

3 Fundamentals of spatial data warehousing for geographic knowledge discovery

Yvan Bédard, Tim Merrett and Jiawei Han

1 Introduction

Recent years have witnessed major changes in the Geographic Information System (GIS) market, from technological offerings to user requests. For example, spatial databases used to be implemented in GISs or in Computer-Assisted Design (CAD) systems coupled with a Relational Data Base Management System (RDBMS). Today, spatial databases are also implemented in spatial extensions of universal servers, in spatial engine software components, in GIS web servers, in analytical packages using so-called 'data cubes' and in spatial data warehouses. Such databases are structured according to either a relational, object-oriented, multi-dimensional or hybrid paradigm. In addition, these offerings are integrated as a piece of the overall technological framework of the organization and they are implemented according to very diverse architectures responding to differing users' contexts: centralized vs distributed, thin-clients vs thick-clients, Local Area Network (LAN) vs intranets, spatial data warehouses vs legacy systems, etc. As one may say, 'Gone are the days of a spatial database implemented solely on a stand-alone GIS' (Bédard 1999). In fact, this evolution of the GIS market follows the general trends of mainstream Information Technologies (IT).

Among all these possibilities, the penetration of data warehouses into the management and exploitation of spatial databases is a major trend as it is for non-spatial databases. According to Rawling and Kucera (1997), 'the term Data Warehouse has become the hottest industry buzzword of the decade just behind Internet and information highway'. More specifically, this penetration of data warehouses allows developers to build new solutions geared towards one major need which has never been solved efficiently insofar: to provide a unified view of dispersed heterogeneous databases in order to efficiently feed the decision-support tools used for strategic decision making. In fact, the data warehouse emerged as the unifying solution to a series of individual circumstances related to providing the necessary basis for global knowledge discovery.

First, large organizations often have several departmental or application-oriented independent databases which may overlap in content. Usually, such systems work properly for day-to-day operational-level decisions. However, when one needs to obtain aggregated or summarized information integrating data from these different

systems, it becomes a long and tedious process which slows down decision making. It then appears easier and much faster to process a homogeneous and unique dataset. However, when several decision makers build their own summarized databases to accelerate the process, incoherences among these summarized databases rapidly appear and redundant data extraction/fusion work must be performed. Over the years, this leads to an inefficient chaotic situation (Inmon *et al.* 1996).

Second, past experiences have shown that re-engineering the existing systems fully in order to replace them with a unique corporate system usually leads to failure. It is too expensive and politically difficult. Then, one must find a solution which can cope as much as possible with existing systems and which does not seek to replace them.

Third, the data structure used today by most decision-support solutions use, partly or completely, the multidimensional paradigm. This paradigm is very different from the traditional, normalized relational structure as used by most transaction-oriented operational-level legacy systems. The problem is that with today's technology, it is almost impossible to keep satisfactory response times for both transaction-oriented and analysis-oriented operations within a unique database as soon as this database becomes very large. One must then look for a different solution which provides short response times for both analytical processing and transaction processing. This very particular need has had a major impact on the way data warehouses are presently built, that is, as an additional database importing data in a read-only mode from the already existing legacy systems (prior to the restructuring of these data for analysis processing).

Fourth, as we mentioned previously, the decision makers need aggregated and summarized data for strategic decisions. This includes past data as well as present and predicted data (when possible), in order to analyse trends over time.

Finally, decision makers are also hoping for fast answers, simple user interfaces, a high level of flexibility supporting user-driven *ad hoc* exploration of data at different levels of aggregation and at different epochs, and finally some automatic analysis capabilities searching for unexpected data patterns systematically.

In other words, the needed solution must support the extraction of useful knowledge from bunches of detailed data dispersed in heterogeneous datasets. Such a goal appears reasonable if we consider data warehousing and automatic knowledge discovery as the 'common-sense' follow-up to traditional databases. This evolution results from the desire of organizations to further benefit from the major investments initially made into disparate, independent and heterogeneous departmental systems. Once most operational-level needs are fulfilled by legacy systems, organizations wish to build more global views that support strategic decision making (the frequent bottom-up penetration of innovations). In fact, this evolution is very similar to the situation witnessed in the 1970s where organizations evolved from the management of disparate flat files to the management of integrated databases.

Such a general evolution of needs and solutions is also taking place in the world of GIS. Innovative solutions are emerging from academia, industry and governmental research centers. R&D continues to be performed to offer spatial data warehouse and Geographic Knowledge Discovery (GKD) solutions that make

better use of automatic spatial data fusion, interoperability, generalized spatial analysis, multiscale mapping combining automatic generalization with multiple representations, navigational cartographic interfaces for spatial online analytical processing, spatio-temporal clustering detection, multi-scale spatio-temporal indexing methods, etc.

The goal of the present chapter is to present a unified overview of the fundamental concepts underlying spatial data warehousing. Such an overview aims to serve three different purposes. First, it introduces efficient means to provide the raw material for GKD and data mining, the main purpose of the present book. Second, in spite of extensive literature about non-spatial data warehousing since the mid-1990s, the large majority of this literature focuses on specific topics. We aim to present a unified view encompassing data warehousing with knowledge discovery concepts such as data mining, OnLine Analytical Processing (OLAP), visualization and so on. Third, this chapter contributes to the emerging literature specialized on spatial data warehouse and GKD.

This chapter includes four sections. After having presented the 'raison d'être' of spatial data warehousing in the previous paragraphs of this section, we introduce the most important concepts of non-spatial data warehousing in the second section in the global context of knowledge discovery. The third section of the chapter deals with the particularities of spatial data warehousing with regard to knowledge discovery, especially spatio-temporal applications. Some innovative research directions are described, as well as the main challenges to solve before reaching the ideal GKD system. In the last section, we conclude with the main spatial data warehouse challenges which remain to be solved to achieve the ideal GKD system.

2 Concepts and architectures of data warehouses

The present section provides a global synthesis of the actual state of data warehousing and of the related concepts of multidimensional databases, data marts, online analytical processing and data mining. Specialized terms such as legacy systems, granularity, facts, dimensions, measures, snowflake schema, star schema, fact constellation, hypercube and N-tiered architectures are also defined. This synthesis is based on the theoretical concepts found in the pioneering literature of the mid-1990s, but it also reflects the most recent trends found in the literature as well as our own experiences.

2.1 Data warehouse

An interesting paradox in the world of databases is that systems used for day-to-day operations store vast amounts of detailed information but yet are very inefficient for decision support and knowledge discovery. The systems used for day-to-day operations usually perform well for transaction processing where minimum redundancy and maximum integrity checking are key concepts, furthermore, this typically

takes place within a context where the systems process large quantities of transactions involving small chunks of detailed data. On the other hand, decision makers need fast answers made of a few aggregated data summarizing large units of work. something transactional systems do not achieve today with large databases. This difficulty of combining operational and decision-support databases within a single system gave rise to the dual-system approach typical of data warehouses.

Although the underlying ideas are not new, the term 'data warehouse' originated ten years ago and rapidly became an explicit concept recognized by the community. It has been defined very similarly by pioneers such as Brackett (1996), Gill and Rao (1996), Inmon et al (1996) and Poe (1995). In general, a data warehouse is an enterprise-oriented, integrated, non-volatile and read-only collection of data imported from heterogeneous sources and stored at several levels of detail to support decision-making. To facilitate the understanding of this definition, let us take each of these characteristics separately:

Enterprise-oriented As explained in the first section of this chapter, one of the aims of data warehouses is to become the single and homogeneous source for the data which are of interest to make enterprise-level strategic decision-making. Usually, no such homogeneous database exists since system development tends to happen in a bottom-up manner within organizations, resulting in several disparate specialized systems. Nor does such a single source exist, since these are detailed data which are stored within operational systems while enterprise-level strategic decision-making requires summarized data, resulting in costly and time-consuming processing to get global information about an enterprise activities.

Integrated This crucial characteristic implies that the data imported from the different source systems must go through a series of transformations so that they evolve from heterogeneous semantics, constraints, formats and codings to a homogeneous result stored in the warehouse. This is the most difficult and time-consuming part of building the warehouse (Kim 1999). In a well-regulated application domain (e.g. accounting or finance), this is purely a technical achievement. However, in other fields of activities, severe incompatibilities may exist among sources, making it impossible to integrate certain data or severely affecting the quality of the result of the integration. To facilitate this integration process, warehousing technologies offer built-in functions. Such functions include semantics fusion/scission, identification matching, field reformatting, file merging/splitting, value recoding, constraints calibration, replacing missing values, measurement scales and units changing, updateness filtering, adaptive value calculation, detecting unforeseen or exceptional values, etc. There also exist third-party software which offer more advanced functions, they are called data cleansing, data scrubbing, data fusion or data integration tools. Adherence to standards and to interoperability concepts also helps minimize the integration problem.

Non-volatile The source systems usually contain only current or near-current data since their out-of-date data are replaced by new values and afterwards

destroyed or archived. On the other hand, warehouses do not replace out-of-date data, they keep these historic (also called 'time-variant') data in order to allow trends analysis and prediction over periods of time (a key component of strategic decision-making). Consequently, legacy data are said to be volatile since they are updated continuously (i.e. replaced by most recent values) while, on the other hand, warehouse data are non-volatile, that is, they are not replaced by new values; they are kept alone, with these new values. However, to be more precise, one can specify about non-volatile data that, 'once inserted, (it) cannot be changed, though (it) might be deleted' (Date 2000). Reasons to delete data are usually not of transactional nature but of enterprise-oriented nature such as the decision to keep only the data of the last five years, to remove the data of a division that has been sold by the enterprise, to remove the data of a region of the planet where the enterprise has stopped doing business, etc. Thus, a data warehouse can grow in size (or decrease on rare occasions) but never be rewritten.

Read-only The warehouses can import the needed data but they cannot alter the state of the source databases, making sure that the original data always rest within the source. Such a requirement is necessary for technical concerns (e.g. to avoid update loops and inconsistencies) but mandatory to minimize organizational concerns (where is the original data? who owns it? who can change it? do we still need the legacy system? etc.) Thus, by definition, data warehouses are not allowed to write back into the legacy systems. However, although a data warehouse is not an Online Transaction Processing (OLTP) system (a system oriented towards the entering, storing, updating, integrity checking, securing and simple querying of data), it can be built to enter directly new information which is of high value for strategic decision-making but which does not exist in legacy systems.

Heterogeneous sources As previously mentioned, the data warehouse is a new, additional system which does not aim at replacing, in a centralized approach, the existing operational systems (usually called 'legacy systems'). In fact, the implementation of a data warehouse is an attempt to get enterprise-level information while minimizing the impact on existing systems. Consequently, the data warehouse must obtain its data from various sources and massage these data until they provide the desired information. Usually, the data warehouse imports the raw data from the legacy systems of the organization, but it does not have to be limited to these in-house systems. In all cases, collecting metadata (i.e. data describing the integrated data and integration process) is necessary to provide the user knowledge about the lineage and quality of the result.

Several levels of detail (also called 'granularity') Decision-makers need to get the global picture, but when they see unexpected trends or variations, they need to drill down to get more details to discover the reason for these variations. For example, when sales drop in the company, one must find out if it is a general trend for all types of products, for all regions and for all stores or if this is for a given region, for a given store or for a specific category of products (e.g. sport

equipment). If it is for a specific category such as sport equipment, one may want to dig further and find out if it is for a certain brand of products since a specific week. Thus, in order to provide fast answers to such multi-level questions, the warehouse must aggregate and summarize data by brand, category, store, region, periods, etc. at different levels of generalization. One such hierarchy could be store-city-region-country, another hierarchy could be day-weeknumber-quarter-year with a parallel hierarchy date-month-year. The term 'database granularity' is frequently used to refer to this hierarchical concept. For example, average sales of individual salespeople is a fine-grained aggregation; average sales by department is coarser; and the sales of the whole company is the most coarse (i.e. a single number). The finest granularity refers to the lowest level of data aggregation to be stored in the database (Date 2000) or, in other words, the most detailed level of information. This may correspond to the imported source data or to a more generalized level if the source data have only served to calculate higher-level aggregations and summarizations before being discarded.

To support decision-making It is the sum of all the previous characteristics which make data warehouses the best source of information to support decision-making. Data warehouses provide the needed data stored in a structure which is built specifically to perform with global, homogeneous, multi-levels and multi-epochs queries from decision-makers. This allows for the use of new decision-support tools and new types of data queries, exploration and analyses which were too time consuming in the past.

The characteristics of data warehouses, in comparison to the usual transaction-oriented systems, are presented in Table 3. 1.

2.2 Multidimensional data structure

Data warehouses are often structured using the multidimensional paradigm. Such structure is preferred by decision-support tools which dig into the data warehouse (e.g. OLAP and data mining tools). The multidimensional paradigm is built to facilitate the navigation within the database, especially within its different levels of granularity. It does so with simple functions such as drill-down (i.e. go to finer granularity), drill-up (i.e. go to coarser granularity) and drill-across (i.e. show

Table 3.1 Legacy system vs data warehouse

<i>Legacy system</i>	<i>Data warehouse</i>
<ul style="list-style-type: none"> • Built for transactions • Original source • Detailed data • Application-oriented • Current data only • Normalized data structure • Run on DBMS. GIS, web servers, CAD 	<ul style="list-style-type: none"> • Built for analysis, decisions • Copy or read-only data • Aggregate/summary data • Enterprise-oriented • Current + historic data • Denormalized, redundant data structure • Run on Super RDBMS, MD-DBMS....

another information at the same level of granularity). The term 'multidimensional' results from the extension to N dimensions of the usual matrix representation where the dependent variable is a cell within a two-dimensional (2-D) space defined by two axes, one for each independent variable (e.g. purchases could be the cells while countries and years the axes, giving immediately in the matrix all the purchases per country per year).

The data models of the multidimensional paradigm are based on three fundamental concepts: facts, dimensions and measures. A measure (e.g. total cost) is the attribute of a fact (e.g. purchase), which represents the state of a situation with regards to the dimensions of interest (e.g. region, date, product). Thus, one can look at the measure of a fact for a desired combination of dimensions (e.g. purchases of \$25,000.000 for Canada + 1999 + skisuit) and say that the measure is the dependent variable while the dimensions are the independent variables. Such an approach is recognized to map more directly with the user's perceptions of his informations and activities (i.e. fixing, the independent variables first, then find what the dependent variable is), thus facilitating the exploration of the database. 'The major reason why multidimensional systems appear intuitive is because they do their business the way we do ours' (Thomsen 1997).

Each dimension has members, each member represents a position on the dimensional axis (e.g. January, February, March, ...). The members of a single dimension may be structured in a hierarchical manner (e.g. year subdivided into quarters, quarters subdivided into months, months subdivided into weeks, weeks subdivided into days), creating the different levels of granularity of information. Alternate hierarchies can also be defined for a same dimension (e.g. year-month-day vs year-quarter-week).

Such a multidimensional paradigm can be modelled using three data structures: the Star Schema, the Snowflake Schema and the Fact Constellation. A star schema contains one central fact table, where each dimension key is linked to one dimension table, a snowflake schema contains one central fact table, where a dimension key is linked to a set of normalized dimensional tables, whereas a fact constellation contains a set of fact tables, connected by some shared dimension tables. In comparison to traditional transactional database implementations, the multidimensional paradigm relies heavily on denormalization of the database (to optimize performance), thus redundancy is high and data volumes much bigger but response times are also much faster.

Since a data warehouse may consist of a good number of dimensions and each dimension may have multiple levels, there could be a very large number of intermediate aggregated data cubes (called cuboids) to be computed. Usually, only a selected set of higher-level cuboids will be computed as shown by Harinarayan *et al.* (1996). Methods have been developed for efficient computation of multidimensional multi-level aggregates, such as Agarwal *et al.* (1996). Several popular indexing structures, including bitmap index and join index structures have been developed for fast access of multidimensional databases. A good overview of implementation methods for multidimensional databases is given by Chaudhuri and Dayal (1997).

2.3 Data mart

The exact definition of data mart is still a controversy (Date 2000), however it is frequently defined as a specialized, subject-oriented, highly aggregated mini-warehouse. It is more restricted in scope than the warehouse and can be seen as a departmental or partial special-purpose warehouse usually dealing with coarser granularity. Several data marts can be created in an enterprise. Most of the time, it is built from a subset of the data warehouse. Figure 3.1 illustrates the distinction between legacy systems, data warehouses and data marts while Table 3.2 highlights the differences between data warehouses and data marts.

However, in face of the major technical and organizational challenge of building an enterprise-wide warehouse, one may be tempted to build subject-specific data marts without building the data warehouse first. This may, accelerate the feeding of the database and the delivery of partial decision-support solutions. This may solve temporary problems with small investments and minimum political struggle. But, there is a risk of seeing several data marts emerging throughout the organization and still having trouble in getting the global picture. It also is useless to say that

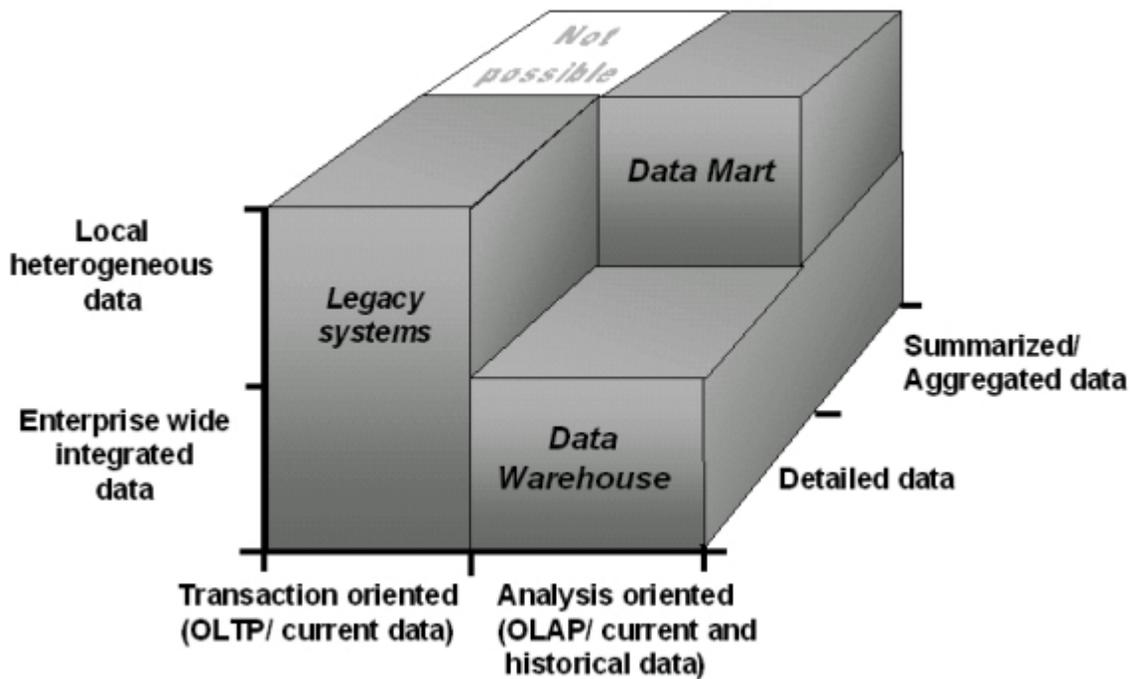


Figure 3.1 Comparison between legacy systems, data mart and data warehouses.

Table 3.2 Data Warehouse vs data mart

<i>Data warehouse</i>	<i>Data mart</i>
<ul style="list-style-type: none"> • Built for analysis • Aggregated/summary data • Enterprise-oriented • Denormalized, redundant data structure 	<ul style="list-style-type: none"> • Built for high-level analysis • Highly aggregated and summarized data • Subject-oriented • Highly denormalized, redundant data structure
<ul style="list-style-type: none"> • VLDB 	<ul style="list-style-type: none"> • Smaller DB

the old chaotic situation prevailing between legacy systems will undoubtedly arise between data marts. In spite of these problems and an unavoidable chaos, this alternative presents several short term advantages. Thus, it is frequently adopted and may sometimes be the only possible alternative.

2.4 Online analytical processing

OLAP is a very popular category of decision-support tools which are typically used as clients of the data warehouse (and of data marts). OLAP provides functions for the rapid, interactive and easy *ad hoc* exploration and analysis of data with a multidimensional user interface. Consequently, OLAP functions include the previously defined drill-down, drill-up, and drill-across functions as well as other navigational functions such as filtering, slicing, dicing, pivoting, etc. (see OLAP Council 1995; Thomsen 1997). Users may also be helped to focus on exceptions or locations which need special attention by methods which mark the interesting cells and paths. This kind of discovery-driven exploration of data has been studied by Sarawagi *et al.* (1998). Also, multi-feature databases which incorporate multiple, sophisticated aggregates can be constructed, as shown by Ross *et al.* (1998), to further facilitate data exploration and data mining.

In all cases, the user interface follows the multidimensional paradigm and offers several visualization capabilities (e.g. pie charts, histograms, bar charts). These capabilities include cross-tabs and three-dimensional (3-D) statistical charts, the latter one allowing the analysis of three dimensions of information at the same time (theoretically represented by cubes instead of matrices). As a result, it is common in the OLAP community to hear the term 'data cube' or 'hypercube' to describe the multidimensional paradigm.

There are several possibilities in building OLAP-capable systems. Each OLAP client can read directly from the warehouse and be used as a simple data exploration tool, or it can have its own data server. Such an OLAP server may structure the data with the relational approach, the multidimensional approach or a combination of both (based on granularity levels and frequency uses of dimensions). These are then respectively called ROLAP (Relational OLAP), MOLAP (Multidimensional OLAP) and HOLAP (Hybrid OLAP) although one may argue that the distinction is not based on sound principles despite use in current practice. Their capability is usually limited to much smaller databases than data warehouses (or even data marts) and they are not used as servers of other knowledge discovery packages like query builders, report builders, executive information systems and data mining.

2.5 Data mining

Another popular client of the data warehouses server is a category of software packages called data mining. This category of knowledge discovery software uses different techniques such as neural network, decision trees, genetic algorithms, rule induction and nearest neighbour to automatically discover hidden patterns or trends in large databases and to make predictions (see Berson and Smith (1997) or

Han and Kamber (2000) for a description of popular techniques). One must focus here on 'large' databases since data mining really shines where it would be too tedious and complex for a human being to use OLAP for the manual exploration of data. In fact, we use data mining to fully harness the power of the computer and of specialized algorithms to help us discover meaningful patterns or trends that would have taken months to find, or that we would have never found because of the large volume of these data and of the complexity of the rules which govern their correlations.

With regard to the data warehouse, it is important to keep in mind that in order to make data mining really work, the warehouse must keep data hierarchies down to a fine level of granularity. 'There must be sufficient types and amount of data in a database, before a mere mortal data mining software can discover any useful pattern' (Kim 1997).

2.6 *Data warehouse architectures*

Data warehouses can be implemented with different architectures depending on technological and organizational needs and constraints. The most typical one is also the simplest: it is called the 'corporated architecture' (Weldon 1997) or the 'generic architecture' (Poe 1995). It is represented in Figure 3.2. In such an architecture, the warehouse imports and integrates the desired data directly from the heterogeneous source systems, stores the resulting homogeneous enterprise-wide aggregated/summarized data in its own server, and lets the clients access these data with their own knowledge discovery software package (e.g. OLAP, data mining, query builder, report generator, executive information system). This two-tiered client-server architecture is the most centralized architecture.

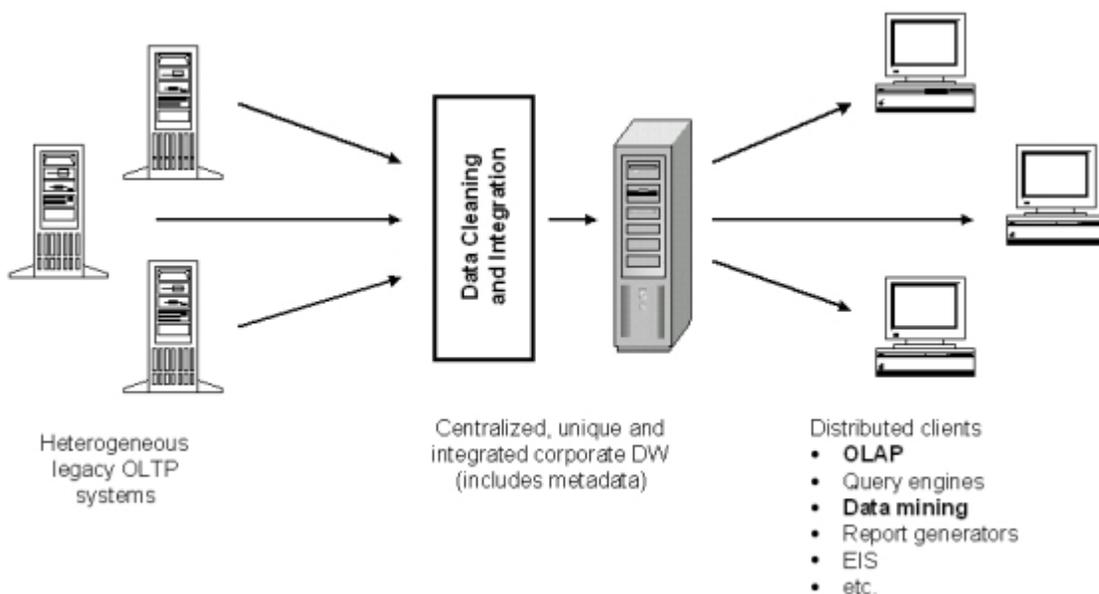


Figure 3.2 Generic architecture of a data warehouse.

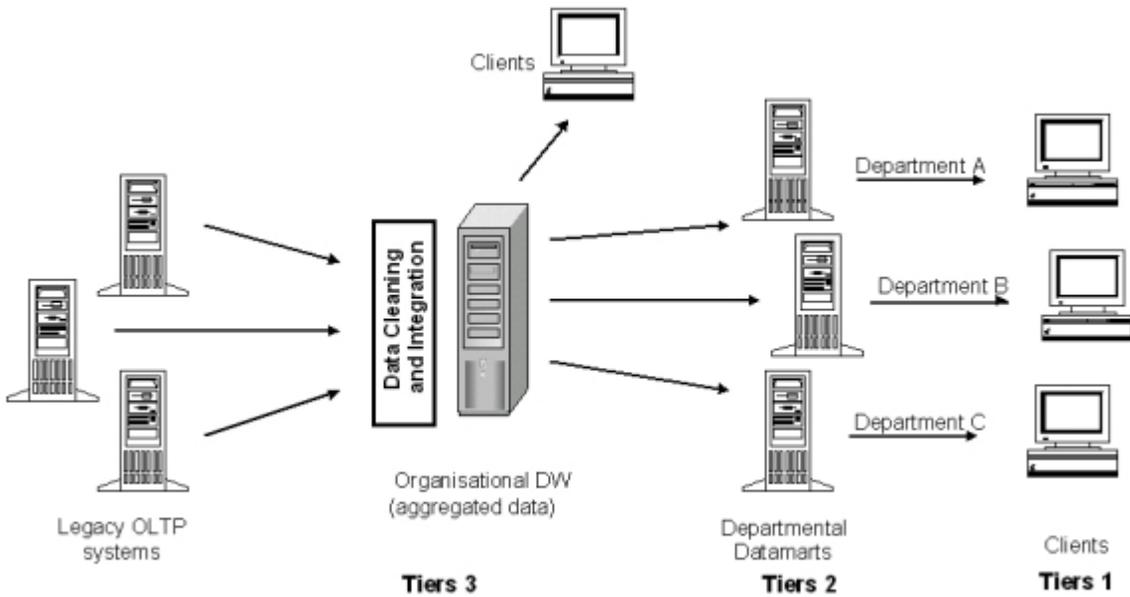


Figure 3.3 Standard federated three-tiered architecture of a data warehouse.

There is a frequently-used alternative which is called 'federated architecture'. It is a partly-decentralized solution and is presented in Figure 3.3. In this example, data are aggregated in the warehouse and other aggregations (at the same or a coarser level of granularity) are implemented in the data marts. This is a standard three-tiered architecture for data warehouses.

While the original concept of data warehouse suggests that its granularity is very coarse in comparison with that for transaction systems, some organizations decide to keep the integrated detailed data in the warehouses in addition to generating aggregated data. In some cases, for example in the four-tiered or multi-tiered architecture shown in Figure 3.4, two distinct warehouses exist. The first one stores the integrated data at the granularity level of the source data, while the second warehouse aggregates these data to facilitate data analysis. Such an architecture is particularly useful when the fusion of detailed source data represents an important effort and the resulting homogeneous detailed database may have a value of its own besides feeding the second warehouse.

Many more alternatives exist for designing warehouse architectures such as the data mart architecture without a data warehouse (Figure 3.5) or the many variations of the multi-tiered architectures. The number of alternatives is multiplied by the possibility offered by some software packages of building virtual data warehouses. In this latter case, integration of data is performed on-the-fly and not stored persistently, resulting in slower response times. Nevertheless, this is sometimes used for data marts or very small warehouses. (See Table 3.3 for a comparison between physical and virtual data warehouses.)

Finally, a major consideration remains: in spite of remarkable advances in computing power, parallel processing and indexing methods, it appears that today's approach to keep day-to-day transactional databases separate from decision-support databases will remain the rule for some time. One has to remember that

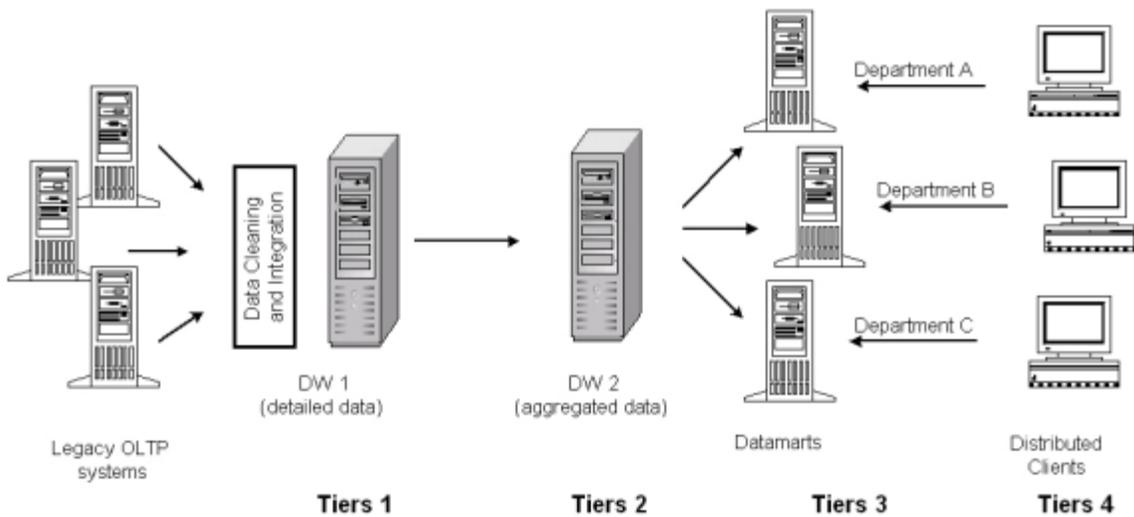


Figure 3.4 Multi-tiered architecture of a data warehouse.

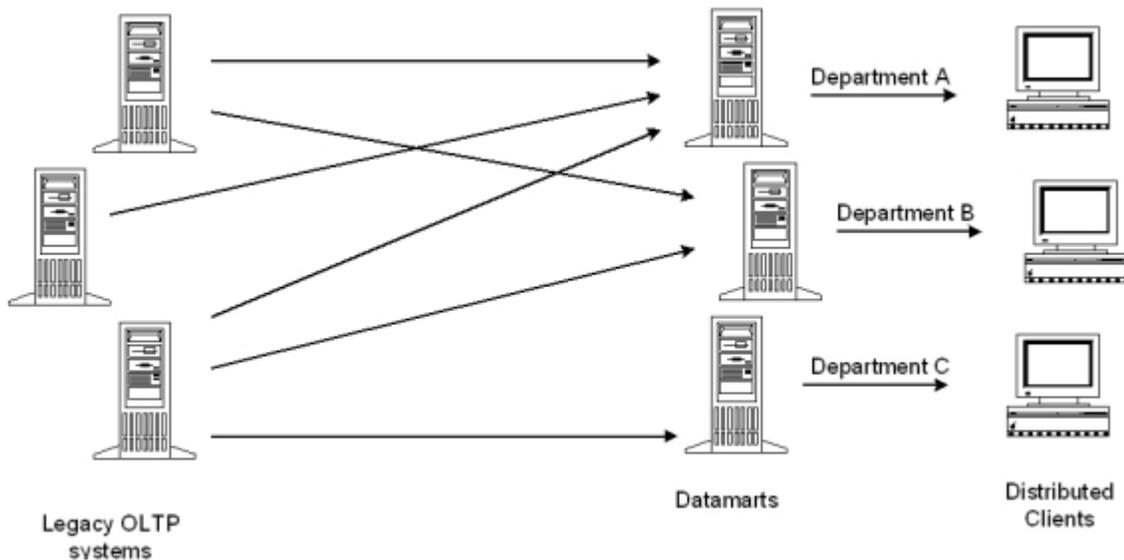


Figure 3.5 Data mart architecture without a data warehouse.

Table 3.3 Physical vs virtual data warehouses

<i>Physical</i>	<i>Virtual</i>
<ul style="list-style-type: none"> • Persistent data • A priori integration • All data integrated • Requires warehouse specific DBMS • Faster response • OK for large DB 	<ul style="list-style-type: none"> • No persistent data • On-the-fly integration • Integrate as required • Requires no DW specific DBMS • Slower response • Not for large DB

warehouses are strongly denormalized databases built to support several levels of aggregated data, thus rapidly resulting in large databases. It is not uncommon to see warehouses over 500GB or warehouses growing by as much as 50 per cent per year (Date 2000). We can nevertheless expect to see operational

and decision-support activities being integrated in a distant future given sufficient computing resources, improved interoperability and the desire to eliminate some problems typical to the dual-system approach (e.g. data copying, data transformation, update inconsistencies and delays, database volumes).

3 Spatial data warehousing

Applying warehousing and knowledge discovery concepts to spatial data yields interesting results, especially when spatial technology is coupled with non-spatial technology (e.g. coupling a GIS with a data warehouse). In addition, this type of solution meets certain needs of the geomatics community. However, it appears that, in spite of interesting results, numerous issues such as slow response times, unsatisfying map navigation capabilities, data fusion bottlenecks, etc. still need to be solved. In fact, today's GIS packages have been designed and used mostly for transaction processing and minimal analysis. It is recognized that GIS *per se* are not efficient for decision-support applications and that alternative tools must be used. However, no unique solution is ideal, and in most cases we must rely on a coupling of spatial and non-spatial technologies. Still, the result appears limited from a user point of view with regard to both functionalities and response time. The present section starts with basic concepts and then focuses on an overview of some of the most pressing requirements for efficient spatial data warehousing.

In spatial databases, semantic data and cartographic data used to be typically stored and processed by separate tools (e.g. a relational database management system and a GIS). Nowadays, with the arrival of universal servers, it has become efficient to store cartographic data and semantics data together and to perform basic analyses with the server. When one wants to perform more advanced analyses, specialized GIS modules are needed. Such solutions remain transaction-oriented and do not satisfy decision-support requirements. Additional concepts are needed for a multidimensional approach. Such basic concepts are presented in the following paragraphs.

When building a spatial data warehouse with the multidimensional paradigm, one may consider that, in addition to the usual semantics and temporal dimensions, there are three types of spatial dimensions (in the multidimensional sense, not the geometric sense) according to the theory of scale measurement (cf. nominal, ordinal, interval and ratio scales where each scale allows for richer analysis than its precedent one; see Chevallier and Bédard 1990; Chrisman 1997). Each type of dimension considers if it deals with a geometric spatial reference such as X, Y coordinates systems (i.e. quantitative data of the interval and ratio scales), with a semantic spatial reference such as place names (i.e. qualitative data of the nominal and ordinal scales), or with a combination of both, such as street addresses (which is a combination of quantitative and qualitative data where the quantitative data can be located precisely or interpolated along the linear axis identified by the qualitative data). The type of spatial reference system supported by the

warehousing/decision-support technology influences the type of spatial dimension one may use, or in other words, the type of hierarchy of a dimension:

1 Non-geometric spatial dimension This is a dimension containing only non-geometric data. For example, 'administrative units' can be constructed for the spatial warehouse as a dimension containing only nominal data to locate a phenomenon in space. Such a dimension could start with the names of municipalities, and its generalizations would also be non-geometric, such as counties and provinces. Such a solution can be implemented with non-spatial technology as long as cartographic representations and navigation are not required. The capacities and limitations of spatio-temporal analysis using a hypercube built only with non-geometric spatial dimensions has been demonstrated by Caron (1998).

2 Geometric-to-non-geometric spatial dimension This is a dimension whose primitive-level data is geometric but whose generalization, starting at a certain high level, becomes non-geometric. For example, a province represented by a polygon in the Canada map, that is, geometric data, is the finest granularity level of this spatial dimension. However, each province can be generalized to some value which is solely nominal, such as Pacific Canada, Atlantic Canada, etc. and its further generalization remains nominal, thus playing a similar role to a non-geometric dimension at coarser granularities of this spatial dimension. Using such design technique allows one to benefit from the simplification potential of the measurement scales, that is, qualitative measurements carry less details than quantitative measurements.

3 Fully geometric spatial dimension This is a dimension whose primitive level and all of its high-level generalizations are geometric. For example, polygons of equi-altitude regions are geometric data, and every generalization, such as regions covering 0-500 m, 500-1,000 m, and so on, are also geometric.

One may also notice that the last two types of spatial dimensions indicate that geometric data may have more than one way of being generalized to high-level concepts, and the generalized concepts can be geometric, such as maps representing larger regions, or non-geometric, such as named areas or general description of the region. These can be used as alternative ways to go from fine granularity to coarse granularity, even within the same spatial dimension (which we call a mixed spatial dimension), when software packages allow alternate paths as they presently do for semantics and temporal data.

We may also distinguish two types of measures (in the multidimensional sense) within a spatial data cube.

- 1 *Numerical measure* This is a measure containing only numerical data. For example, one measure in a spatial data warehouse could be monthly revenue of a region, and a roll-up may get the total revenue by year, by county, etc.
- 2 *Spatial measure* This is a measure which contains a collection of pointers to spatial objects. For example, during the generalization (or roll-up) in a spatial

data cube, the regions with the same range of temperature and altitude will be grouped into the same cell, and the measure so formed contains a collection of pointers to those regions.

There are at least three possible choices regarding the computation of spatial measures in spatial data cube construction:

- 1 Collect and store the corresponding spatial object pointers but do not perform precomputation of spatial measures in a spatial data cube: This can be implemented by storing, in the corresponding cube cell, a pointer to a collection of spatial objects (possibly represented by pointers). This choice indicates that the (region) merge of a group of spatial objects, when necessary, may have to be performed on-the-fly. It is still a good choice if only spatial display is required (i.e. no real spatial merge has to be performed), or if there are not so many regions to be merged in any pointer collection (thus online merge is not very costly). If the OLAP results are just for viewing, display-only mode could be useful. However, OLAP results can be used for further spatial analysis and spatial data mining, such as association, classification, etc. It is thus important to merge a number of spatially connected regions for such analysis.
- 2 Precompute and store some rough approximation/estimation of the spatial measures in a spatial data cube: This choice is good for a rough view or coarse estimation of spatial merge results under the assumption that it takes little storage space to store the coarse estimation result. For example, the Minimum Bounding Rectangle (MBR) of the spatial merge result (representable by two points) can be taken as a rough estimate of a merged region. Such a precomputed result is as small as a non-spatial measure and can be presented quickly to users. If higher precision is needed for specific cells, the application can either fetch precomputed high-quality results, if available, or compute them on-the-fly. Methods for efficient polygon amalgamation methods for region merge has been studied recently in Zhou *et al.* (1999).
- 3 Selectively precompute some spatial measures in a spatial data cube: The question is how to select a set of spatial measures for precomputation. The selection can be performed at the cuboid level, that is, either precompute and store each set of mergeable spatial regions for each cell of a selected cuboid, or precompute none if the cuboid is not selected. Since a cuboid usually consists of a large number of spatial objects, it may involve precomputation and storage of a large number of mergeable spatial objects but some of them could be rarely used. Therefore, it is recommended that the selection be performed at a finer granularity level by examining each group of mergeable spatial objects in a cuboid to determine whether such a merge should be precomputed. Methods for selective precomputation of the mergeable spatial objects in spatial data cubes have been studied in Stefanovic *et al.* (2000). The experiments have shown that the best choice is to selectively precompute some aggregated spatial regions and then perform efficient online polygon amalgamation operations.

A typical application example of such a spatial data warehouse construction and online analytical processing is to do multidimensional analysis of regional weather patterns. For example, in British Columbia (B.C.) of Canada, there are about 3,000 weather probes distributed in different small regions of the province. Each probe records temperature, precipitation, wind velocity, and many other weather-related measures for a designated small area and transmits signals to a provincial weather data analysis centre. A user may like to view weather patterns on a map by month, by region, and may even like to drill-down or roll-up dynamically along any dimension to explore desired patterns, such as wet and hot regions in the Fraser Valley in July, 1997. This creates scenarios which require many consolidated data cuboids or fast spatial aggregation computation for online flexible multidimensional analysis.

In a spatial warehouse, both dimensions and measures may contain spatial components. These considerations have immediate impact on the efficiency and usability of the warehouse. In spite of the interesting characteristics of the purely multidimensional structure, one must not reject the star/snowflake schema model as it is still considered to be a good choice providing a concise and organized warehouse structure easier to connect to a GIS.

However, since spatial warehouses may grow very rapidly in size, implementing a virtual warehouse seems inappropriate for most applications as human interventions are frequently required when extracting, cleansing, aggregating, generalizing and integrating the geometric data. In order to avoid repeating these costly efforts, it is a good practice to physically keep the results of these steps so that there is no need to repeat them, for example in a four-tiered architecture. Virtual warehousing for spatial data then becomes interesting only in a context of such architecture, or fully automatic interoperability or small databases (e.g. for spatial data marts).

In spite of all these possibilities, it rapidly becomes evident that integrating spatial data requires additional processing in comparison to non-spatial data. For example, one must make sure that each source map is topologically correct before integration and that it respects important spatial integrity constraints, that the overlay of these maps in the warehouse is also topologically correct (e.g. without slivers and gaps) and coherent with regard to updates, that the warehouse maps at the different scales of analysis are consistent especially with regard to spatial precision, that spatial reference systems are properly transformed to fit the warehouse spatial reference system, that the geometry of objects is appropriate for several levels of granularity, that we deal properly with fuzzy spatial boundaries, etc.

Recent experiments at Laval CRG in the building of spatial data warehouses in forestry, transportation, environmental health, digital libraries and avalanche prevention led us to realize that there are major problems in cleaning and integrating spatial data from different sources and epochs. Trade-offs have to be made and different types of decision-support analysis have to be left out because basic spatial units have been redefined over time, because historical data have not been kept, because data semantics and codification have been modified over time and are not directly comparable, because legacy systems are not documented according to good software engineering practices, because spatial reference systems have

changed, because of the fuzziness in spatial boundaries of certain natural phenomena which are remeasured at different epochs, because the spatial precision of measuring technologies has changed, and so on. One may wonder if building the theoretical multi-source, multi-scale and multi-epoch spatial data warehouse is feasible.

One typical example of such problems is presented by Rebut (1998). This experiment at Laval University Forêt Montmorency (70 km North of Quebec City) shows how the need for information may evolve rapidly over a period of twenty years and seriously affect the possibility of successfully integrating temporal data for natural phenomena. In this specific case, forest management practices have evolved dramatically during this period to better include environmental considerations. As a result, only, 12 per cent of today's data remained defined exactly the same way over the twenty-year period. The remaining 88 per cent were redefined, recoded or completely new types of data. The basic spatial units were also redefined and their delineation was remeasured completely from aerial photographs for every map production (1973, 1984, 1992). Thus, the initial steps of spatial data fusion were very labour-intensive and have partial results. In spite of these efforts, certain temporal analyses cannot be executed, limiting the usefulness of the data warehouse. However, one must say that these are non-technical constraints and that it is impossible to do better. The problem is not with data warehousing, OLAP or data mining, but with the very nature of data which changed over this period of time, both semantically and spatially.

We still believe that in certain cases, good data warehousing remains possible, for example with administrative data which are highly, regulated and which are not redefined every five to ten years (e.g. cadastre, property assessment) or with data which have always been collected according to strictly defined procedures of known quality (e.g. topographic databases). With such datasets, we assume that the problems are minimal. However, with databases about natural phenomena or with databases which do not keep track of historical data, we must live with major limitations such as a non-temporal data warehouse (which is not strictly a data warehouse), or a semi-temporal data warehouse (historical data exist for given epochs but are not comparable over time), or non-matching maps of different scales, epochs and themes, or comparing data of unknown quality, etc. It is our experience that about 80 per cent of the efforts to build such databases to support GKD goes to building the spatial data warehouse, and that in spite of such efforts, the result is not as elegant as dreamed of (although it remains very useful for what it does, especially if temporal data are not needed).

In spite of such a situation, we can expect important improvements in the near future as the research community is working hard on better understanding the requirements of efficient spatial data warehousing and knowledge discovery. Both academia and industry are developing solutions, and innovative products have recently appeared on the market. The last section of this chapter will present the research issues still pending for efficient spatial data warehousing supporting knowledge discovery.

4 Discussion and conclusion

We have presented a unified overview of the fundamental concepts of spatial data warehousing in the context of GKD. In general,

- Data warehouses are an important new class of information repository which meet the need of data integration from dispersed heterogeneous databases and facilitates data exploration and information discovery for strategic decision-making.
- As a strategically important information system, the spatial data warehouse will play an important role in the new generation spatial information systems since it provides a unified view of integrated spatial data from heterogeneous spatial databases, stores cleansed, integrated and transformed data, and facilitates multiple dimensional spatial data analysis and GKD.
- Spatial OLAP is a highly desirable data exploration facility in spatial data warehouse. It provides fast, flexible, and multidimensional ways for spatial data analysis. However, it also poses great challenges to efficient implementation of such mechanisms in large spatial databases. Recent research has made good progress along this direction.

As an emerging field, spatial data warehousing has also posed great challenges, especially in the context of GKD. Here we identify a list of R&D issues on construction and improvement of spatial data warehousing for effective and efficient GKD. Some of these issues have been identified by Bédard *et al* (1997).

Spatial data interoperability, integration and processing Automatically selecting the best data available for a given region regardless of the legacy system they are stored in, automatically integrating the heterogeneous data solely by using the proper metadata, automatically checking the spatial integrity constraints among these previously disparate data, and automatically generalizing data or building multiple representations to produce multi-scale warehouses, are examples of issues still not completely solved.

Adapting needs analysis and system design methodologies Better planning of the exact operations needed to build the warehouse, and better identification of the limitations of the expected result prior to its implementation are necessary to save time, costs and frustrations. Traditional system design methods have been developed for transaction-oriented systems and for non-spatial systems, not for spatial data warehouses and GKD. Although R&D has explored this latter direction in recent years, we must still adapt these methods for analytical systems and spatio-temporal databases in order to improve the quality of the results and to accelerate the development process.

Metadata management Describing spatial data fusion operations using metadata, increasing user awareness of the quality of the integrated information by supplying

proper metadata, balancing the quantity of metadata with regard to the volume of spatial data, properly attaching metadata to geometric primitives, exploiting metadata properly to facilitate the importation of updated legacy data, etc.

Handling very large spatial databases The addition of geometric data, historical data, metadata and aggregate data rapidly expand the size of spatial data warehouses, thus creating very strong needs for spatio-temporal indexing and partitioning methods, query optimizers, efficient spatio-temporal topological operators, excellent data update mechanisms, etc.

Scalability, of spatial knowledge discovery As mentioned previously, the size of the spatial data warehouse increases rapidly. Consequently, the ability to keep the same performance independently of the size of the data base is important. Efficient spatial data mining methods must be developed to cope with the huge size of spatial data, the nature of incremental updates of spatial data warehouses, and different granularities of spatial data mining requirements. A specific effort must be made to deal efficiently with cascading spatio-temporal updates for multi-scale maps.

Query building and data navigation Spatio-temporal querying is complex and today's user interface does not keep up with this complexity. In addition, multi-dimensional cartographic navigation operators are needed if one needs to benefit fully from the multi-scale multi-theme data structure of spatial data warehouses.

Web-based spatial data warehouse and spatial data mining With the rapid progress of web technology and increasing popularity of e-business, it is expected that there will be increasing requirements for integrating spatial data warehouses and spatial data mining with the web technology. It is likely that the future generations of spatial data warehouses and spatial data mining will be web-based. XML technology will be ready for future spatial data and information exchange. Therefore, it is important to investigate how to develop web-based spatial data warehouses and GKD for future generations of spatial information systems.

Acknowledgements

The authors are supported in part by Natural Science and Engineering Research Council of Canada, and GEOIDE Networks of Centres of Excellence, and the third authors is also supported by NCE-IRIS (institute of Robotics and Intelligent Systems).

Bibliography

Agarwal, S., Agrawal, R., Deshpande, P. M., Gupta, A., Naughton, J. F., Ramakrishnan, R. and Sarawagi, S. (1996) 'On the computation of multidimensional aggregates'. in *Proceedings of the 1996 International Conference on Very Large Data Bases*. Bombay, India: 506-21.

- Bédard, Y. (1999) 'Principles of spatial database analysis and design', Chap. 29. in: Paul A. Longley, Michael F. Goodchild, David J. Maguire and David W. Rhind (eds) *Geographical Information Systems: Principles, Techniques, Application and Managements*. 2nd edn, New York: Wiley, 413-24.
- Bédard, Y., Larrivée, S., Proulx, M.-J., Caron, P.-Y. and Létourneau, F. (1997) Étude de l'état actuel et des besoins de R&D relativement aux architectures et technologies des data warehouses appliquées aux données spatiales, *Research report for the Canadian Defense Research Center in Valcartier* Centre for Research in Geomatics, Laval University, Quebec City: 94 pages.
- Berson, A and Smith, S. J. (1997) *Data Warehousing, Data Mining & OLAP*. McGraw-Hill. 612.
- Brackett, M. H. (1996) *The Data Warehouse Challenge: Taming Data Chaos*. John Wiley & Sons, 579 pages.
- Caron, P. Y. (1998) *L'étude du potentiel de OLAP pour supporter l'analyse spatio-temporelle*. M.Sc. thesis, Centre for Research in Geomatics, Laval University, Quebec City, Canada. 132 pages.
- Chaudhuri, S. and Dayal, U. (1997) 'An overview of data warehousing and OLAP technology'. *ACM SIGMOD Record*, 26: 65-74.
- Chevallier, J.-J. and Bédard, Y. (1990) *Classification des types de références spatiales utilisées dans les systèmes d'information à référence spatiale (SIRS)*, Mensuration. Photogrammétrie et Génie rural. Switzerland, 638-42.
- Chrisman, N. (1997) *Exploring Geographic Information Systems*. John Wiley & Sons, 320.
- Codd, E. F. (1993) 'Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate', *Technical Report*. E. F. Codd and Associates.
- Date, C. J. (2000) *An Introduction to Database Systems*. Seventh Edition. Addison-Wesley 938 pages.
- Ester, M., Kriegel, H.-P. and Sander, J. (1997) 'Spatial data mining: A database approach'. in *Proc. Int. Symp. Large Spatial Databases (SSD'97)*. Berlin, Germany: 47-66.
- Fayyad, U. M., Piatetsky-Shapiro, G., Smyth. P. and Uthurusamy, R. (1996) (eds) *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press.
- Gill, S. H. and Rao, P. C. (1996) *The Official Client/Server Computing Guide to Data Warehouse*. QUE Corporation, 382 pages.
- Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart. D., Venkatrao, M., Pellow. F. and Pirahesh, H. (1997) 'Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals'. *Data Mining and Knowledge Discovery. 1:* 29-54.
- Han, J. and Kamber, M. (2000) *Data Mining: Concepts and Techniques*. Morgan Kaufmann.
- Han, J., Nishio, S., Kawano. H. and Wang. W. (1998)'Generalization-based data mining in object-oriented databases using an object-cube model'. *Data and Knowledge Engineering*, 25(1-2): 55-97.
- Harinarayan, V., Rajaraman, A. and Ullman, J. D. (1996) 'Implementing data cubes efficiently', in *Proceedings of the 1996 ACM-SIGMOD International Conference on Management of Data*. Montreal, Canada: June, 205-16.
- Inmon, W. H., Richard, D. and Hackathorn, D. (1996) *Using the Data Warehouse*. John Wiley & Sons, 285.
- Inmon, W. H. (1997) *Building the Data Warehouse*. John Wiley & Sons, 410 pages.
- Kim, W. (1997) 'OLTP versus DSS/OLAP/data mining', *Journal of Object-Oriented Programming*. SIGS Publications, November-December. 68-77.
- Kim, W. (1999) 'I/O problems in preparing data for data warehousing and data mining'. Part I. *Journal of Object-Oriented Programming*. SIGS Publications, 13-7.

- OLAP Council (1995) *OLAP Council White Paper*. <http://www.olapcouncil.org>, January, 4 pages.
- Poe, V. (1995) *Building a Data Warehouse for Decision Support*. Prentice Hall.
- Rawlings, J. and Kucera, H. (1997) 'Trials and tribulations of implementing a spatial data warehouse', in *Proceedings of the 11th Annual Symposium on Geographical Information Systems*, Vancouver: 510-513.
- Rebout, C. (1998) *Adaptation d'une base de données pour une application SOLAP (Spatial OLAP) pour l'aide à l'aménagement intégré des ressources forestières*. Thesis made at Laval University Centre for Research in Geomatics for the DESS diploma, Joseph-Fourier University, Grenoble, France, 72 pages.
- Ross, K. A., Srivastava, D. and Chatziantoniou, D. (1998) 'Complex aggregation at multiple granularities', in *Proceedings of the International Conference of Extending Database Technology (EDBT'98)*. Valencia, Spain: 263-77.
- Sarawagi, S., Agrawal, R. and Megiddo, N. (1998) 'Discovery-driven exploration of OLAP data cubes' in *Proceedings of the International Conference of Extending Database Technology (EDBT'98)* Valencia. Spain: 168-82.
- Stefanovic, N., Han, J. and Koperski, K. (2000) 'Object-based selective materialization for efficient implementation of spatial data cubes'. *IEEE Transactions on Knowledge and Data Engineering* (to appear).
- Thomsen, E. (1997) *OLAP Solutions: Building Multidimensional Information Systems*. Wiley Computer Pub.
- Weldon, J. L. (1997) 'State of the Art: Warehouse Cornerstones', *Bvte*, January. pp. 87.
- Zhou, X., Truffet, D. and Han, J. (1999) 'Efficient polygon amalgamation methods for spatial olap and spatial data mining', in *Proceedings of the 6th International Symposium on Large Spatial Databases (SSD'99)*. Hong Kong: 167-87.