

MRDB APPROACH TO HANDLE AND VISUALISE MULTIPLE DLM'S IN A CONSISTENT WAY

K.-H. Anders^{a,*}, I.Ö. Bildirici^b

^a Institute for Cartography and Geoinformatics, University of Hannover, Germany –
Karl-Heinrich.Anders@ikg.uni-hannover.de

^b Selcuk Universitesi, Mühendislik-Mimarlik Fakültesi Jeodezi ve Fotogrametri Müh. Bol., Konya, Turkey –
bildirici@selcuk.edu.tr

Commission VI, WG IV/3, IC WG II/IV

KEY WORDS: Cartography, GIS, Multiresolution, Generalisation, Matching, Database System, Visualization

ABSTRACT:

A Multi-resolution/representation-database (MRDB) can be described as a spatial database, which can be used to store the same real-world-phenomena at different levels of precision, accuracy and resolution. Furthermore these phenomena can be stored in different ways of presentation or symbolisation. In an MRDB, different views on the same physical objects or phenomena can be stored and linked. This variety of sights can stem from different views of the world, different applications, as well as different resolutions. These lead to differences in the objects as such, i.e. in the semantics and in the geometry. There are several reasons for introducing a MRDB: On the one hand it allows a multi-scale analysis of the data: Information in one resolution can be analysed with respect to information given in another resolution. On the other hand a major reason for National Mapping Agencies to investigate and implement MRDB is the possibility of propagating updates between the scales, which is also called “incremental generalisation”. In a cooperation with the German Federal Agency for Cartography and Geodesy (BKG) we are developing at the ikg an automatic model generalisation and update (data revision) system for the German ATKIS (Authoritative Topographic-Cartographic Information System). In this paper we will describe the structure and functionality of our ATKIS MRDB, which is based on an Oracle database system as storage component and the ESRI ArcGIS system as the user interface to the MRDB.

1. INTRODUCTION

National mapping agencies (NMA's) in general and nowadays also commercial spatial data provider have to provide digital map series with different scale of the same area. These leads to the problem to update all the digital map series in a consistent way. Updating all related maps independently is an time and money consuming process because the amount of work to guarantee the consistency of all related data sets is high. To make this process more efficient it should be possible to change manually only the data set with the highest resolution (largest scale) and all other data sets with a smaller scale will be generated automatically from this base data set.

In a project financed by the German Federal Agency for Cartography and Geodesy (BKG) we are developing at the institute for cartography and geoinformatics (ikg) together with the institute for database systems (dbs) an automatic model generalisation and update (data revision) system for the German ATKIS (Authoritative Topographic-Cartographic Information System). The ATKIS database includes four digital landscape models (DLM): BaseDLM (1:10.000-1:25.000), DLM50 (1:50:000), DLM250 (1:250.000), DLM1000 (1:1000.000). All these four DLM's have to be stored in one logical database with tools for the automatic update of the DLM's from the BaseDLM. Therefore we are developing integration tools, model generalisation tools, and visualisation components.

2. MRDB

An MRDB can be described as a spatial database, which can be used to store the same real-world-phenomena at different levels of precision, accuracy and resolution (Devoegele, 1996; Weibel, 1999). It can be understood both as a *multiple representation database* and as a *multiple resolution database*. In an MRDB, different views on the same physical objects or phenomena can be stored and linked. This variety can stem from different views of the world, different applications, as well as different resolutions. These lead to differences in the objects as such, i.e. in the semantics and in the geometry. Also the graphic representation can be taken into account, leading to geometric, semantic and graphic multiplicities (Bédard, 2002). There are two main features that characterise an MRDB:

1. Different levels of detail (LoD's) are stored in one database and
2. The objects in the different levels are linked.

The first feature can be compared to the analogue map series of the NMA's: these maps of different scales exist separately, only implicitly linked by the common geometry. In the second case, however, individual objects are explicitly linked with each other and thus each object “knows” its corresponding objects in the other representations.

* Corresponding author.

2.1 Federated Database

The MRDB approach we have developed is based on the architecture of a *federated database system (FDBS)* (Conrad, 1997; Sheth, 1990). Starting point for a FDBS are several existing databases which should work together to provide a global service, but keep their own local autonomy. A FDBS can be classified by three main characteristics: *Distribution*, *Heterogeneity*, and *Autonomy*. Distribution means that the data is stored on different database systems (DBS), which can run on a single computer system or in a distributed computing environment. Heterogeneity, because the DBS can use different database management systems (DBMS) and finally Autonomy, because the different DBS can be designed independently from each other and it is possible that each DBS runs independently from the *Federation Layer* (no changes of local application programs are necessary). Figure 1 shows the general architecture of a federated database system.

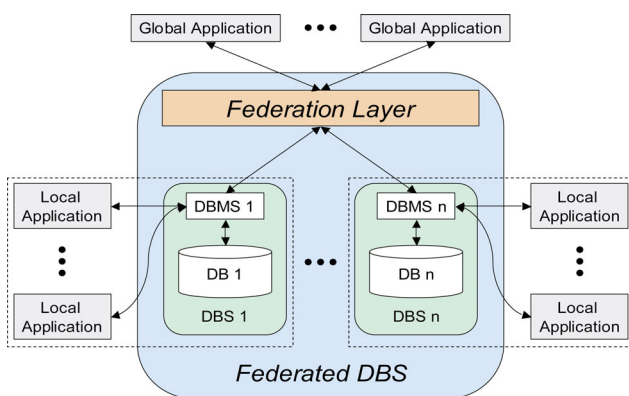


Figure 1. General architecture of a federated database system.

2.2 System architecture

Figure 2 shows the complete system architecture of the MRDB. The Federation Layer of the FDBS is lower green block in Figure 2. Every component DBS in WIPKA stores one specific representation layer (BaseDLM, DLM50, DLM250, or DLM1000). The Federation Layer integrates the component DBS's to an MRDB and provides the user interface to the MRDB. Therefore the Federation Layer contains a *working database* to store all needed meta data. These meta data includes the registration of the component databases and the links between them. The working database stores also all information for the object identification and propagation of updates through the component databases.

Access to the Working database and the link information is provided by a PL/SQL interface which can be used by an database connector (e.g., ADO, ODBC, JDBC). Access to the spatial data can be done directly using oracle interfaces or by ArcSDE from Esri for ArcGIS. ArcSDE is a generic database interface which can be used to connect ArcGIS to the most commercial database systems. The most functionality of this interface layer is implemented as *stored procedures* directly on the Oracle database system which is an problem for ArcSDE if the called stored procedure has a return value. That is the case if one is querying the linked objects to a given object id. Therefore one has to use in VBA (Visual Basic for Applications) the ADO (Active Data Object from Microsoft

interface to call such kind of stored procedures. VBA is the main macro programming interface of ArcGIS.

Based on this interface to the working database all MRDB applications are implemented like matching procedures, generalisation tools, and graphical user interfaces.

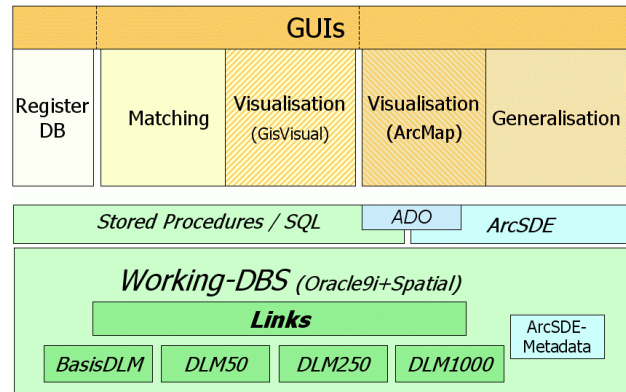


Figure 2. System architecture of the MRDB.

3. GENERATION

In general there are three possibilities to generate data for an MRDB: manual linking, linking by matching, and linking by generalisation.

3.1 Manual linking

Manual linking means that the MRDB system provides interactive tools for selection and linking / unlinking of data objects with different representations from different data sets. The manual linking will always be the case if two or more existing data sets have to be integrated into an MRDB and if no automatic method is available or successful.

3.2 Linking by matching

The integration of several existing large data sets into an MRDB completely by hand would be to inefficient, therefore one has to use automatic matching tools to create links between the data sets. In our system we will provide different kind of matching tools. There will be basic *semantic filters*, *geometric filters*, *topological filters*, and *relational matching procedures*. A semantic filter is a kind of simple matching routine, because he is comparing only the attribute values of objects, e.g. object types like residential area, industrial area, motorway, or highway. A geometric filter compares only the geometry (shape and position) of the objects and a topological filter compares the topological relations between simple and aggregated objects. The relational matching procedures are combination of all the basic filters to complex matching routines which are taking into account all information available for an object. For now a relational matching procedure for line objects was implemented and a simple geometric matching routine for area objects. The relational matching routine is using semantic filters to create a first selection of possible matching candidates to reduce the search space. The simple area matching procedure is based on the size of the intersection area of two possible candidates. The shape of both objects is not used for the moment. We have tested this matching routine by linking

building ground plans with settlement areas of the ATKIS BaseDLM (Figure 3).

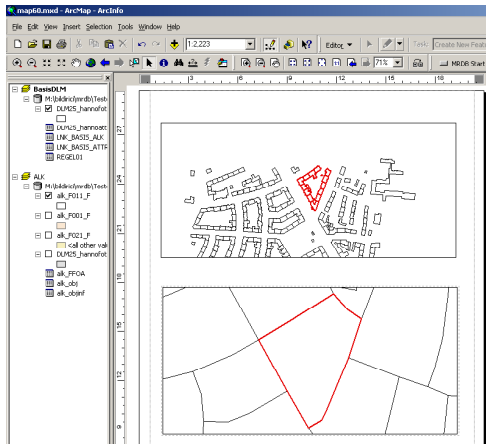


Figure 3. Ground plans linked with an ATKIS settlement area.

3.3 Linking by generalisation

The third possibility to generate MRDB data is the generation of object links during the automatic creation of another object representation from a reference object. This is needed if a completely new data set has to be derived from a base data set or in the case of automatic updating. As an example for the medium to low resolution model generalisation we studied the automatic derivation of an ATKIS DLM50 (1:50.000) database from an ATKIS BaseDLM (1:10.000-1:25.000) database. The ATKIS DLM50 object catalogue describes the object types which should be captured in the DLM50 and which geometries types should be used to represent the real world objects. Therefore in general we have to deal with area, line, and point generalisation, but for our first studies we concentrated on the subject of area generalisation. These operations are implemented for the derivation of the DLM50, they are however generic enough to be transformed to representations of arbitrary scales. The area generalisation process consist of three steps:

1. Reclassification of the object types
2. Aggregation of adjacent areas with equal object type
3. Shape generalisation

The first step is needed because some object types of the ATKIS BaseDLM do not exist anymore in the ATKIS DLM50 landscape model. The second step is needed because in the ATKIS DLM50 the minimum size criteria for capturing of certain object types as area objects has increased. The third step handles all cases of area objects which are still not big enough after the second step to be captured as areas in the ATKIS DLM50 model. In such cases according to the ATKIS DLM50 object catalogue these areas has to be represented by a point or have to be eliminated completely. In both cases one has to establish a reclassification and an additional aggregation step to these areas. This additional reclassification and aggregation step can be done in different ways. Four possible solutions are shown in Figure 4. The replacement of an area can be done “by definition” which means that a priority list of new object types is given that describes which new object type has to be used to replace the old object type relative to the adjacent areas (Figure 4a).

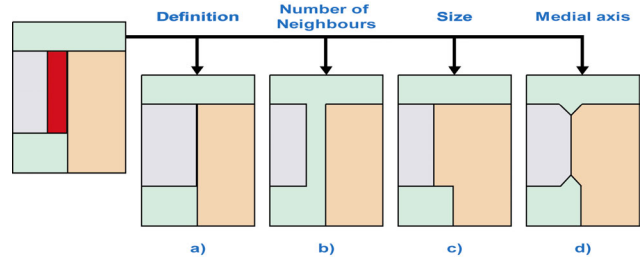


Figure 4. Possible solutions for aggregation of area objects.

E.g., if an area with object type *farmland* has to be replaced then a possible priority list can be: 1. grassland, 2. garden area, 3. area without vegetation, and so on. That means if an adjacent area has the type grassland then the area will be reclassified as grassland. If no adjacent grassland can be found then one has to look for an area without vegetation. If there is no such area then may be there is an area of type indefinable area, etc. The priority list has to make sure that always a new object type can be found. Another way is to choose the most frequent object type of the adjacent neighbour areas (Figure 4b) or to choose the object type of the largest adjacent neighbour area (Figure 4c). A more sophisticated approach is to compute the skeleton (e.g., medial axis, straight skeleton) of the area which has to be replaced and to increase all adjacent areas according to the computed skeleton (Figure 4d) (Bader, 1997). The maximum number of equal neighbours and the maximum size approach have the same drawback, that this maximum number must not be unique. E.g., an area can have as many neighbours of object type X as of object type Y. It is also possible that an area has more than one adjacent neighbour areas with the same size but different object types. These ambiguities can be solved in the most cases by combining the criteria's (number and size of), but in general there can be still ambiguous cases. The approach with the skeletons takes all neighbour areas in account and increases all neighbour areas relative to the shape of the area which has to be replaced. The drawback of this approach is that it is more complicated to be implemented and it can produce artificial shapes, like the two small triangles in figure Figure 4d. We have implemented the first approach using priority lists from the NMA's and a straight skeleton approach to aggregate areas without prior knowledge. Figure 5a and 5b shows an example for this type of area aggregation. During this aggregation process the links between the original areas of medium resolution and the new areas of low resolution are stored in the link table of the MRDB federation layer.

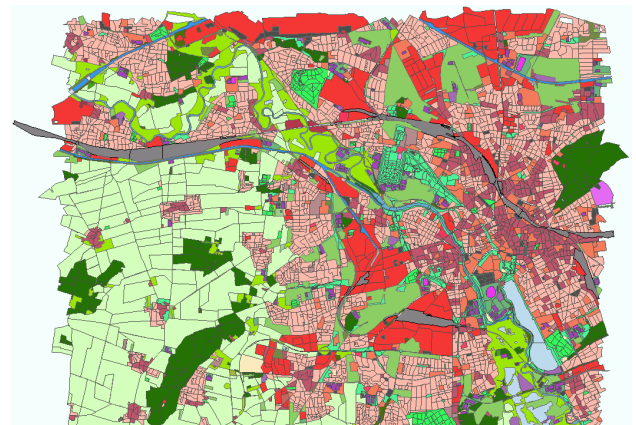


Figure 5a. Example: Before rule based aggregation of areas.

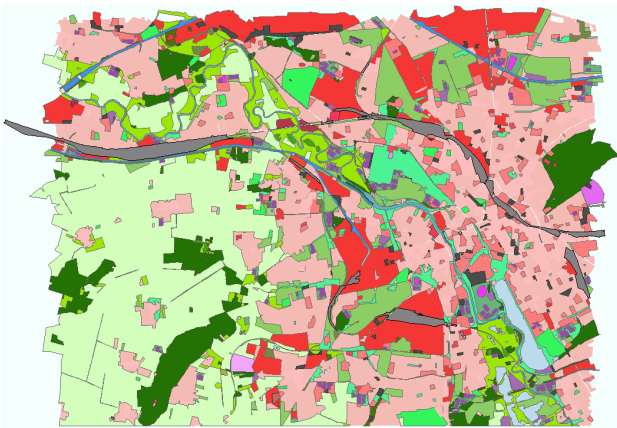


Figure 5b. Example: Result of rule based aggregation.

4. VISUALISATION

Commercial GIS systems like the used ArcGIS system from ESRI did not provide standard tools to display and handle MRDB data. Therefore we have developed at the ikg a tool for ArcGIS in VBA (Visual Basic for Applications from Microsoft). The MRDB tool (Figure 6) allows to display all four ATKIS DLM's in one overview and the selection and linking of ATKIS objects. One can select one or more objects in one of the four DLM's manually or by a query and the corresponding objects (linked objects) in another DLM will be displayed automatically.

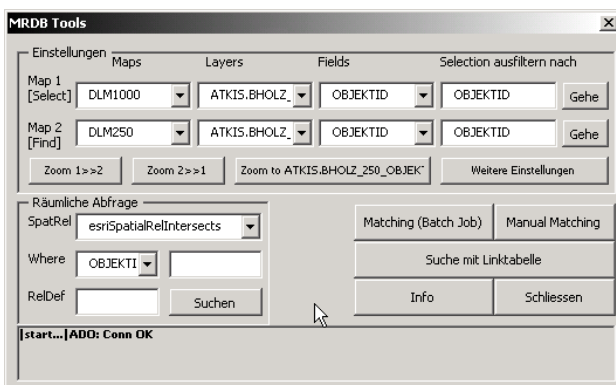


Figure 6. Main window of the MRDB tool.

The connection between the MRDB tool and the MRDB is based on ADO (Active Data Objects from Microsoft) and ArcSDE. The technical problem was how to display several DLM's at the same time in ArcGIS. There are two views in ArcGIS. The *data view* which should be used for interactive processing and the *layout view* which is used to control the map layout for printing purposes. The problem which occurred was that the more flexible data view only displays the active data set (the data set the user can select something). Several windows in the data view at the same time like a multi document interface is not supported by ArcGIS in version 8.3. Fortunately the layout view allows to display several data sets at the same time in so called *data frames*. In a data frame one can select objects also manually and one can control a data frame with a program. Therefore the layout view is used in the MRDB tool. Figure 7 shows the presentation of the four ATKIS DLM's in the layout view of ArcGIS.

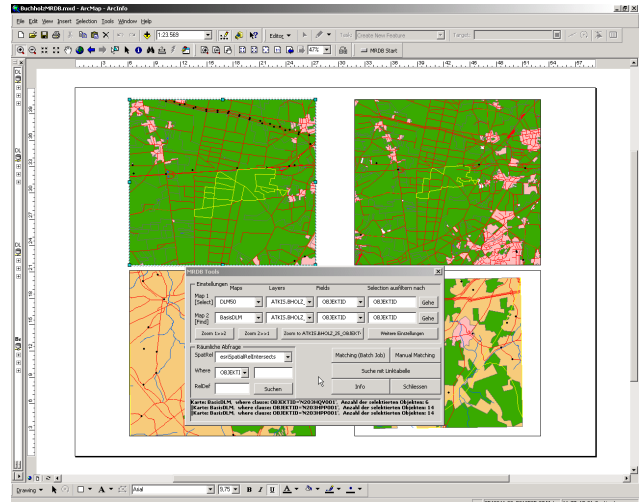


Figure 7. Layout view of the MRDB tool.

5. UPDATE

The goal of the WIPKA project is to implement an MRDB to enable an automatic incremental update process of all representation levels in the MRDB. The idea is to update only the lowest level with the highest geometric and semantic resolution (in our case is that the ATKIS BaseDLM). All other levels should then be updated automatically at most as possible to make the data revision process faster and more efficient and consistent. We are now developing a message passing system which will allow us to control the update process by propagating the changes through all levels according to the stored MRDB links and to trigger appropriated generalisation operators. We are in the early stage of these project, we can only describe the rough concept which we want to implement. An MRDB can be seen as a directed Graph (Figure 8 shows this situation schematically) which provides information about the relationship between objects. These relationships can be used to propagate updates bottom up through the representation levels. In our concept we define three main types of update events which are propagated through the network:

- Insert: A new object was created.
- Remove: An object was deleted.
- Change: An object has changed.

These changes can be divided up in:

- Change Attribute: Only attribute values are changed.
- Change Geometry: Only the shape of the object has changed.
- Change Attribute and Geometry: The attribute values and the shape have changed.

After the update process of a lower level is completely finished all changes are propagated to the next level according to the stored links. The update propagation has to be understood as calling an appropriate generalisation method for the linked object at the next level. For the update process the system will provide a set of generalisation operators which are selected by a rule based system. That means the user has to maintain a set of rules which describes what operator should be used under certain conditions. These conditions are the type of the update

event, the type of the updated object and the object which has to be changed, and thresholds for the attribute and geometry changes. Conflict situations which can not be handled automatically by the system are stored and after the processing graphically presented to an user in order that he can decide to solve a conflict situation interactively or by changing the update rule set. The update propagation will be repeated until we have reached the final representation level (in our case the DLM1000). The rule based system which controls the update process will be part of the federation layer in the MRDB.

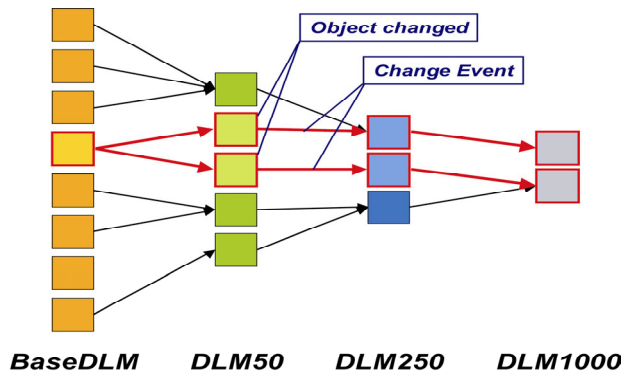


Figure 8. Update propagation in the MRDB.

6. CONCLUSION

In this paper we have described the structure of an MRDB system based on Oracle and ESRI software that establish a consistent data structure for topographic data sets in Germany, and the possibility of propagating updates and thus greatly simplifying the update process. For now we have no complete system implemented. We have implemented the MRDB database schema, some matching tools, some generalisation tools, and a graphical user interface. In the moment we are implementing the automatic update system based on the described schema. Additionally the matching and generalisation tools are extended to cover all object and geometry types of ATKIS.

7. REFERENCES

- Bader, M., 1997. Methoden zur Erkennung und Lösung von metrischen Konflikten in der Generalisierung von Polygonmosaiken, MSc Thesis, Department of Geography, University of Zurich.
- Bédard, Y. and Bernier, E., 2002. Supporting multiple representation with spatial databases views management and the concept of "VUEL", ISPRS / ICA Joint Workshop on Multi-Scale Representations of Spatial Data, Ottawa, Canada.
- Conrad, S., 1997. Föderierte Datenbanksysteme : Konzepte der Datenintegration. Springer-Verlag, Berlin..
- Devogele, T., Trevisan, J. and Raynal, L., 1996. Building a Multiscale Database with Scale-Transition Relationships: In Proc. Of the 7th Int. Symposium on Spatial Data Handling, Advances in GIS Research II, pp. 6.19-6.33, Delft.
- Sheth, A. and Larson, J., 1990. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Surveys 22:3, pp. 183-236.

Weibel, R. & Dutton, G., 1999. Generalising spatial data and dealing with multiple representations. In: P.A. Longley, M.F. Goodchild, D.J., Maguire & D.W. Rhind, editors, Geographic Information Systems – Principles and Technical Issues, volume 1., John Wiley & Sons, 2 edition, pp. 125–155.

8. ACKNOWLEDGEMENT

The funding of the Federal Agency of Cartography and Geodesy, Germany, is gratefully acknowledged.