Data Replication and Data Sharing - Integrating Heterogeneous Spatial Databases

Mark Stoakes and Katherine Irwin
Professional Services, Safe Software Inc.

Abstract

Many companies are faced with the problem of maintaining more than one database; with a different spatial area of interest, a different coordinate system, or a different GIS system. A master database is created to track all changes. Data replication is used to ensure all the working databases have the most current data.

Safe Software has worked with ESRI to develop a methodology to propagate only changes in the working databases to the master database. Each working database can pull updates from the master. This presentation will address the different issues that arise in a large data replication project.

Introduction

Location and spatial data is becoming a core part of business databases and decision-making. This growth in the use of spatial data has increased the need to share data with other organizations. Data sharing is driven by the need to maintain more accurate and up-to-date spatial databases, but at the same time reduce data acquisition and maintenance costs. In other cases, organizations may maintain identical databases at different locations in order to reduce network loads and improve response times for the data users who are spread over a wide area. In these cases, data replication is used to ensure all users are working from the most current data. Data may also be shared by linking several heterogeneous spatial databases through a common data access portal over a LAN or intranet. In all cases, the goal is to improve the accessibility of the spatial data, improve data quality and reduce the cost of maintaining the datasets involved.

The three broad approaches to sharing data are:
- **Data Sharing**. Data sharing is a data warehousing approach to making data available to a wider range of users. Data is acquired from several data owners and loaded into a centralized warehouse. Data can then be distributed to members of the data-sharing consortium through Web-Based data viewers, or delivered in different formats to the various data users.
- **Data Replication**. Data replication is used generally used where large numbers of data users who are spread over a wide geographic area require real-time access to the same data. To reduce network loads and improve data access performance the data is replicated over several databases at different locations. The databases are synchronized on a regular basis, usually nightly.
- **Distributed Data Access**. In this case the data warehouse simply acts as a node for data distribution. Data is held on the data owner's server, and the data warehouse acts as a live link to the data provider's datasets across a LAN, WAN or Intranet. Since the different databases may be in different formats (ESRI ArcSDE, Oracle Spatial, etc.) the Spatial ETL tool must be capable of reading all the formats to be accessed and served. There is no need to maintain multiple copies of the data, as is the case in data replication and data sharing.

This paper illustrates some of the issues that arise when undertaking data replication, data sharing or distributed data access.

The Challenge of Sharing Data and Replication

Once organizations agree to share or replicate their spatial data, they face the challenge of maintaining up-to-date datasets. Spatial data is changing continuously as new infrastructure, subdivisions or more accurate data is collected. To maintain up-to-date databases the various data "owners" must exchange their most current datasets with those they share their data with. This can be done in one of two ways:

- **Complete data load**. This is the most straightforward approach. The current dataset is removed and completely replaced with the new dataset. However, this approach is often impractical due to volume of data, which may be difficult to distribute and take a prohibitively long time to reload, resulting in the database being inaccessible to the users for extended periods of time.
- **Change only updates**. This approach requires smaller data volumes to be distributed as only the records that have changed (modifies, deletions and additions) are exchanged. Change only updates also reduce the time for the data load because of the smaller data volume. The update process is more complex than the complete data load approach.

For organizations that share data with external users (those outside their administrative sphere of influence) "change only updates" result in a number of potential challenges that may include data ownership, feature ID conflicts, coordinate accuracy and versioning. Often the data warehouse is a different format to the working databases (ESRI Geodatabase, Oracle Spatial, MapInfo TAB, GeoMedia, AutoCAD, etc.). Also, the warehouse and working databases may have different data models (or schemas). *Posting scripts* are used to control the transfer of the data between the different databases, and these scripts must be capable of handling these different configuration issues and formats, refer to Figure 1.
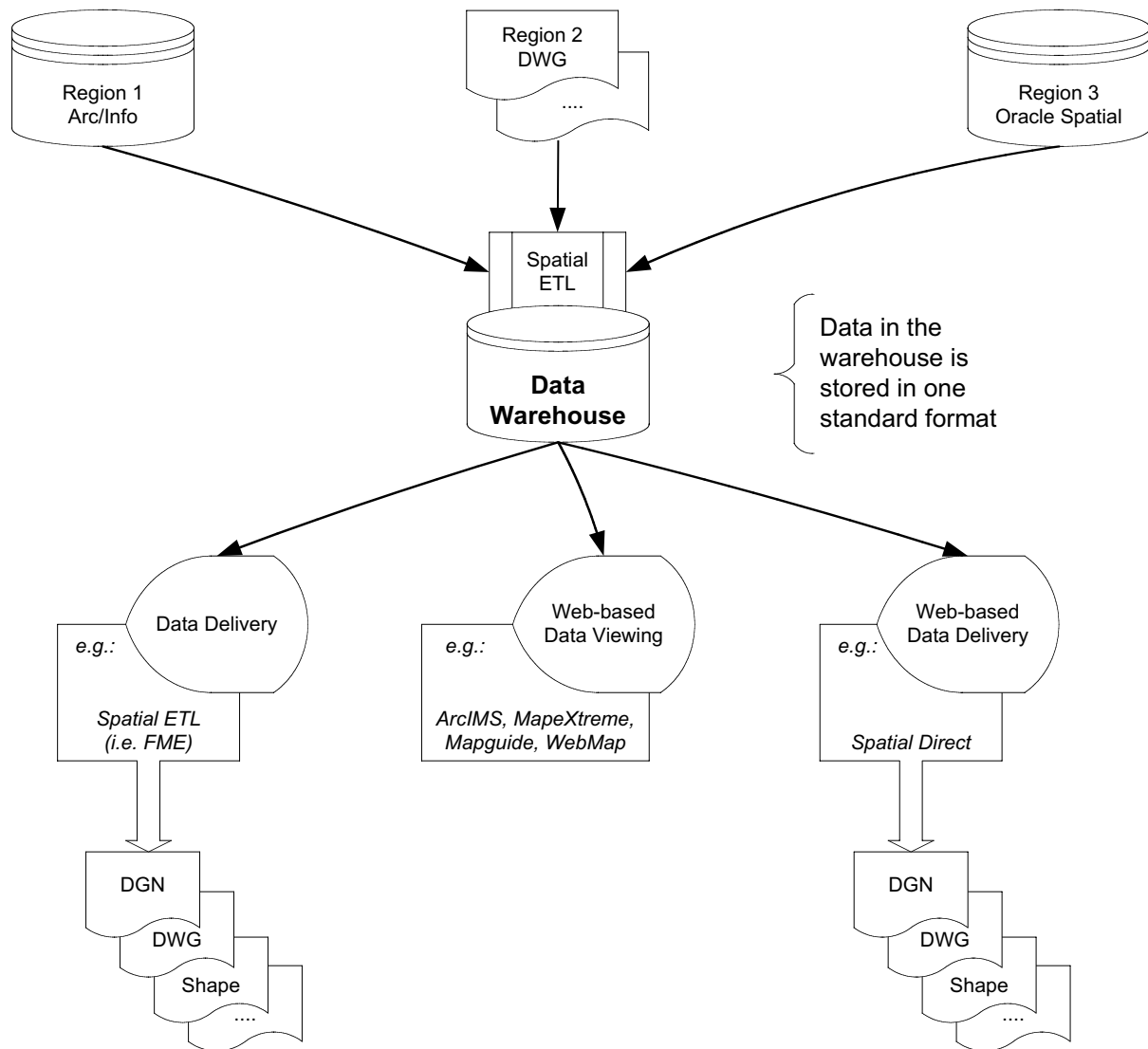
Figure 1: Example Data Sharing Environment

Data warehouses may have a different "focus" than the data supplier databases – e.g. a utility will require more detail of their infrastructure than, say, a municipality who's interest is perhaps restricted to the facility locations. Some organizations in the data-sharing consortium may be less reliable than others so data validation may be required before the data warehouse accepts the data.

Although data replication is a similar process to data sharing, the issues are more straightforward because the databases are replicas and data ownership is less complex, refer to Figure 2. Data sharing and replication environments often go hand in hand with data distribution for example web-based data viewing (ArcIMS, MapXtreme, Mapguide, WebMap) as well as web-based data delivery (SpatialDirect).

Data is the same:
- area of interest
- format
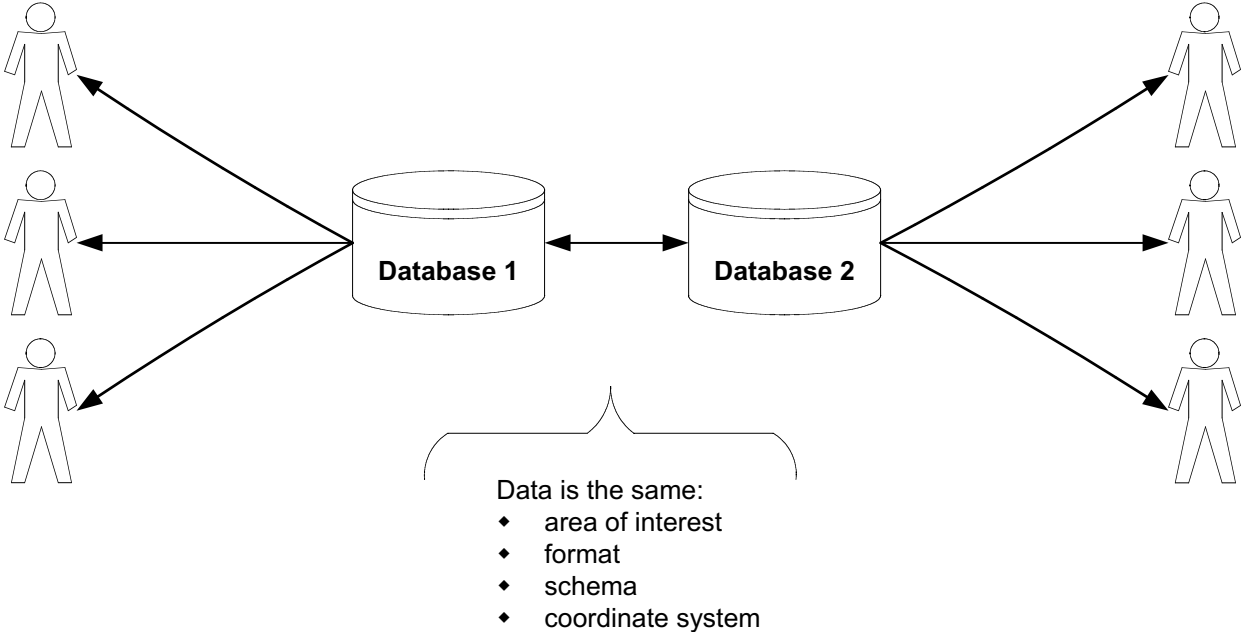- schema
- coordinate system

Figure 2: Example Data Replication Environment

On the surface, distributed data access appears to be more straightforward. Because the warehouse acts as a live link to several external data servers, there is no need to transfer data for update purposes, since the database owners are responsible for data maintenance. Hence the data flow is unidirectional, from the servers to the warehouse, refer to Figure 3. The warehouse may actually be eliminated completely and replaced by an application – which is the basis for a web feature server.
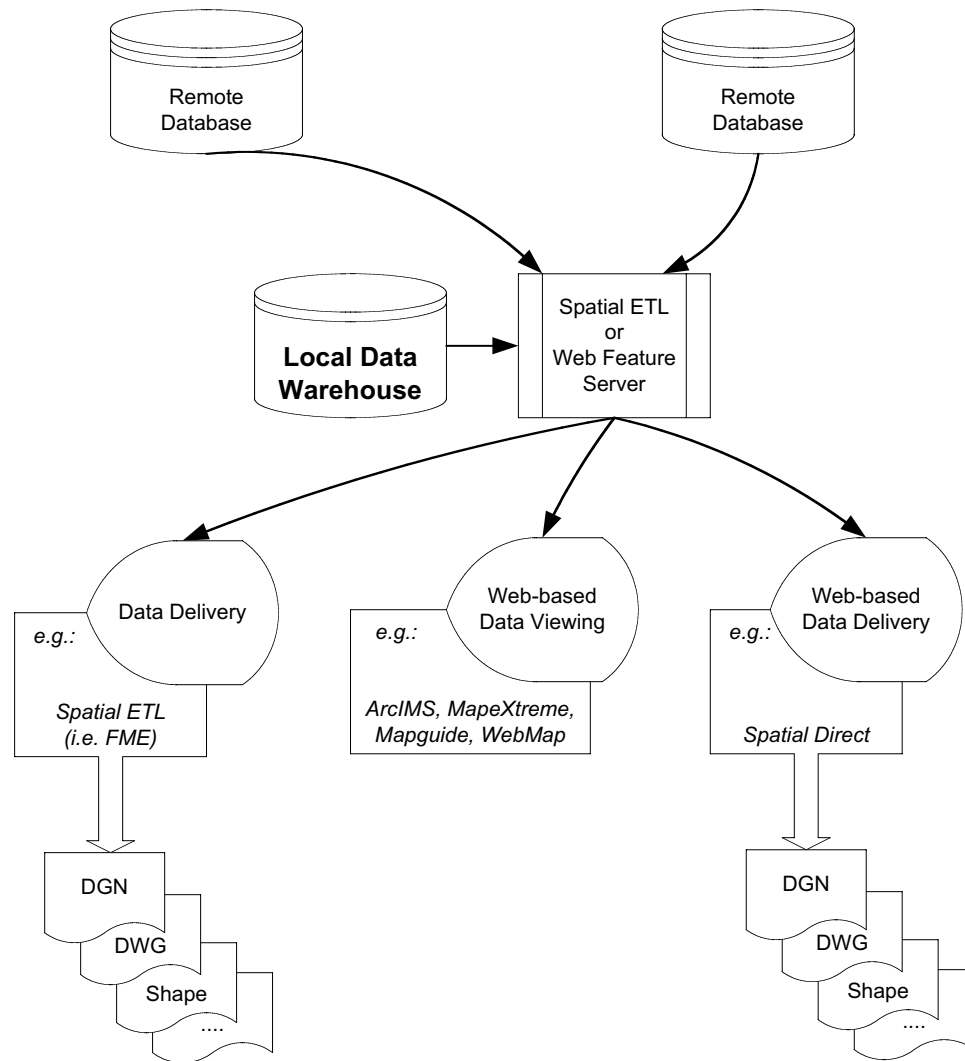
Figure 3: Distributed Data Access

Issues with Data Sharing and Replication

Depending on the requirements of the organization, and the type of data sharing involved, one or more of the following issues may have to be addressed to successfully set-up a data sharing or replication environment:

Data Formats: If the warehouse and working databases have different formats then the posting scripts must also handle the format translation. If some of the working databases are CAD formats (AutoCAD, MicroStation) then translations can be more complex as users will have expectations of maintaining symbology and text placement. CAD type formats often require tiling, as they are not capable of handling the large data extents found in databases.

Schema Mapping: In the case of data replication, the database schemas are typically the same – i.e. there is a one to one table and attribute mapping between a features in the different databases. Data sharing often involves complex schema mappings, with a single feature class mapping to several destination feature classes and vise versa. Often attribute

values are used as mapping keys, for example a "Type" attribute may control how features in a source table map to several different tables in the destination database, refer to Figure 4.
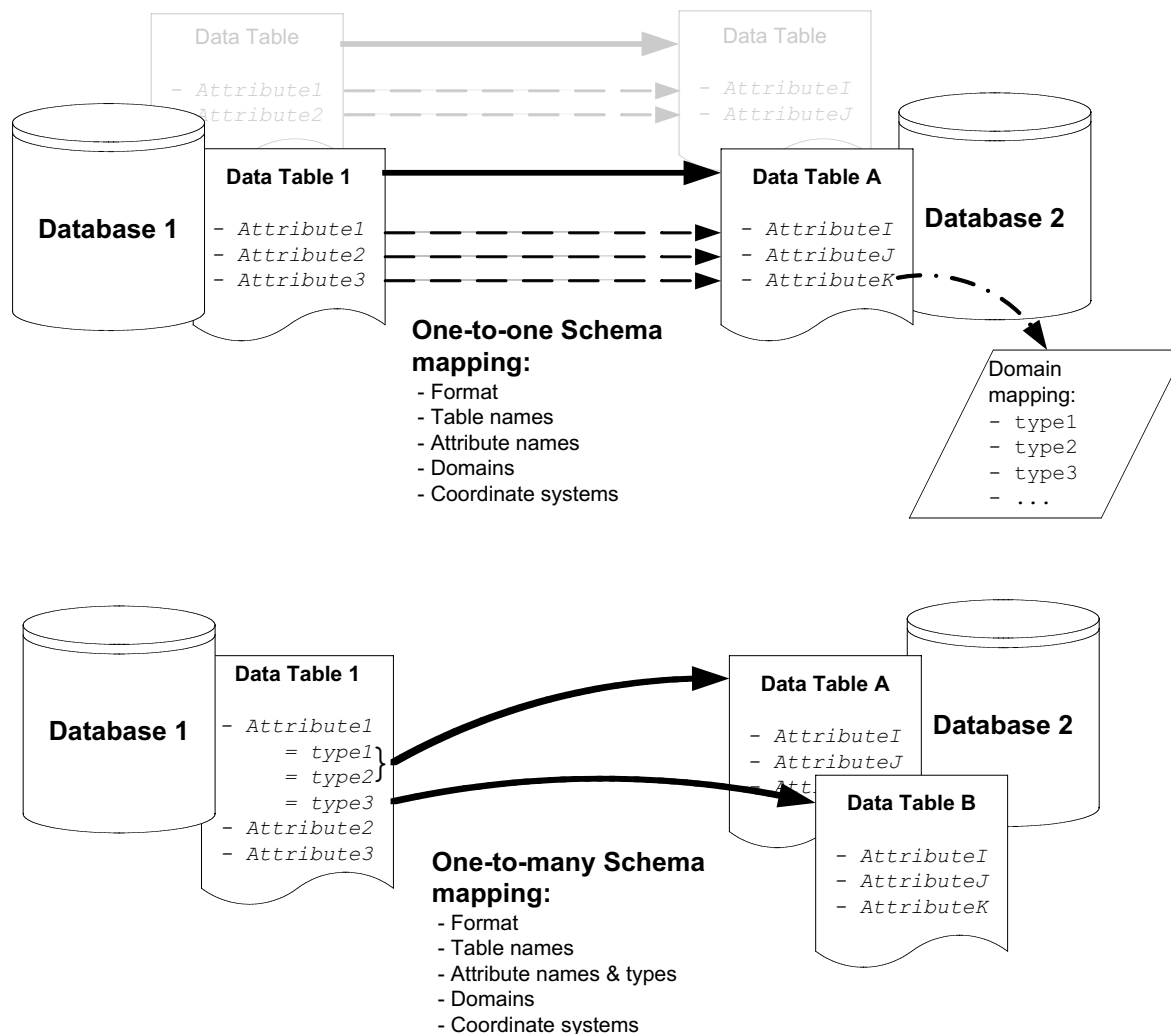

Figure 4: Schema Mapping Examples

Schema mappings must be reversible so that a feature pushed to a working database can be pulled back as the same feature in the data warehouse.

Unique Feature Identifiers: To simplify the update process, ID's are used to keep track of which features have changed. In an ideal world, all features would have a unique identifier such as the UK OS TOID, Microsoft's UUID or Geographic Data BC GOID, and these identifiers would be common across all the datasets to be shared. For data replication this is usually possible since the databases are replicas and the data is owned by the same organization. However, common ID's are not always possible when sharing data with external organizations, or if the database formats or schemas differ. Although most databases do require a unique feature ID, these are often not compatible between databases, due to different typing (integer, character) or incompatible lengths. In this case it may be necessary to maintain an ID cross-reference table so that ID's in a working database can be matched to the ID's used in the warehouse. Ownership of the ID's must also be addressed. If several

working databases are able to generate new features, then a methodology must be setup to ensure all the ID's are unique.

If it is not possible to maintain common ID's throughout the data sharing consortium then it may be necessary to revert to feature matching, where incoming features are matched (attributes and geometry) with existing features in the warehouse. This may be almost as computationally expensive as a full data load.

Data Ownership: It's important to clarify data ownership to eliminate potential conflicts. For example, who owns the "poles"; the utility, cable company or telco? Which organization is responsible to maintaining the pole location and any attribute data? Data ownership may have to be shared. For example, a municipality may require accurate pole locations, whereas a utility only needs the relative location of the poles – the attribute data being more important.

Coordinate Systems and Accuracy: Often the working dataset will have a different coordinate system to the data warehouse. For example, if the data warehouse covers a larger extent than the working database's area of interest, then the former may be in a geographic (lat/long) coordinate space and the working database in a Cartesian coordinate system such as UTM. Although there will be some coordinate shift due to the mathematics of the coordinate transformation, this is usually negligible. More significant is the resolution of the databases coordinates. Many databases store the coordinates as integer values, and the resolution of the coordinates are often dictated by the extent of the spatial data. If the resolution of the working databases and the warehouse differ, significant coordinate drift can occur as the data is moved back & forth. A related issue is that as coordinates are rounded to the integer grid, adjacent points may collapse, resulting in duplicate points. Features with duplicate points are often invalid and may be rejected during the data load. It also possible that different databases have different coordinate dimensions, which will result in the loss of Z values, and measures, if they exist.

Transaction Tables: Transaction tables are the change record tables used to track changes occurring in the dataset – the list of features that have been modified, added or deleted. Typically these are tables that records the type of change (modify, delete or add) and any other metadata. Transaction tables can also track the actual changes to the feature (whether it's a geometric change or attribute change) and often both the original record and changed record are maintained in an open format such as GML2 and stored as a BLOB in the database. The sequence in which the change records are processed is very important since the same feature may undergo several changes before the transaction table is processed. There are products available that help generate the transaction tables such as ESRI's Job Transaction Server or GeoMedia Transaction Manager, but these may require customization.

Versioning: Many spatial databases support long transaction versioning. Decisions have to be made as to what constitutes a "version", particularly in the warehouse where the number of versions may become excessive if every update cycle from a working database is treated as a separate version.

Data Validation: Different data suppliers in the data-sharing consortium will have different QA standards. Posting scripts can be configured to ensure that source data meets agreed schema and data standards. Checks can also be made if any relationships between features

are important, such as lot polygons having a corresponding lot point, and vice versa. Posting scripts can also be configured to clean-up data, such as removing unneeded feature types, removing redundant attribution and eliminating CAD data (such as legends and title bars).

Data Security and Proprietary Data: Often in a data-sharing consortium, some members will only have access to a restricted set of the data. Although this does not affect the update process, the processes that distribute the data must be set-up to accommodate these data restrictions and the data owners must be confident that only qualified users have access to more restricted datasets.

Implementation

Examples of organizations using Spatial ETL (extract, transform, load) tools to control their data replication and data sharing needs are:
- NIMA   Data Sharing & Replication
- Southwest Bell   Data Replication
- Geographic Data Technologies   Data Sharing
- BC Integrated Cadastral Information Society   Data Sharing
- IHS Energy   Distributed Data Access

The solution to implementing data sharing and data replication is a configurable Spatial ETL tool, such as FME®. Historically Spatial ETL tools have been used as direct data translators. Posting scripts control the mapping of the source format to destination format and the schema mapping. In addition, it is beneficial if the Spatial ETL tool can manipulate the data in the translation pipeline including coordinate transformation, clipping, polygon formation, point thinning, attribute manipulation, etc. For data replication and sharing, the posting script should be configurable so that the issues described previously can be addressed; reading the transaction table, mapping schemas and processing the different update types (change, add or delete).

The amount of effort required to develop the posting scripts depends almost entirely on the differences between the schemas of the data warehouse and the working databases/datasets and which of the many issues described above that have to be addressed. The more alike the schemas of the working databases and data warehouse, the simpler the posting scripts.

When sharing data from a wide variety of data suppliers it is often useful to define a data delivery method using a neutral, standards based format such as GML2. The schema of the data delivery format closely mirrors the schema of the data warehouse, simplifying QA and the data load. Figure 5 shows an example of data sharing using an intermediate data delivery format. Data suppliers are responsible for exporting their data into the data delivery format using any Spatial ETL tools they have available. The data load is carried out using standardized posting scripts, which only have to deal with the schemas of the data delivery format and the data warehouse. This approach can be applied to either a complete data load, or change only updates.
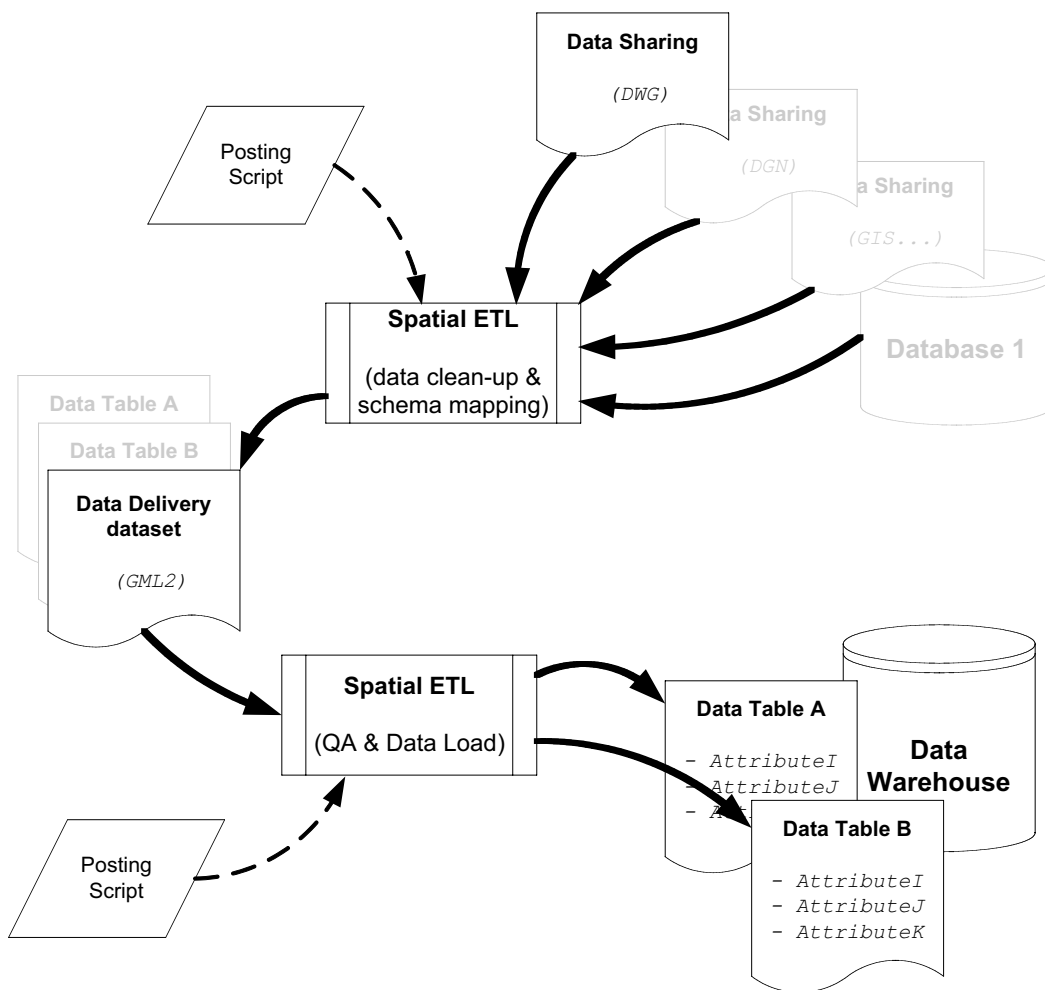
Figure 5: Data Sharing using a Data Delivery Format

When replicating data, the changes that have been made in all of the working databases are posted to the data warehouse. For working databases, only changes for the area of interest are extracted from the data warehouse and pushed back into each working database. The spatial ETL tool should be capable of taking full advantage of the spatial or attribute (SQL) queries offered by the different spatial databases such as ESRI SDE, Oracle Spatial, etc., so that only update features are transferred, refer to Figure 6. In the example shown, transaction tables are used to track the type of update (modify, add, delete), feature ID and the original and update feature as a GML2 "blob". If the ID's between the databases are incompatible then a key cross-reference table can be used.

In addition to the posting scripts that handle the update process there is generally a requirement for an initial load, where all the features that lie within the area of interest (AOI) are loaded from the warehouse into the working databases. When extracting features that cross the AOI boundary, they can either be clipped – in which case they cannot be edited in the working database – or they can be transferred as a complete feature resulting in a hairy tile where features extend outside the AOI boundary. This is the approach taken by UK Ordnance Survey when distributing MasterMap tiles.