Geospatial Data Integration Open Service Bus Architecture

Authors

Erel Rosenberg
Correlation Systems Ltd. Israel
1 Hamlacha St. Or-Yehuda, 60372, Israel
Erel@cs.co.il

Dieter Schneider
Tel-Aviv University
Studweidstrasse 40, CH 3700 SPIEZ
d.schneider@bluewin.ch

Abstract
In order to process and analyze data, which was received form one or more sources, one must compare the received information, identify similar and dissimilar elements, and create a unified model, which reliably depicts and predicts the state of the real world without losing important information.

The following paper proposes a novel approach for a flexible platform, which is based on industry standards and which enables the fusion, processing, and analysis of geographic data, originating from multiple sources (heterogeneous or identical). The platform is based on a unified model for the reporting geographic data, as well as a software architecture, which is designed especially to enable the activation of geospatial analytical services and the management and dissemination of their products.

The main advantage of the proposed platform is its flexibility, which is expressed in its capability to add supplementary data sources as well as additional analytic services without any modification to the core software. This flexibility makes the proposed architecture an ideal platform for the implementation of geospatial analytics in different and varying fields, such as: security, safety, army and military organizations, consumer behavior analysis, environmental protection, public transport and many more.

1. **Introduction**

The proliferation of data and knowledge in the modern world has shifted the focus of information technology from overcoming the lack of information to efficiently dealing with the flood of available data. This predicament of modern life may be reduced through the use of Data Fusion and Data Mining technologies.

This general problem is especially true in the field of geospatial information. Today's world introduces an abundance of geographic data sources. From GNSS services to Cell Phone location based services and from RFID locators to passive sensors and video object detection, the abundance of spatio-temporal information hold the promise of new and exciting business opportunities, but at the same time requires overcoming the challenge of integrating large amounts of information and processing it in a way which will provide valuable, high quality information on time.

The approach, presented in this paper, includes two facets:
1.1. A unified model for the reporting of geographic data, which is applicable to most relevant geographic data sources/sensors.

1.2. A SOA based software architecture, which is designed especially for geospatial analytic applications, and which includes:
1.2.1. **Enterprise Service Bus middleware,** which converts geographic data structures into the format of the unified model (the conversion is only necessary if the data is not originally received in the required format), as well as controls the flow of data in the proposed system architecture.
1.2.2. A **Toolkit**, which includes a set of available web services. These services include a wide range of data fusion and integration processes, tracking and positioning algorithms, as well as geographic data mining and behavior analysis capabilities. These processes automate the knowledge discovery process and enable the detection of behaviors and geospatial phenomena and relationship.
1.2.3. An **Analysis Server**, which controls the activation of the services available in the Toolkit

1.2.4. A **Reporting Server**, which provides a mechanism for the reporting and handling of the service bus products (as messages and alerts sent to the different services subscribers). The reporting Server also enables the activation of services which use the products as an input, thus allowing further and higher levels of processing.

2. The Unified Data Model

2.1. General

When creating a data structure for the reporting of the measured geographic attributes of an object, it is always luring to develop a specific structure, which is specially adapted to the characteristics of the described objects and to the needs and constraints of a specific application. On the other hand, a general structure may offer greater levels of flexibility and reuse potential, but, at the same time, may prove to be inadequate for some applications and for the description of certain objects.

In this chapter, we attempted to create a general model for describing the geospatial attributes of real world objects. The model includes the measured geospatial data as well as additional information which will enable the intelligent and effective integration and fusion of the data.

It is important to note that this model can be broadened to include additional attributes that will make the model relevant to additional non geographic applications.

2.2. The Model

The following are the proposed attributes of the unified model:

2.2.1. **Identification** – The object's unique identification.

2.2.2. **Type** - The objects class: soiltype, builtup objects, etc

2.2.3. **Relations** – the object's relations to other object and entities including: organizational hierarchies, manning, system component relations, etc.

2.2.4. **Temporal Attributes** – the temporal data describing the object's activity.

2.2.5. **Geographic Attributes** – any geographic information relevant to the object including: location, height, heading, velocity.

2.2.6. **Data Source** – the identification of the data source.

2.2.7. **Quality** – the perceived quality of the reported data, including: importance, certainty, and accuracy.

The attributes will be reported using a set of eXtended Markup Language (XML) and Geography Markup Language (GML) extensions, adapted for the presentation of real world object elements and which provide a common XML layer. The elements will be defined in a way which is independent of the specific application that is being used.

2.3. Geographic Data Integration – General Principals

The general model described above is designed to allow many data sources to report geographic related data in a way which will enable the fusion of the data and will be suitable for a wide variety of applications and purposes.

Although different attributes may require different fusion algorithms and at the same time several fusion algorithms may be applicable for a single reported attribute, several general principals apply to the fusion and integration of all the above mentioned attributes:

2.3.1. For each fusion algorithm a temporal window may be defined, in which inconsistencies will have to be resolved. Out side the defined window, conflicting reports will not be considered as inconsistencies. The default window may be changed by the specific application user.

2.3.2. Based on the hierarchy information, available as a result of fusing the relations attribute, it is possible to perform the data fusion in two dimensions:

2.3.2.1. **Same Level Fusion** – fusing data emanating from reports describing the object itself.

2.3.2.2. **Multi Level Fusion** – determining the values of an object's attributes based on the fusion of its subordinate object's attributes.
It should be noted that in many cases both types of fusion will be possible and the relation between the results of each process must be determined by the user.

2.3.3. In a similar manner, data describing an object can be derived from data reported on another object, which has a relation to the described object. Unlike the multi level fusion, the relation in this case does not have to be hierarchical and not even permanent. For example, a car fleet management system can infer the past position of its cars by using fueling report sent by each car. Even if the relation between the car and a gas station is temporary, at the moment of fueling the location of the car is identical to the location of the station.

2.3.4. The quality attribute, provided with each report, may be used to improve the data fusion process (by including only high accurate/certain/important data in the process) and solve cases of inconsistencies, ambiguity, etc.

2.4. **Non-cooperative data collection**
It is assumed that in most cases the reported data is collected in a cooperative manner, meaning that the correlation between the reported information and the object's identification is clear. However, in some cases, information regarding an object or objects is collected without cooperation (e.g. by radar systems), in which case the data fusion processes must include the following stages in order to overcome the lack of positive identification: target number estimation, based on previously known data and new data measurements (the results of the data fusion in step N become the known objects for step N+1), as well as the association of new data to those targets (data fusion per se).
Data fusion processes which are based on non-cooperative data collection may result in ambiguous results. Such ambiguity may be resolved over time (as more data is collected) or by fusing the results with additional external data.

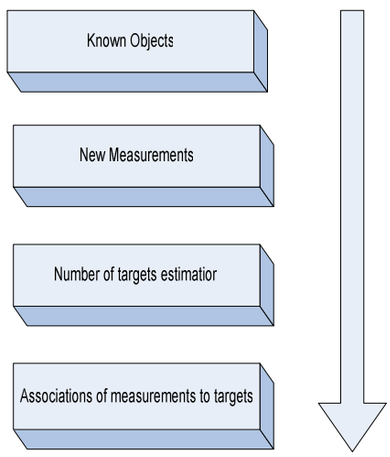

Figure 1: The general scheme of non-cooperative data fusion

In some cases the last two phases of the process, described in figure 1, can be performed simultaneously, or iteratively, however, the basic data flow is identical for most cases.
Several factors are taken into consideration when performing data fusion which is based on non-cooperative data collection:

2.4.1. **Execution time** – typically more accurate algorithms require more computation power.

2.4.2. **Time Frame** – longer data collection usually leads to better results. However, longer collection periods lead to delays in the reporting of results; therefore an operational decision regarding the desirable reporting delay is required.

2.4.3. **Association type** – an association of a measurement can be categorized as a "hard decision", in which a specific measurement is associated to one target, or a "probabilistic decision" (or "soft decision"), in which the measurements is associated to several targets with a certain degree of confidence (probability). The selection of an association type is usually influenced by the data fusion algorithm; however operational requirements may lead to preferring one type over the other.

2.5. **Error Resolution**

During the process of data fusion, situations may arise in which no clear result may be reached, for example: due to inconsistencies in the reported data, due to ambiguities arising from the non cooperative nature of the data collection, or as a result of reaching different results when applying same level fusion and multi level fusion (these situations will be referred to hereinafter as "errors"). In these cases the error may be resolved using predefined rules and/or additional external data and/or using the provided quality Meta data. If an error is not resolved, even after applying all available relevant means, it would be beneficial to allow the sending of an error report to the relevant data source in order for it to try and improve or correct the data. Sending the error report can be either automatic (in a predefined format according to predefined rules) or controlled by the user.

It should be noted that the time of the error reporting may vary due to the nature of the data (some data may not require immediate error resolution) and due to attempts to resolve the errors locally (before sending an error report to the data source). Any delay in the reporting of an error should not exceed the time frame in which the specific data remains relevant.

2.6. **Result Analyses**

After the data fusion and integration processes, analyses can be applied to the data, resulting in additional data, which was not previously available. The analysis can be parallel (processing the same data) or sequential (where each analysis processes the results of a previous analysis).

2.7. **Geographic Data Fusion Techniques**

The following chapter provides a list of the commonly used methods for the estimation of the location (or track) of an object, based on cooperative and non cooperative measurements or reports. The list is not exhaustive, but rather details the main process groups in wide use today.

2.7.1. **Partitioning Methods**

Partitioning algorithms share the same general approach in which the algorithms first try to find the center or distribution that will optimize an objective criterion. Once it is found, the membership of n objects within k clusters is automatically determined. The algorithms adopt an iterative relocation technique in order to find a local optimal

2.7.2. **Hierarchical Methods**

A Hierarchical method creates a hierarchical decomposition of the given set of data objects, forming a dendrogram – a tree which splits the database recursively into smaller subsets. The dendrogram can be formed in two ways: 'bottom – up' or 'top - down'.

2.7.3. **Density Based Methods**
Most partitioning methods perform clustering based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulties when trying to discover clusters of an arbitrary shape. Other clustering methods have been developed based on the notion of density. These typically regard clusters as dense regions of objects in the data space which are separated by regions of low density (representing noise).
Density based methods can be used to filter out noise (outliers), and discover clusters of arbitrary shape.

2.7.4. **Grid Based methods**
Density base methods are indexed – based methods that face a breakdown in efficiency when the number of dimensions is high. In order to enhance the efficiency of clustering, a grid – based clustering approach uses a grid data structure. It quantizes the space into a finite number of cells, which form a grid structure, upon which all the clustering operations are performed. The main advantage of the approach is its fast processing time which is typically independent of the number of data objects, and depends only on the number of cells in each dimension in the quantizes space.

2.7.5. **Constraint-Based Cluster analysis**
The development of spatial clustering on large databases has provided many useful tools for the analysis of geographic data. However, most of these algorithms provide very few avenues for users to specify real life constraints that must be satisfied with the clustering. In order for spatial clustering to be more useful, additional research must be done on further developing this type of method in order to provide users with the ability to incorporate real life constraints in to clustering algorithm.

3. **The proposed Software Architecture**
3.1. **Overview**
In this Chapter we propose a software architecture for the implementation of a Service Oriented Architecture (SOA) based Geospatial Data Integration and Analysis System.
The proposed architecture is not designed to provide a solution for a specific need but is rather designed to be a general and flexible solution. The architecture's flexibility is expressed in its capability to add additional data sources as well as additional services without major modifications to the core software. This flexibility makes the system an ideal platform for different and varying fields, such as: security, safety, consumer behavior analysis, environment protection, public transport and many more. In addition, the generic solution, provided by the proposed system, is preferable for organizations that do not have the capability (technical and/or financial) to define proprietary interfaces between their relevant systems.
The reliance of the proposed system on the unified model described in chapter 2 enables the same system to be used by different users (a multi user system) with different requirements by allowing each system user to choose the services relevant to him.

3.2. **Prerequisites**
The prerequisites of the successful implementation of the architecture are:
3.2.1. That all data sources (when fusing data emanating from more than one source) observe the same environment.

3.2.2. That all relevant data is harmonized according to the format of the proposed unified model. If this prerequisite does is not fulfilled at the data source level, it is required to implement

a harmonization process, which will ensure that all relevant data is harmonized before the data fusion process begins.

3.2.3. That all data sources and the system will be synchronized. This doesn't mean that the data from all the sources should arrive to the system simultaneously, but rather that a parameter will determine the maximum allowed time difference. The higher the level of fusion (refer to the chapter 3.3) the longer the allowed limit becomes.

3.3. **The Cluster Model**

When integrating data form a large number of sources it is preferable to organize the data fusion process in "clusters" of systems, which operate in a common environment (view the same part of the world), or share common characteristics (e.g. same hierarchical level). The fused data from each cluster may be sent to a higher level cluster for higher level data fusion. By optimizing the results at the cluster level, the results of the higher level data fusion process improve immensely.
The clusters (and data-sources/sensors) are typically organized as a DAG (Directed Acyclic Graph), which is designed to ensure that a single record will not be reported twice.
As was stated in the last chapter, the systems which are joined in the clusters must be synchronized, although the data itself does not have to arrive simultaneously from various sources. A parameter will limit the allowed time frame for data arriving to each cluster. The higher the level of the cluster and data fusion process the longer the allowed time frame will be.

3.4. **Conceptual Architecture**

3.4.1. **Overview**

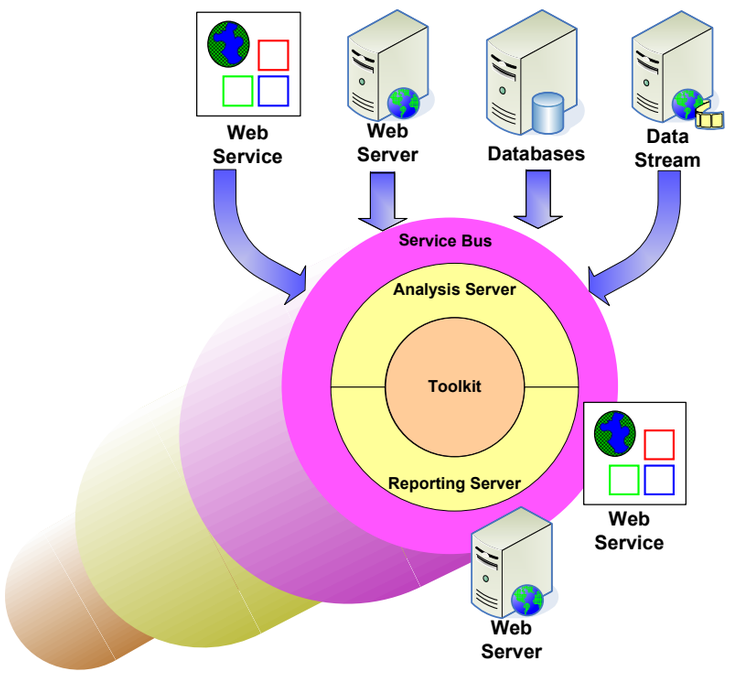

Figure 2: Conceptual Architecture

The software architecture is based on the following elements:

3.4.2. **Enterprise Service Bus middleware**, which provides tools for the conversion of data structures into the proposed unified model, for the management of data flows to and from the processing layers (including error reporting), and for the management of the flow of processing results. In addition, the service bus provides a set of agents that are able to receive data via standard communication protocols (such as: web services, http, database, TCP/IP), perform data alignment, and transfer the data for further processing based on user defined rules.

3.4.3. A **Toolkit**, which includes the set of available system web services. These services include a wide range of data fusion and integration processes, tracking and positioning algorithms, as well as geographic data mining and behavior analysis capabilities. These processes automate the knowledge discovery process and enable the detection of behaviors and geospatial phenomena and relationship.

3.4.4. An **Analysis Server**, which controls the activation of the services available in the Toolkit.

3.4.5. A **Reporting Server**, which provides a mechanism for the reporting and handling of the service bus' products (as messages and alerts sent to the different services and subscribers) as well as information regarding the technical status of the system. The reporting Server also enables the activation of services, which will use the products as input, thus allowing further and higher levels of processing.

3.5. Enterprise Service Bus Components

3.5.1. Data Fusion Services

Data fusion services are Web Services that receive data reports from various sources in the unified model format, and are responsible for the fusion of the different data reports into a single data set. A data fusion service can be one of a pre-existing library of services or can be written especially for the specific purposes/application, and added to the library.

3.5.2. Analyses

Similar in their function and characteristics to date fusion services, analyses can be applied to the fused data sets, resulting in additional data, which was not previously available. The analysis can be parallel (processing the same data set) or sequential (where each analysis processes the results of a previous analysis).

3.5.3. Data Collectors

Data collectors are special services that function as a temporary database for intermediate processing results. The data collectors are used in order to store the information until a sufficient amount of data (or time) is collected.

3.5.3.1. Single Queue Data Collectors

In a single queue data collector the data is stored until one of the following release condition is met:

- **Timeout** – a predefined period of time expires.
- **Quantity** – the amount of the data in the data collector reaches a pre-defined amount.

It is important to note that meeting a release condition means that the data in the collector is ready to execute the next service; however, the decision of actually executing that service is made at the service bus level and not by the data collector itself. As a result, low priority data collectors may be ready to release their stored data but will have to continue the data collection until the system has the required resources to perform the relevant service.

3.5.3.2. Multiple Queue Data Collectors

A multiple queue data collector is design to split the collected information based on a single attribute. The collected data is split into different queues which are operated autonomously.
Another function of the multiple queue data collector is to manage the release of information from its various queues. When the ESB decides to execute a service, which is relevant to more then one queue in the collector, the collector will decide which queue will release its data first, according to predefined parameters.

3.5.4. **Filters**
Filters are "simple" services that filter the received data.
Basic filters include:

- Selection of specific records based on their values and numeric operators, such as: equal, graters etc.
- Selection of specific records based on the geospatial attributes (i.e. "crossing a polygon" or "located near an area").

3.5.5. **Split / Merge**
The Split / Merge element enable users to either:

- Duplicate and split a data stream without affecting its content
- Merge two streams into one by means of simple aggregation and without affecting its content.

3.5.6. **Hubs**
Hubs are convectors designed in order to "translate" data received from the various sources into the format of the proposed unified model. Hubs can convert several data types:

- TCP/IP binary or XML streams.
- Data retrieved from external (relational) database.
- Data retrieved from HTML web sites.
- Data retrieved from web services that are not compatible with the proposed model

4. Bibliography
4.1. **Books**
4.1.1. Vicenc Torrs. 2003. *Information Fusion in Data Mining*. Springer.

4.1.2. Harvey J. Miller and Jiawei Han. 2001. *Geographic Data Mining and Knowledge Discovery*. Taylor and Francis.

4.1.3. Eli Brookner. 1998. *Tracking and Kalman Filtering Made Easy*. Wiley-Interscience.

4.1.4. David L. Hall and James Llinas. 2001. *Handbook of Multi Sensor Data Fusion*. CRC Press.

4.2. **Articles**
4.2.1. Technical panel for C3, U.S. department of defense, data fusion subpanel of the Joint Directors of Laboratories. 1998. *Data Fusion Lexicon*.

4.2.2. A. Gad, M. Farooq, J. Serdula, and D. Peters. 2004. The 7th International Conference on Information Fusion, *Multitarget Tracking in a Multisensor Multiplatform Environment*. http://www.fusion2004.foi.se/papers/IF04-0206.pdf (accessed April 25, 2006).

4.2.3. Rajnish Kumar, Matthew Wolenetz, Bikash Agarwalla, JunSuk Shin, Phillip Hutto, Arnab Paul, and Umakishore Ramachandran. 2005. *DFuse: A Framework for Distributed Data Fusion*. College of Computing Georgia Institute of Technology.

4.3. **Internet Sites**

4.3.1. The World Wide Web Consortium (W3C) page on XML. http://www.w3.org/XML (accessed April 25, 2006)

4.3.2. The Open Geospatial Consortium. http://opengis.net/gml/ (accessed April 25, 2006)