Chapter Title

Fundamentals of Spatial Data Warehousing for Geographic Knowledge Discovery

YVAN BÉDARD[1] and JIAWEI HAN[2]

[1] Canada NSEC Industrial Research Chair in Spatial Database for Decision Support, Centre for Research in Geomatics, Laval University, Quebec City, Canada

e-mail: yvan.bedard@scg.ulaval.ca

[2] Database and Information Systems Research Lab, Department of Computer Science, University of Illinois at Urbana-Champaign, USA

email: hanj@cs.uiuc.edu

Chapter Outline

1. Introduction

Recent years have witnessed major changes in the Geographic Information (GI) market, from interoperable technological offerings to national spatial data infrastructures, web-mapping services and mobile applications. The arrival of new major players such as Google, Microsoft, Nokia, or TomTom for instance has created tremendous new opportunities and geographic data have become ubiquitous. Thousands of systems are geo-enabled every week, including data warehouses. As a special type of databases, data warehouse aims at providing organizations with an integrated, homogeneous view of data covering a significant period of time in order to facilitate decision-making. Such a view typically involves data about geographic, administrative or political places, regions, or networks organized in hierarchies. Data warehouses are separated from transactional databases and are structured to facilitate data analysis. They are built with either a relational, object-oriented, multidimensional, or hybrid paradigm although it is with the two latter ones that they bring the most benefits. Data warehouses are designed 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. In fact, the evolution of spatial data warehouses fits within the general trends of mainstream Information Technologies (IT).

Data warehouses provide these much-needed unified, global and summarized views of the data dispersed into heterogeneous legacy databases over the years. Organizations invest millions of dollars to build such warehouses in order to efficiently feed the decision-support tools used for strategic decision making, such as dashboards, executive information systems, data mining, report makers, and OLAP (On-Line Analytical Processing). In fact, data warehouse emerged as the unifying solution to a series of individual circumstances impacting 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 need 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, incoherencies 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, Richard and Hackathorn 1996).
- Second, past experiences have shown that fully reengineering the existing systems 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

but does not seek to replace them. With this regard, data warehouses add value to existing legacy systems rather than attempting to replace them since the unified view of the warehouse is built from an exact or modified copy of the legacy data.

- Third, the data structure used today by most decision-support solutions adopt, 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 transactional technologies, 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 has resulted into the concept of data warehouse, that is an additional read-only database typically populated with analysis-oriented aggregated or summarized data obtained from the extraction, transformation, and loading (ETL) of the detailed transactional data imported from existing legacy systems. After the ETL process and the new structuring of the resulting data, one typically finds only aggregated data in the warehouse, not the imported detailed legacy data.
- Fourth, strategic decision-making requires not only different levels of aggregated and summarized data but also direct access to past data as well

as present and future data (when possible) to analyze trends over time or predictions. The multidimensional paradigm frequently used in data warehouses efficiently supports such needs.

- 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 different epochs, and finally automatic analysis capabilities searching for unexpected data patterns.

In other words, the needed solution must support the extraction of useful knowledge from 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.

The goal of the present chapter is to introduce fundamental concepts underlying spatial data warehousing. It includes four sections. After having presented the “raison d'être” of spatial data warehousing in the previous paragraphs of this section, we introduce key concepts of non-spatial data warehousing in the second section. The third section deals with the particularities of spatial data warehouses, which in fact are typically spatiotemporal. In the last section, we conclude and present future R&D challenges.

Section 2. Key concepts and architectures for 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, on-line 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.

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 few aggregated data summarizing large units of work, something transactional systems do not achieve today with large databases. This difficulty to combine 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 in the early nineties 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, Richard and Hackathorn (1996) and Poe (1995). In general, a data warehouse is an enterprise-oriented, integrated, non-volatile, read-only collection of data imported from heterogeneous sources and stored at several levels of detail to support decision-making. Since this definition has been loosely interpreted and implemented in several projects and, consequently, not always delivered the promised returns on investments, it is highly important to explain every key characteristics:

- Enterprise-oriented: 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. Similarly, such single source does not exist since the data stored within operational systems describe detailed transactions (*e.g.*, the amount of cash withdrawal by one person in a given ATM at a precise moment) while enterprise-level strategic decision-making requires summarized data (e.g., increase in monetary transactions by our clients in all our ATMs of the province for the last month), resulting in costly and time-consuming processing to get global information about an enterprise’s 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 coding into a set of homogeneous results stored in the warehouse. This is the most difficult and time-consuming part of building the warehouse. 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 different sources, or within the same source through several years due to semantics evolutions, making it impossible to integrate certain data or to produce high quality results. To facilitate this integration process, warehousing technologies offer ETL capabilities. Such ETL functions include semantics fusion/scission, identification matching, field reformatting, file merging/splitting, field merging/splitting, value recoding, constraints calibration, replacing missing values, measurement scales and units changing, updates filtering, adaptive value calculation, detecting unforeseen or exceptional values, smoothing noisy data, removing outliers, and applying integrity constraints to resolve inconsistencies. These ETL capabilities are sometimes called data cleansing, data scrubbing, data fusion, or data integration. Adherence to standards and to interoperability concepts helps minimizing the integration problem. Of particular interest is the so-called “data reduction” process where one produces a reduced volume of representative data that provides the same or similar analytical results than a complete warehouse would provide (Han and Kamber, 2006).

- Non-volatile: The transactional 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 keep these historic (also called “time-variant”) data 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, *i.e.*, they are not replaced by new values; they are kept for a long period along with the 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 to do business, *etc*. Thus, a data warehouse can grow in size (or decrease in rare occasions) but never be rewritten.
- Read-only: The warehouses can import the needed detailed data but they cannot alter the state of the source databases, making sure that the original data always rest within the sources. Such requirement is necessary for technical concerns (*e.g.*, to avoid update loops and inconsistencies) but is mandatory to minimize organizational concerns (such as “where is the

original data?" "who owns it?" "who can change it?" and "do we still need the legacy system?"). Thus, by definition, data warehouses are not allowed to write back into the legacy systems. However, although a data warehouse is conceptually not meant to act as an OLTP system (On-Line Transaction Processing system oriented towards the entering, storing, updating, integrity checking, securing, and simple querying of data), it is sometimes built to allow direct entry of new data 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 higher-level 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 processes) is necessary to provide the required knowledge about the lineage and quality of the result. Recently, the quality of the produced high-level analytical data has become one of the main concerns of warehouse users; consequently, metadata have become more important and recent projects have introduced risk management approaches in data warehouse design methods as well as automatic context-sensitive user warnings (see chapter written by Gervais *et al.* in this book).

- Several levels of detail (also called “granularity” or “abstraction” levels): 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 of these variations. For example, when sales drop in the company, one must find 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 that 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 could be day-weeknumber-quarter-year with a parallel hierarchy date-month-year. The term “granularity” is frequently used to refer to this hierarchical concept. For example, average sales of individual salesperson 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. Inversely, when talking about the most summarized levels, users sometimes talk about “indicators”, especially when the quality of the source data is low. Such indicators give an approximate view of the global picture which is often sufficient for decision-making purposes.

- Support of decision-making: It is the sum of all the previous characteristics that makes data warehouse the best source of information to support decision-making. Data warehouses provide the needed data stored in a structure which is built specifically to answer global, homogeneous, multi-level, and multi-epoch 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.

Table 3.1. Legacy system vs. data warehouse

Legacy Systems	Data Warehouse
• Built for transactions, day-to-day repetitive operations	• Built for analysis, decisions and exploratory operations
• Built for large number of simple queries using few records	• Built for ad hoc complex queries using millions of records
• Original data source with updates	• Exact or processed copy of original data, in a read-only mode
• Detailed data	• Aggregated/summary data
• Application-oriented	• Enterprise-oriented
• Current data	• Current + historic data
• Normalized data structure and built with the transactional paradigm/concepts	• Denormalized data structure and typically built with the multidimensional paradigm/concepts
• top performance for transactions	• top performance for analysis

2.2 Multidimensional data structure

Data warehouses are typically structured using the multidimensional paradigm. Such structure is preferred by decision-support tools which dig into the data warehouse (*e.g.*, OLAP, dashboards, and Data Mining tools). The multidimensional paradigm is built to facilitate the interactive navigation within the database, especially within its different levels of granularity, and to instantly obtain cross-tab information involving several themes of analysis (called *dimensions*) at multiple levels of granularity. It does so with simple functions such as drill-down (*i.e.,* go to finer granularity within a theme), drill-up (*i.e.,* go to coarser granularity) and drill-across (*i.e.,* show 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 dependant variable is a cell within a 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). In the literature, a multidimensional database is usually represented by a cube with three axes (since it is not possible to represent more dimensions), and accordingly the multidimensional database is usually called *data cube* (or hypercube when N > 3).

The data models of the multidimensional paradigm are based on three fundamental concepts: facts, dimensions, and measures (Rafanelli, 2003, Kimball and Ross 2002). A measure (*e.g.,* total cost, number of items) is the attribute of a fact (*e.g.,* sales), which represents the state of a situation with regards to the themes or dimensions of interest (*e.g.,* region, date, product). Thus, one can look at the measures of a fact for a given combination of dimensions (*e.g.,* sales of 123,244 items and $25000000 for Canada in 2006 for sport equipment) and say that a measure is the dependent variable while the dimensions are the independent variables. Such an approach appears to be cognitively more compatible with the users' perception, thus facilitating the exploration of the database (*i.e.,* selecting the independent variables first, then seeing what the dependent variable is). “The major reason why multidimensional systems appear intuitive is because they do their business the way we do ours.” (Thomsen 2002). One can simply define a multidimensional query by saying “I want to know this (a measure) with regards to that (the dimensions elements)”.

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. Alternative hierarchies can also be defined for the same dimension (*e.g.*, year-month-day vs. year-quarter-week). A hierarchy where every child member has only one parent member is called a strict hierarchy. A non-strict hierarchy has a M:N relationship between parent members and child members, leading to implementation strategies which create summarizability constraints. A hierarchy can be balanced or not; it is balanced when the number of aggregation levels remains the same whatever the members selected in the dimension.

Such a multidimensional paradigm can be modeled using three data structures: the Star Schema, the Snowflake Schema, and the Fact Constellation. A Star schema contains one central fact table made of the measures and of one foreign key per dimension to link the fact with the dimension's member's (cf. using the primary key of the dimension tables) which are stored in one table per dimension, independent of a member's hierarchical level; a Snowflake schema contains one central fact table similar to the Star fact table, but the fact table foreign keys are linked to normalized dimension (typically, one table per hierarchical level); whereas a Fact Constellation contains a set of fact tables, connected by some shared dimension tables. It is not uncommon to see hybrid schemas where some dimensions are normalized and others are not.

Since a data warehouse may consist of a good number of dimensions and each dimension may have multiple levels, there could be a large number of combinations of dimensions and levels, each forming an aggregated multidimensional “cube” (called *cuboids*). For example, a database with 10 dimensions, each having 5 levels of abstraction will have 6^{10} cuboids. Due to limited storage space, usually, only a selected set of higher level cuboids will be computed as shown by Harinarayan *et al.* (1996). There have been many methods developed for efficient computation of multidimensional multi-level aggregates, such as Agarwal *et al.* (1996), and Beyer and Ramakrishnan (1999). Moreover, if the database contains a large number of dimensions, it is difficult to precompute a substantial number of cuboids due the explosive number of cuboids. Methods have been developed for efficient high-dimensional OLAP, such as Li et al. (2004). Furthermore, several popular indexing structures, including bitmap index and join index structures have been developed for fast access of multidimensional databases, as shown in Chaudhuri and Dayal (1997). An overview of the computational methods for multidimensional databases is given by Han and Kamber (2006).

2.3 Data mart

It is frequent to define *data mart* 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 special-purpose sub-warehouse usually dealing with coarser granularity. Typically, the design of data marts relies more on users' analysis needs while a data warehouse relies more on available data. Several data marts can be created in an enterprise. Most of the time, it is built from a subset of the data warehouse, but it may also be built from an enterprise-wide transactional database or from several legacy systems. As opposed to a data warehouse, a data mart does not aim at providing the global picture of an organization. Within the same organization, it is common to see the content of several data marts overlapping. In fact, when an organization builds several data marts without a data warehouse, there is a risk of inconsistencies between data marts and of repeating, at the analytical level, the well-known chaotic problems resulting from silo databases at the transactional level. Figure 3.1 illustrates the distinctions between legacy systems, data warehouses, and data marts whereas Table 3.2 highlights the differences between data warehouses and data marts.

Table 3.2. Data warehouse vs. data mart

Data warehouse	**Data mart**
• Built for global analysis	• Built for more specific analysis
• Aggregated/summarized data	• Highly aggregated/summarized data
• Enterprise-oriented	• Subject-oriented
• One per organization	• Several within an organization
• Usually multidimensional data structure	• Always multidimensional data structure
• Very Large DB	• Large DB
• Typically populated from legacy systems	• Typically populated from warehouse

[Figure 3.1.]

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

In face of the major technical and organizational challenges regarding the building of enterprise-wide warehouses, one may be tempted to build subject-specific data marts without building a data warehouse. This solution involves smaller investments, faster return on investments and minimum political struggle. But, there is a long-term risk to see several data marts emerging throughout the organization and still have no solution to get the global organizational picture. Nevertheless, this alternative presents several short term advantages. Thus, it is frequently adopted and may sometimes be the only possible alternative.

2.4 On-line analytical processing

On-line analytical processing (OLAP) is a very popular category of decision-support tools which are typically used as clients of the data warehouse and 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, and pivoting. (see OLAP Council 1995, Thomsen 2002, Wrembel and Koncilia 2006). Users may also be helped by more advanced functions such as to focus on exceptions or locations which need special attention by methods which mark the interesting cells and paths. Such kind of discovery-driven exploration of data has been studied by Sarawagi, Agrawal and Megiddo (1998). Also, multi-feature databases which incorporate multiple, sophisticated aggregates can be constructed, as shown by Ross, Srivastava and Chatziantoniou (1998), to further facilitate data exploration and data mining.

OLAP technology provides a high-level user interface that applies the multidimensional paradigm not only to the selection of dimensions and levels within data cubes, but also to the way we navigate within the different forms of data visualization (Fayyad, Grinstein and Wierse 2001). Visualization capabilities include for instance 2D or 3D tables, pie charts, histograms, bar charts, scatter plots, quantile plots, and parallel coordinates where the user can navigate (*e.g.,* drill-down in a bar of a bar chart).

There are several possibilities to build OLAP-capable systems. Each OLAP client can be reading directly 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 of the uses of dimensions) (Imhoff, Galemmo and Geiger 2003). These are then respectively called ROLAP (relational OLAP), MOLAP (multidimensional OLAP) and HOLAP (hybrid OLAP) although most users do not have to care such distinctions since they are at the underlying implementation techniques Alternatively, one may use specialized SQL servers that support SQL queries over star/snowflake/constellation schemas (Han and Kamber 2006).

One may also encounter so-called “Dashboard” applications with capabilities that are similar to OLAP. Although a dashboard can use OLAP components, it is not restricted to present aggregated data from a data cube, it may also display data from transactional sources (*e.g.,* from a legacy system), web RSSs, streaming videos, ERP systems, sophisticated statistical packages, *etc.* Dashboards wrap different types of data from diverse sources and present them in very simple, pre-defined panoramas and short repetitive sequences of operations to access, day-after-day, the same decision-support data. Strongly influenced by performance management strategies such as balanced scorecards, they are typically used by high-level strategic decision-makers who rely on indicators characterizing the phenomena being analyzed. Being easier to use than OLAP, dashboards are not meant to be as flexible or as powerful as OLAP since they support decision processes that are more structured and predictable. They are very popular for top managers but are highly dependent on the proper choice of performance indicators.

2.5 Data mining

Another popular client of the data warehouses server is a category of software packages or built-in functions called Data Mining. This category of knowledge discovery tools uses different techniques such as neural network, decision trees, genetic algorithms, rule induction and nearest neighbor to automatically discover hidden patterns or trends in large databases and to make predictions (see Berson & Smith 1997 or Han & Kamber 2006 for a description of popular techniques). 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 or when there are possibilities to discover highly unexpected patterns. 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 data and of the complexity of the rules which govern their correlations. Data mining supports the discovery of new associations, classifications, or analytical models by presenting the results with numerical values or visualization tools. Consequently, the line between OLAP and Data Mining may seem blurred in some technological offerings, but one must keep in mind that Data Mining is algorithm-driven while OLAP is user-driven and that they are complementary tools. Kim (1997) compares OLTP with DSS, OLAP and Data Mining. A good direction for combining the strengths of OLAP and Data Mining is to study OLAM (On-Line Analytical Mining) methods where mining can be performed in an OLAP way, i.e., exploring knowledge associated with multidimensional cube spaces by drilling, dicing, pivoting, and other user-driven data-exploration functions (Han and Kamber 2006).

2.6 Data Warehouse architectures

Data warehouses can be implemented with different architectures depending on technological and organizational needs and constraints (Kimball and Ross 2002). The most typical one is also the simplest, 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, dashboards). This two-tiered client-server architecture is the most centralized architecture.

[Figure 3.2.]

Figure 3.2. Generic architecture of a data warehouse

There is a frequently-used alternative 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.

[Figure 3.3.]

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

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 4-tiered architecture shown in Figure 3.4, two distinct warehouses exist. The first stores the integrated data at the granularity level of the source data, while the second warehouse aggregates these data to facilitate data analysis. Such architecture is particularly useful when the fusion of detailed source data represents an important effort and that the resulting homogeneous detailed database may have a value of its own besides feeding the second warehouse.

[Figure 3.4.]

Figure 3.4. Multi-tiered architecture of a data warehouse

Many more alternatives exist such as the *no-warehouse architecture* (Figure 3.5) which may have two variations to support the data marts: with and without OLAP servers. Similarly, some variations of the previous architectures could also be made without an OLAP server. In this case, a standard DBMS supports the star/snowflake schemas and the OLAP client does the mapping between the relational implementation and the multidimensional view offered to the user. In the short term, it results into easier data cube insertion within the organization (such as no software acquisition, smaller learning curve) but on the longer term, it results into increased workloads when building and refreshing data cubes (such as workmanship cost, repetitive tasks). Short-term contingencies, used technologies, personnel expertise, cube refreshment frequencies, existing workloads and long-run objectives must be considered when building the warehouse architecture. Further variations exist when one takes into account the possibility of building virtual data warehouses. In this latter case, integration of data is performed on-the-fly and not stored persistently, which results in slower response times but smaller data cubes.

[Figure 3.5]

Figure 3.5. Data mart architecture without a data warehouse

Finally, we believe that data warehouse is a very useful infrastructure for data integration, data aggregation, and multidimensional online data analysis. The advance of new computer technology, parallel processing, and high-performance computing as well as the integration of data mining with data warehousing will make data warehouses more scalable and powerful at serving the need of large-scale, multidimensional data analysis.

Section 3. Spatial data warehousing

Spatially enabling data warehouses leads to richer analysis of the positions, shapes, extents, orientations, and geographic distributions of phenomena. Furthermore, maps facilitate the extraction of insights such as spatial relationships (adjacency, connectivity, inclusion, proximity, exclusion, overlay, etc.), spatial distributions (concentrated, scattered, grouped, regular, etc.) and spatial correlations (Bedard, Rivest and Proulx 2007). When we visualize maps displaying different regions, it becomes easier to compare; when we analyze different maps of the same region, it becomes easier to discover correlations; when we see maps from different epochs, it becomes easier to understand the evolution of a phenomena. Maps facilitate understanding of the structures and relationships contained within spatial datasets than tables and charts, and when we combine tables, charts and maps, we increase significantly our potential for geographic knowledge discovery. Maps are natural aids to make the data visible when the spatial distribution of phenomena does not correspond to predefined administrative boundaries. Maps are active instruments to support the end-users thinking process, leading to a more efficient knowledge discovery process (more alert brain, better visual rhythm, more global perception) (Bedard, Rivest and Proulx, 2007).

Today's GIS packages have been designed and used mostly for transaction processing. Consequently, GIS is not the most efficient solution for spatial data warehousing and strategic analytical needs of organizations. New solutions have been developed, in most cases they rely on a coupling of warehousing technologies such as

OLAP servers with spatial technologies such as GIS. Research started in the mid-90s in several universities such as Laval (Bedard *et al.* 1997, Caron 1998, Rivest, Bedard and Marchand 2001), Simon Fraser (Stefanovic 1997, Han, Stefanovic and Koperski 1998), and Minnesota (Shekhar *et al.* 2001) and nowadays, several researchers and practitioners have become active in spatial data warehousing, spatial OLAP, spatial data mining, and spatial dashboards. Several in-house prototypes have been developed and implemented in government and private organizations, and we have witnessed the arrival of commercial solutions on the market.

This coupling of geospatial technologies with data warehousing technologies has become more common. Some couplings are loose *(e.g.,* import-export-reformatting of data between GIS and OLAP), some are semi-tight (*e.g.,* OLAP-dominant Spatial OLAP, GIS-dominant Spatial OLAP) while others are tight (*e.g.,* fully integrated SOLAP technology) (Rivest *et al.* 2005, Han and Kamber 2006). See the other chapters in the present book and these recent publications for a description of these solutions: Rivest *et al.* 2005, Han and Kamber 2006, Damiani and Spaccapietra 2006, Bedard, Rivest and Proulx 2007, Malinowski and Zimanyi 2008. For the remaining of this chapter, we focus on some fundamental spatial extensions of warehousing concepts: spatial data cubes, spatial dimensions, spatial measures, spatial ETL (Extract, Transform, Load) and spatial OLAP operators (or spatial multidimensional operators).

3.1 Spatial data cubes

Spatial data cubes are data cubes where some dimension members or some facts are spatially referenced and can be represented on a map. There are two categories of spatial data cubes: feature-based and raster-based (Figure 3.6). Feature-based spatial data cubes include facts which correspond to discrete features having geometry (vectors or cells) or having no geometry (in which case dimensions members must have a vector-based or raster-based geometry). Such fact geometry may be specific to this fact (in which case it may be derived or not from the dimensions) or it may correspond to the geometry of a spatial member. Raster spatial data cubes are made of facts which correspond to regularly subdivided spaces of continuous phenomena, each fine-grained fact being represented by a cell and every fine-grained cell being a fact.

[Figure 3.6]

Figure 3.6. Feature-based and raster-based spatial data cubes (adapted from McHugh 2008)

Traditionally, transactional spatial databases consisted of separated thematic and cartographic data (*e.g.,* using a relational database management system and a GIS). Nowadays, it is frequent to have both thematic and cartographic data stored together in a universal server or spatial database engine. Similarly, spatial data cubes can use thematic and cartographic data that are separated into different datasets (*e.g.,* using an OLAP server and a GIS) or they can store natively cartographic data and offer built-in spatial aggregation/summarization operators. Insofar, practical spatial warehousing applications have been based on the coupling of spatial and non-spatial technologies as this is still the only solution commercially available.

3.2 Spatial dimensions

In addition to the usual thematic and temporal dimensions of a data cube, there are spatial dimensions (in the multidimensional sense, not the geometric sense) which can be of three types according to the theory of measurement scales (*cf.* qualitative = nominal and ordinal scales, quantitative = interval and ratio scales, each scale allowing for richer analysis than the precedent one). These three types of dimension are:

- *Non-geometric spatial dimension* contains only nominal or ordinal location of the dimension members, such as place names (*e.g.*, St-Lawrence River), street addresses (134 Main Street) or hierarchically structured boundaries (*e.g.*, Montreal → Quebec → Canada → North America). Neither shape, nor geometry, nor cartographic data is used. This is the only type of spatial dimension supported by non-spatial warehousing technology (*e.g.*, OLAP). The possibilities and limitations of such dimensions have been demonstrated by Caron (1998); they can only offer a fraction of the analytical richness of the other types of spatial dimensions (Bedard, Rivest and Proulx 2007).
- *Geometric spatial dimension* contains a vector-based cartographic representation for every member of every level of a dimension hierarchy to allow the cartographic visualization, spatial drilling or other spatial operation of the dimension members (Bédard, Rivest and Proulx 2007). For example, every city in North-America would be represented by a point, every province/state of Canada, USA or Mexico would be represented as polygons, every North-American country would also be represented as polygons, as well as North-America itself. Similarly, polygons could represent equi-altitude regions in British-Columbia, and

every generalization, such as regions covering 0-500 meters, 500-1000 meters, and so on, would also be represented by a polygonal geometry.

- *Mixed geometric spatial dimension* contains a cartographic representation for some members of the dimension, and nominal/ordinal locators for the other members. This can be for instance all the members of certain levels of a dimension hierarchy (e.g., a point for every city, a polygon for every province/state, but only names for the countries and for North-America, *i.e.*, no polygon for these latter two levels of the hierarchy). Then, the non-geometric levels can be the finest grained one (to reduce the map digitizing efforts), the most-aggregated ones (when users know exactly where they are), anywhere in between, in any number and in any sequence. A mixed spatial dimension can also contain a cartographic representation for only some members of the same hierarchy level (*e.g.*, all Canadian cities, but not all Mexican cities). The mixed spatial dimension offers some benefits of the geometric spatial dimension while suffering from some limitations of the non-geometric spatial dimension, all this at varying degrees depending on the type of mixes involved.

Furthermore, spatial dimensions relate to different ways to use geometry to represent a phenomenon: discrete feature-oriented topological vector data vs. continuous phenomena-oriented raster data (McHugh 2008). Depending on the type of geometry used, the users' potential to perform spatial analysis and geographic knowledge discovery changes significantly. As a result, users have the choice among seven categories of spatial dimensions as presented in Figure 3.7.

[Figure 3.7]

Figure 3.7. Raster-based and feature-based spatial data cubes with different examples of spatial dimensions

The four additional categories of spatial dimensions are presented hereafter:

- *Raster spatial dimension*: Every level of the dimension hierarchy uses the raster structure, the highest spatial resolution being used for the finest-grained level of the hierarchy. For instance, one could use 100km cells for North-America, 10km cells for countries, and 1km cells for province/states.
- *Hybrid spatial dimension*: Some levels of the dimension hierarchy use the raster structure while other levels use the vector structure. This can be for instance polygonal geometries for North-America and for countries while the raster structure is used for province/states. Inversely, this could be points for cities, polygonal geometries for provinces/states, 100km raster cells for countries and 1000km cells for North-America. All levels must be represented cartographically.
- *Mixed raster spatial dimension*: Such a dimension contains raster data for some members of the dimension and nominal/ordinal locators for the other members (*i.e.*, no geometry). This can be for instance all the members of certain levels of a dimension hierarchy (*e.g.*, cells for the province/state level, names for the country and North-America levels, *i.e.*, no cartographic representation for these latter two levels of the hierarchy).

Then, the same mixing possibilities exist as for the mixed geometric spatial dimension. A mixed raster spatial dimension can also contain raster cells for only some members of the same hierarchy level (*e.g.,* all Canadian provinces, but not all American states). The mixed raster dimension offers some benefits of the raster spatial dimension while suffering from some limitations of the non-geometric spatial dimension, all this at varying degrees depending on the type of mixes involved.

- *Mixed hybrid spatial dimension*: Such a dimension contains raster data for some members of the dimension, vector data for other members, and nominal/ordinal locators for the remaining ones (*i.e.,* no geometry). This can be for instance all the members of certain levels of a dimension hierarchy (*e.g.,* names for cities, polygons for the province/state level, raster cells for the country level and a name only for the North-America level, *i.e.,* no cartographic representation for the finest and most aggregated levels of the hierarchy). Then, the same types of mixing possibilities exist as for the mixed geometric spatial dimension, without restriction. A mixed hybrid spatial dimension can also contain raster cells for some members of the same hierarchy level, polygons for other members and no geometry for the remaining ones of this level (*e.g.,* all Canadian provinces using raster cells, all American states using polygons, and Mexican states using names). The mixed hybrid dimension offers some benefits of the raster and of the geometric spatial dimensions while suffering from some limitations of the non-geometric spatial dimension, all this at varying degrees depending on the type of mixes involved.

More than one spatial dimension may exist within a spatial data cube (see Figure 3.7).

3.3 Spatial measures

In addition to the non-spatial measures that still exist in a spatial data warehouse, we may distinguish three types of spatial measures (in the multidimensional sense):

- *Numerical spatial measure*: single value obtained from spatial data processing (*e.g.,* number of neighbors, spatial density). Such measure contains only a numerical data and is also called *non-geometric spatial measure*.
- *Geometric spatial measure*: set of coordinates or pointers to geometric primitives that results from a geometric operation such as spatial union, spatial merge, spatial intersection, or convex hull computation. For example, during the summarization (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.
- *Complete spatial measure*: combination of a numerical value and its associated geometry. For example, the number of epidemic clusters with their location.

3.4 Spatial ETL

In spite of all these possibilities, it rapidly becomes evident that integrating and aggregating/summarizing spatial data requires additional processing in comparison to non-spatial data. For example, one must make sure that each source dataset is topologically correct before integration and 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 regards to updates, that the warehouse maps at the different scales of analysis are consistent, that spatial reference systems and referencing methods are properly transformed, that the geometry of objects is appropriate for each level of granularity, that there is no mismatch problem between the semantic-driven abstraction levels of the dimension hierarchies and the cartographic generalization results (Bedard, Rivest and Proulx 2007), that we deal properly with fuzzy spatial boundaries, etc. Consequently, spatial ETL requires an expertise about the very nature of spatial referencing (*e.g.*, spatial reference systems and methods, georeferencing and imaging technologies, geoprocessing, and mapping) and one must not assume this process can be executed 100% automatically. Furthermore, there are issues related to the desire of users to clean and integrate spatial data from different 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 re-observed at different epochs, because the spatial precision of measuring technologies has changed; and so on (see Bernier and Bedard 2007 for problems with spatial data, and Kim 1999 for non-spatial data). One must realize that building and refreshing the multi-source, multi-scale and multi-epoch spatial data warehouse is feasible but requires efforts, strategic trade-offs and a high level of expertise. In some cases, properly dealing with metadata, quality information and context-sensitive users warnings becomes a necessity (Levesque *et al.* 2007).

Spatial ETL technologies are emerging. One may combine a warehouse-oriented non-spatial ETL tool or the built-in functions of OLAP servers with a spatial technology such as a GIS, an open-source spatial library or a commercial transaction-oriented spatial ETL. One may also look for fully integrated spatial ETL prototypes that are in development in research centers. This new category of spatial ETL tools will include spatial aggregation/summarization operators to facilitate the calculation of aggregated spatial measures.

In spite of these difficulties, it remains possible to develop simple spatial warehousing applications if one keeps the cartographic requirements at a reasonable level. Many applications are running today and succeeded to minimize the above issues. It is the case for example with administrative data which are highly regulated and which are not redefined every five to ten years (*e.g.,* cadastre, municipalities) or with data which have always been collected according to strictly defined procedures of known quality (*e.g.,* topographic databases). With such datasets, the problems are minimal. However, with databases about natural phenomena or with databases which do not keep track of historical data, we must face some of the above-mentioned issues and choose to develop non-temporal data warehouses, semi-temporal data warehouse (historical data exist for given epochs but the data are not comparable over time), warehouses displaying non-matching maps of different scales, and warehouses of varying data quality. It is our experience that a majority of the efforts to build spatial data warehouses goes to spatial

ETL, and that the quality of existing legacy spatial data has an important impact on the design and building of the warehouse.

3.5 Spatial OLAP operators

Spatial data cubes can be explored and analyzed using spatial OLAP operators. Spatial operations allow the navigation within the data cube with regards to the spatial dimensions while keeping the same level of thematic and temporal granularities (Bedard, Rivest and Proulx 2007). The SOLAP operators are executed directly on the maps and behave the same way as non-spatial operators. Basic operators include spatial drill-down, spatial roll-up, spatial drill-across, spatial slice and dice while the most advanced operators include spatial open, spatial close, views synchronization, etc. A detailed definition with examples is provided in (Rivest *et al.* 2005). A recent survey of the commercial technologies proposed to develop Spatial OLAP applications (Proulx, Rivest and Bedard 2007) has shown that only the most tightly integrated SOLAP technologies properly support the basic spatial drill operations; the loosely-coupled technologies rather use traditional “zoom” or “select layer” functions of the traditional transactional paradigm to simulate a change of abstraction level in a spatial dimension.

Section 4. Discussion and Conclusion

We have presented an overview of the fundamental concepts of spatial data warehousing in the context of Geographic Knowledge Discovery (GKD). Such spatial data warehouses have become important components of an organization infrastructure. They meet the needs of data integration and summarization from dispersed heterogeneous databases and facilitate interactive data exploration and geographic knowledge discovery for power analysts and strategic decision makers. Spatial data warehouses provide fast, flexible, and multidimensional ways to explore spatial data when using the appropriate client technology (*e.g.*, SOLAP, spatial dashboards). Several applications have been developed in many countries. However, as it is still the case with transactional geospatial systems, there remain challenges to populate efficiently such warehouses. Recent research is making significant progress along this direction as there are several universities, government agencies and private organizations now involved in spatial data warehousing, SOLAP, and spatial dashboard. Today's research issues include the followings:

- Spatial data cube interoperability (Sboui *et al.* 2007, Sboui *et al.* 2008);
- Spatially-enabling OLAP servers and ETL tools for spatial data cube (e.g., open-source GeoMondrian and GeoKettle projects by Dube, Badard and Bedard);
- Developing spatial aggregation/summarization operators for spatial data cubes;
- Improving spatial data cube and SOLAP design methods and modeling formalism;
- Developing web services for spatial warehousing (Badard *et al.* 2008);
- Quasi real-time spatial warehousing;
- Mobile wireless spatial warehousing and on-the-fly creation of spatial mini-cubes (Badard *et al.* 2008, Dube 2008)
- Formal method to select the best quality legacy sources and ETL processes;
- Formal methods and legal issues to properly manage the risk of warehouse data misuse (see the chapter by Gervais *et al.* in this book);

- Improving the coupling of spatial and non-spatial technologies for spatial warehousing;
- Improving the client tools that exploit the spatial warehouses;
- Enriching spatial integrity constraints for aggregative spatial measures;
- Improving spatial data mining approaches and methods;
- Improving the coupling between spatial data mining and SOLAP;
- Facilitating the automatic propagation of legacy data updates towards spatial data warehouses;
- Increasing the capacities of raster spatial data cubes for interactive SOLAP analysis (McHugh 2008);
- Integrating spatial data mining algorithms, such as spatial clustering, classification, spatial collation pattern mining, spatial outlier analysis methods, with spatial OLAP mechanisms;
- Handling high dimensional spatial data warehouse and spatial OLAP, where a data cube may contain a quite large number of dimensions on categorical data (such as regional sensus data) but other dimensions are spatial data (such as maps or polygons);
- Integrating data mining methods as preprocessing methods for constructing high quality data warehouses, where data integration, data cleansing, clustering, feature selection can be first performed by data mining methods before a data warehouse is constructed.
- Integrating space with time to handle spatial temporal data warehouses, sensor-based or RFID-based spatial data warehouses.

As it is the case with information technology in general, this field is evolving rapidly as new concepts are emerging, new experimentations successful and a larger community becomes involved. If one looks back five years ago, Googling "spatial data warehouse" or "SOLAP" didn't give hundreds of hits while nowadays, it is the case. The scientific community is adopting the datacube paradigm to exploit spatial data and the number of papers is rapidly increasing. In a similar manner, the industry has recently started to offer commercial solutions and their improvements will drive the more general adoption by users in a short term.

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 *Proc. 1996 Int. Conf. Very Large Data Bases*, pp. 506-521,Bombay, India, September.

BADARD, T., BEDARD, Y., HUBERT, F., BERNIER, E., AND E. DUBE, 2008. *Web Services Oriented Architectures for Mobile SOLAP Applications.* International Journal of Web Engineering and Technology (accepted).

BÉDARD Y., LARRIVÉE S., PROULX M-J, CARON P-Y, 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.

BÉDARD, Y., RIVEST, S., AND M.-J. PROULX, 2007. Spatial On-Line Analytical Processing (SOLAP): Concepts, Architectures and Solutions from a Geomatics Engineering Perspective. Chapter 13 in (Wrembel and Koncilia, 2007), pp. 298-319.

BERNIER, E., Y. BÉDARD, 2007, A Data Warehouse Strategy for on-Demand Multiscale Mapping. Chapter 9 in: Mackaness, W., Ruas, A. and Sarjakoski T. (Eds), *Generalisation of Geographic Information: Cartographic Modelling and Applications*, pp. 177-198.

BERSON A. AND SMITH S.J., 1997, *Data Warehousing, Data Mining & OLAP*. McGraw-Hill, 612 p.

BEYER, K., AND R. RAMAKRISHNAN, 1999. Bottom-Up Computation of Sparse and Iceberg Cubes. *Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99)*, Philadelphia, PA, pp. 359-370, June.

BRACKETT M.H., 1996, *The Data Warehouse Challenge: Taming Data Chaos*. John Wiley & Sons, 579 pages.

CARON, P.Y., 1998, *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.

DAMIANI M.L., AND SPACCAPIETRA, S., 2006. Spatial Data Warehouse Modelling. Chapter 1 in: *Processing and Managing Complex Data for Decision Support*. J. Darmont and O. Boussaid (Eds), Idea Group, pp. 1-27.

DATE C.J., 2000, *An Introduction to Database Systems*, Seventh Edition. Addison-Wesley, 938 pages.

DUBE, E., 2008. *Developpement d'un service web de constitution en temps reel de mini cubes SOLAP pour clients mobiles.* M.Sc. thesis (draft), Centre for Research in Geomatics, Dept. Geomatics Sciences, Laval University, Quebec City, Canada.

FAYYAD, U.M., G. GRINSTEIN AND A. WIERSE, 2001. *Information visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, 442 pages.

GILL S.H. and RAO P.C., 1996, *The Official Client/Server Computing Guide to Data Warehouse*. QUE Corporation, 382 pages.

HAN, J., STEFANOVIC, N. AND KOPERSKI, K., 1998. Selective Materialization: An Efficient Method for Spatial Data Cube Construction. *Proceedings of the Second Pacific-Asia Conference, PAKDD'98* (PP. 144-158).

HAN J. and KAMBER M., 2006, *Data Mining: Concepts and Techniques*, 2nd Edition. Morgan Kaufmann.

HARINARAYAN V., RAJARAMAN A., and ULLMAN J. D., 1996, Implementing Data Cubes Efficiently. In Proc. 1996 ACM-SIGMOD Int. *Conf. Management of Data*, pp. 205-216, Montreal, Canada, June.

IMHOFF, C., GALEMMO, N., AND J. G. GEIGER, 2003. *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. John Wiley, 456 pages.

INMON W.H., RICHARD D., and HACKATHORN D., 1996, *Using the Data Warehouse*. John Wiley & Sons, 285 p.

KIM W., 1997, OLTP versus DSS/OLAP/Data Mining. *Journal of Object-Oriented Programming,* SIGS Publications, Nov-Dec, pp. 68-77.

KIM W., 1999, I/O Problems in Preparing Data for Data Warehousing and Data Mining, Part 1. *Journal of Object-Oriented Programming*, SIGS Publications, February, pp. 13-17.

KIMBALL, R., AND ROSS, M., 2002. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition).* Wiley, 464 pages.

LEVESQUE, M.-A., Y. BÉDARD, M. GERVAIS, R. DEVILLERS, 2007, Towards a Safer Use of Spatial Datacubes: Communicating Warnings to Users, *Proceedings of the 5th International Symposium on Spatial Data Quality*, June 13-15, Enschede, Netherlands.

X. LI, J. HAN, AND H. GONZALEZ, 2004. HIGH-DIMENSIONAL OLAP: A MINIMAL CUBING APPROACH, *PROC. 2004 INT. CONF. VERY LARGE DATA BASES (VLDB'04)*, TORONTO, CANADA, PAGES 528-539, AUGUST.

MCHUGH, R. 2008. *Etude du potentiel de la structure matricielle pour optimizer l'analyse spatial de données géodécisionnelles.* M.Sc. thesis (draft version), Dept. Geomatics Sciences, Laval University, Quebec City, Canada. N.P.

MALINOWSKI, E., AND E. ZIMANYI, 2008. Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications. Springer. In press.

OLAP COUNCIL, 1995, *OLAP Council White Paper*. http://www.olapcouncil.org, 4 pages. January.

POE V., 1995, *Building a Data Warehouse for Decision Support*. Prentice Hall, 210 p.

PROULX, M., S. RIVEST, AND Y. BÉDARD, 2007. *Évaluation des produits commerciaux offrant des capacités combinées d'analyse multidimensionnelle et de cartographie*. Research report for the partners of the Canada NSERC Industrial

Research Chair in Geospatial Databases for Decision-Support. Centre for Research in Geomatics, Laval University, Quebec City, Canada, November, 65p.

RAFANELLI, M., 2003. *Multidimensional Databases: Problems and Solutions*. London: Idea Group Publishing. 478 pages.

RIVEST, S., BÉDARD, Y., AND P. MARCHAND, 2001. Towards Better Support for Spatial Decision-making: Defining the Characteristics of Spatial On-Line Analytical Processing. *Geomatica*, Journal of the Canadian Institute of Geomatics, Vol. 55, No. 4, pp. 539-555.

RIVEST, S., Y. BÉDARD, M.-J. PROULX, M. NADEAU, F. HUBERT AND J. PASTOR, 2005. SOLAP: Merging Business Intelligence with Geospatial Technology for Interactive Spatio-Temporal Exploration and Analysis of Data. *Journal of International Society for Photogrammetry and Remote Sensing* (ISPRS), Vol. 60, No. 1, pp. 17-33.

ROSS K. A., SRIVASTAVA D., and CHATZIANTONIOU D., 1998, Complex Aggregation at Multiple Granularities. Proc. Int. *Conf. of Extending Database Technology (EDBT'98*), Valencia, Spain, pp. 263-277, March.

SARAWAGI S., AGRAWAL R., and MEGIDDO N., 1998, Discovery-Driven Exploration of OLAP Data Cubes. Proc. Int. *Conf. of Extending Database Technology (EDBT'98).* Valencia, Spain, pp. 168-182, March.

SBOUI, T., Y. BÉDARD, J. BRODEUR, T. BADARD, 2007, A Conceptual Framework to Support Semantic Interoperability of Geospatial Datacubes. Proceedings of ER/2007 SeCoGIS workshop, November 5-9, Auckland, New Zealand, Lecture Notes in Computer Sciences, Springer, pp. 378-387.

SBOUI, T., Y. BÉDARD, J. BRODEUR, T. BADARD, 2008, Risk Management for the Simultaneous Use of Spatial Datacubes: a Semantic Interoperability Perspective. International Journal of Business Intelligence and Data Mining (submitted)

SHEKHAR, S., LU, C.T., TAN, X., CHAWLA, S., AND VATSAVAI, R., 2001. Map Cube: A Visualization Tool for Spatial Data Warehouses. Chapter in: Geographic Data Mining and Knowledge Discovery. H.J. Miller and J. Han (Eds), Taylor & Francis.

STEFANOVIC, N., 1997. *Design and Implementation of On-Line Analytical Processing (OLAP) of Spatial Data*. MSc thesis, Simon Fraser University, Canada.

THOMSEN, E., 2002. *OLAP Solutions: Building Multidimensional Information Systems (Second Edition)*. John Wiley & Sons. 688 pages.

WELDON J.L., 1997, State of the Art: Warehouse Cornerstones. *Byte*, pp.87, January.

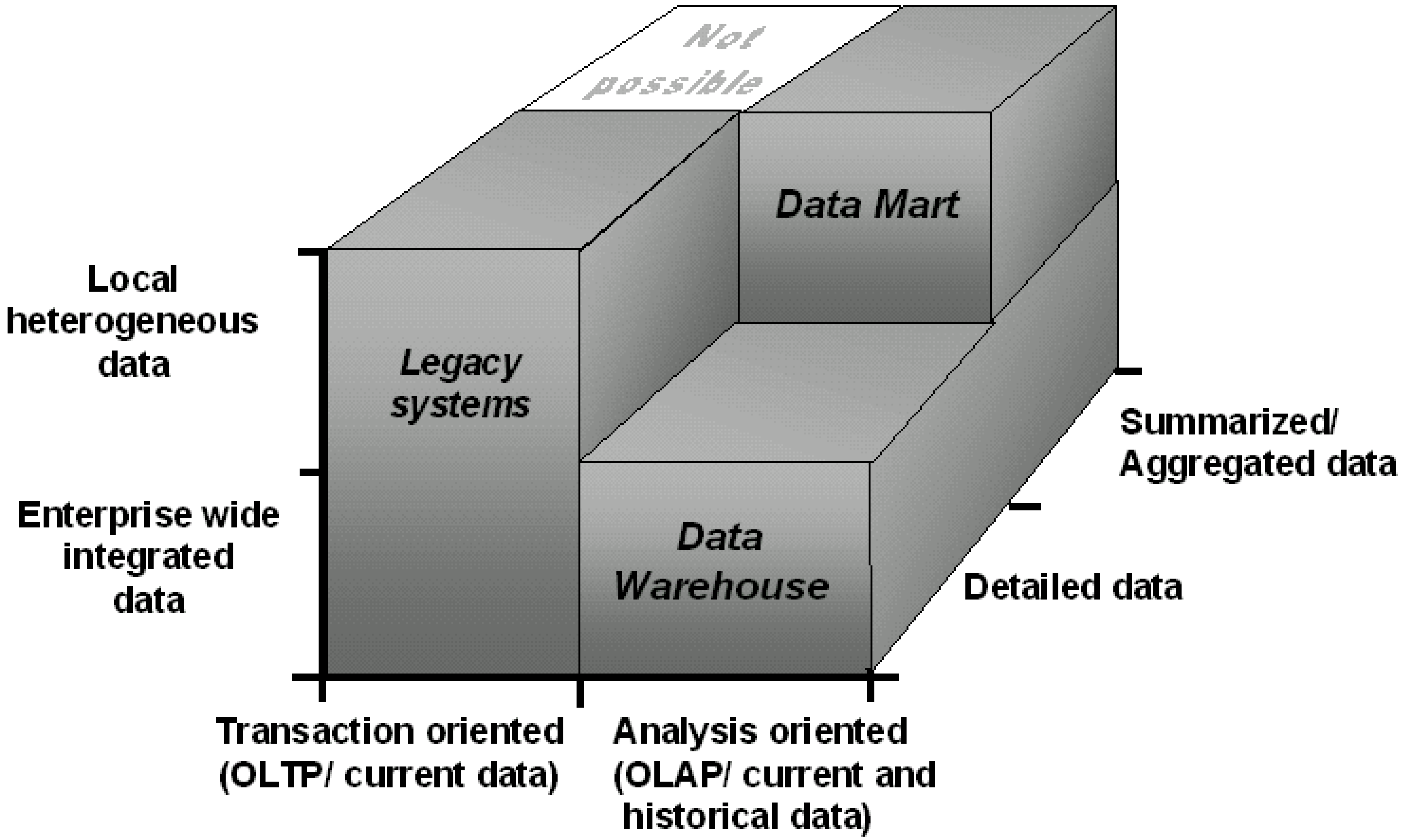
WREMBEL, R. AND C. KONCILIA, 2006. *Data Warehouses and OLAP: Concepts, Architectures and Solutions.* London: IRM Press (IDEA Group Publishing), London, UK, 361 pages.

ZHOU X., TRUFFET D., and HAN J., 1999, Efficient Polygon Amalgamation Methods for Spatial OLAP and Spatial Data Mining. Proc. *6th Int. Symp. on Large Spatial Databases (SSD'99)*, Hong Kong, pp. 167-187, July.

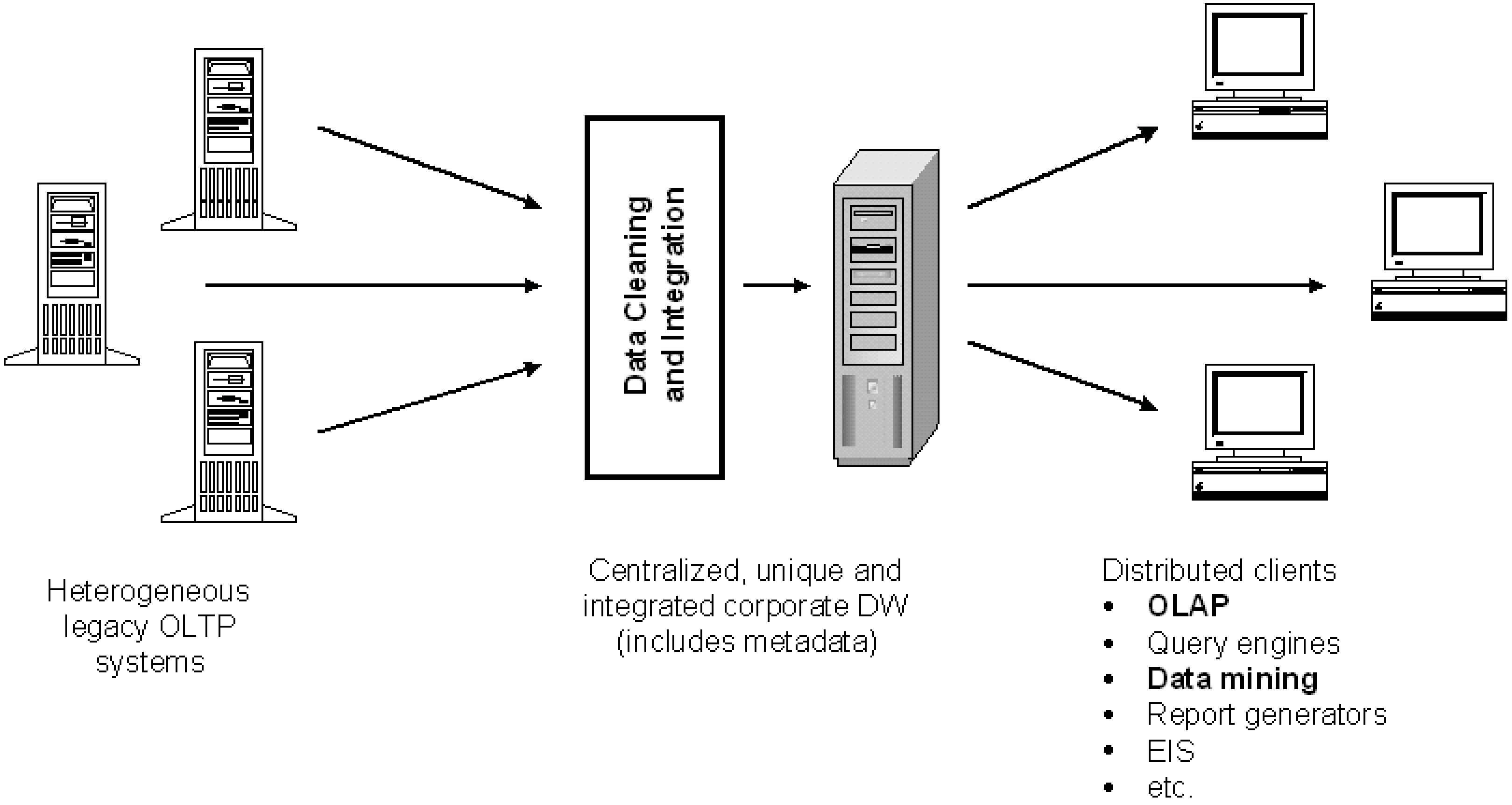
Acknowledgements

This research has been supported by Canada NSERC Industrial Research Chair in Spatial Databases for Decision-Support, by the Canadian GEOIDE Network of Centres of Excellence in Geomatics, and by the U.S. National Science Foundation NSF IIS-05-13678 and BDI-05-15813.

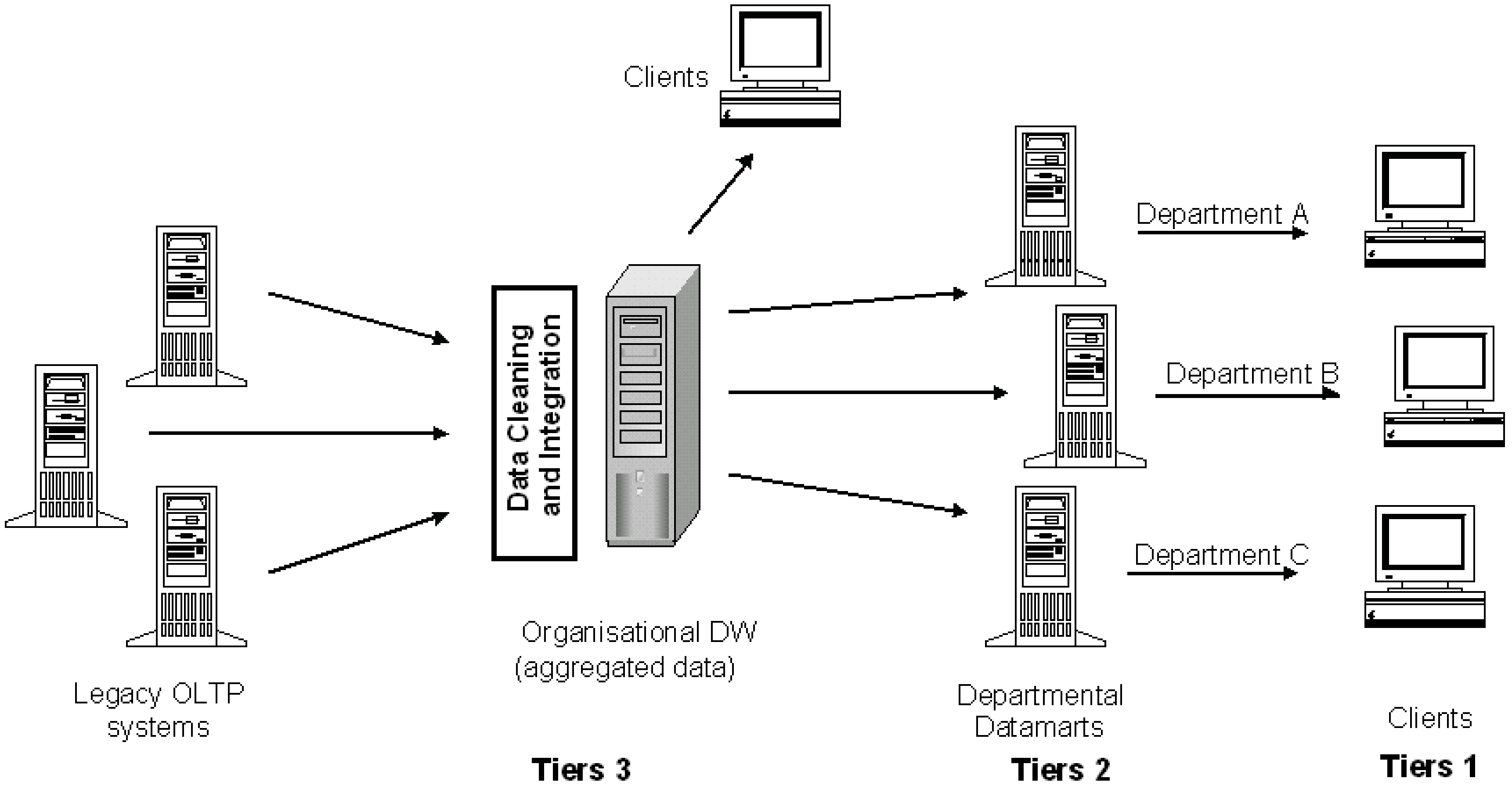
[Figure 3.1]


Not possible
Data Mart
Local heterogeneous data
Legacy systems
Summarized/ Aggregated data
Enterprise wide integrated data
Data Warehouse
Detailed data
Transaction oriented (OLTP/ current data)
Analysis oriented (OLAP/ current and historical data)

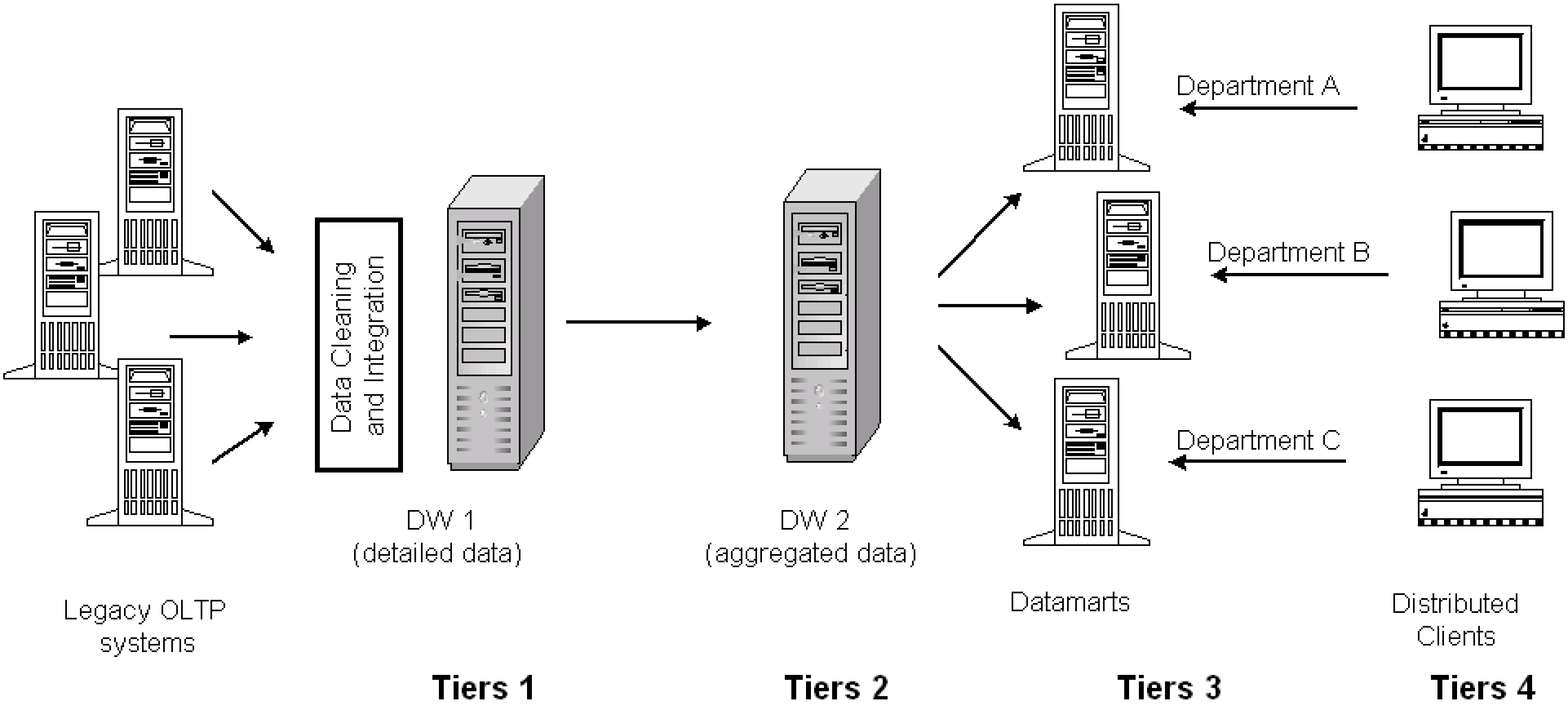
[Figure 3.2.]


Data Cleaning and Integration
Heterogeneous legacy OLTP systems
Centralized, unique and integrated corporate DW (includes metadata)
Distributed clients
• **OLAP**
• Query engines
• **Data mining**
• Report generators
• EIS
• etc.

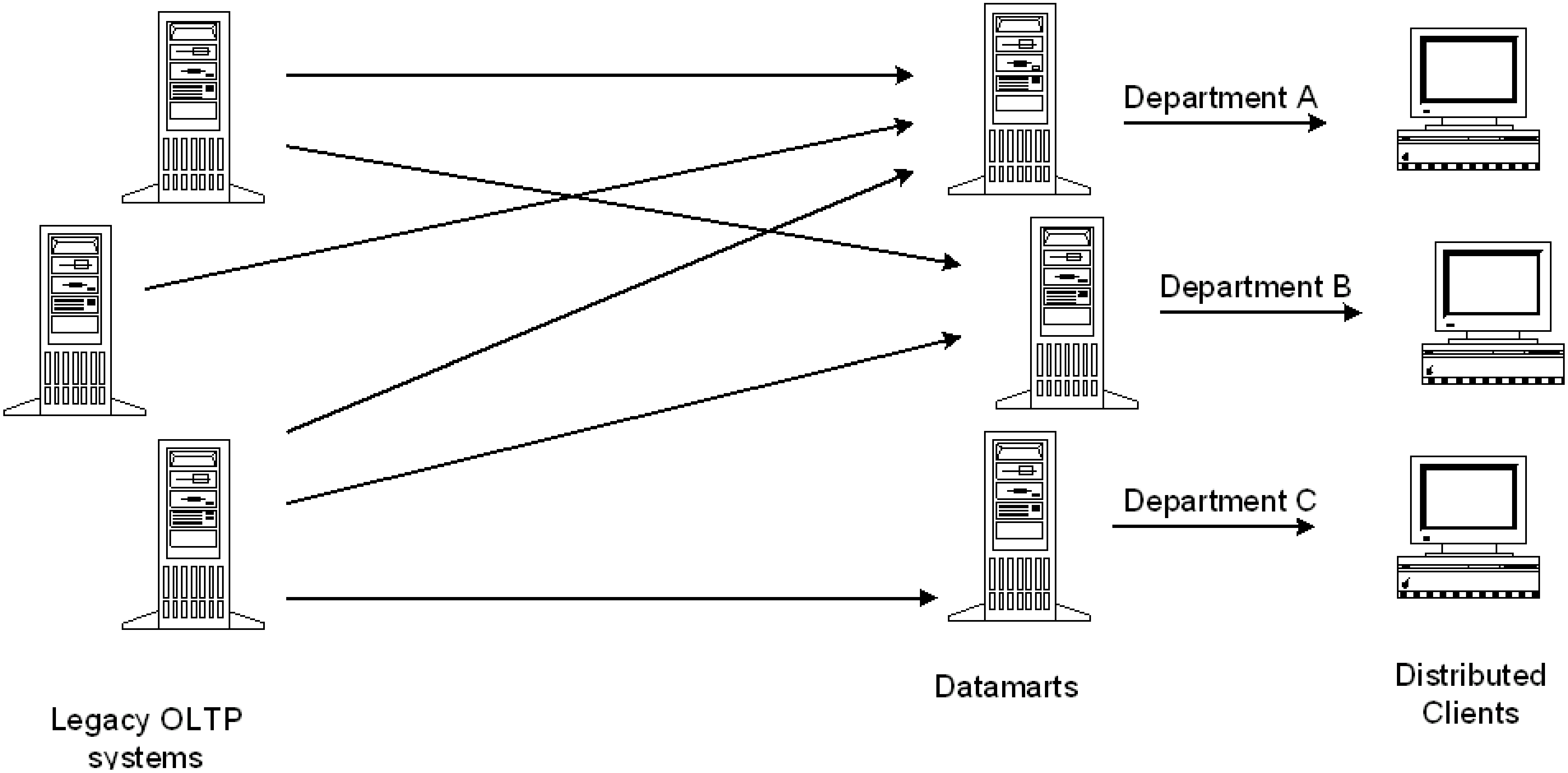
[Figure 3.3]


Clients
Data Cleaning and Integration
Department A
Department B
Department C
Legacy OLTP systems
Organisational DW (aggregated data)
Departmental Datamarts
Clients
Tiers 3
Tiers 2
Tiers 1

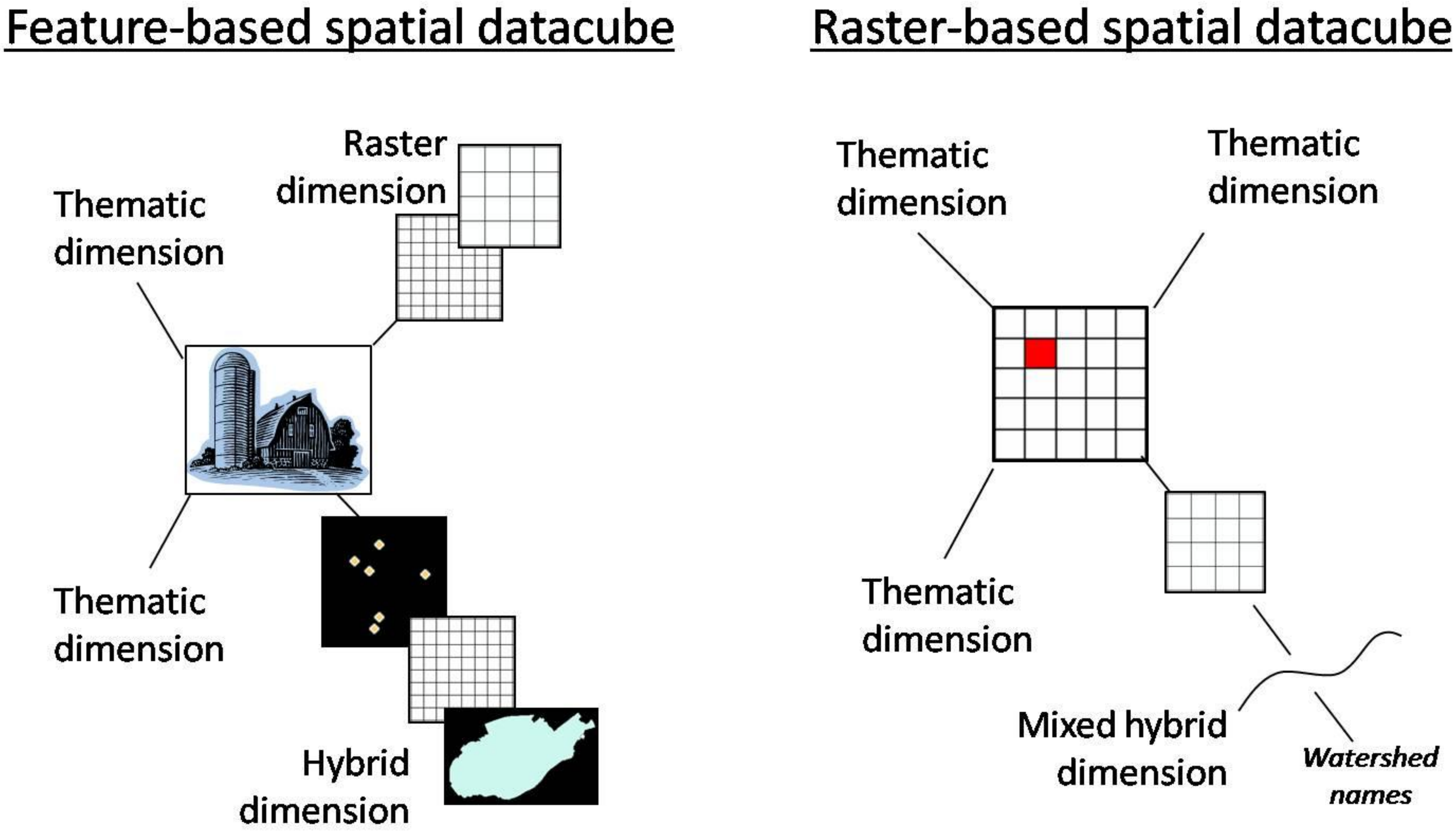
[Figure 3.4]


Department A
Department B
Department C
Data Cleaning and Integration
DW 1
(detailed data)
DW 2
(aggregated data)
Datamarts
Distributed Clients
Legacy OLTP systems
Tiers 1
Tiers 2
Tiers 3
Tiers 4

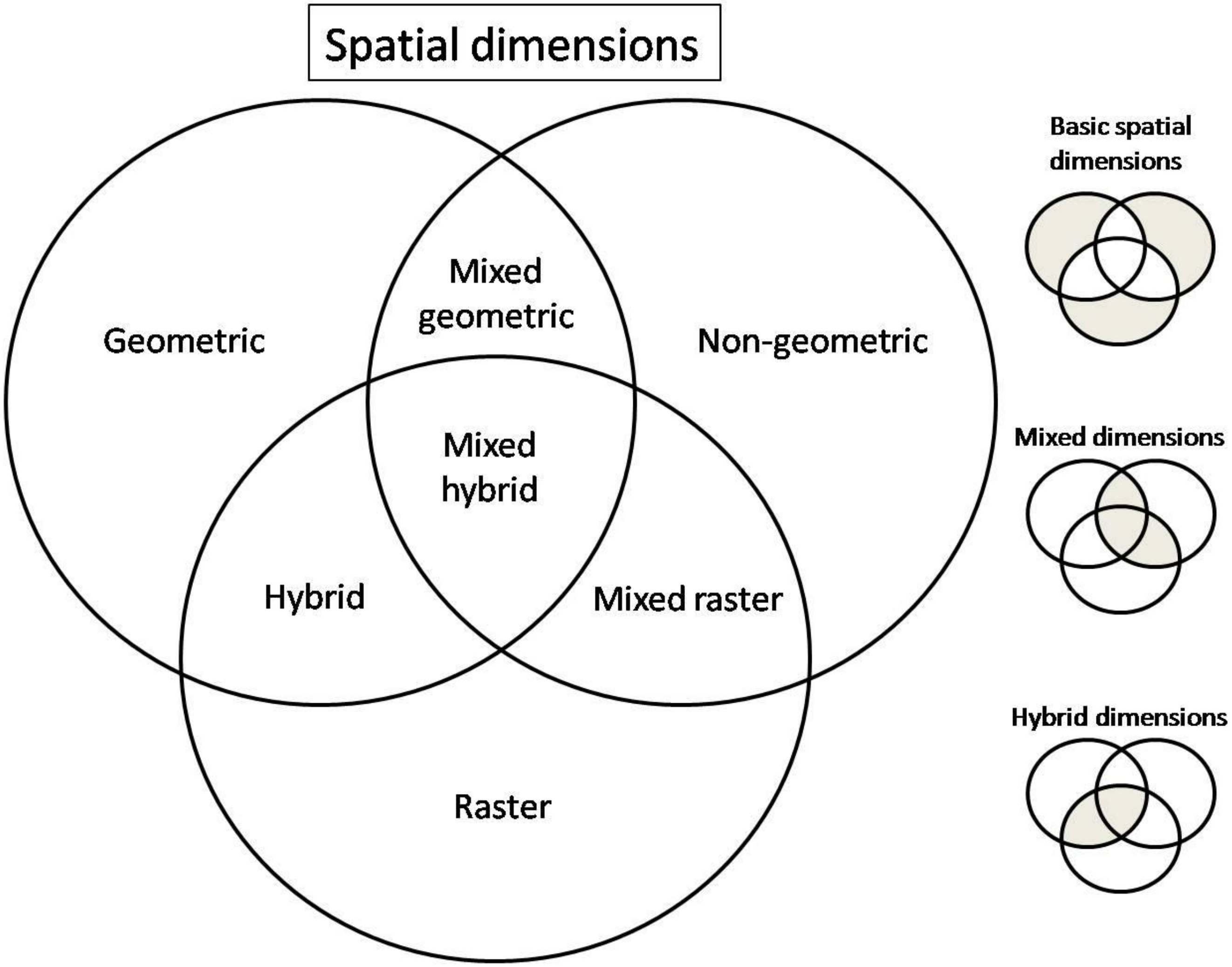
[Figure 3.5]


Department A
Department B
Department C
Datamarts
Distributed Clients
Legacy OLTP systems

[Figure 3.6]


Feature-based spatial datacube
Raster dimension
Thematic dimension
Thematic dimension
Hybrid dimension
Raster-based spatial datacube
Thematic dimension
Thematic dimension
Thematic dimension
Mixed hybrid dimension
Watershed names

[figure 3.7]


Spatial dimensions
Geometric
Mixed geometric
Non-geometric
Mixed hybrid
Hybrid
Mixed raster
Raster
Basic spatial dimensions
Mixed dimensions
Hybrid dimensions