings between an integrated schema and a selected local schema. The mapping process itself is a very simple click-and-point operation using a mouse. The user of the schema mapper simply clicks on a component in an integrated schema and points to the corresponding local schema component using a graphical user interface.

For example, let us assume that the user wants to establish mappings between an integrated schema and local schemas. Figure 5 shows a hypothesized example of two heterogeneous local schemas and an integrated schema. It also illustrates the relationships between semantically related attributes using linked lines between the integrated schema and local schemas. Let us say that the user wants to assert a mapping between the attribute population in entity class CENSUS from the integrated schema and the attribute size in entity class POPULATION from local schema A. The user then simply clicks the integrated schema component population in entity class CENSUS and then points to the corresponding local schema component size in entity class POPULATION from local schema A. The schema mapper then confirms a new schema mapping assertion that the user just created. Figure 6 shows that the left window pane contains the integrated schema of the two local schemas and the right window pane shows the local schema A. Note that the user can also establish schema mappings between any combination of schema components, if the two schema components are semantically equivalent (e.g., from an entity class to an entity class, from an attribute to an entity class, from an attribute to a relationship, from an entity class to a relationship, from a relationship to a relationship, etc.).



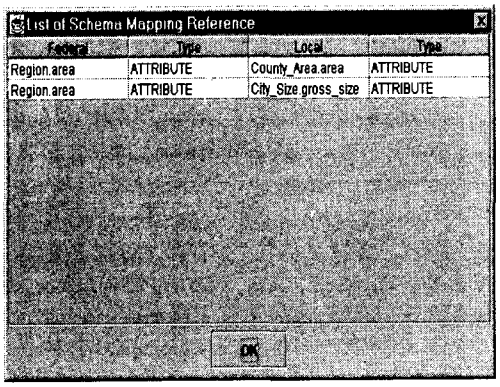
<Figure 5> Hypothesized Example of Schema Mapping



<Figure 6> Establishing Mapping between Schema Components using the Schema Mapper

After completing the mapping process, the user can browse mapping information for each schema component.

Figure 7 for example, shows the mapping information of the attribute area in entity class REGION from the integrated schema. The user can also examine the same information from any local schema that has been mapped to the integrated schema. In addition, the user can browse and update the mapping information. Capturing relationships among schemas through the interschema identification process is extremely important for the system to detect and resolve various schema conflicts.



<Figure 7> Browsing Schema Mapping Information

The last component, *conversion function library manager*, handles all kinds of conversion processes for spatial and analytical data. Conversion functions include spatial overlay, network analysis, polygon calculation, statistical surface estimation, topology analysis, raster-to-vector and vector-to-raster conversion, integer-to-string and string-to-integer conversion, etc. The conversion function library manager can automatically invoke the schema designer and schema translator for the user if the appropriate conversion function is not found. In this way, the user or human integrator can add or create new conversion functions to the library.

6. Conclusion

Schema integration is one of the most difficult issues in the heterogeneous and distributed geographic databases. Most schema integration techniques described in the literature, however, do not address the problems of managing heterogeneities among complex objects that contain both spatial and non-spatial information. A comprehensive framework for understanding semantic heterogeneity among geographic databases does not exist. This work is the first attempt to provide a schema integration methodology in the geographic database area. The use of existing integration methodologies that have been developed for heterogeneous databases, without considering heterogeneity among spatial information, may result in an incomplete integration of complex spatial data (Nyerges 1989). The intention of the proposed methodology is to fill the gap that exists between geographic and conventional database schema integration.

Another contribution of this research is the development of a prototype integration toolkit to demonstrate the feasibility and features of our proposed methodology. We have described and implemented a usable prototype system in order to demonstrate the practicality and performance of our methodology. The usability of such tools through several experimental studies is reported elsewhere. Most schema integration phases are manual processes and research in this area has focused on the problems of defining suitable

methodologies and languages. Software tools that support schema integration processes have recently started to emerge. However, whether such tools can help automate the entire integration process is still in question. Full automation may be impossible to support the integration process. However, the increasing complexity of the schema integration process will demand the development of some form of automated supports as the heterogeneous and distributed database systems grow and the data components become more complex. The practical implication of this paper is the development of the semi-automated tools for the spatial schema integration process. Our tools can help reduce a complex and time-consuming integration process and relieve the human integrator from such laborious tasks. An important aspect of the toolkit is that it supports human integrators through all phases of the integration process under a common working environment.

Future work includes the development of an advanced spatial query language capable of handling semantic conflicts in multiple heterogeneous geographic databases. The global spatial query language should allow users to seamlessly access a large number of geographic databases without requiring them to be familiar with the contents and structure of the heterogeneous data sources when users want to obtain an answer for a particular query.

References

[1] Batini, C., M. Lenzerini and S. B. Navathe, A Comparative Analysis of Methodologies for Database Schema Integration, ACM Computing Surveys, Vol. 18, No. 4, December 1986, pp. 323-364.

[2] Breitbart, Y., Multidatabase Interoperability, SIGMOD Record, Vol. 19, No. 3, September 1990, pp. 53-60.

[3] Dangermond, J., "A Classification of Software Components Commonly Used in Geographic Information Systems," in Introductory Readings in Geographic Information Systems, D. J. Peuquet and D. F. Marble (Eds.), Taylor & Francis, 1993, pp. 30-51.

[4] Dayal, U. and H. Hwang, View Definition and Generalization for Database Integration in a Multidatabase System, IEEE Transactions on Software Engineering, Vol. SE-10, No. 6, November 1984, pp. 628-645.

[5] Frank, A., Integrating Mechanisms for Storage and Retrieval of Land Data, Surveying and Mapping, Vol. 46, No. 2, June 1986, pp. 107-121.

[6] Frank, A. U., Spatial Concepts, Geometric Data Models, and Geometric Data Structures, Computers & Geosciences, Vol. 18, No. 4, 1992, pp. 409-417.

[7] Goodchild, M. F., Geographical Data Modeling, Computers & Geosciences, Vol. 18, No. 4, 1992, pp. 401-408.

[8] Guptill, S. C., "Multiple Representations of Geographic Entities Through Space and Time," in Proceedings of the 4th International Symposium on Spatial Data Handling, Zürich, Switzerland, July 2327, 1990, p. 859868.

[9] Güting, R. H., An Introduction to Spatial Database Systems, The VLDB Journal, Vol. 3, No. 4, October 1994, pp. 357-399.

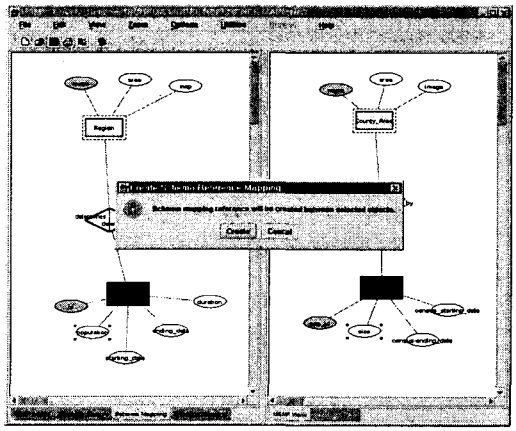
[10] Hayne, S. and S. Ram, "Multi-User View Integration System (MUVIS): An Expert System for View Integration," in Proceedings of the 6th International Conference on Data Engineering, Los Angeles, CA, February 59, 1990, p. 402409.

[11] Kashyap, V. and A. P. Sheth, Semantic and Schematic Similarities Between Database Objects: A Context-based Approach, The VLDB Journal, Vol. 5, No. 4, December 1996, pp. 276-304.

[12] Kim, W. and J. Seo, Classifying Schematic and Data Heterogeneity in Multidatabase Systems, IEEE Computer, Vol. 24, No. 12, December 1991, pp. 12-18.

[13] Larson, J. A., S. B. Navathe and R. Elmasri, A Theory of Attribute Equivalence in Databases with Application to Schema Integration, IEEE Transactions on Software Engineering, Vol. 15, No. 4, April 1989, p. 449463.

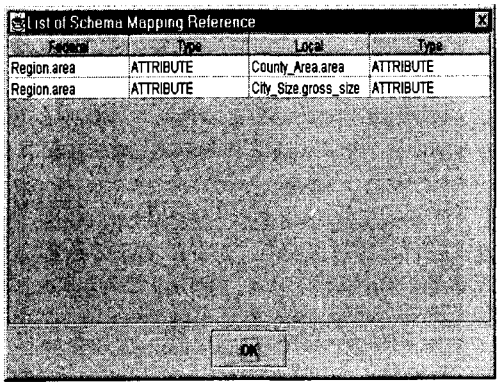
[14] Lavathe, S. and M. Schkolnick, "View Representation in Logical Database Designs," in Proceedings of ACM SIGMOD International Conference



<Figure 6> Establishing Mapping between Schema Components using the Schema Mapper

After completing the mapping process, the user can browse mapping information for each schema component.

Figure 7 for example, shows the mapping information of the attribute area in entity class REGION from the integrated schema. The user can also examine the same information from any local schema that has been mapped to the integrated schema. In addition, the user can browse and update the mapping information. Capturing relationships among schemas through the interschema identification process is extremely important for the system to detect and resolve various schema conflicts.



<Figure 7> Browsing Schema Mapping Information

The last component, *conversion function library manager*, handles all kinds of conversion processes for spatial and analytical data. Conversion functions include spatial overlay, network analysis, polygon calculation, statistical surface estimation, topology analysis, raster-to-vector and vector-to-raster conversion, integer-to-string and string-to-integer conversion, etc. The conversion function library manager can automatically invoke the schema designer and schema translator for the user if the appropriate conversion function is not found. In this way, the user or human integrator can add or create new conversion functions to the library.

6. Conclusion

Schema integration is one of the most difficult issues in the heterogeneous and distributed geographic databases. Most schema integration techniques described in the literature, however, do not address the problems of managing heterogeneities among complex objects that contain both spatial and non-spatial information. A comprehensive framework for understanding semantic heterogeneity among geographic databases does not exist. This work is the first attempt to provide a schema integration methodology in the geographic database area. The use of existing integration methodologies that have been developed for heterogeneous databases, without considering heterogeneity among spatial information, may result in an incomplete integration of complex spatial data (Nyerges 1989). The intention of the proposed methodology is to fill the gap that exists between geographic and conventional database schema integration.

Another contribution of this research is the development of a prototype integration toolkit to demonstrate the feasibility and features of our proposed methodology. We have described and implemented a usable prototype system in order to demonstrate the practicality and performance of our methodology. The usability of such tools through several experimental studies is reported elsewhere. Most schema integration phases are manual processes and research in this area has focused on the problems of defining suitable

methodologies and languages. Software tools that support schema integration processes have recently started to emerge. However, whether such tools can help automate the entire integration process is still in question. Full automation may be impossible to support the integration process. However, the increasing complexity of the schema integration process will demand the development of some form of automated supports as the heterogeneous and distributed database systems grow and the data components become more complex. The practical implication of this paper is the development of the semi-automated tools for the spatial schema integration process. Our tools can help reduce a complex and time-consuming integration process and relieve the human integrator from such laborious tasks. An important aspect of the toolkit is that it supports human integrators through all phases of the integration process under a common working environment.

Future work includes the development of an advanced spatial query language capable of handling semantic conflicts in multiple heterogeneous geographic databases. The global spatial query language should allow users to seamlessly access a large number of geographic databases without requiring them to be familiar with the contents and structure of the heterogeneous data sources when users want to obtain an answer for a particular query.

References

[1] Batini, C., M. Lenzerini and S. B. Navathe, A Comparative Analysis of Methodologies for Database Schema Integration, ACM Computing Surveys, Vol. 18, No. 4, December 1986, pp. 323-364.

[2] Breitbart, Y., Multidatabase Interoperability, SIGMOD Record, Vol. 19, No. 3, September 1990, pp. 53-60.

[3] Dangermond, J., "A Classification of Software Components Commonly Used in Geographic Information Systems," in Introductory Readings in Geographic Information Systems, D. J. Peuquet and D. F. Marble (Eds.), Taylor & Francis, 1993, pp. 30-51.

[4] Dayal, U. and H. Hwang, View Definition and Generalization for Database Integration in a Multidatabase System, IEEE Transactions on Software Engineering, Vol. SE-10, No. 6, November 1984, pp. 628-645.

[5] Frank, A., Integrating Mechanisms for Storage and Retrieval of Land Data, Surveying and Mapping, Vol. 46, No. 2, June 1986, pp. 107-121.

[6] Frank, A. U., Spatial Concepts, Geometric Data Models, and Geometric Data Structures, Computers & Geosciences, Vol. 18, No. 4, 1992, pp. 409-417.

[7] Goodchild, M. F., Geographical Data Modeling, Computers & Geosciences, Vol. 18, No. 4, 1992, pp. 401-408.

[8] Guptill, S. C., "Multiple Representations of Geographic Entities Through Space and Time," in Proceedings of the 4th International Symposium on Spatial Data Handling, Zürich, Switzerland, July 2327, 1990, p. 859868.

[9] Güting, R. H., An Introduction to Spatial Database Systems, The VLDB Journal, Vol. 3, No. 4, October 1994, pp. 357-399.

[10] Hayne, S. and S. Ram, "Multi-User View Integration System (MUVIS): An Expert System for View Integration," in Proceedings of the 6th International Conference on Data Engineering, Los Angeles, CA, February 59, 1990, p. 402409.

[11] Kashyap, V. and A. P. Sheth, Semantic and Schematic Similarities Between Database Objects: A Context-based Approach, The VLDB Journal, Vol. 5, No. 4, December 1996, pp. 276-304.

[12] Kim, W. and J. Seo, Classifying Schematic and Data Heterogeneity in Multidatabase Systems, IEEE Computer, Vol. 24, No. 12, December 1991, pp. 12-18.

[13] Larson, J. A., S. B. Navathe and R. Elmasri, A Theory of Attribute Equivalence in Databases with Application to Schema Integration, IEEE Transactions on Software Engineering, Vol. 15, No. 4, April 1989, p. 449463.

[14] Lavathe, S. and M. Schkolnick, "View Representation in Logical Database Designs," in Proceedings of ACM SIGMOD International Conference

on Management of Data, May 31-June 2, 1978, pp. 144-156.

[15] Lee, D. and Y. Hwang, Extracting Semantic Metadata and Its Visualization, ACM Crossroads, Vol. 7, No. 3, Spring 2001, pp. 19-27.

[16] Lee, Y. C. and M. Isdale, "The Need for a Spatial Data Model," in Proceedings of Canadian Conference on GIS-91, March 1822, 1991, p. 530530j.

[17] Litwin, W. and A. Abdellatif, Multidatabase Interoperability, IEEE Computer, Vol. 19, No. 12, December 1986, pp. 10-18.

[18] Medeiros, C. B. and F. Pires, Databases for GIS, SIGMOD Record, Vol. 23, No. 1, March 1994, pp. 107-115.

[19] Morehouse, S., "The Role of Semantics in Geographic Data Modelling," in Proceedings of the 4th International Symposium on Spatial Data Handling, Zürich, Switzerland, July 2327, 1990, p. 689698.

[20] Nyerges, T. L., Schema Integration Analysis for the Development of GIS Databases, International Journal of Geographical Information Systems, Vol. 3, No. 2, 1989, pp. 153-183.

[21] Papazoglou, M. P., L. Marinos and N. G. Bourbakis, "Distributed Heterogeneous Information Systems & Schema Transformation," in Proceedings of International Conference on Databases, Parallel Architectures, and Their Applications, Miami Beach, Florida, March 79, 1990, p. 388397.

[22] Park, J., "Spatial Data Modeling: Issues and Implications on Geographic Information Systems," in Proceedings of the 2000 International Conference on E-Transformation and and E-Business with Coming of Digital Economy, Seoul, Korea, November 17, 2000, pp. 108-121.

[23] Peuquet, D. J., Its About Time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems, Annals of the Association of American Geographers, Vol. 84, No. 3, 1994, pp. 441-461.

[24] Ram, S., Intelligent Database Design Using the Unifying Semantic Model, Information and Management, Vol. 29, No. 4, October 1995, pp. 191-206.

[25] Ram, S. and J. Park, "Modeling Spatial and Temporal Semantics in a Large Heterogeneous GIS Database Environment," in Proceedings of the 2nd Americas Conference on Information Systems (AIS '96), Phoenix, AZ, August 1618, 1996, pp. 683-685.

[26] Ram, S., J. Park and G. Ball, Semantic Model Support for Geographic Information Systems, IEEE Computer, Vol. 32, No. 5, May 1999a, pp. 74-81.

[27] Ram, S., J. Park, K. Kim and Y. Hwang, "A Comprehensive Framework for Classifying Data- and Schema-Level Semantic Conflicts in Geographic and Non-Geographic Databases," in Proceedings of the 9th Workshop on Information Technologies and Systems, Charlotte, North Carolina, December 11-12, 1999b, pp. 185-190.

[28] Ram, S. and V. Ramesh, A Blackboard-Based Cooperative System for Schema Integration, IEEE Expert, June 1995, pp. 56-62.

[29] Ram, S. and V. Ramesh, "Schema Integration: Past, Current and Future," in Management of Heterogeneous and Autonomous Database Systems, A. Elmagarmid, M. Rusinkeiwicz and A. P. Sheth (Eds.) San Francisco, Morgan Kaufmann, 1999, pp. 119-155.

[30] Sheth, A. P., S. K. Gala and S. B. Navathe, On Automatic Reasoning For Schema Integration, International Journal of Intelligent and Cooperative Information Systems, Vol. 2, No. 1, 1993, pp. 23-50.
Sheth, A. P. and J. A. Larson, Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, ACM Computing Surveys, Vol. 22, No. 3, September 1990, pp. 184-236.

[31] Worboys, M. F. and S. M. Deen, Semantic Heterogeneity in Distributed Geographic Databases, SIGMOD Record, Vol. 20, No. 4, December 1991, pp. 30-34.

박 진 수

1991년 계명대학교 졸업 (BA)
1994년 University of Pittsburgh (MS, MBA)
1999년 University of Arizona (Ph.D, MIS 전공)
1999년 - 현재 University of Minnesota (조교수)

Journal of Database Management (Editorial Member)
Workshop on Information Technologies and Systems (Program Committee)

관심분야: Heterogeneous DB Management and Integration, Knowledge Sharing and Coordination, Data Modeling, Spatial DB systems, Geographic Information Systems, Intelligent Agents for Information Resource Management