Applying DLM and DCM concepts in a multi-scale environment

Martijn Meijers and Peter van Oosterom

1 Introduction

Although the separation between Digital Landscape Model (DLM) and Digital Cartographic Model (DCM) is considered as state of the art, data producers, like national mapping agencies, still wrestle with the question what to store explicitly in order to efficiently maintain their geographic databases and maps (cf. Stoter et al., 2008). In this paper we will try to show that explicit storage of both models, up to the data instance level, leads to more redundancy in multi-scale data models and makes it more difficult to manage geographic databases. To streamline the process of data production for both analysis and map making purposes, we propose to maintain only the data instances of the DLM, including minor 'distortions' to apply visualization rules easier, and to investigate variable scale data storage.

2 Where to put the balance between DLM and DCM

In a nutshell, geographic data producers are dealing with data capture, data management and visualizations. For creating a digital geographic database (the DLM), objects will be captured in the real world with certain rules applicable for the data capturing process. Besides rules for geometry and topology, like minimum size, geometric accuracy and connectivity of objects, these rules also include object classification and population of attributes carrying thematic semantics. From the database, objects can be selected to produce digital maps. The transformation of objects from the DLM to a visual end-product is described in a DCM. However, as this transformation process is not in all cases straightforward, data producers face the problem of what to store and maintain, the geographic objects or also the map objects resulting from this transformation? To make the different choices clear, we will first consider the different ways that exist to treat the objects at one scale only:

1. It is possible to only store the digital map objects. This is not an optimal solution in the sense that it mixes the representation of 'real world' objects and its visualization and makes it difficult to use the objects for geographic analysis.
2. One can store the geographic objects persistently and derive the map objects in an automated way when needed. In this case, the geographic objects are thus not adapted *at all* for any kind of visualization. Although a lot of research has been carried out, a fully automated solution still has not been reached for such a setup. Operations that are difficult to manage in the transformation from geographic objects to map objects are for example displacement and typification.
3. Another option is to store the geographic objects as well as the map objects explicitly. Both models are thus instantiated and made persistent. This allows for fast access to both the geographic objects, as well as the map objects, but comes with a price of redundancy. In order to maintain consistency more easily, links can be created between the counterpart instances in both representations.
4. The last option is to adapt the geographic objects for a default visualization. This differs from the second solution, because non-straightforward visualization aspects, e.g. displacements and typification, are performed and the results of those operations are explicitly stored *as* the geographic objects. This change of geographic objects should take place in accordance with tolerances specified in the capturing rules with respect to the desired quality of the dataset. Our motivation to allow this kind of distortion is that other geometric distortions take place within a dataset anyhow; e.g. simplification, aggregation, or complete removal of objects. The map objects are still generated on-the-fly by applying relatively simple visualization rules. A downside of this solution is that problems might still arise when a visualization significantly different from the default visualization is used with the geographic objects.

3 DLM and DCM in a multi-scale database

In a multi-scale database the choice how to deal with the geographic objects and map objects is a bit more complicated in comparison when looking at a single scale only. For example, when method three is adapted, both models have to be maintained. In a multi-scale setup this doubles the amount of objects that need to be maintained (per scale the geographic and the map objects). As storage space tends to become cheaper and cheaper

Dagstuhl Seminar Proceedings 09161
Generalization of spatial information
http://drops.dagstuhl.de/opus/volltexte/2009/2135

this is not really a problem. However, there is also a trend for data producers having to provide higher rates of updates: maintaining more data means more work, so it would be beneficial to minimize the amount of redundancy. Another problem is that of inconsistencies between the different datasets: if maintenance takes place in a not-fully-automated fashion, it is possible for geographic and map objects to become out of sync with each other (and tell different stories about the same reality). All in all, the fourth solution we sketched above seems more appropriate: Within the quality bounds required for a geographic dataset, the geographic objects should be adapted to make a default visualization easily possible. If a user requires higher (geometric or thematic) accuracy, then the multi-scale setup allows to go down the scale dimension to a larger scale dataset and select a more detailed dataset for the application.

4 Variable-scale as ultimate solution?

From a data producers perspective, this multi-scale setup might still not be ideal. We can extend the line of thinking for the scales stored in the database: If certain features are present at multiple scales, then why store these representations redundantly? Variable-scale data structures have been proposed (Van Oosterom et al., 2006) and tested (Hofman et al., 2008), to provide an answer. Two advantages of variable-scale data structures are: 1. no, or at least very little, redundancy between scales and 2. also the possibility of 'in-between scales' representations, not only the fixed, stored representations.

Although variable-scale structures might seem to be an optimal solution, in case of certain generalization events it still can be necessary to include a second representation, because otherwise the representation in the structure (and/or selection at the required scales) will become too complicated. A few examples: 1. it is possible to store both a road area (at the large scales) and a road centerline (at the smaller) scales and link these representations in the structure to the same real world object, 2. certain generalization operations require contextual information and are relatively expensive to compute, in these cases it may be more effective to store a second representation, such as a displaced house, and 3. certain types of concepts occur not at the largest scale, but only at smaller scales (e.g. roundabout composed by several road areas, or building block composed by individual houses and optionally gardens).

In current map products the decisions when to add extra representations are 'black-white' and bound to the maintained scales: road areas are present on 1:1,000 and road areas and roundabouts on 1:10,000, individual houses on 1:1,000, individual houses and building blocks on 1:10,000 and only building blocks in urban area on 1:50,000. We plan to investigate how to embed these semantic aspects in the variable-scale data structures. The nature of objects changes in a continuous way related to geographic scale, and thus there is no reason to take only black-white decisions. It may be far more natural to gradually move from individual houses to building blocks when moving from large to smaller map scales (and thus being able to provide smooth zooming to end users).

Acknowledgements

We like to thank prof. dr. Menno-Jan Kraak for the fruitful discussions we had with him and which provided us with more insights on the topic. Also thanks to drs. Theo Tijssen for proof reading the paper.

References

Hofman, A., Dilo, A., Van Oosterom, P. J. M., and Borkens, N. (2008). Using the constrained tGAP for generalisation of IMGeo to Top10NL model. In Mustiere, S., Mackaness, W., and Oosterom, P. J. M. van, editors, *Proceedings of the ICA Commission on Map Generalisation and Multiple Representation*, pages 1–23, Montpellier.

Stoter, J. E., Morales, J. M., Lemmens, R. L. G., Meijers, B. M., Van Oosterom, P. J. M., Quak, C. W., Uitermark, H. T., and Van den Brink, L. (2008). A Data Model for Multi-Scale Topographic Data. In Ruas, A. and Gold, C., editors, *Lecture Notes in Geoinformation and Cartography*, pages 233–255. Springer.

Van Oosterom, P. J. M., De Vries, M. E., and Meijers, B. M. (2006). Vario-scale data server in a web service context. In Ruas, A. and Mackaness, W., editors, *Proceedings of the ICA Commission on Map Generalisation and Multiple Representation*, pages 1–14, Paris.