Management and Conflation of Multiple Representations within an Open Federation Platform

Steffen Volz

University of Stuttgart, Institute for Photogrammetry, Geschwister-Scholl-Str. 24D, 70174 Stuttgart, Germany
steffen.volz@ifp.uni-stuttgart.de

Abstract. Building up spatial data infrastructures involves the task of dealing with heterogeneous data sources which often bear inconsistencies and contradictions, respectively. One main reason for those inconsistencies emerges from the fact that one and the same real world phenomenon is often stored in multiple representations within different databases. It is the special goal of this paper to describe how the problems arising from multiple representations can be dealt with in spatial data infrastructures, especially focusing on the concepts that have been developed within the Nexus project of the University of Stuttgart that is implementing an open, federated infrastructure for context-aware applications. A main part of this contribution consists of explaining the efforts which have been conducted in order to solve the conflicts that occur between multiple representations within conflation or merging processes to achieve consolidated views on the underlying data for the applications.

1 Introduction

Whenever there is the need to perform a geospatial query, an adequate data source has to be found from which a correct answer to the query can be derived. In many cases, however, such a single data source does not exist. Instead, the geospatial world is split into pieces (i.e. heterogeneous geospatial databases) and the appropriate pieces have to be assembled like a mosaic or a puzzle to form the required data set. In principle, it is the goal of geospatial data infrastructures to provide algorithms which execute this assembly or database integration, respectively, automatically on demand to achieve a common view on the underlying data for the applications. However, the integration process is a huge challenge since the existing geospatial data sets which have been acquired by different institutions according to different conceptual schemas and data models, in different formats and scales, with different accuracies or at different dates, etc. are highly inconsistent. The biggest problem with respect to inconsistency results from the fact that the same real world object can have multiple contradictory representations in different databases. The integration of these multiple representations (MRep) requires at first their identification within the infrastructure and eventually, the conflicts between them have to be resolved in order to generate one consistent and consolidated data set.

Dagstuhl Seminar Proceedings 06101
Spatial Data: mining, processing and communicating
http://drops.dagstuhl.de/opus/volltexte/2006/588

Usually, spatial data infrastructures are realized as federated database systems which hide the heterogeneity of the different databases taking part in the system and act as a single, homogeneous database to the global applications [6]. This approach is also pursued by the Nexus project of the University of Stuttgart which develops an open, federated infrastructure for context-aware applications [17]. The main part of the information stored within the Nexus databases, so-called Context Servers, consists of geospatial data. In order to facilitate the federation of the associated data sources, a common, federated schema, the Augmented World Schema (AWS), is provided. Thus, the data of existing data sources or local databases first have to be mapped onto the AWS. Once the data have been transformed into the AWS format, they are stored as so-called Augmented Areas (AA) on geospatial data servers within the Nexus platform. Augmented Areas consist of objects of at least one object class and must not contain multiple representations or other inconsistencies. For exchanging AAs, they can be represented in the Augmented World Modeling Language (AWML), the common data format to serialize objects within Nexus. In case that the same real world object is represented in different Augmented Areas, the problem of multiple representations occurs. It is the goal of this contribution to explain the Nexus approach on

- how to model and manage multiple representations within a federated platform
- how to handle inconsistency issues between multiple representations
- how to identify and merge (or conflate) multiple representations into one consolidated data set

The paper is organized in 5 sections. In the following section 2, related work concerning the modeling, management, matching and merging of multiple representations is discussed. Section 3 describes the Nexus infrastructure and its mechanisms to deal with multiple representation issues. In section 4, our approach to the conflation of geospatial data within open, federated platforms is proposed and its results are presented. Finally, section 5 concludes this paper and gives an outlook on future issues.

2 Related Work

Currently, many spatial data infrastructures are evolving on the global ([11], [12], [17]), on the national ([1], [8]) and also on regional or city levels. However, mechanisms to deal with multiple representations within these geospatial infrastructures are only hardly available. Instead, such mechanisms are still the subject of ongoing research initiatives. The following sections will discuss the concepts that have been proposed up to now.

2.1 Modeling and Managing Multiple Representations

In [3] the situation in France is presented where they have three coexisting spatial databases containing the same real world objects at different levels of detail. The authors point out that there are basically two approaches to organize multiple representations: on the one hand they can be instances of one object type. For each database view and level of detail, one different representation of the object type can be created. On the other hand multiple representations can be kept separately as instances of different object types and linked by binary correspondence relationships (like is-a, equivalence, aggregation and set-to-set). It is illustrated how these two approaches can be used to derive a unified database schema from different databases containing similar object types that allows global querying and ensures global consistency by enforcing consistency rules. The mentioned contribution is related to the MurMur project that is especially dealing with concepts for the management of multiple representations in geospatial databases [20]. Here, also the temporal aspect of multiple representations is reflected where each representation corresponds to one point or interval in time. Introducing time in spatial modeling is realized using timestamps that express the temporal validity of an object, and by introducing temporal relations between multiple representations (e.g. one representation existed before another representation etc.).

The work of [10] proposes the so-called Multiple Representation Management System (MRMS) that can maintain consistency over autonomous databases containing multiple representations. It uses the Multiple Representation Schema Language (MRSL) to model MRep entities and the Multiple Representation Schema to specify consistency rules. The MRMS operates on top of the database management system and thus does neither influence the mechanisms within the database nor does it – unlike Nexus – require a complete integrated schema. The focus of this work is different to our approach since the authors mainly describe how the MRep objects correspond to the real world entity they represent, rather than evaluating the consistency amongst multiple representations. For each occurrence of multiple representations they introduce a superior object, the integration object, that represents the real world entity and specify the correspondences between the representations and the integration object, i.e. multiple representations can be seen as roles of the integration object. The integration object is responsible for keeping its representations consistent. This approach facilitates that consistency among more than two representations can be controlled.

2.2 Matching and Merging Multiple Representations

Each conflation process basically consists of two steps: at first, multiple representations have to be identified during a matching phase and then a merge (or fusion) operation integrates corresponding objects into one single representation.

Many efforts have been made with respect to the matching of multiple representations. One of the most fundamental approaches has been presented in [27] for street network data of Geographic Data Files (GDF) [5] and the German Authoritative Topographic Cartographic Information System (ATKIS) [28]. In a first step, the algorithm finds all potential correspondences of topologically connected line elements in two source data sets by performing a buffer operation. The matching candidates are stored in a list which contains a large amount of ambiguous matching pairs. Representations with a low likelihood of correspondence are eliminated using topological and geometric information and the remaining matching pairs are evaluated with a merit function to derive the solution of the matching task. This is a combinatorial problem which is solved by an A* algorithm. The buffer algorithm of [27] has recently been adopted by several other authors. The work of [16] extends the algorithm to be able to apply it in a symmetric fashion for the matching of cartographic objects. In [22], the Java Conflation Suite developed by the Jump Project [14], is extended by 3 different modules, one of which also uses the buffering approach to optimize matching procedures between route data derived from navigation systems and road data provided by national mapping agencies. Also, [29] applied the buffer algorithm while matching street networks. They developed a method to adjust the buffer parameters during the matching process to find an optimal solution. The authors of [9] have proposed new topology-based approaches for the matching of linear objects in order to perform a conflation.

The issue of conflation has first been addressed by [21] who worked within a joint project between the United States Geological Survey and the Bureau of the Census that aimed at consolidating separate map data of both institutions. Fundamental techniques with respect to geometric conflation like data adjustment by rubber-sheeting or 0-cell matching by the spider function have already been proposed there. In [4] conflation is understood as a combination of two digital maps to produce a third map that is better than each of its sources. The merging part is identified as feature deconfliction during which data quality of the source data sets plays a major role. A linearly weighted algorithm is used that combines object attributes, i.e. also the geometries, on the basis of their believability. The discrete Fréchet distance has been chosen by [7] to identify homologous objects. For the fusion of geometries a weighting function is applied that allows merging homologous lines and polygons while considering adjacency relations in the process. In [15] a new distance measure is developed for polygonal lines. Conflation is understood here as the process of optimally adjusting two polygonal lines by an arbitrary combination of translation and rotation so that the distance between them is minimal. The author proposes an iterative algorithm which is able to compute an optimal translation followed by an optimal rotation with a complexity of O(n). The contribution of [13] uses a rubber-sheeting transformation for the purpose of adjusting corresponding data sets. In this work a huge number of control points is created. For each corresponding pair of polygonal lines each vertex is transferred to the corresponding object, i.e. new control points are generated so that each vertex has its corresponding vertex in the corresponding object. Then an interpolation of coordinate differences is defined that allows calculating the corrections of coordinates for the objects.

3 Handling Multiple Representations within Nexus

In the following sections, the basic concepts of Nexus are briefly outlined at first. Then, our approach to deal with multiple representations and the related consistency issues is presented.

3.1 Introduction to Nexus

Nexus is designed to be an open platform that has a three-tiered architecture: the application layer, the federation layer and the context layer (see Fig. 1). The applications access the federation via a standardized interface: data can be exchanged using the Augmented World Modeling Language (AWML) and queries can be formulated in the Augmented World Query Language (AWQL). Both languages are XML-based.

Any application request is directed to a *Nexus node* within the federation layer. On each node, a federation component is running. The Nexus nodes are responsible for developing a strategy to respond to application requests. They distribute the queries on the appropriate components of the Context Tier, i.e. the Context Servers (CS). There can be different kinds of CS, specifically tailored to the needs of different data types. For example, a main-memory solution was implemented for mobile objects whose positions have to be updated frequently. Most of the CS are realized as regular (geospatial) databases.

In order to create a strategy to answer application requests, a Nexus node must first identify which geospatial data or Augmented Areas (AA), respectively, have to be used for query processing. Therefore, it addresses the *Area Service Register* which can be understood as a metadata repository or a spatial search engine, respectively, storing general information about AAs (like spatial extent, level of detail, stored object classes etc.) and the addresses of the Context Servers where they are located. After having queried the appropriate CS, the Nexus nodes integrate the data that is returned from the Context Layer to one consistent and coherent result set by conflating multiple object representations (see section 4). Then typical geospatial services like map production can be performed on these data on the federation level. Eventually, the results are propagated to the application.

In order to enable the federation of heterogeneous and distributed data, a global schema has to exist. This global schema is called the Augmented World Schema (AWS) according to the Nexus terminology. The term "augmented" reflects the fact that not only information can be represented in the AWS which physically exists in the real world but also virtual objects like virtual blackboards or virtual information towers can be modeled. The AWS has an object-oriented structure and supports multiple inheritance. It contains the basic object classes necessary for location- and context-aware services in the so-called Standard Class Schema (SCS), e.g. classes to represent geospatial objects like buildings or roads, classes to store relations (e.g. topological or temporal relations) or classes for mobile objects like cars, etc. If an application needs to introduce new object types, they can be derived from any type that is already available in the SCS to form an Extended Class Schema (ECS) [18].

All spatial objects within the SCS inherit from the class SpatialObject which ensures that every AWS object has a globally unique identifier, the Nexus Object Locator (NOL), and a geographic position. A NOL basically consists of three parts: the address of the server on which it is stored, the identifier of the Augmented Area that contains the object, and the unique ID within the data set. The SCS defines further attributes and specifies their semantics, but most of them are optional, i.e. a data provider does not have to supply them to take part in the system.

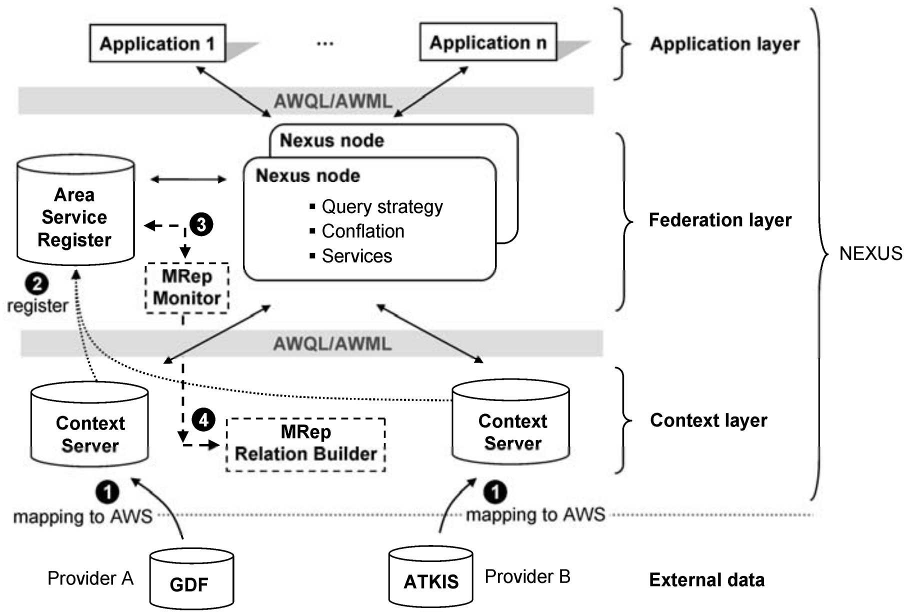

Fig. 1. The Nexus architecture is organized in three tiers; anyone who wants to participate can provide data if there is a mapping function from existing formats onto the AWS and if the data have been registered

3.2 Management of Multiple Representations

Just like in the World Wide Web, anyone who wants to participate in Nexus can supply data to the system independently from other providers. The only requirements are that

- the data that is originally available in the source schemas has to be mapped onto the AWS and stored within newly set up or already existing context servers (see step (1) in Fig. 1). Such mapping functions have been implemented in Nexus for street data of GDF and ATKIS, i.e. these data can be serialized as AWML
- the provided data sets have to be registered at the Area Service Register (see step (2) in Fig. 1)

If these two steps have been carried out, a service, the MRep Monitor, is triggered which automatically checks if data of the same object classes and geographical regions as the newly input data are already present in the Nexus infrastructure (see step (3) in Fig. 1). Whenever this is the case another service starts that tries to establish explicit relations, so-called MRep Relations, between corresponding objects by performing an automatic matching (see step (4) in Fig. 1). This service is called the MRep Relation Builder. The matching algorithm that builds up MRep Relations has been proposed in [25] and is briefly sketched in the following. It is the prerequisite of the conflation mechanism that is explained in section 4. After the matching approach has been illustrated the structure of MRep Relations is outlined and our approach regarding the update of multiple representations using MRep Relations is described.

3.2.1 The Matching Approach

The matching algorithm during which MRep Relations are being built has up to now been developed for linear street data of approximately the same scale and the same accuracy. The test data sets stem from different street databases available in Germany, namely GDF [5] and ATKIS [28]. A mapper has been developed both for GDF and for ATKIS which is capable of transforming the data from the source formats into AWML.

Let us assume that Nexus does not comprise any street data at all. Now, if a street data set of a specific area is registered as an Augmented Area at the Area Service Register, the MRep Monitor does not detect any potential multiple representation problems since it is the first AA covering street data within Nexus. The case is different if a second Augmented Area covering the same or at least an overlapping geographical area as the first AA is introduced: then, the MRep Monitor triggers the MRep Relation Builder.

The matching algorithm that is implemented in the MRep Relation Builder has been presented in [25] in detail but shall be repeated here very briefly to give the reader an idea of the process. At first, the two datasets to be matched are geometrically adjusted by a rubber sheeting transformation. Then, an algorithm topologically splits the data sets in a way that a maximum number of 1:1 matches can be achieved. This results in an optimization with respect to computing time since the number of 1:n and n:m matches (for the detection of which combinatorial problems have to be solved) is reduced. After the two pre-processing steps, the actual matching begins with a seed node screening process, identifying homologous nodes in the multiply represented street networks that show a high likelihood of correspondence. It establishes MRep Node Relations between them and stores them in a seed node list. Starting from the seed nodes, their adjacent nodes and incident edges are investigated and geometric as well as topological similarities between them are detected. If the similarity values exceed well-defined thresholds, MRep Node Relations are being set up for the adjacent nodes of the seed nodes and MRep Edge Relations are created for the edges emanating from the seed nodes. The newly detected MRep nodes are then added to the seed nodes list and the process starts again. The algorithm runs in multiple iterations and both the seed node detection and the following node and edge matching procedures are repeated applying relaxed constraints. The algorithm was used with multiply represented street data of Stuttgart and achieved a correctness rate of approximately 97% in the test area.

3.2.2 The Structure of MRep Relations

Basically, relationships between multiple representations could only be expressed as simple pointers within a bidirectional list, displaying that an object or an object set of data set A can be assigned to an object or an object set of data set B and vice versa ($a_A \leftrightarrow x_B$, $\{c_A, m_A\} \leftrightarrow y_B$, $\{l_A, r_A\} \leftrightarrow \{n_B, s_B\}$, etc.). However, the relations could also be defined in a more explicit way as it has already been proposed by [23]. The notion to store explicit relations between multiple representations, so-called MRep Relations, relies on the fact that during matching more information can be derived. Besides the identifiers of the instances which make up a representation in data set A and the identifiers of the instances constituting the corresponding representation in data set B, the cardinality of an MRep Relation can be implicitly deduced. It can either be *1:1*, *1:n*, *n:1* or *n:m*.

Furthermore, the degree of similarity between two corresponding representations can be determined and stored in an MRep Relation. Since geospatial objects do have geometric, topological and thematic properties, similarities between representations can be assessed on these three levels. Within this work, only geometric and topological similarity indicators were generated. We investigated street networks where crossings were stored as point features (nodes) and streets were represented by linear geometries (edges). For the crossings, we used proximity and geometric properties of incident edges in order to determine similarity. The geometric similarity measures for the streets were based on the comparison of angle and length differences and also on distance values like the average line distance and the Hausdorff distance. Topological similarity was detected using adjacency relations of corresponding street representations. Thus, we calculated different partial similarity measures both for nodes and edges which were aggregated into a total similarity value to express the overall degree of consistency of two representations (see Fig. 3).


```
<mrepedgerelation>
    <mrepedgerelation_id>0x0786mrepedgerel_307</mrepedgerelation_id>
    <mrepedgerelation_pos>34.777654, 54.008631</mrepedgerelation_pos>
        <attributes>
            <general_atts>
                <source_ids>
                    <id>atkis_A02MZNE3RE</id>
                </source_ids>
                <target_ids>
                    <id>gdf_0x09c658b10c9e11d9901aed0fa2996570</id>
                </target_ids>
                <cardinality>1:1</cardinality>
                <total_similarity>93.75</total_similarity>
            </general_atts>
            <geometric_atts>
                <length_difference>3.00</length_difference>
                <angle_difference>0.05</angle_difference>
                <hausdorff_distance>11.71</hausdorff_distance>
                <avg_line_distance>11.50</avg_line_distance>
            </geometric_atts>
            <topological_atts>
                <startnode_deg_diff>0</startnode_deg_diff>
                <endnode_deg_diff>1</endnode_deg_diff>
            </topological_atts>
        </attributes>
</mrepedgerelation>
```


Fig. 3. Excerpt of an MRRL File showing the basic structure of an MRep Relation

MRep Relations are designed as so-called heavyweight relationships according to the terminology of OGC [19]. This means that they are modeled as object classes within the AWS and exist as individual objects that can have attributes. Just like other Nexus objects they have a Nexus Object Locator (NOL). MRep Relations are always symmetrical and have a multiplicity of 2 (see [2]), i.e. they can only relate two corresponding objects of the same geometric type, one of which is considered as the source and the other one as the target representation. Since MRep Relations do have a geometric component, the centroid of the geometries of the corresponding objects, they can be efficiently found within the Nexus infrastructure.

MRep Relations are exchanged within Nexus using an XML-based format called **M**ulti**R**epresentational **R**elation **L**anguage (MRRL) which has been specified previously [24]. The above extract of MRRL shows the basic structure of an MRep Relation by means of an MRep Edge Relation (see Fig. 3).

3.2.3 Performing Updates using MRep Relations

In principle, one could say that building MRep Relations or matching, respectively, is only needed when a client query requires the integration of multiple, overlapping data sets which contain representations of the same real world phenomenon. This is basically true, but establishing MRep Relations has two main advantages: first, the matching process can be tedious and time-consuming, thus increasing the response time to client queries, i.e. if MRep Relations have been set up beforehand, the efficiency of the federation can be improved since only the conflation step has to be performed. On the other hand, MRep Relations allow for a mechanism to ensure that an update that has been carried out on a representation can be propagated to its corresponding representations within the platform and thus global consistency within the Nexus infrastructure can be achieved.

In order to provide a mechanism that enables consistency restoration, the objects which are related by an MRep Relation are referenced from the relation and possess an attribute themselves which references the MRep Relation object. This attribute is called backward reference ("back_ref" in Fig. 4) and allows recognizing how many corresponding counterparts an object has within the Nexus infrastructure (notice that each SpatialObject can have more than one backward reference since it might participate in more than one MRep Relation, i.e. it can carry a set of references to MRep Relations). Thus, when an update is performed on a representation that has corresponding representations within the infrastructure, all necessary MRep Relations can be derived. Using these MRep Relations, the corresponding representations of the updated representation can be found and efficiently changed themselves. The update procedure is depicted in Fig. 4 where the Boolean attribute "under_construction" of object "Street_A" is changed. Using the relevant MRep Relations, the update can be propagated to all related multiple representations, namely "Street_C" and "Street_M" within the Nexus infrastructure.

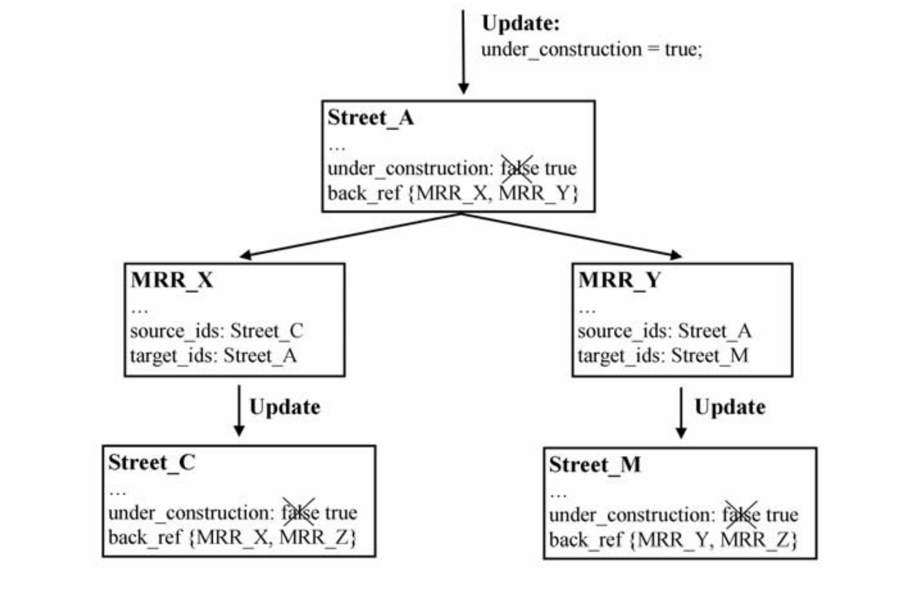

Fig. 4: Updating multiple representations within Nexus

4 Automatic Conflation within Nexus

The first step of a conflation process always consists of a matching process. This matching process is in Nexus done beforehand and the result is stored as MRep Relations. As it has been shown in [25], there are applications like navigation within multiply represented street networks which do not need a consolidated, conflated data set but exploit MRep Relations during processing. However, for some applications like map production a merged data set only containing one single representation (SRep) for each real world object is essential. Therefore, conflation has to be supported by Nexus. In the following sections first a conflation scenario is outlined. Then the conflation approach itself is presented and its results are presented.

4.1 Conflation Scenario

When a client query is directed to a Nexus node, the federation looks for data that is required to process the query. Now, let us assume that a client wants to receive a street map for a query area Q. In Nexus, though, no Augmented Area is available that covers this area completely, but there are two areas X and Y which are overlapping and partly contain data of the required area. If they can be added or merged, respectively, the whole query area could be represented as a map (see Fig. 5). This is the basic scenario we are trying to solve in the following sections.

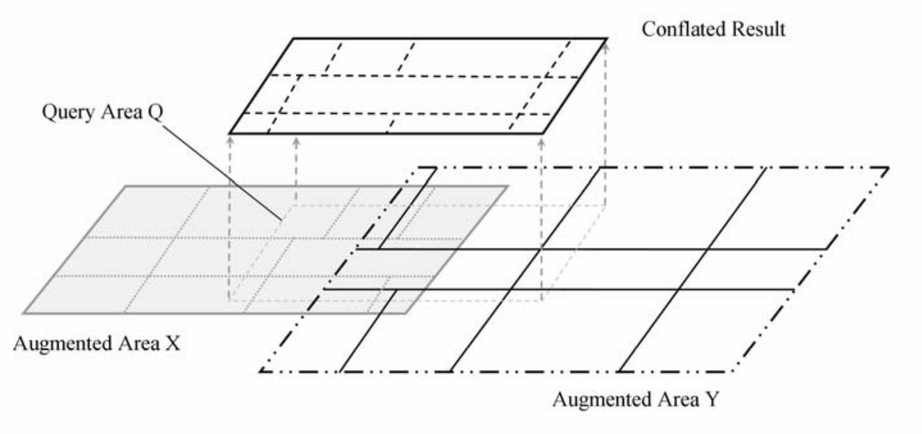

Fig. 5. The scenario for a conflation operation

4.2 Conflation Approach

After contacting the Area Service Register the federation finds out that there is no data set within Nexus covering the whole query region by simple intersection operations between the query area and the extents of the Augmented Areas available. Then, those AAs are determined that best fit the requirements of the query and that demand the minimal effort with respect to conflation. In the scenario above, these are the Augmented Areas X and Y. Up to now, we have only investigated conflation operations between two data sets, but basically the approach could also be applied if multiple AAs had to be merged.

When the relevant AAs have been determined, the federation requests the appropriate clippings of these data from the responsible Context Servers. Additionally, the MRep Relations which are referencing objects of the two data sets are loaded by the federation. As it was mentioned before, they are spatially indexed and thus can be found efficiently. With the collected information, the federation node can perform the conflation process and create the result desired by the client.

In the following sections, first the client interface is presented and then the conflation algorithm and its results are presented.

4.2.1 Client Interface

When the client queries the Nexus platform, some information has to be provided in case a conflation operation is necessary to produce the appropriate result. A sample application has been developed which illustrates some of the options of the user (see Fig. 6). These options are:

- Geometry options:
  - G_INCLUDE_ALL_UNMERGED: All geometries of representations from both Augmented Areas that are not linked by MRep

Relations, i.e. that do not have a counterpart in the other AA as well as the multiple representations are included in the result set. No merge operation is performed.

- G_INCLUDE_ALL_MERGED (MERGE_PARAMETER): All geometries of representations from both Augmented Areas that are not linked by MRep Relations, i.e. that do not have a counterpart in the other AA as well as the merged geometries of multiple representations are included in the result set. How this merge is to be carried out can be further specified by a parameter. E.g. of the two geometries provided by multiple representations, the one is selected as the resulting geometry that has the higher positional accuracy, that is the most current version, that has the most vertices or that stems from a certain source. Also, a mean geometry could be computed.

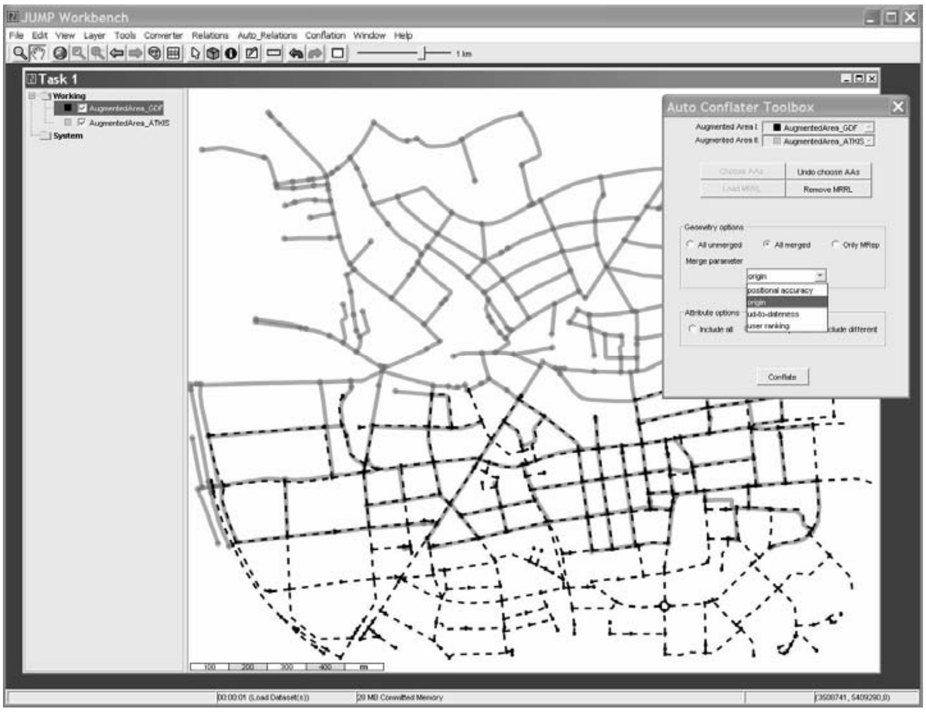

Fig. 6. Sample application showing the client interface for the conflation process as well as the test data (ATKIS grey, GDF black and dotted); MRep Relations have been created for the multiple representations in the overlapping area

- G_INCLUDE_MREP (MERGE_PARAMETER): Only the merged geometries of multiple representations are contained in the result set, i.e. single representations both in the overlapping and in the non-overlapping areas are eliminated. Again, the result of the merge operation can be specified by a parameter. This case appears when the query area is covered by both AAs.

- Attribute options:
  - A_INCLUDE_ALL: The result set contains all attributes of MRep source objects; notice that the same attributes of multiple representations have equal attribute values due to the consistency restoration mechanisms within Nexus.
  - A_INCLUDE_EQUAL: Only equal attributes are contained in the resulting objects.
  - A_INCLUDE_DISPARATE: Only those attributes are contained in the result set which are different in both representations

4.2.2 Conflation Algorithm

The algorithm that performs the merge operation is realized as the G_INCLUDE_ALL_MERGED option of 4.2.1 for realizing the sample scenario and takes both the Augmented Area of ATKIS origin (AA_ATKIS) and the Augmented Area of GDF origin (AA_GDF) as well as the necessary MRep Relations as input information.

If these data are available, the conflation algorithm starts (see Fig. 7, stage I). First of all, the multiple edge representations in both Augmented Areas are used in order to align the Augmented Areas in the overlapping region. For this purpose, the arithmetic means of the point coordinates of the start and end nodes belonging to multiple edge representations are calculated. Of course, this could also be done by a weighted function instead of the arithmetic mean, taking the positional accuracy of the representations into account. In the following step, the transformation vectors are computed, which are directed both from the start and end points of the multiple representation edges to the respective arithmetic mean points. If these transformation vectors have been derived for all multiple representations, a rubber sheeting transformation is performed to achieve that all corresponding representations do have equal start and end points. Finally, it is decided which of the two representations yields the final geometry and is picked as the resulting representation in the merged data set. As it was mentioned in the previous section, different strategies are applicable here based on the merge parameter. Figure 7, stage II, shows potential results: either the rubber sheeted geometry representations of ATKIS or GDF origin could be taken or a mean (or average) geometry could serve as the geometry of the resulting object. In a first, simple approach, we chose to take the geometry of ATKIS origin.

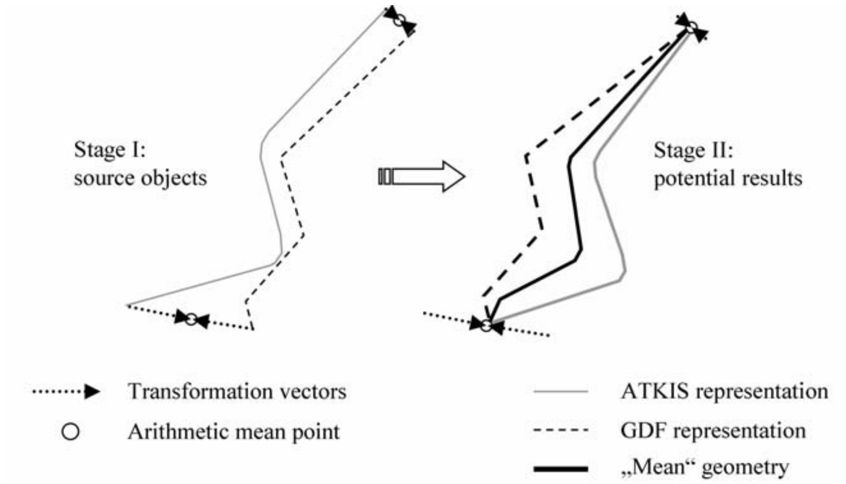

Fig. 7. The merging algorithm computes a single representation out of two multiple representations

4.2.3 Results of the Conflation

The algorithm that performs the conflation has been implemented within the JUMP environment [14]. The result of the process is shown in Fig. 8 both for the whole test data set (also depicted in Fig. 6) and for an enlarged test scene within the test data, displaying the state before (A1, B1) and after (A2, B2) the conflation process.

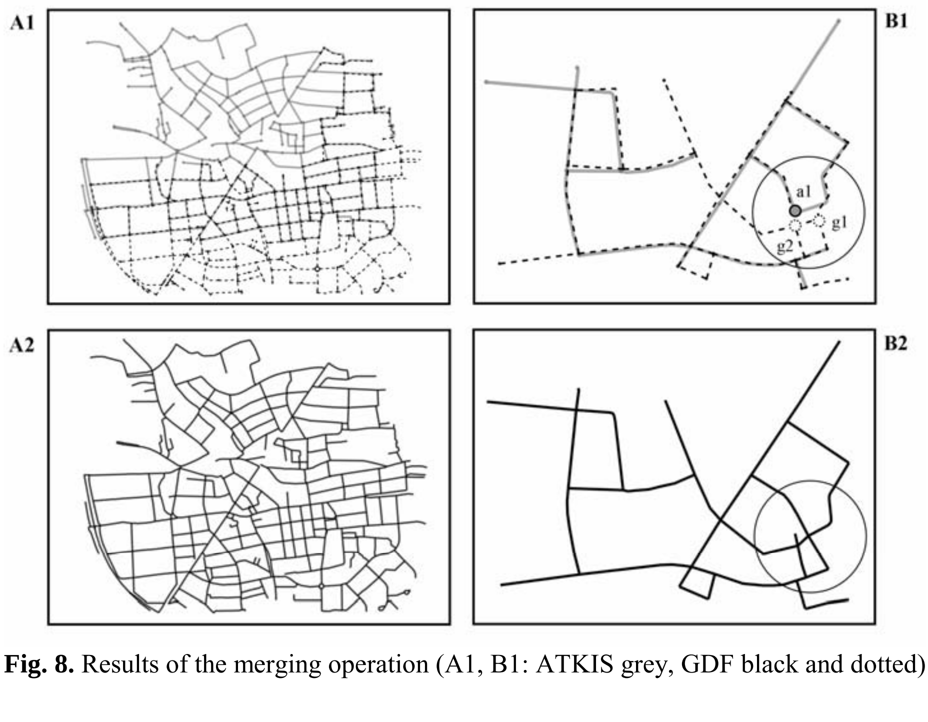

Fig. 8. Results of the merging operation (A1, B1: ATKIS grey, GDF black and dotted)

Basically, the approach works well and yields a visually acceptable result. However, we have not derived criteria based on which the quality of the conflation process can be evaluated, yet. Performing an evaluation of the conflation process and measuring the uncertainty that arises within conflation operations are some of the primary issues with respect to future research.

What can be said by now is that clearly the most difficult situations arise whenever the data sets are strongly inconsistent. In those cases, the matching very likely does not yield a reliable result or cannot produce a result at all, thus providing only a weak basis for the following conflation process. When no matching result can be generated, redundant objects are introduced in the conflated data set. In cases where an erroneous matching result is produced, also the conflation cannot work properly. We encountered one such situation in the test area, marked by the circle in B1 and B2. Here, both the GDF nodes g1 and g2 were incorrectly assigned to the same ATKIS node a1, resulting in a wrong rubber sheeting transformation and thus producing a wrong conflation result. More of such problems will have to be revealed in the future by applying the presented approach to larger data sets and to data sets of different types (e.g. polygonal geometries of buildings) in order to obtain a more generic solution.

5 Conclusion and Outlook

In this paper it has been demonstrated how multiple representations can be dealt with in open geospatial data infrastructures which are organized as federated systems. We especially focused on the Nexus approach to relate multiple representations by explicit relations (MRep Relations) that contain similarity measures for describing the degree of consistency of corresponding geospatial objects. In order to keep the Nexus databases consistent, MRep Relations are utilized for update procedures. For generating consistent and consolidated data sets that merely contain one single representation (SRep) for each real world phenomenon, MRep Relations are exploited in conflation algorithms based on rubber sheeting techniques. It is obvious that the quality of such a data fusion is very much depending on the result of the matching phase. By now, mechanisms that are able to evaluate the quality of the conflation process and the uncertainty of the conflation result are not available. It is one of the main future goals to develop adequate approaches for this purpose.

Furthermore, in this work only the merging of representations which are available in the same scale has been addressed. In future approaches, multi-scale issues shall be handled as well, both with respect to matching and to conflation. Additionally, research on updating multiple representations reveals further challenges: first, propagating updates from a large-scale to its corresponding small-scale representation will be a major research objective. Second, the fact must not be neglected that inconsistencies can be caused while modifying properties of multiply represented objects that might trigger updates themselves (like in a chain reaction), e.g. if the geometry of a street has been changed and this new geometry intersects a building in the Augmented Area, this conflict has to be solved, too.

References

1. ANZLIC: Homepage of the Spatial Information Council for Australia and New Zealand, http://www.anzlic.org.au, accessed: 04.2005
2. Arbter, B., Drosdol, T., Dürr, F., Großmann, M. Hönle, N. Volz, S.: Das Nexus Relationen- und Topologiekonzept, University of Stuttgart: Sonderforschungsbereich SFB 627, Technical Report Nr. 2005/01, (2005), 32 p.
3. Balley, S., Parent, C., Spaccapietra, S.: Modelling Geographic Data with Multiple Representations, Int. J. of Geographic Information Systems 18 (4), (2004), pp. 327-352.
4. Cobb, M., Chung, M., Miller, V., Foley, H., Petry, F., Shaw, K.: A rule-based approach for the conflation of attributed vector data, GeoInformatica 2(1), (1998), pp. 7-35.
5. Comité Européen de Normalisation (CEN): Geographic Data Files Version 3.0, TC 287, (1995), GDF for Road Traffic and Transport Telematics.
6. Conrad, S.: Schemaintegration – Integrationskonflikte, Lösungsansätze, aktuelle Herausforderungen, Informatik – Forschung & Entwicklung, 17, (2002), pp. 101-111.
7. Devogele, T.: A new Merging Process for Data Integration Based on the Discrete Fréchet Distance. In: Richardson, D., van Oosterom, P. (eds.), Proceedings of the 10th International Symposium on Spatial Data Handling (SDH) 02, Ottawa, Springer, (2002), pp. 167-181.
8. FGDC: Homepage of the Federal Geographic Data Committee of the United States, http://www.fgdc.gov, accessed: 04.2005
9. Filin, S., Doytsher, Y.: Detection of Corresponding Objects in Linear-Based Map Conflation. Surveying and Land Information Systems, Vol. 60, No. 2, (2000), pp. 117-128.
10. Friis-Christensen, A., Skogan, D., Jensen, C. S., Skagestein, G., Tryfona, N.: Management of Multiply Represented Geographic Entities. In: Proceedings of the 6th International Database Engineering & Applications Symposium, Edmonton, Canada, July 17-19th, (2002), pp. 150-159.
11. Google Earth: Homepage of Google Earth, http://earth.google.com/, accessed 02.2006
12. GSDI: Homepage of the Global Spatial Data Infrastructure Association, http://www.gsdi.org, accessed: 04.2005
13. Haunert, J.-H.: Link based Conflation of Geographic Datasets. In: Proceedings of the ICA workshop on Generalisation and Multiple Representation, A Coruña, Spain, (2005).
14. JUMP, Homepage of the Java Unified Mapping Platform, http://www.jump-project.org/, accessed 02.2006
15. Kundu, S.: Conflating two Polygonal Lines. Pattern Recognition 39(3), (2006), pp. 363-372.
16. Mantel, D., Lipeck, U.: Matching Cartographic Objects in Spatial Databases. In: Proceedings of the XXth ISPRS Congress, Comm. IV, Istanbul, Turkey, (2004), pp. 172-176.
17. Nexus: Homepage of the Nexus Project of the University of Stuttgart, http://www.nexus.uni-stuttgart.de, accessed: 02.2006
18. Nicklas, D., Mitschang, B.: On Building Location Aware Applications Using an Open Platform Based on the Nexus Augmented World Model, Software and Systems Modeling, Vol. 3, No. 4, (2004), pp. 303-313.
19. OpenGIS Consortium: OGC Abstract Specification, Topic 8: Relationships between Features, Version 4, Project Document Number: 99-108r2, (1999), Datum: 1999-03-26. http://www.opengis.org/techno/abstract/99-108r2.pdf
20. Parent, C., Spaccapietra, S., Zimanyi, E.: MurMur: Database Management of Multiple Representations. In: Proceedings of the AAAI-2000 Workshop on Spatial and Temporal Granularity, Austin, Texas, July 30th, (2000)
21. Saalfeld, A.: Conflation: Automated Map Compilation. Bureau of the Census, Statistical Research Division report series, report number: Census/SRD/RR-87/24, (1987)

22. Stigmar, H.: Matching Route Data and Topographic Data in a Real-Time Environment. In: Proceedings of the 10th Scandinavian Research Conference on Geographical Information Science (SCANGIS) ’05, June 13th to 15th Stockholm, Sweden, (2005), pp. 89-107.
23. Volz, S., Walter, V.: Linking different geospatial databases by explicit relations. In: Proceedings of the XXth ISPRS Congress, Comm. IV, Istanbul, Turkey, pp. 152-157. (2004), pp. 445–473.
24. Volz, S.: Data-driven Matching of Geospatial Schemas. In: Cohn, A.G., Mark, D.M. (eds.): Spatial Information Theory. Proceedings of the International Conference on Spatial Information Theory (COSIT '05), Ellicottville, NY, Lecture Notes in Computer Science 3693, (2005), pp. 115-132.
25. Volz, S.: Shortest Path Search in Multi-Representation Street Databases, In: Proceedings of the 3rd Symposium on Location Based Services and TeleCartography, Vienna, Austria, (2005b), pp. 125-130.
26. Volz, S.: An Iterative Approach for Matching Multiple Representations of Street Data. Proccedings of the JOINT ISPRS Workshop on Multiple Representations and Interoperability of Spatial Data, Hannover, Germany, February 22nd to 24th ’06, (2006), pp. 101-110.
27. Walter, V., Fritsch, D.: Matching Spatial Data Sets: a Statistical Approach. Int. J. Geographical Information Science 13(5), (1999), pp. 445–473.
28. Working Committee of the Surveying Authorities of the States of the Federal Republic of Germany: Authoritative Topographic Cartographic Information System (ATKIS), (1988), http://www.atkis.de
29. Zhang, M., Shi, W., Meng, L.: A Generic Matching Algorithm for Line Networks of Different Resolutions. In: Proceedings of the ICA Workshop on Generalisation and Multiple Representation, A Coruña, Spain, (2005).

Acknowledgements

The research presented here is supported within a Center of Excellence called “Spatial World Models for Mobile Context-Aware Applications” under grant SFB 627 by the German Research Council. The test data have kindly been provided by the NAVTEQ company and the state survey office of the federal state of Baden-Wuerttemberg.