GENERIC FRAMEWORK AND KEY ISSUES FOR UPDATES PROPAGATION BETWEEN HETEROGENEOUS SPATIAL DATABASES

WANG Yuhong [a, b] CHEN Jun[b]

[a] School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo, China - wyh3003@tom.com
[b] National Geomatic Center of China, Zizhuyuan, Beijing, China

KEY WORDS: GIS, Database, Updates Propagation, Schema Matching, Updates Retrieval, Semantic Transformation, Update Integration, Consistency Maintenance.

ABSTRACT:

Data updating is a significant stage in the life of a geographical information system (GIS). When geodata producers have finished updating their own database (named Master Database, MDB), the problem of how to propagate the updates in new version of MDB to users' database (named Client Database, CDB) has become a research focus. Although many works have been done to resolve this problem, it may be underlined that the common weak point of these different works is their lack of genericity. In this paper, we firstly analyze heterogeneities between MDB and CDB. Afterward, the Generic framework for updates propagation is presented. Several key issues within the proposed framework are discussed in detail, mainly including schema matching, updateing information retrieval, semantic transformation, update Integration, consistency maintenance, and so on. Finally, an implemented tool based on analysis of the above issues is presented.

1. INTRODUCTION

With the development of geographic information technology, a large number of GIS application systems have been established by various users to answer their specific tasks such as town planning, water resource management, etc. To reduce data acquisition costs and accelerate system construction, users often obtain geodata from producers and then do some reengineering and value-adding disposals on them to meet their particular needs. These disposals generally involve more or less: 1) Adjusting a data model (or schema) to facilitate efficient implementation of the target applications, including selectively transferring the exiting schema elements, newly defining some new schema elements. 2) Loading existing produces' data into user database according to the new schema, including reclassifying, filtering, transformation of the existing features and data. 3) Adding new classes of data necessary to user's specific tasks and completing information of features from producer's dataset, etc [16], [25]. For convenience of discussion and emphasis the dependence relationship, we call the producer's database as master database (MDB) and correspondingly the user's database as client database (CDB).
After the above handling process, there is some commonness, but at the same time, some discrepancies between MDB and CDB in some ways such as feature category, abstract level, label naming, data contents, geometrical precision, etc. Some researchers have discussed different kinds of potential discrepancies among multiple geo-spatial datasets [4], [24]. In general, these discrepancies can be categorized using two orthogonal classifications [29]. On the one hand, conflicts are classified as data values conflicts, schema conflicts and data model conflicts according to the abstraction level; On the other hand, conflicts may be viewed as syntactic conflicts and semantic conflicts from the point of view of representation and interpretation.

Data updating is a significant stage in the life of a geographical information system (GIS). At present, a lot of geodata gathering and producing organizations all over the world are all actively adopting various strategies and technologies to update or upgrade their own geodata products. Obviously, updates in producer's database should be in time propagated into users' databases to enable them to have the most realistic image of geographic reality. Currently, the whole updated MDB are usually disseminated to end users in bulk for updates propagation. Due to the discrepancies between MDB and CDB in terms of content and structure, updates propagation will be very difficult to perform. In order to achieve efficient and effective updates propagate from MDB to CDB, we analyze a series of issues related to it and suggest some solutions to them.
The remainder of this paper is organized as follows. In section 2, a generic framework for updates propagation are presented. Afterwards, several issues related to updates propagation, including schema matching, updating information retrieval, semantic transformation and updates integration, updating consistency maintenance, are respectively discussed in section 3, section 4, section 5 and section 6. Finally, in section 7, draw our conclusions with our developed tool for updates propagation and give the future works.

2. A GENERIC FRAMEWORK FOR UPDATE PROPAGATION

Clearly, it is impractical for user to reconstruct their database based on the updated MDB because it is a very time-taking, onerous and expensive process. Moreover, user cannot realize updates propagation simply by means of replacing old dataset with new dataset due to the coexisting of user's data with producer's data; otherwise, this means will cause the loss of user's value-added data. Therefore, some operations during updates propagation, different from ones used during database construction, should be adopted in order to avoid the loss of user's value-added data and keep the updated user's database autonomous, complete, correct, consistency as much as before. "Autonomous" means that the application system on user's database can still run normally and independently after it had been updated. "Complete" means that there should not be data

omission in the updated user's database, such as missing attribute value. "Correct" means that the data in the updated user's database should be rightly expressed on both sides of content and format according to its requirements. "Consistent" refers to the absence of apparent contradictions in the updated user's database. Moreover, many of these operations should be achieved as automated bulk processes to improve the efficiency of updates propagation.

In order to meet the above-mentioned requirements, many researchers had discussed a variety of issues related to updates propagation, for example, schema matching [16], [3], updating information retrieval [33], updating inconsistency detection [1], etc. However, it may be underlined that the common weak point of these different works is their lack of genericity. Although they precisely address one or several of these questions they left the others unconsidered or unanswered.
In this section, we draw up a complete review of updates propagation (illustrated in Figure 1) based on the existing research fruits, and then some solutions to key issues within the framework are summarize and present in the subsequent sections.

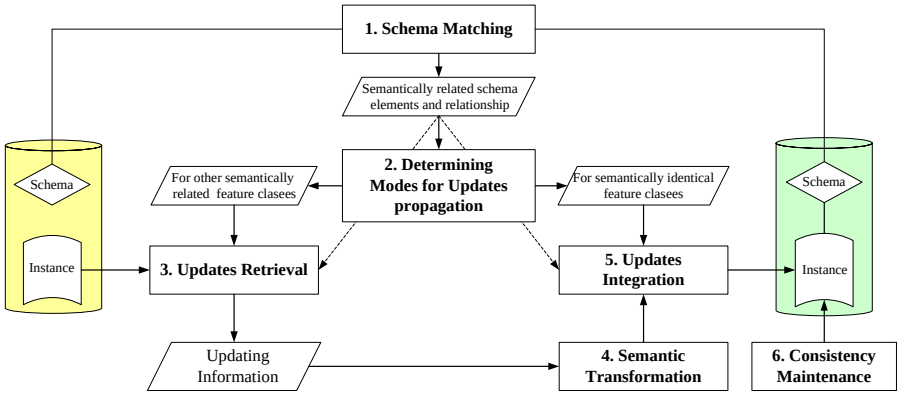


Figure 1 the Generic framework for updates propagation

3. SCHEMA MATCHING

Schema matching is the task of finding semantic correspondences between elements of two schemas, which takes two schemas as input and determines a mapping indicating which elements of the input schemas logically or semantically correspond to each other [11]. During updates propagation, the problem of which kinds of data in MDB can be utilized to update the corresponding data in CDB can be firstly resolved according to the related schema elements determined by schema matching. Afterwards, two updating methods can be adopted by analyzing semantic relationship between the related feature classes to keep updates propagation efficient and correct. For semantically and structurally identical feature classes, the updates propagation between them can be realized by replacing the old dataset in CDB with the new dataset in MDB. For semantically overlapping or containing feature classes the updates propagation can be realized only at granularity (or level) of feature to avoid the loss of user's value-added data. In this case, the schema mapping can be used to integrate the updates in CDB correctly.

At present, the schema matching tasks in commonly used GIS software, such as ArcGIS and MapInfo, are usually realized according to a default rule. It is that two schema elements (especially for attributes) are thought to be semantically same if their name labels are identical in spelling. Although this method can be used for quick data transfer from source dataset to target dataset, it will also result in some incomplete or incorrect condition due to synonymy and homonymy between the name labels of schema elements. To overcome the disadvantage of the above method and improve its flexibility, certain graphical user interfaces in a few software systems are designed to facilitate the manual and interactive customizing of schema mapping relationships between different spatial datasets, e.g., the attribute transfer mapping functions in ArcGIS 9.0, workbench component in FME (Feature Manipulation Engine). Obviously, manually specifying schema matches is a tedious, time-consuming, error-prone, and therefore expensive process and the level of effort is linear in the number of matches to be performed. In addition, manual schema matching also requires operators to have enough knowledge of different datasets. Thus, an automated and less labor-intensive schema matching approach is needed.

However, to the best of our knowledge, there are yet not approaches or prototypes for automatic schema matching between structural spatial databases. Several researchers only discussed the basic concepts related to schema matching problem for different application purposes [3], [8], [32]. Fortunately, there already exist a lot of prototypes and approaches aiming at automatically performing the task of schema matching between non-spatial databases, such as [9], [18], [21], [28]. A detailed classification of these existing approaches is given [11]. In general, each approach has its strengths and weaknesses. Many efforts are still required to achieve a generic and automatic approach to schema matching.

4. UPDATING INFORMATION RETRIEVAL

Just as the above discussed, the updates propagation between semantically overlapping or containing feature classes can be realized only at granularity (or level) of feature to avoid the loss of user's value-added data. On this condition, the actual changed features and the related information in MDB must be explicitly detected, recognized, identified and extracted to ensure that the updates to be propagated into CDB is really meaningful. In general, four methods (differential snapshots, time stamps, triggers and archive logs) are currently available for updating information retrieval between two structured datasets.

– **Differential-snapshots-based Methods**

In these methods, the modifications are inferred by comparing a current source snapshot with an earlier one [15]. The key problem of differential snapshots is how to match data segments of same semantic between two versions of dataset. [33] proposed the update retrieval mechanism based on topologic and geometrical data matching tools, it allows the automatic extraction of the evolutions between two versions of a same geographic database. According to the matching cardinalities, eight kinds of changes can be identified and extracted. Although such an approach is well adapted in a general context where no hypothesis on the data model is assumed, it is based on complex algorithms and needs considerable effort to be implemented.

– **Timestamp-based Methods**

If the tables in MDB have columns containing timestamps, then the latest data can easily be identified using the timestamp columns. For example, Ordnance Survey of Great Britain launched MasterMap in 2001. MasterMap encompasses new ways of managing and providing large-scale digital geodata to customers, and enables end-users to on-line or off-line select and receive change-only updates taken place since a specified day. The basic principle behind it is that timestamp columns (e.g. versiondate, changedata, etc) are introduced when the old geodata product, Land-line, are reengineer into MasterMap.

– **Trigger-Based Methods**

Trigger is a mechanism that initiates an action when an event occurs such as reaching a certain time or date or upon receiving some type of input. A trigger generally causes a program routine to be executed. Trigger-based Change Capture installs insertion, modification and deletion triggers on all data tables to monitor changes taken place on them and capture data into separate queues. For extract delta in MDB, this method only can be implemented at the side of producer. Trigger-based techniques might affect performance on MDB systems, and this impact should be carefully considered prior to implementation on MDB system.

– **Log-based Methods**

In Log-based Change Capture, changes to database are written to the log files (e.g. redo log), and then these changes are extracted from logs by an application logic (e.g. oracle streams). [33] proposed an updating information delivery mode named “updating delta”, this is, besides the new and old versions of the updated objects, the log files specifying the nature of the evolutions are also delivered to user. Of course, the log files he discussed is not dynamically created during updating of database, but created by differential snapshots after finishing updating of the new version database.

According to the above discussion, each of four methods for updating information retrieval has advantages and limitations. Differential-snapshots-based method is more generic due to no demands for other accessories, but its performance efficiency is a considerable problem especially there are no common entity identifier between two snapshots. Timestamps-based can be used to extract efficiently changes, whereas the structures of the database, in which timestamp information is not available, have to be modified to include timestamps. The trigger-based and log-based methods require no changes to the application and can in time capture changes, but it only can be implemented on database system with triggers or transaction logs. Although some databases do have logs, they do not publish their formats and APIs, which means that database-specific log sniffers and readers are not only more difficult to code, but are likely to be unsupported by the database vendor. In the Table 1, we make further comparisons among these four methods.

To decrease the data transmission amount, the above four methods can be firstly implemented in MDB on the site of producers, and then the extracted incremental (or change-only) information is transferred to end users. To improve the readability and accessibility of incremental information and make their integration in CDB easier, various exchange formats based on XML/GML are designed to store and manage incremental information [26].

Methods	Accessories	State	Modify system	Efficiency	Implementation Sites
Differential-snapshots	No	Static	No	general	producer and user
Timestamps-based	Timestamps	Static	possible	excellent	producer and user
Trigger-based	Triggers	dynamic	No	good	producer
Log-based	Logs	dynamic	No	good	producer and user

Table 1. Comparison among four methods for Retrieving Updates in MDB

5. SEMANTIC TRANSFORMATION AND UPDATES INTEGRATION

Once the updates have been retrieved from MDB, each relevant update can be integrated in CDB. However, due to their different semantic specifications on both sides of geodata geometry and attributes, some updates should be transformed in advance according to the CDB’s requirements to avoid inputting inappropriate data. Moreover, the execution process of transformation will become more complex along with the increase of difference degree on data specifications. Thus, the crucial task of semantic transformation is not only to design efficient geometrical transformation algorithms, but also to find attribute conversion functions or rules in a generic way. Fortunately, many relevant works have been done in the other research fields such as geographical data generalization [22], semantic sharing, interoperability and integration between geodata [23], spatial data warehouse, etc. Currently, the problems needed to be solved are how to filter our needed from their fruits and seamlessly associate them with other processes of updates propagation.

After appropriate updates is temporally hold through semantic transformation, three basic updating operations (Addition, Modification, Deletion) and/or their different combinations can

be performed on CDB to integrate updates in it. The concrete operations for updates integration mainly depend on three factors as follows:

– **Evolution types of features in MDB**

Several methods have been proposed to model and store the real world evolutions undergone by the geographical entities in the databases. Some approaches are closest to the implementation of time in GIS; some are closest to the modelling of real world evolutions, but are more difficult to implement [33]. For instance, the only two operations (addition and deletion) are considered in the case that two evolution types (apparition and disappearance), are modeled in MDB. Any modification of an entity implies the apparition of the new one [17].

– **Correspondence relationships between features from MDB and CDB**

In order to perform updating operations correctly, it is necessary to analyze the correspondence relationships between sets of geographical entities that represent the same phenomenon in two representations of the real world. These relations are often implicit (i.e. not explicitly stored in the database) and have to be retrieved on the fly. Several works propose various methods to establish these relations. In general, these methods can be divided into three kinds, that is, geometry-based method, attribute-based method, topology-based method. According to the cardinalities of relationships derived a certain method, different operations should be performed, e.g. 1:1 may correspond to modification; 1:n may correspond to combination of n-1 deletions and one modification.

– **Whether or not history data are required to be kept in CDB**

If history data are required to be kept, deletion operation is often prohibited. For example, the data about one disappeared feature are not deleted but modified in the timestamp-based legacy database.

6. UPDATING CONSISTENCY MAINTENANCE

An updating operation in a spatial database may cause simultaneous updates in a large number of records [3]. Due to the specificity of some updates or the lack of relevant integration information and operations, the integration processes of updates may result in inconsistencies or impossibilities. Consequently, the proper updating of a user's database implies handling these inconsistencies in an effort to keep it consistent as much as possible. Several methods have been proposed for maintaining consistency of spatial database when it is updated [1], [13], [19]. In general, these methods can be categorized into two types according to the maintenance scheduling: deferred and immediate, and the basic dataflow of them can be respectively illustrated in Fig.2.

– **Deferred method for maintaining consistency**

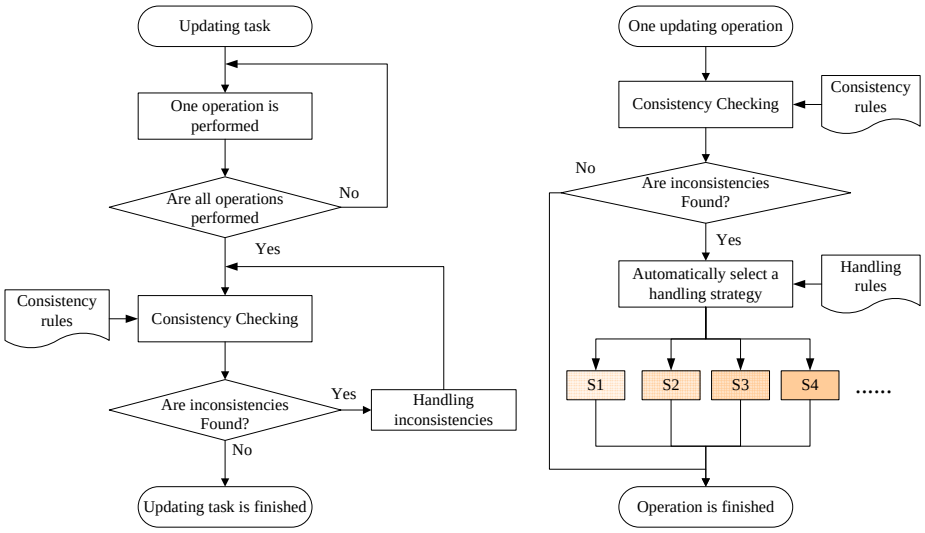
In the deferred method, consistency is checked and inconsistency is handled (usually manually) after all update operations needed in an updating task. The inconsistency caused by individual operation is temporally ignored. For example, ArcGIS 9.0 provides an "Error Inspector" to find the error caused by the edits violating the topology rules.

– **Immediate method for maintaining consistency**

In the immediate method, consistency is immediately tested d after the individual update operation is finished. If inconsistency violating the predefined rules is found, various strategies are optional to make an automated response to it, such as transaction rollback, error warning, exception treatment, and cascade updating. For example, ArcGIS 9.0 also provides a "Topology Edit" tool allowing operator to modify the shared edge or node among several features concurrently. If the shared geometry is edited using the traditional (nontopological) editing tools, only one feature is modified at a time.

From the above discussion, it can be perceived that rules and constrains play a very important role in preserving consistency of a user's database when it is being updating. Although these two concepts are usually used in confusion, strictly speaking, there are still certain differences between them. Constrains are conditions and relationships that must always be true, or must always be false. They are specified at database creation time and enforced by the database management system rather than at application or object level. Rules contain not only internal constrains in DBMS and external conditions residing in an application logic, but also the ways that database responds when a given condition is satisfied.

To implement the above-mentioned methods, despite requirements for further improvement some significant works have been done in various aspects of spatial data consistency maintenance, mainly including: classification of spatial consistency constraints [31], [32]; fomal description of consistency constrains [20]; calculating and detection of inconsistency [10], [12], [19], adjusting and handling of detected inconsistency [6], [7].



a. Deferred Methodology b. Immediate Methodology

Figure 2. The basic data flow of two methodologies for maintaining consistency

7. CONCLUSION AND FUTURE WORK

Aiming at the practical demands, we propose a generic framework for updated propagation and discuss comprehensively several key issues within it. Based on the analysis of these key issues, we have designed as prototype tool named UPBuilder (**U**pdates **P**ropagation **Builder**) for updates propagation between MDB and CDB. As Shown in Fig.3, there are three command components (i.e. Schema Matcher, Change Detector, Updates Integrator) developed especially for efficient updates propagation besides the usual commands for spatial data visualization and manipulation in UPBuilder. Schema mappings between MDB and CDB can be automatically derived through Schema Matcher and are visualizes in schema mapping window (SMW). Change Detector can be used to retrieve the updated features in MDB and the corresponding feature in CDB and output this information in Change Information Window (CIW). Finally, Updates Integrator is responsible for updating CDB according to the schema mappings and change information validated by user.

In the future, we will extend our works in two directions. One is to complete our prototype system and to enable it to perform geometrical semantic transformation and updating consistency checking and handling. The other direction is to increase the practical application experiments and improve the stability and reliability of UPBuilder.

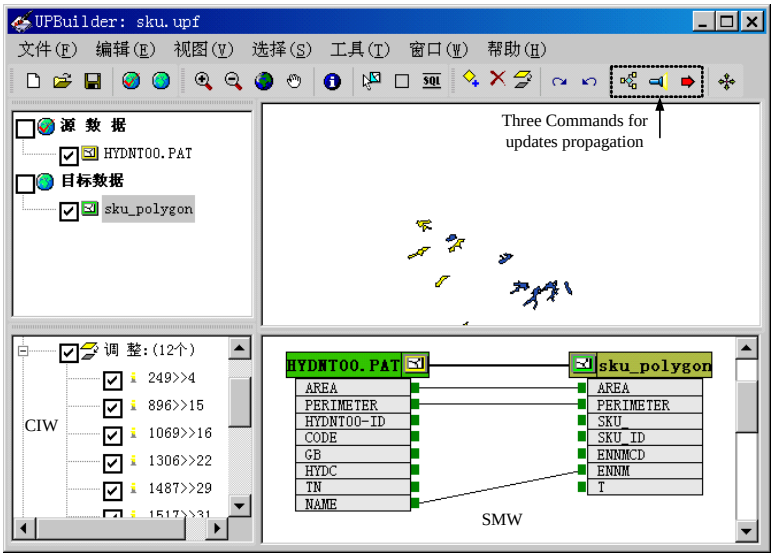


Figure 3. Main interface of UPBuilder

ACKNOWLEDGEMENTS

The work described in this paper is supported by the Natural Science Foundation of China (No.40337055).

REFERENCES

Ally Peerbocus, Geneviève Jomier, Thierry Badard, A Methodology for Updating Geographic Databases using Map Versions, Advances in Spatial Data Handling, Richardson, D. and van Oosterom, P. (Eds.), Springer-Verlag, Berlin, (2002) 363-376..

Andrea Rodrıguez, Inconsistency Issues in Spatial Databases, Inconsistency Tolerance, Lecture Notes in Computer Science, (2005).

Arnaud Braun, From the Schema Matching to the Integration of Updating Information into User Geographic Database, Geoinformaticas Gvale University Press, (2004) 211-218.

Bishr Y., Overcoming the semantic and other barriers to GIS interoperability, IJGIS, 12(4), (1998) 299-314.

Chao Ching-Ming, Chen Po-Zung and Yang Shih-Yang, Change Detection and Maintenance of an XML Web Warehouse, Tamkang Journal of Science and Engineering, 8(4), (2005) 299-312.

Claudia Bauzer Medeirors, Mariano Cilia, Maintenance of Binary Topological Constraints through active database, Proceedings of 3rd ACM International Workshop on Advances in Geographic Information Systems, Baltimore, Maryland USA, (1995) 127-133.

David Gadish, Inconsistency and Adjustment of Spatial Data Using Rule Discovery, PHD Dissertation of the University of Guelph, Canada (2001).

Devogele T., Parent C. and Spaccapietra S., On spatial database integration, International Journal of Geographical Information Science, 12(4), (1998) 335-352.

Do Honghai, Rahm Erhard, COMA - A system for flexible combination of schema matching approaches, Proceedings of the 28th VLDB Conference, Hong Kong, China, (2002) 610-621.

Egenhofer M., Clementini E., and Felice P., , Evaluating Inconsistencies Among Multiple Representations, Sixth International Symposium on Spatial Data Handling, Edinburgh, Scotland, (1994) 901-920.

Erhard Rahm, Philip A. Bernstein, A survey of approaches to automatic schema matching, VLDB Journal, 10(4), (2001) 334-350.

Gong Peng, Mu Lan, Error detection through consistency checking, Geographic Information Sciences, 6(2), (2000) 188-193.

Hakima Kadri-Dahmani, Updating Data in GIS Towards a More Generic Approach, Proceedings of 20th International Cartographic Conference, Beijing, China, (2001) 1463-1471.

Lars Harrie, Anna-Karin Hellström, A Case Study of Propagating Updates between Cartographic Data Sets, Proceedings of the 19th International Cartographic Conference of the ICA', Ottawa, Canada, (1999).

Labio W. and Garcia-Molina H., Efficient Snapshot Differential Algorithm for Data Warehousing, Proceeding of Twenty-Second International Conference on Very Large Data Bases, Zurich, Switzerland, (1995) 63-74.

Laurent Spery, A Framework for Update Process in GIS, Proceedings of the 3rd International Conference on GeoComputation (1998).

Laurent Spery, , Spatial Data Transfer in the case of Update, International Archives of Photogrammetry and Remote Sensing, Part. 4, GIS - Between Visions and Applications, Vol. 32, (1998) 586-593.

Li Wen-Syan and Clifton Chris, SEMINT：A tool for identifying attribute correspondences in heterogeneous databases using neural networks, Data & Knowledge Engineering, 30(1), (2000) 49–84.

Liu Wan-zeng, An automatic method for spatial conflicts detection in spatial database updating, PHD Dissertation of China University of mining & Technology Beijing, Xuzhou (2005). (In Chinese)

Mas, Stephan, Wang, Fei, Reinhardt, Wolfgang, Using Ontologies for Integrity Constraint Definition, Proceedings of the 4th International Symposium On Spatial Data Quality'05, Beijing, China (2005).

Melnik S., Garcia-Molina H., and Rahm E., Similarity flooding: A versatile graph matching algorithm and its Application to Schema Matching, In Proceedings of ICDE 2002, (2002) 117–128.

Meng Liqiu, Automatic Generalization of Geographic Data, Technical Report, SWECO, Stockholm, Swedish Armed Forces, (1997).

Open GIS Consortium Technical Committee, The OpenGIS® Guide（Third Edition）Introduction to Interoperable Geoprocessing, edited by Kurt Buehler and Lance McKee, (1998).

Park Jinsoo, Facilitating interoperability among heterogeneous geographic database systems, PHD Dissertation of Arizona University, (1999).

Paul Hardy, Active Objects and Dynamic Topology for Spatial Data Reengineering and Rich Data Modeling, Dagstuhl Seminar on Computational Cartography and Spatial Modelling (2001).

Peter Woodsford, A Prototype Spatial Object Transfer Format (SOTF), 6th EC-GI & GIS Workshop, Lyon, France (2000).
Prabhu Ram and Lyman Do, Extracting Delta for Incremental Data Warehouse Maintenance, Proceeding of the 16th International Conference on Data Engineering, (2000) 220-229.

Qiang Baohua, Wu Kaiwei, Wu Zhongfu, A data-type-based approach for identifying corresponding attributes in heterogeneous databases, Proceedings of the Second

International Conference on Machine Learning and Cybernetics, Xi'an, (2003) 299-344.

Renato Fileto, Issues on Interoperability and Integration of Heterogeneous Geographical Data, In III Brazilian Symposium on Geoinformatics - GEOINFO, (2001) 33–140.

Servigne, S. Ubeda, T. and Puricelli, A. A methodology for spatial consistency improvement of geographic databases. Geoinformatica, 4(1), (2000) 7-34.

Sophie Cockcroft, A Taxonomy of Spatial Data Integrity Constraints, GeoInformatica, 1(4), (1997) 327–343.

Steffen Volz, Volker Walter, Linking Different Geospatial Databases by Explicit Relationships, Proceedings of the XXth ISPRS Congress, Comm. IV, Istanbul, Turkey, (2004) 152-157.

Thierry Badard, On the automatic retrieval of updates in geographic databases based on geographic data matching tools, In Proceedings of the 19th International Cartographic Conference and the 11th General Assembly of ICA, Ottawa, (1999) 47-56.