International Journal of Geographical Information Science
Vol. 25, No. 5, May 2011, 739–763

A semantic-rich multi-scale information model for topography

Jantien Stoter[a,]*, Thomas Visser[b], Peter van Oosterom[a], Wilko Quak[a] and Nico Bakker[c]

[a]*GIStechnology, OTB, Delft University of Technology, Delft, The Netherlands;* [b]*University of Twente, Enschede, The Netherlands;* [c]*Kadaster, Apeldoorn, The Netherlands*

(*Received 3 July 2009; final version received 27 March 2010*)

National mapping agencies maintain topographic data sets at different scales. Keeping the data sets consistent, for example by means of automated update propagation, requires formal knowledge on how the different data sets relate to each other. This article presents a multi-scale information model that, first, integrates the data *states* at the different scales and, second, formalises semantics on scale *transitions.* This is expressed using the Unified Modelling Language (UML) class diagrams, complemented with Object Constraint Language (OCL). Based on a requirement analysis using the needs of the Netherlands' Kadaster as case study, this article examines several modelling alternatives and selects the optimal modelling approach for a multi-scale information model for topography. The model is evaluated through a prototype database implementation. The results show that UML/OCL provides an appropriate formalism to model rich semantics on both multi-scale data content and scale transitions, which can be used for guarding consistency based on automated generalisation of updates. Further research is required to express generalisation specifications that are currently not formalised and that are only available in software code or as cartographers' knowledge.

Keywords: multi-scale; spatial data modelling; generalisation; knowledge formalisation; multi-scale semantics

1. Introduction

Traditionally, national mapping agencies (NMAs) produce maps at different scales for which they maintain separate data sets. Today's use of topographic data requires consistency between the different data sets, which is not straightforward under conditions of growing data sets and increasing update frequencies.

An important instrument for keeping multi-scale data consistent is a semantically rich multi-scale information model. This model formalises the meaning of topographic objects at different scales. Additionally, it formalises how the object representations at different scales relate to each other.

The realisation of such a multi-scale *i*nformation *m*odel *top*ography (IMTOP) is the topic of this article. IMTOP accomplishes the two main aspects of a multi-scale information model. First, it integrates the data *states* at the different scales. Second, it formalises semantics on scale *transitions*. These two aspects together form what could be called the scale-aware data product specification (i.e. goal of generalisation) and is the important novel contribution of this article. This is expressed using the Unified Modelling Language (UML) class diagrams, complemented with Object Constraint Language (OCL).

*Corresponding author. Email: j.e.stoter@tudelft.nl

ISSN 1365-8816 print/ISSN 1362-3087 online

DOI: 10.1080/13658816.2010.490218
http://www.informaworld.com

Apart from ensuring consistency, formal semantics on scale transitions also supports automated generalisation of topographic data. For current map generalisation processes, no central place exists where all knowledge on scale transitions can be found: part is captured in written map specifications, part is implicitly available in software code and part is present in the mind of the cartographers who interactively perform map generalisation. As a result formal requirements for automated generalisation are not directly available, despite that some automation in generalisation has been introduced by many NMAs (Stoter 2005).

The reasons for these missing requirements are first that currently no formalism has proven to be adequate for capturing the specifications of a map. Second, not all requirements are easy to be formalised. Finally much knowledge on generalisation requirements and processes still needs to be revealed.

IMTOP addresses the first issue by evaluating the use of UML/OCL to add semantics on scale transitions to a multi-scale information model. To design the optimal multi-scale topographic information model, we conducted a requirement analysis using the needs of the Netherlands' Kadaster as case study. We designed, implemented and evaluated several alternatives to meet the identified requirements.

It is important to be aware that the focus of this article is knowledge modelling for consistent multi-scale databases. Therefore the formalisation is meant to evaluate data against the formal specifications that contain knowledge on multi-scale aspects. In a next step (future work) the data that is identified as not valid according to the specification can be further processed in a generalisation system that can use the formal multi-scale knowledge as input to implement the best process. Although it is important to define rich semantics on multi-scale aspects in the model to optimally support generalisation processes and to assure that the generalisation output meets the expectations, the formal definition of the generalisation process itself (i.e. which operators, where and how to apply) is outside the scope of the presented research.

This article is structured as follows. Section 2 presents relevant previous studies on single- and multi-scale data modelling. Section 3 presents the results of the requirements analysis. Section 4 describes the design of IMTOP distinguishing in modelling multi-scale data content and modelling semantics on scale transitions. To evaluate the design of the IMTOP, it is implemented in a prototype in Section 5. The article ends with conclusions in Section 6.

2. Previous studies on single- and multi-scale data modelling

Section 2.1 presents the main approaches for single-scale data modelling. Section 2.2 presents previous studies on multi-scale data modelling.

2.1. Approaches for single-scale spatial data models

Three approaches can be identified for modelling spatial data: (1) geometry/topology-first approach, (2) feature-first approach and (3) a hybrid approach.

To appreciate the differences between the approaches it is important to understand the difference between topological structure, in which topology is embedded in a data structure (e.g. a node–arc–face structure) and topological relationships which is a special case of relationships between features as defined in Abstract Specifications (AS) Topic 8: Relationships Between Features of the Open Geospatial Consortium (OGC 1999). According to these specifications, relationships can be specified through relationship types, degree, roles, role types, feature types, cardinality, directionality and so on. However, embedding these relationships in a data structure is not part of the AS. Also the OGC Implementation Specifications for Simple Features (OGC 2006) only address the topological relationships which can be obtained by 'comparing two geometric objects and then to make pair-wise tests of the intersections between the interiors, boundaries and exteriors'. This is in contrast to a topological structure supporting explicit relationships, for example 'left-polygon', 'right-polygon', 'start-node', 'end-node'.

The geometry/topology-first approach uses the geometry as main entrance, often structured in a topological structure (e.g. a linear network or a partition of space). Attributes are added to these geometries to classify the objects. This approach is for example implemented in the shape file format of ESRI.

The feature-first approach models the feature classes first (i.e. the entrance of the modelling approach is the feature) with added geometry attributes. An example of this approach is the feature model of the Open Geospatial Consortium (see above) and ESRI's geodatabase. Every class may have its own set of thematic attributes, which may vary for the different classes, and every class has its own geometric description. No data structure exists that explicitly specifies spatial relationships between features. Therefore the feature-first model does not contain explicit topological relationships captured in a data structure although these relationships can be specified in the model as constraints and calculated in the database. These relationships are very important for generalisation: for example, What are the neighbours of this instance (candidates for aggregation)? Is the network connectivity damaged when this road segment is removed?

The third (hybrid) approach treats geometry/topology and the feature class equally. It combines the strengths of both approaches: the thematic attributes are specifically designed for every feature class, but the model also enables shared geometry and use of embedded topological structures. The spatial domain is a full partition and is described using tables for nodes, edges and faces (and solids in three dimensions). The instances are modelled in the same way as in the feature-first approach with the exception that objects do not have their own independent geometry attributes, but refer to primitives in the geometry/topology part of the model (node, edge, face, etc.). This approach is described in the 'formal data structure' (FDS) theory of Molenaar (1989) and more recently implemented in products such as 1Spatial's Radius Topology and Oracle's spatial topology (first introduced in version 10g).

It cannot be claimed that one model is 'better' than another model. This depends on the application context and use. It is only possible to identify the optimal approach, after specifying a number of important characteristics of the application domain and typical use (Stoter *et al.* 2008).

2.2. Approaches for multi-scale data models

A multi-scale data model is a specific type of a multi-representation data model (MR-DM). An MR-DM models the support of different database representations of the same real-world object. A multi-scale data model models different database representations that differ mainly in the level of detail.

The issue of multi-representation was introduced in a research program of the National Center for Geographic Information and Analysis (Buttenfield and Delotto 1989, NCGIA 1989). Since then many researchers have focused on MR-DMs.

The multiple representation management system (MRMS) of Friis-Christensen and Jensen (2003) provides MR methods such as 'checkConsistency' and 'restoreConsistency', as well as triggers to execute those methods in case of updates, insertions and deletes, modelled with UML and OCL and implemented on top of Oracle. The *m*odelling of *a*pplication *d*ata with *s*patio-temporal features (MADS) of Parent *et al.* (2006) is based on stamps. One or several stamps can be assigned to classes, relationships, attributes, values and so on to indicate for which map scale the class and so on is relevant. 'Perceptory' (Bédard *et al.* 2004) extends existing UML editors with spatio-temporal icons for modelling multi-representation concepts in methods of the object classes. However, no independent description of multi-representation concepts exists, as in MADS. Jones *et al.* (1996) propose a conceptual model for a multi-representation database as a single database that is capable of storing spatial objects with multiple geometries. This approach does not take into account the complexity of the relationships that can exist in multi-representation (and multi-scale) data sets. For example, two objects that are merged in a generalisation step cannot be adequately modelled. The work of Devogele *et al.* (1996) models map scale transitions, but only between pairs of objects (roads in this case); it does not consider scale transitions influencing objects of different types. The work of Kilpelainen (1997) focuses on the link between object instances when there is an exact dependence among the object classes (e.g. building as complex polygon, building as simple polygon, building as point, building as part of a building area).

Whereas previous studies focused mainly on integration of multi-representations in a multi-scale environment, that is integrating the state aspects of data, our research proposes a formalism also including how a set of topographic object classes should change over scale. This enables meaningful integration of geographic information at different scales as well as automated generalisation.

3. Requirement analysis

This section presents the results of the requirements analysis for IMTOP. Section 3.1 presents the case study and requirements that follow from it. Section 3.2 lists further aspects that outline the scope of IMTOP.

3.1. The case study

The multi-scale environment of the Dutch Kadaster was used to define requirements for IMTOP to identify which multi-scale knowledge should be formally specified with the UML/OCL notation so that a multi-scale database or generalisation system can maintain or process the data accordingly.

The Kadaster produces vector data sets at scales 1:10K, 1:50K, 1:100K, 1:250K, 1:500K and 1:1000K. For map production, the vector data sets for each scale are processed with special visualisation software. The 1:25K map is directly derived from TOP10vector data without generalisation.

Interactive generalisation was introduced in the 1990s and is still the process of today. Cartographers digitally 'draw' the smaller scales from the next larger scale map using written map specifications (Kadaster 2006). Apart from the larger scale map, aerial photos are used as source for cartographers' interpretation of the landscape.

Since 2007, a TOP10NL data set, which is the object-oriented version of TOP10vector, is available for the whole of the Netherlands. The content of TOP10NL is specified in a UML class diagram (TOP10NL 2005) using the feature-first approach. Object-oriented equivalents at the smaller scales are currently being created.

IMTOP must integrate all object-oriented vector data sets, starting from TOP10NL, and it should also support production of the new 1:1000K map. In the future, it should also incorporate the large-scale base map of the Netherlands, defined in the Information Model Geography (IMGEO). Whereas the 1:10K and smaller scale data are all produced by

Kadaster, this large-scale data is mainly produced and used by municipalities. Integrating topographic data collected by two different organisations offers other challenges for IMTOP (Stoter *et al.* 2009a).

The main classes in TOP10NL, all subclasses of one parent class (i.e. GeoObject), are listed in Table 1. TOP10NL further specialises the Relief class into five subclasses. The 'part of . . .' classes model the division of whole objects into several geometries in an object-oriented approach.

Information reduction across the scales has never been harmonised (Figure 1). Consequently, differences in the level of detail are partly caused by the (original) purposes of the different scales. TOP10NL is used for GIS analyses, orientation, as background for thematic information and is printed as paper and digital raster map at scale 1:25K. The original aim of 1:50K map was to give as 'much' detail as possible according to international military standards, because it was the most important map for military use. On the other hand the 1:100K served as an overview map for militaries and less detail was needed and even desired. Consequently, the 1:25K and 1:50K maps are very detailed, whereas the 1:100K is very open and much more generalised than expected.

The legends of the 1:25K, 1:50K and 1:100K maps are rather well harmonised. However, the 1:250K and 1:500K maps are visualised with other colours and symbols. The underlying vector databases at all scales (i.e. TOPxxvector databases) do have the same structure and coding system.

The differences between the data at several scales were analysed to show which generalisation knowledge should be captured in the data model IMTOP, that is which semantics on scale transition is required to model a multi-scale database. The model should define these differences in such a way that a generalisation process knows exactly what it should achieve. GIS systems currently contain operators to carry out the required processes; however, they need human interaction to set the appropriate parameter values and to select the features that comply to a specific input context (Stoter *et al.* 2009c). When this knowledge is captured in a multi-scale data model, a machine process that implements the model can do the process without any (or with minimum) human interaction.

Some global differences between the data sets that were identified from the case study and that require a formal definition are as follows:

- The road classifications at 1:10K, 1:50K and 1:100K are very similar, although the road classes are merged in 1:100K to fewer classes. The 1:250K has a different road classification, based on other (international military) specifications. As the databases

Table 1. Main classes in TOP10NL; Dutch terms are added in brackets.

Main classes in TOP10NL
PartOfRoad (*Wegdeel*)
Land Use (*Terrein*)
PartOfWater (*Waterdeel*)
PartOfRailway (*Spoorbaandeel*)
Layout Element (*Inrichtingselement*)
Registration Area (*Registratief Gebied*)
Building (*Gebouw*)
Geographical Area (*Geografisch gebied*)
Functional Area (*Functioneel gebied*)
Relief (*Reliëf*)

at several scales were designed independently, the classifications are currently not aligned in a data model. IMTOP should model this reclassification knowledge so that classification schemes can be changed accordingly.
- Roads are represented by polygons and lines at 1:10K and only as lines at smaller scales. Collapsing polygons to lines is not always trivial. For example, roundabouts are introduced at smaller scales as points because of the linear road network replacing road polygons. However, in 1:10K database roundabouts are not encoded, but can be interpreted by humans from the polygons. Consequently, roundabouts need a more formal approach in IMTOP.
- Land use, roads and water form a planar partition at scale 1:10K (buildings are located on top of the planar partition). Data sets at 1:50K and smaller contain a planar partition of land use (including built-up area) and water features. For these scale transitions the multi-scale specifications should make clear at first which feature types at what scales constitute the planar partition. In addition it should be clear how former area objects are converted to line objects. For example, collapsing road polygons to lines might require additionally that former road areas are assigned to (which?) neighbouring area objects.
- Roads that are kept on smaller scales are the most important roads, that is ‘dead end’ roads are deleted and ‘through’ roads are selected. Because of a lack of semantics (important roads and dead ends are not encoded in the data), the reclassification requires human interpretation. More formal knowledge enables more automation.
- Maps at scale 1:250K and smaller do not show individual buildings. IMTOP should contain such knowledge so that the process “understands” which objects to delete and/or to convert.

3.2. The scope of IMTOP

Several aspects outline the scope of IMTOP. First, ‘general’-purpose topographic maps at different scales are the object of study. Second, IMTOP models a multi-scale database in which data at several scales is maintained because on-the-fly automated generalisation is not considered feasible.

Also we do not distinguish between database (also called model) representations and cartographic representations at several scales (Brassel and Weibel 1988, Gruenreich 1992) as the Kadaster only maintains a database representation and 1:10K scale, where the geometries in the vector data sets at scale 1:50K and smaller take into account the way they appear on the map. For example, a motorway at 1:50K is portrayed with a line symbol of width 1.5 mm, which is 75 m in reality, which caused the displacement of other features such as buildings. The map is created by adding symbols to the geometries in the vector data sets (see Figure 2).

Another aspect that defines the scope of our research is the assumption that meaning of concepts does not change across the scales.

The last aspect that defines the scope of IMTOP is that we identified the hybrid approach (Section 2.2) as the most optimal modelling approach. On the one hand IMTOP requires functionality supported in the feature-first approach (also supported in the hybrid approach): bridge over water should be allowed; administrative area can overlap topographical objects; multi-geometry of objects should be possible (preferably all captured in planar or linear topological structures as supported in hybrid approach), for example both centrelines and polygons for roads. On the other hand the model should contain support for topological structure as available in the geometry-first (and also in the hybrid approach). Therefore, IMTOP follows the hybrid approach and topological primitives are used to model

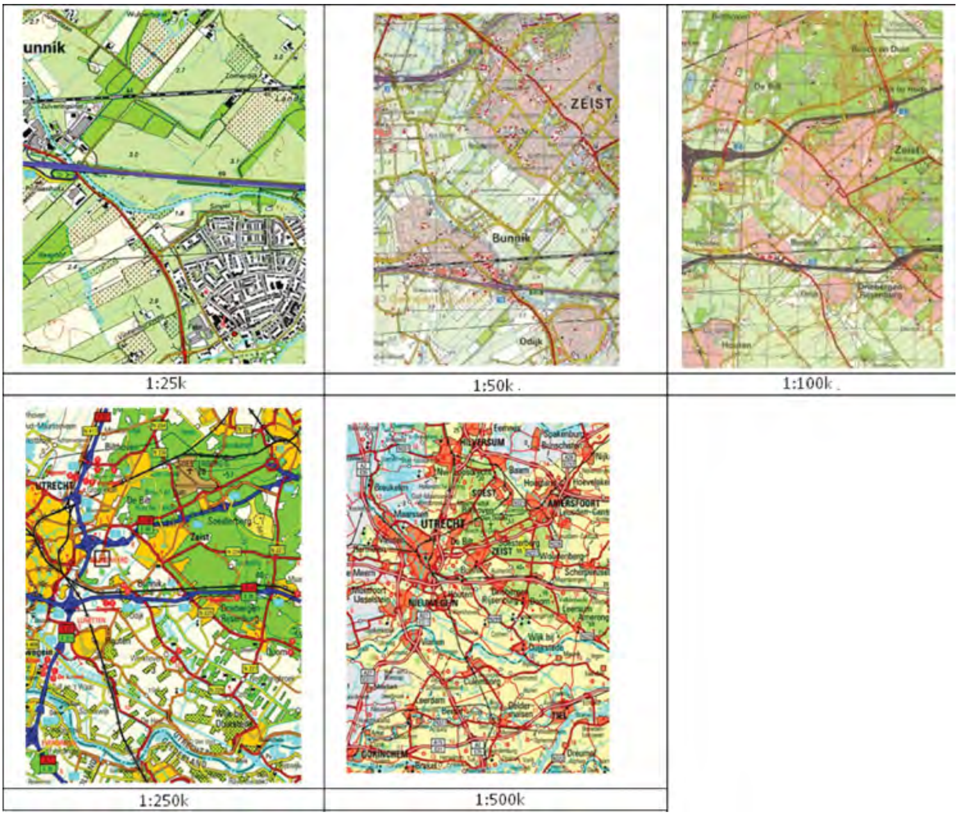

Figure 1. Cut outs of maps at different scales as produced by Kadaster.

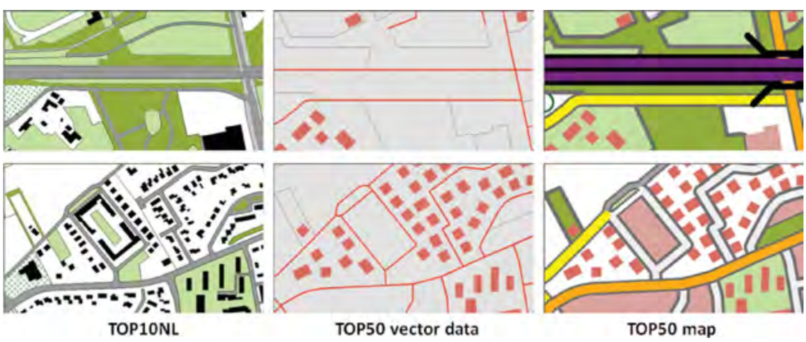

Figure 2. Displacement of features in TOP50vector triggered by graphical conflicts of symbolised features in 1:50K map.

geometries. Also multiple topology layers (manifolds) are supported, for example to represent the base topography and the administrative units in different layers.

The model uses ISO 19107 spatial schema (and related Open Geospatial Consortium standards), for example via TP_Node, TP_Fdge and TP_Face.

4. Design of multi-scale information model topography

IMTOP is a result of two main design steps:

(1) Modelling the multi-scale data content (Section 4.1)
(2) Modelling semantics on scale transitions (Section 4.2)

Enterprise Architect (EA 2009) was used as UML design tool for IMTOP.

4.1. Modelling multi-scale data content

To model multi-scale data content, first, required content at separated map scales was determined. The required classes, associations, attributes, attribute values and geometry types (possibly multiple geometries) were identified per map scale. The required data content was obtained from current product specifications for all TOP vector products (Kadaster 2002).

From this analysis we can conclude that the TOP10NL classes (see Table 1) occur in all scales, with the exception of buildings which are not available at scale 1:250K and smaller. In addition, at scale transitions, mostly information is reduced, for example less attributes and attribute values are available, but sometimes information is added. An example is the attribute value 'roundabout' for TypeOfInfrastructure. This information is not registered at scale 1:10K because roads are represented by polygons. It is, however, required at scale 1:50K and smaller, where roads are collapsed and smaller roundabouts are represented by points.

To find the best way of modelling changes across scales, we developed three alternative UML class diagrams which are evaluated in Sections 4.1.1–4.1.3.

4.1.1. Constraints for scale-specific attributes and attribute values

In the first alternative, attributes and attribute values are allowed or disallowed at specific scales by means of constraints. For example, 'if 1:50K then geometry type of "secondary roads" is line'. This approach appeared not to be a good solution because many constraint expressions have to be inspected to know what a class looks like in one scale. In addition, constraints are used both for modelling valid data content (for non-scale-dependent situations) and for modelling scale transitions (see Section 4.2).

4.1.2. Inheritance, derivation relationships and derived attributes for scale-specific information

The second alternative uses inheritance, derivation relationships and derived attributes to model a class at specific scales. Every class that occurs in topographic reality is modelled by a super class containing only attributes that are valid at the base scale (TOP10NL in our case). For every scale a subclass is modelled via the specialisation association. In addition, classes at every scale are modelled as derived from the previous large-scale class using a 'derivedFrom' association ('*afgeleidVan*' in Dutch). Figure 3 shows an example for the 'Road' class for TOP10NL, TOP50NL and TOP100NL.

It should be noted that this 'derivedFrom' relationship between a class at different scales is not a UML 'generalisation–specialisation' relationship. Instead it is an association expressing that a TOP50NL object is derived from TOP10NL, that is details are removed which can be the deletion of attributes (which is often not the main operation) or deletion of objects with certain

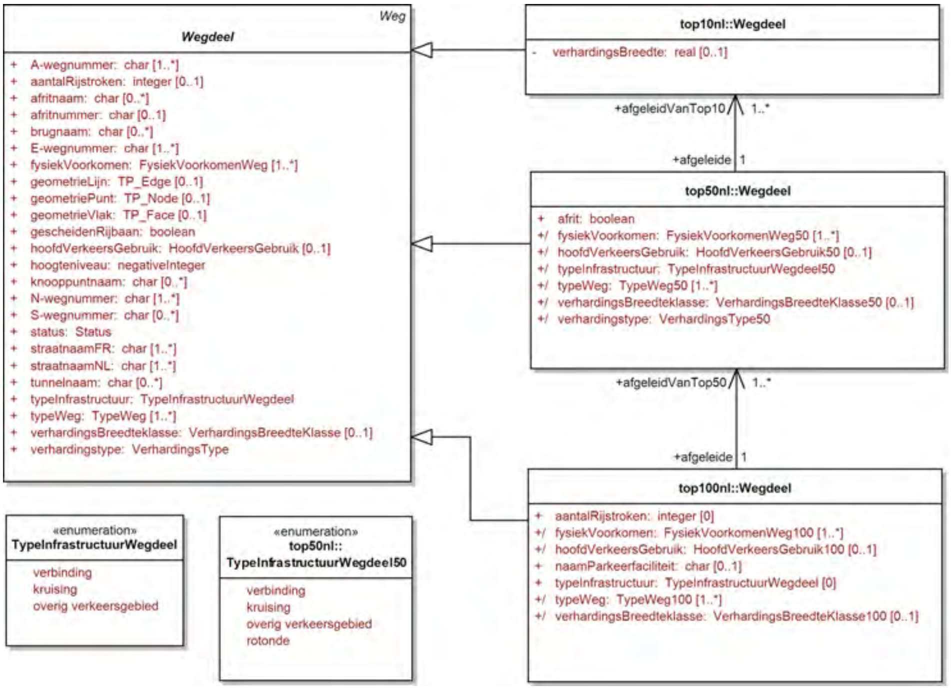

Figure 3. Using inheritance, derivation relationships, derived attributes and multiplicity on attributes to define what is valid on every scale.

attribute values. This is not the same as UML inheritance. The inheritance that is modelled is used to name attributes of a class that are shared by all scales only once. Attributes at a specific scale may therefore be either shared by all scales (modelled in the superclass), derived (indicated with '/') or introduced (e.g. '*afrit*' ('exit') for Roads in TOP50NL).

It should further be noted that in the superclass there may be optional attributes (with multiplicity [0. . .1] or [0 . . .*]). In a subclass the multiplicity of attributes is further specified, and sometimes even set to [0], effectively removing the attribute from the subclass. This last aspect may be counterintuitive as normally subclasses get only more attributes. The reason for having the optional attributes in the superclass is that when two or more subclasses need such an attribute, the attribute is specified only once. This avoids 'model redundancy' and assures consistency for the different scales: same attribute name and data type.

The derivation rules for derived attributes can be modelled with OCL, for example: 'derive: derivedFromTOP50NL.typePavement' for typePavement in TOP100NL. This inheritance approach is only used for classes, as inheritance is not defined for enumerations (see ISO 19103).

This approach for modelling multi-scale topography is advantageous for several reasons: the model is easy to read and it is easy to derive a model per map scale by showing the relevant classes for that map scale only.

4.1.3. Stereotypes for multi-scale semantics

In the third alternative the underlying meta-model of UML is extended with multi-scale semantics through user-specific UML stereotypes and tagged values. The stereotype <<MultiScale>> is used to indicate that the given class has different representations at

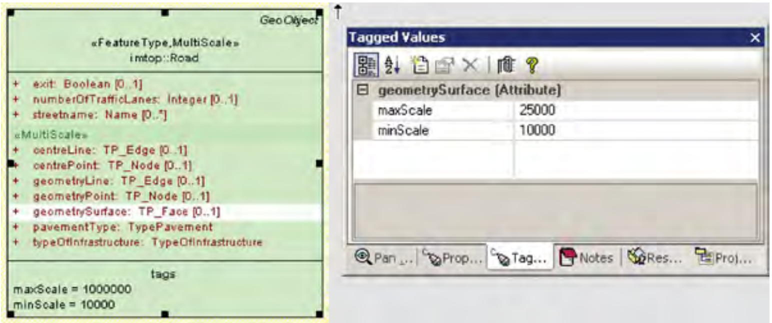

Figure 4. Example of using a MultiScale stereotype to model map scale specific information. The tags are only available in the GUI of EA.

different map scales (similar to stamps in MADS, see Section 2.1). Attributes and attribute values of MultiScale classes (labelled with the <<MultiScale>> stereotype), get a 'minScale' and 'maxScale' tag to indicate on which map scales they are valid. In the example of Figure 4, all spatial attributes are stereotyped with the <<MultiScale>> stereotype and the appropriate minScale and maxScale tags are added.

Although the approach enables to model multi-scale semantics, a few aspects make the approach less appropriate for IMTOP. First, the tags are not visible in the class diagram itself, as can be seen in Figure 4. It is also not easy to automatically generate separate UML models per map scale from this model. Furthermore, the model is very compact and it is therefore not easy to read: every concept (e.g. 'Road') is modelled as a single class covering all map scales, and multi-scale aspects are only visible when analysing the specific attributes and attribute values. However, the main problem with this approach is that it models only one class per feature type for all scales, which does not express the huge semantic differences between for example a 1:10K road and a 1:500K road.

Based on the considerations outlined above the second approach was selected to model multi-scale data content in IMTOP. In Figure A1.1 (Appendix 1) the approach is applied on three classes (building, land use and roads) and for three scales.

4.2. Modelling semantics on scale transitions

Now that we have modelled the multi-scale data content for IMTOP we can enrich it with semantics on scale transitions. Several considerations justify the modelling approach used for modelling scale transitions.

First, to implement research theories on generalisation, we expressed the semantics on scale transitions as 'generalisation constraints' to be respected (sometimes called 'hard constraints') and 'goals to be optimized' (sometimes called 'soft constraints'). In research on generalisation, the use of constraints and optimisation goals are common methods to define requirements and to control and evaluate the automated generalisation process. Examples are McMaster and Shea (1988), Beard (1991), Bard (2004), Barrault *et al.* (2001), Ware *et al.* (2003), Burghardt and Neun (2006) and Sester (2000). Constraints express what generalisation output should look like without addressing the way this result should be achieved, for example by defining sequences of operations.

It is important to understand the difference between meaning of constraints in generalisation theory and in database theory. Generalisation 'constraints' are either requirements to which a map should be optimised (e.g. minimum distance between two objects) or requirements which should be strictly met (e.g. the data should be topologically correct) as the constraints in database theory. The requirement meant for optimisation may be partly ignored while still obtaining a satisfying result, because of two main reasons. First, it is not possible for humans to distinguish between parameter values mentioned in the constraints (e.g. minimum size of a building) and the value plus/minus a flexibility range and therefore the values are used with a notion of flexibility (Ruas 1999, Bard 2004). Second, situations may be overconstrained and therefore a constraint may need to be relaxed to meet a more important constraint, for example 'keep important buildings'. To address both generalisation constraints (soft) and database (hard) constraints at the same time we use the term 'generalisation specification'. These specifications cover both the optimisation constraints and the 'hard' constraints that should always be met. An importance value (showing which specification should be satisfied above the other) indicates which type of generalisation constraint it concerns.

One should be aware that IMTOP formalises the *goal* of generalisation and not the generalisation process itself. Although IMTOP could suggest an action to meet the goal (as this knowledge can also help the process), prescribing the process in the model would prohibit a certain flexibility for the systems, as each system will have its own strategy. Generalisation would, however, be served by formal description of operators that implement the process as web services, see Foerster *et al.* (2008). This is outside the scope of this article.

To help find the best process more knowledge is required than the goal itself. For example, we can add the above-mentioned importance value or a suggested action (mostly it is known at NMAs how to best achieve the goal). Weighting different specifications against each other and finding successful procedures is, however, outside the scope of this article. Also the examples in this section are used to explain the formalism and they are not meant to model a full-scale transition.

A last consideration for the applied modelling approach is to distinguish between two types of generalisation specifications: specifications for assuring legibility and specifications for preserving the source data. Preservation specifications prescribe preservation of topology, position, orientation, shape and distribution/statistics and are completely satisfied at scale transitions. They may become violated when operations are applied to ensure legibility (minimal dimensions and granularity). The distinction is introduced by Harrie (2001) and Burghardt *et al.* (2007) and is useful here because the two types require different modelling approaches: legibility specifications are defined within one scale (the target scale) and require explicit relationships between objects in one scale if more than one object is involved, whereas preservation specifications are defined in correlation with the source data and hence require relationships between objects and object groups at different scales. Steiniger and Weibel (2007) and Bobzien *et al.* (2008) identify relationships between and within scales as vertical and horizontal relationships, respectively.

For modelling generalisation specifications, we used UML/OCL. The formal language OCL is an extension of UML that can be used to enrich data models in UML with more semantics (Warmer and Kleppe 2003). As mentioned above a difference can be made between data model constraints (used to define data content) and constraints that specify generalisation specifications, that is rules related to the topographical presentation at a certain scale. Both can be expressed in OCL. In this section we address the second types of constraints.

To be able to understand the OCL syntax as used in this article, we will briefly introduce the basics of OCL in Section 4.2.1. Sections 4.2.2 and 4.2.3 present the modelling of legibility and preservation specifications in OCL, respectively. Section 4.2.4 closes with final notes on our approach.

4.2.1. Syntax of OCL

A typical expression specifies either an invariant constraint that must hold for the data or a query on the model that returns the objects meeting a certain condition.

An OCL constraint expresses an invariant and has a scope and a condition. The scope (context) defines which objects should comply with the constraint, for example all buildings, all building–road combinations or all roads of type 'highway'. The OCL condition (indicated with inv) specifies when an object in the scope is valid. The structure of a typical constraint is as follows:

```
context top50nl::Building
inv: geometry.area() >= 0.4
```

Within OCL we use spatial functions to address spatial properties and relationships as area, distance and intersection of geometrical objects (see Hasenohr and Pinet 2006, Pinet *et al.* 2007). In our constraints, we also use methods defined in the UML model, such as getNearestBuildings(). This is allowed as long as the methods have no side effects.

The OCL invariant constraints specify valid system states and can be implemented in a database with a check constraint that rejects an object when an update, insertion or delete does not endure the check. Apart from invariants, constraints may also be expressed to support scale transitions, that is as *pre-conditions* and *post-conditions* of the transition process, for selecting appropriate large-scale input (i.e. instances) and for checking resulting smaller scale output which might trigger another event. However, constraints express valid data content and are therefore not appropriate to express the 'optimisation' nature of generalisation specifications. Therefore, we used OCL queries instead to acknowledge that valid system states may coincide with partly and intentionally violated generalisation specifications.

An OCL query may look like

```
context top10nl::PartOfRoad::oneLanePartOfRoads(): PartOfRoad
body: top10nl::PartOfRoad.allInstances()->select(numberOfLanes = 1)
```

This query specifies the result of the oneLanePartOfRoads () method, which is a member of the Road class in the TOP10NL package. Note that the context element is extended with the return type of the query (PartOfRoad). The body describes which objects will be returned, in this case all PartOfRoad objects for which the numberOfLanes attribute is equal to 1. Note that this form of OCL is used to specify queries, that is 'virtual' object–methods that do not modify the objects.

To return objects that do not meet generalisation specifications, we used the following template:

```
context package::GeoObject::method() : ReturnType
body: GeoObject.allInstances()->select(specificationScope &&
        !specificationCondition)
```

This template query iterates over all GeoObjects and selects the objects that are in the scope of the generalisation specification but do not meet the specification, for example the object is a building but has an area less than 0.4 map mm^2. In the next step the returned objects can be further processed, as will be seen in Section 5.

4.2.2. Expressing legibility specifications in OCL

We distinguish between three types of legibility specifications to be expressed: specifications defined for single objects, specifications defined for two objects, specifications defined for groups of objects within one scale. To specify the last two types of associations, the *ruimtelijkeRelatie* (spatial relationship) associations were added to IMTOP (see Figure A1.1).

An example of a specification on a single object is ‘area of all TOP50NL buildings should be at least 0.4 map mm^2’. In an OCL query following the template above this has the following result:

```
context top50nl::Building::tooSmallBuildings() : Building
body: top50nl::Building.allInstances()->select(
        b.geometry.area() < 0.4)
```

The query specifies a method tooSmallBuildings() that returns all buildings smaller than 0.4 map mm^2.

An example of a specification defined for two objects is ‘a road and a building should have a minimum distance of 0.4 map mm’. Expressed in an OCL query this could look as

```
context GeoObject::minPartOfRoadBuildingDistance():GeoObject
body:  GeoObject.allInstances()->select(r, b |
       r.oclIsKindOf(top50::PartOfRoad) &&
       b.oclIsKindOf(top50::Building) &&
       r.geometry.distance(b.geometry) < 0.4)
```

The third type of legibility specifications is defined on groups of objects, for example group of buildings. Groups are formed based on some common aspects of the member objects, for example they are all positioned within the same area partitioned by the road network. An example of such specification is that buildings within one land-use object of type ‘other’ should never exceed 10% of the land-use object area. This specification assures converting dense groups of buildings to ‘built-up area’. A query that returns all land-use objects that are covered for more than 10% with buildings is described in OCL as

```
context top50::LandUse::tooCoveredLandUseObjects():LandUse
body: LandUse.allInstances()
      ->select(l | type = ‘other’ && top50:Building.allInstances()
            ->select( b | b.intersects(l))
            ->collect(b|b.geometry.intersection(l.geometry).
  area())
            ->sum() / self.geometry.area()
                 <= 0.1)
```

4.2.3. Expressing preservation specifications in OCL

Expressing preservation specifications is less straightforward than expressing legibility specifications as it requires relationships between objects at different scales.

To model preservation specifications, we incorporated a modelling framework into IMTOP that contains three main aspects.

First, we extended IMTOP with richer inter-scale relationships because the ‘derived from’ relationships (see Figure A1.1) are limited to the same topographic class, that is a

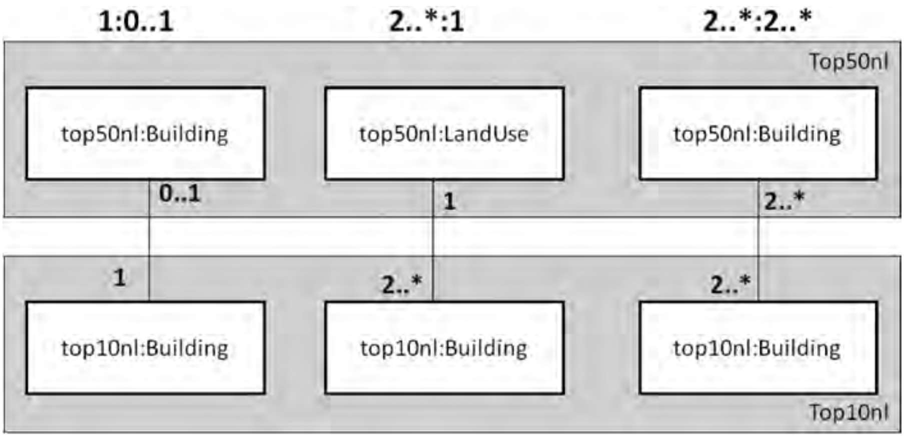

Figure 5. Three types of inter-scale relationships, exemplified for TOP10NL–TOP50NL relationships.

building only has an inter-scale relationship with other buildings. Preservation specifications may be defined between instances of different classes in source and target scale.

We distinguish between three types of inter-scale relationships (Figure 5):

- one-to-at most one, for example as a result of elimination or selection;
- many-to-one (at least two-to-one), for example as results of aggregating, amalgamating or merging two or more objects into one object;
- many-to-many (at least two-to-at least two), as a result of replacing a group of objects by a new group of objects, which is called typification.

As the model is meant to enforce consistency between scales, the 0:1 option (where objects do appear in small scale but not in large scale because of different update cycles) is not considered as it is not a valid option in a consistent multi-scale database.

To add these inter-scale relationships to the model, we defined a relationship on the GeoObject level which links different GeoObjects. The relationship is inherited by all subclasses enabling relationships between any GeoObject subclasses (see Figure 6). Note that these relationships may replace the '*ruimtelijkeRelatie*' associations modelled to express legibility specifications between objects and group of objects.

The second aspect of the framework for modelling preservation specifications is a method for handling inter-scale relationships. A getDerivatives() method is added to GeoObject to operate on these relationships (see Figure 7).

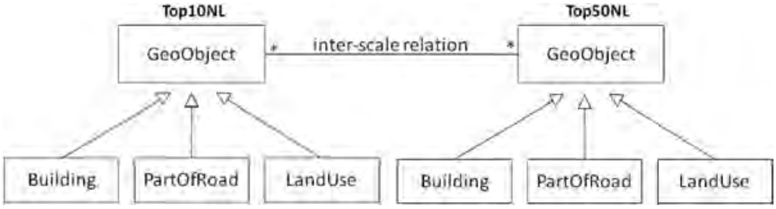

Figure 6. Objects on different scales can be connected regardless of their type as the relation is specified in GeoObject and inherited by its child classes.

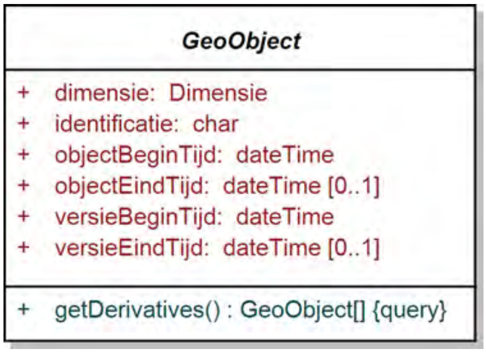

Figure 7. Extending GeoObject with new method.

This method returns a set of GeoObjects (at the next smaller scale level) that were derived from this GeoObject. Calling this method on an object in the smallest scale data set will return an empty set.

The last aspect of the framework is to distinguish between two levels of constraint complexity to express all aspects of generalisation specifications:

- Existence level: a constraint evaluating the existence of an object or objects
- Property level: a constraint evaluating the geometrical, topological and/or structural properties of an object or objects

To query the properties of an object (e.g. its position or area) an object should exist. Thus, the property level is an extension of the existence level. In other words, a violation on the existence level is a special case of a violation on the property level. As preservation constraints deal with properties, they have an underlying constraint at the existence level. The OCL examples that express preservation specifications will illustrate this.

In this section, we give OCL examples for preservation specifications for the first and last type of inter-scale relationships shown in Figure 6. The OCL examples use TOP10NL as source data set and TOP50NL as target (i.e. derived) data set.

An example specification of the first type (between one and at most one object) is that an ‘important’ building never should be displaced. This also implies that an ‘important’ building should exist, as removal is the extreme case of displacement. The query has to iterate over all important building objects in the TOP10NL scale and ‘travel’ over the inter-scale object relationships to the corresponding object in TOP50NL to check whether their centroids are equal (for simplicity reasons this example ignores possible rotation). The OCL query looks like

```
context top10::Building::displacedImportantBuildings(): Building
body: top10::Building.allInstances()
       ->select(b1 | type = ‘important’ &&
              b1.getDerivatives()->select(b2 |
              b1.geometry.centroid() == b2.geometry.centroid())
         ->isEmpty())
```

An example specification of the last inter-scale relationships (between at least two and at least two objects) prescribes that an aligned group (or chain) of buildings in the source data set should remain aligned in the target data set. A method isAligned is introduced that returns true when the two objects are aligned. Also, a method top10::Building.getAlignedGroups() is introduced that will return a set of all aligned buildings (a set containing groups of aligned Buildings):

```
context top10::Building::nonalignedGroupOfBuildings(): Building
body: top10::Building.getAlignedGroups()
    ->select( s | !s->collect( b | b.getDerivatives() )
        ->forAll( d1, d2 | d1 == d2
            || d1->intersect(d2)->forAll(db1,db2,db3 | db1
  == db2 || db2 == db3 || db1 == db3 || db1->isAligned(db2,db3))))
```

The OCL query returns every group of buildings from the set of aligned Building groups (returned by getAlignedGroups()) of which its derived buildings are not aligned. Alignment is checked by making sure that, for every pair of Buildings on the source scale, its shared derivatives (intersection of d1 and d2) contain no three buildings db1, db2, db3 that are not aligned.

4.2.4. Final notes on defining semantics on scale transitions in OCL

The presented examples that express generalisation specifications in OCL may seem very simple for the complex generalisation problems that may occur in practice. However, as mentioned earlier, it is important to understand that OCL is used here to define specifications (i.e. goal of generalisation) so that a process that implements the model has the knowledge to apply its functionality to obtain valid data content. For example, we may have a polygon river with a polygon lake at largest scale. In the medium scale, a polygon river will be collapsed to line segment partly and simplified to polygons and a polygon lake is still polygon but simplified. In OCL the knowledge supporting the optimal generalisation solution can be formalised as follows: (1) geometry of rivers should be line geometry, (2) hydrographic network should be kept, (3) minimum granularity of lake polygons, (4) minimum dimensions of lake polygon, (5) shape preservation of rivers and lakes. What operators to apply and in what way to best meet the specifications as defined in the model is up to the system and also an interesting research topic, but outside the scope of this article.

5. Implementation and results

We converted IMTOP as designed in the previous section to a prototype database implementation to evaluate the model. This was done with the open source DataBase Management System PostgreSQL with PostGIS extension (PostgreSQL 2009). The prototype consists of two parts corresponding to the modelling steps presented in Section 4: an implementation of a simplified IMTOP model in database tables that contain multi-scale data content (Section 5.1) and the implementation of OCL queries expressing semantics on scale transitions as database views (Section 5.2).

5.1. Transformation of IMTOP classes to database tables

We limited the prototype implementation of IMTOP to three classes: Building (*Gebouw*), PartOfRoad (*Wegdeel*) and Land Use (*Terrein*) and two map scales: TOP10NL and TOP50NL. The UML classes were transformed into database definition language (DDL) scripts that created a set of tables, as shown in Figure 8. Note that a many-to-many

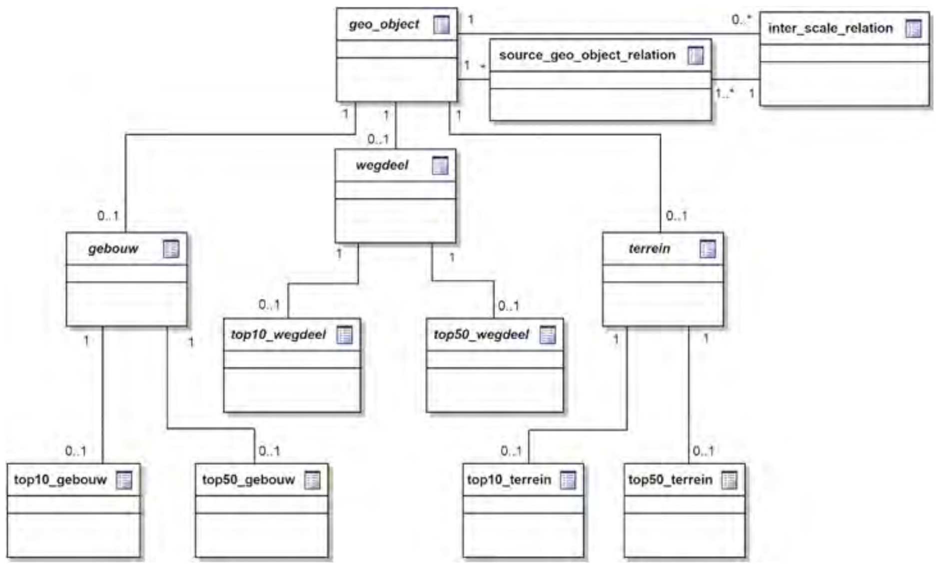

Figure 8. Diagram of database tables of the simplified IMTOP model (table attributes are omitted).

association of two classes results in three database tables. Apart from transforming the classes into tables, the database prototype materialises two other aspects of IMTOP.

First, the relationships between two classes are materialised through primary and foreign keys. Second, the inheritance principles are materialised through the tables. The inheritance of the classes *Gebouw, Wegdeel* and *Terrein* from GeoObject can be realised either by adding the attributes from GeoObject to all three tables or by creating a separate table for GeoObject and adding a foreign key linking to a geo_object table row in the inheriting tables. The prototype implements the second method to accommodate the inter-scale relationships. The same principle is applied to inherit a specific topographic class at each scale from the concerning superclass (e.g. of *top10_gebouw* from *gebouw*). The result is that every row in *geo_object* belongs to one row in one of the three tables (i.e. *gebouw, wegdeel* or *terrein*) and that every object within one map scale is represented with three rows in three separate tables. For example, a TOP10NL *Gebouw* object has a row in top10_gebouw, relating to a row in *gebouw*, which relates to a row in *geo_object*.

The prototype was filled with sample data provided by the Kadaster. Because of the inheritance implementation as described above, each object in the sample data resulted in three rows in three tables.

5.2. *Translation of OCL expressions to database views*

We limited the prototype implementation to database views that capture the OCL query results. These queries select objects that do not meet the goal as formalised in the data model and therefore may require further processing or may be part of a good generalisation solution as explained in the introduction of Section 4.2. In the future the resulting set could either be offered to a generalisation system calculating the optimal solution or be used to evaluate the output of automated generalisation.

OCL can be used to generate SQL, for example OCL2SQL (Demuth *et al.* 2005). Also Pinet *et al.* (2007), Heidenreich *et al.* (2007) and Hespanha *et al.* (2008) showed automated conversion from OCL to SQL. However, we translated the OCL expressions manually into database views. The main reason is that the automated conversion required considerable programming to translate all the specific aspects of our domain, whereas manual translation of the OCL expressions into database views already proves the concept of formalising semantics on scale transitions with OCL.

From the example queries that were presented in Section 4, we will show for one legibility and one preservation specification how they were implemented in the prototype.

The legibility query defining maximum coverage of land-use objects with buildings can be translated to the following database view:

```
CREATE VIEW too_covered_terrain AS
        SELECT top50_terrein_id, geometry
         FROM top50_terrein t5
         LEFT JOIN terrein t ON t.terrein_id = t5.terrein_id
         WHERE t.type_terrein = 1 AND (
SELECT SUM(area(intersection(g.geometry,t5.geometry)))
        FROM top50_gebouw g
        WHERE intersects( g.geometry, t5.geometry))
 / area(t5.geometry) > 0.1
```

All LandUse instances that appear in the view are selected in Figure 9.

To implement the complex inter-scale relationships (see Figure 6) for preservation specifications at database level, we added a relation_group_id attribute in a inter_scale_relation table that groups relationships between objects represented in different scales (see Figure 10). These extra classes and relationships do not cause incompatibility as they are a superset of, and thus compatible with, the original IMTOP implementation. Note that geometrical relationships within one scale can be derived directly from the database, but the inter-scale relationships as modelled here have to be maintained explicitly as additional information. This information can

Figure 9. Result of the too_covered_terrain view.

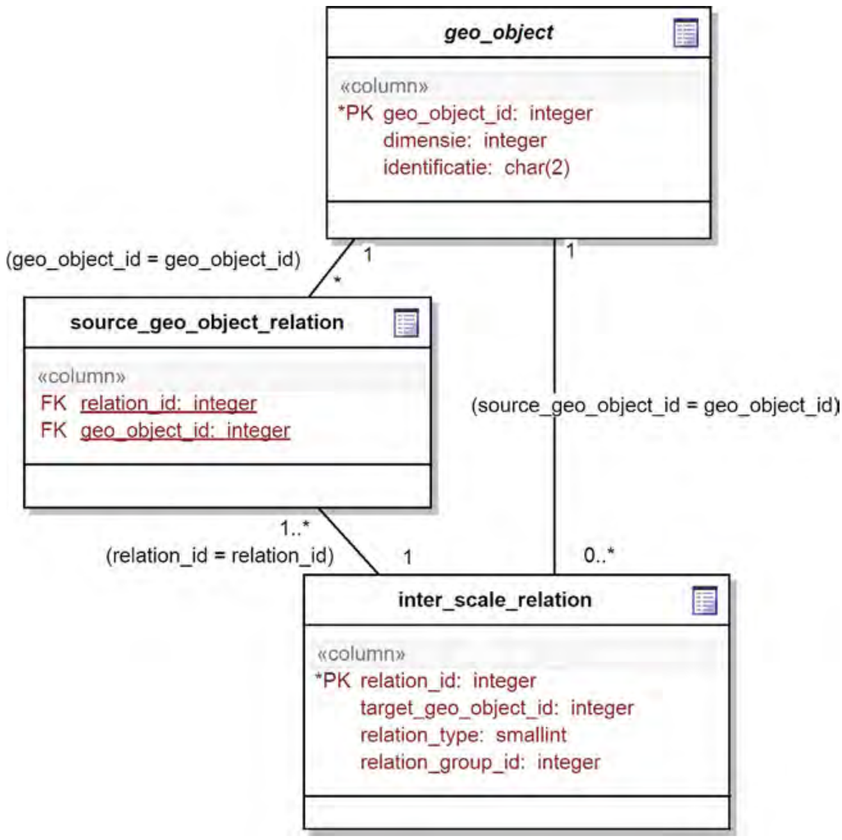

Figure 10. Complex relationships between different scales in IMTOP.

be either a result of calculating the relationships between existing data sets (i.e. data matching) or an additional result of a generalisation process.

The OCL query that we implemented for preservation specification defines that important buildings should not be displaced (important buildings are of type_building '2'):

```
CREATE VIEW displaced_important_buildings AS
    SELECT g10.top10_gebouw_id, g10.geometry
    FROM top10_gebouw g10
    LEFT JOIN gebouw g ON g10.gebouw_id = g.gebouw_id
    LEFT JOIN source_geo_object_relation sgor
       ON g.geo_object_id = sgor.geo_object_id
    LEFT JOIN inter_scale_relation isr
       ON sgor.relation_id = isr.relation_id
    LEFT JOIN gebouw g2
      ON isr.target_geo_object_id = g2.geo_object_id
    LEFT JOIN top50_gebouw g50
      ON g50.gebouw_id = g2.gebouw_id
    WHERE g.type_gebouw = 2 AND
       (centroid(g50.geometry) = centroid(g10.geometry))
```

As can be seen, the SQL statement becomes a lot more complex because of the inter-scale relationships. Information from all tables between top10_gebouw and top50_gebouw is needed to travel over the inter-scale relationships to test the condition.

6. Discussion and conclusion

This article designed an information model for a multi-scale topographic database containing rich semantics both on multi-scale data content and on scale transitions. The IMTOP ensures consistency between data sets. In addition, because of explicit relationships between objects at different scales, automated generalisation of updates is supported. Because of these relationships, objects in a smaller scale that have a relationship with updated objects in a larger scale can be identified as well as objects that do occur in a larger scale but do not have derivates in smaller scales. Those objects are candidates for further processing (Harrie and Hellström 1999, Uitermark *et al.* 1999, Uitermark 2001).

The results of instantiating (part of) the model and evaluating it against model requirements show that UML/OCL is an appropriate formalism for modelling a multi-scale database and for modelling scale transitions. Besides supporting the workflow of NMAs (as the Kadaster in our case study), the semantic-rich model may serve external users and applications of multi-scale topography. This has important added value compared to independent maintenance and provision of data sets at different scales.

Several conclusions can be drawn from our experiments that identify areas for further research.

An important conclusion is the limited use of database views to implement semantics on scale transitions. These views return the objects that do not meet the individual specifications but that still may be valid objects. To serve automated generalisation, the set of OCL queries should be input for a generalisation orchestration process that calculates the best solution rather than that they are translated query per query into database views. For the OCL part, this raises several questions.

- How to determine and formalise the domain of satisfaction and violation values for generalisation specifications? This also implies dealing with the limitations of OCL queries. An OCL query can return either a set of violating objects or a measure of violation for a certain object, but it cannot return both. On the implementation level, an SQL statement would be capable of returning both, but by replacing every query in the prototype with two new queries, a lot of extra complexity will be added to the model and the transformation process. Specifying satisfaction and violation values has been addressed in previous research, see for example Ruas (1998) and Bard (2004).
- How to weight, prioritise and aggregate different specifications and how to express this in OCL? Until now generalisation specifications mainly focus on geometric and/or thematic properties. Better understanding of impact and interdependencies of individual specifications may come from cognitive science. Weighting and prioritising generalisation specifications have previously been addressed in the domain of constraint-based optimisation, see Ruas (1998), Bard (2004) and Mackaness and Ruas (2007).
- How to enrich the OCL queries with more semantics on generalisation processes, for example suggesting generalisation operation to solve violating specifications, or name operations that should never be performed on certain types of objects? This information might help to find the best generalisation solution. A relevant question here is how much information should (and can) be covered in the model (to define as concisely as

possible what is expected) and how much should be left to the implementation (to leave flexibility to the systems)?

- How to extend OCL syntax with spatial functionalities and functionalities to detect (i.e. formalise) spatial concepts, such as 'urban structure' and 'shape' and their allowed changes at scale transitions and how to translate these into DBMS queries? For many of these functionalities, appropriate measures are lacking. Previous research on measuring generalisation concepts and cartographic conflicts was conducted by Regnauld (1996), Harrie (2001), Christophe and Ruas (2002), Steiniger *et al.* (2008), Lüscher *et al.* (2008), Thomson and Béra (2008) and Xiang *et al.* (2008).

Another conclusion is that annotating the sample data with a number of inter-scale object relationships was time-consuming. Calculating these relationships is not trivial because of the complex relationships, see Figure 6 and also Féchir and De Waele (2005) and Mustiere (2006). Building inter-scale relationships is also not trivial because the smaller scale data may be displaced as influenced by symbolisation (see Figure 2).

A topic for further research is the possible change of meaning at scale transitions despite the fact that these object classes do have the same name ('label'). In the current approach, it is assumed that the meaning of a concept does not change at scale transitions. However, as the different scales are targeting different information demands, it can be expected that meanings of concepts do change at scale transitions. For example, TOP10NL road centrelines represent different lanes of a road, whereas at the smaller scales centrelines represent full road construction, including small parallel roads and verges. These concept differences need further study.

After having the framework completed, the next challenge is to transform existing map specifications enriched with cartographers' knowledge into the formalism (see Stoter *et al.* 2009b). This is not easy as much knowledge on generalisation requirements and processes still needs to be revealed. In addition, not all requirements are easy to be formalised.

Future research will also focus on how IMTOP can be prepared for use in the Semantic Web. UML class diagrams are specifically useful to agree with users on data content and to formalise the generalisation specifications that are nowadays often only available to direct the cartographer. However, transferring IMTOP into an ontology language such as resource description framework (RDF) or Web Ontology Language (OWL) will make more semantics known at machine level required for wide reuse of the data. In addition a formal ontology of IMTOP may enforce to model even more semantics on data content and scale transitions. Because of improved compatibility this will provide better possibilities for automated generalisation.

Although the hybrid approach was identified as best approach for modelling multi-scale data content, at first instance we have used the feature-first type approach for modelling generalisation specifications, that is the OCL queries were all defined on the feature classes. Further improvements are expected when applying the hybrid approach to model semantics on scale transitions, as it will be easier to express some of the generalisation specification in OCL and it will also better support the subsequent generalisation operations.

Finally, the concept of defining semantics on scale transitions should be applied to a web mapping context. The legibility and preservation constraints are well applicable to web maps by expressing parameter values of constraints in pixels rather than in map units. The challenge in a web mapping context will therefore be embedding the multi-scale data as defined in IMTOP in vario-scale data structures that can efficiently be disseminated to clients via progressive transfer (van Oosterom 2005).

References

Bard, S., 2004. Quality assessment of cartographic generalisation. *Transaction in GIS*, 8 (1), 63–81.

Barrault, M., *et al*., 2001. Integrating multi-agent, object-oriented, and algorithmic techniques for improved automated map generalisation. *In*: *Proceedings of the 20th international cartographic conference (ICC 2001)*, 6–10 August 2001, Beijing, China, 2110–2116. [CD-ROM].

Beard, K., 1991. Constraints on rule formation. *In*: B.P. Buttenfield and R.B. McMaster, eds. *Map generalisation: making rules for knowledge representation*. London: Longman, 121–135.

Bédard, Y., *et al*., 2004. *Modelling geospatial databases with plug-ins for visual languages: a pragmatic approach and the impacts of 16 years of research and experimentations on Perceptory*. LNCS 3289. Berlin: Springer, 17–30.

Bobzien, M., *et al*., 2008. Multi-representation databases with explicitly modelled horizontal, vertical and update relations. *Cartography and Geographic Information Science*, 35 (1), 3–16.

Brassel, K.E. and Weibel, R., 1988. A review and conceptual framework of automated map generalisation. *International Journal of Geographic Information Systems*, 2 (3), 229–244.

Burghardt, D. and Neun, M., 2006. Automated sequencing of generalisation services based on collaborative filtering. *In*: M. Raubal, *et al*., eds. *Geographic information science: 4th international conference on geographical information science* 2006, IfGIprints 28, 41–46.

Burghardt, D., Schmidt, S., and Stoter, J.E., 2007. Investigations on cartographic constraint formalisation. *In*: *10th ICA workshop of ICA commission on generalisation and multiple representation*, 2–3 August, Moscow, Russia.

Buttenfield, B.P. and Delotto, J.S., 1989. *Multiple representations: National Center for Geographic Information and Analysis (NCGIA)*. Scientific Report for the Specialist Meeting, Technical paper 89–3, 87 p.

Christophe, S. and Ruas, A., 2002. Detecting building alignments for generalisation purposes. *In*: D.E. Richardson and P. van Oosterom, eds. *Advances in spatial data handling: 10th international symposium on spatial data handling*, Berlin: Springer, 419–432.

Devogele T., Trevisan, J. and Raynal, L., 1996. Building a multi-scale database with scale transition relationships. *In*: *International symposium on spatial data handling*, London: Taylor & Francis, 337–351.

Demuth, B., Hussmann, H., and Konermann, A., 2005. Generation of an OCL 2.0 Parser. *In*: *Proceedings of the models'05 workshop on tool support for OCL and related formalisms-needs and trends*..

EA, 2009. *Enterprise architect* [online]. Available from: http://www.sparxsystems.com.au/ [Accessed 4 May 2009].

Féchir, A. and De Waele, J., 2005. The future production of generalised maps at IGN Belgium. *8th ICA (International Cartographic Association) workshop on generalisation and multiple representation*, 7–8 July, A Coruña, Spain [online]. Available from: http://aci.ign.fr/Acoruna/Papers/Fechir_DeWaele.pdf.

Foerster, T., Morales, J.M., and Stoter, J.E., 2008. A classification of generalisation operators formalised in OCL. *In*: E. Pebezma and T. Bartoschek, eds. *Proceedings of GI-days 2008: interoperability and spatial processing in GI applications*, 16–17 June. Münster, Germany: Institute for Geoinformatics, 141–156.

Friis-Christensen, A. and Jensen, C.S., 2003. Object-relational management of multiply represented geographic entities. *In*: *Proceedings of the 15th international conference on scientific and statistical database management*, 9–11 July. Cambridge, MA, 183–192.

Gruenreich, D., 1992. ATKIS – a topographic information system as a basis for GIS and digital cartography in Germany. From digital map series to geo-information systems, *Geologisches Jahrbuch Reihe A* (Heft 122), 207–216.

Harrie, L., 2001. *An optimisation approach to cartographic generalisation*. Thesis (PhD). Lund Institute of Technology.

Harrie, L. and Hellström, A.K., 1999. A prototype system for propagating updates between cartographic data sets. *The Cartographic Journal*, 36 (2), 133–140.

Hasenohr, P. and Pinet, F., 2006. Modelling of a spatial DSS template in support to the common agricultural policy. *Journal of Decision Systems*, 15 (2–3), 181–196.

Heidenreich, F., Wende, C., and Demuth, B., 2007. A framework for generating query language code from OCL invariants. *In*: *UML/MoDELS conference*, Dresden

Hespanha, J., *et al*., 2008. The MDA approach applied to the land administration domain model with focus on constrains specified OCL. *In*: *Proceedings of FIG working week*, 14–19 June, Stockholm, Sweden.

Jones, C.B., *et al*., 1996. Database design for a multi-scale spatial information system. *IJGIS*, 10(8), 901–920.

Kadaster, 2002. *Specificaties TOPxxvector (Specifications TOPxxvector)*. Emmen, The Netherlands: Topografische Dienst.

Kadaster, 2006. Generalisatievoorschriften TOP50vector (Generalisation regulations TOP50vector). Emmen, The Netherlands: Topografische Dienst.

Kilpelainen, T., 1997. *Multiple representation and generalisation of geo-databases for topographic maps*. Thesis (PhD). Finnish Geodetic Institute.

Lüscher, P., Weibel, R., and Burghardt, D., 2008. Alternative options of using processing knowledge to populate ontologies for the recognition of urban concepts. *In*: *Workshop on international cartographic commission on generalisation and multi-representations*, June, Montpellier, France [online]. Available from: http://aci.ign.fr/montpellier2008/papers/22_Luescher.pdf

Mackaness, W.A. and Ruas, A., 2007. Evaluation in map generalisation process, Chapter 5. *In*: W.A. Mackaness, Ruas, A. and Sarjakoski, L.T. eds. *Generalisation of geographic information: cartographic modelling and applications*. Amsterdam: Elsevier, 89–112.

McMaster, R.B. and Shea, K.S., 1988. Cartographic generalisation in a digital environment: a framework for implementation in a geographic information system. *In*: *GIS/LIS proceedings*, San Antonio, TX, 240–249.

Molenaar, M., 1989. Single valued vector maps: a concept in geographic information systems. *Geo-Informations systeme*, 2 (1), 18–26.

Mustiere, S., 2006. Results of experiments on automated matching of networks at different scales. *In*: ISPRS WG II/3, II/6 Workshop 'Multiple representation and interoperability of spatial data', 22–24 February Hanover, Germany [online]. Available from: http://www.isprs.org/commission2/hannover06/hannover06.pdf.

NCGIA, 1989. The research plan of the National Center for Geographic Information and Analysis. *International Journal of Geographical Information Systems*, 3 (2), 117–136.

Open Geospatial Consortium (OGC), 1999. The openGIS™ abstract specification. topic 8: relationships between features. Version 4.openGIS™ Project Document Number 99–108r2.doc.

Open Geospatial Consortium (OGC), 2006. *In*: J. Herring, ed. *OpenGIS® implementation specification for geographic information – simple feature access – Part 1: Common architecture*. OGC 06–103r3.

Parent, C., Spaccapietra, S., and Zimányi, E., 2006. *Conceptual modelling for traditional and spatio-temporal applications: the MADS approach*. Heidelberg: Springer.

Pinet, F., Duboisset, M., and Soulignac, V., 2007. Using UML and OCL to maintain the consistency of spatial data in environmental information systems. *Environmental Modelling and Software*, 22 (8), 1217–1220.

PostgreSQL, 2009. [online]. Available from: http://www.postgresql.org/ [Accessed 8 June 2009].

Regnauld, N., 1996. Recognition of building clusters for generalisation. *In*: M.-J. Kraak and M. Molenaar, eds. *Advances in GIS research II: proceedings of the 7th international symposium on spatial data handling*. Vol. 4B. London: Taylor & Francis, 1–14.

Ruas, A., 1998. OO-Constraint modelling to automate urban generalisation process. *In*: *Proceedings of the 8th international symposium on spatial data handling*, 12–15 July Vancouver, Canada, 225–235

Ruas, A., 1999. *Modèle de généralisation de données géographiques à base de contraintes et d'autonomie*. Thesis (PhD), Université de Marne-la-Vallée.

Sester, M., 2000. Generalisation based on least-squares adjustment, *The XIXth international congress, commission 4th, international archives of photogrammetry and remote sensing, Amsterdam*, The Netherlands, 931–938.

Steiniger, S., and Weibel, R., 2007. Relations among map objects in cartographic generalization. *Cartography and Geographic Information Science (CaGIS)*, 34 (3), 175–197.

Steiniger, S., *et al*., 2008. An Approach for the classification of urban building structures based on discriminate analysis techniques. *Transactions in GIS*, 12 (1), 31–59.

Stoter, J.E., 2005. Generalisation: the gap between research and practice. *In*: *Proceedings of the 8th ICA workshop on generalisation and multiple representation*, 7–8 July, A Coruña, Spain, 10 p.

Stoter, J.E., Quak, C.W., and Hofman, A., 2009a. Harmonisation and integration of topographical concepts for a multi-scale base register topography. *In*: van Loenen, Besemer and Zevenbergen, ed. *SDI convergence, research, emerging trends, and critical assessment*. The Netherlands: Netherlands Geodetic Commission.

Stoter, J.E., *et al*., 2008. A data model for multi – scale topographical data. *In*: A. Ruas and C. Gold., eds. *Proceedings of the 13th international symposium on spatial data handling, SDH 2008* Berlin: Springer, 233–254.
Stoter, J.E., *et al*., 2009b. Specifying map requirements for automated generalisation of topographic data. *The Cartographic Journal*, 46(3), 214–227.
Stoter, J.E., *et al*., 2009c. Methodology for evaluating automated map generalization in commercial software. *Computers, Environment and Urban Systems*, 33 (5), 311–324.
Thomson, M.K. and Béra, R., 2008. A methodology for inferring higher level semantic information from spatial databases. *In*: *Proceedings of the GIS Research UK 16th Annual Conference (GISRUK 2008)*, Manchester, UK.
TOP10NL, 2005. *TOP10NL versie 2.3, februari 2005* [online]. Available from: http://www.kadaster.nl/top10nl/gegevensmodel_top10nl_2%203.pdf. (In Dutch).
Uitermark, H., 2001. *Ontology based geographic data set integration*. Thesis (PhD), Twente University.
Uitermark, H., *et al*., 1999. Ontology-based spatial data set integration. *In*: *proceedings of STDBM'99, workshop on spatio-temporal database management*, 10–11 September, Edinburgh, Scotland, 60–79.
van Oosterom, P.J.M., 2005. Variable-scale topological data structures suitable for progressive data transfer: the GAPface tree and GAP-edge forest. *Cartography and Geographic Information Science*, 32(4), 331–346.
Ware, J.M., Jones, C.B., and Thomas, N., 2003. Automated map generalisation with multiple operators: a simulated annealing approach. *International Journal of Geographical Information Science*, 17 (8), 743–769.
Warmer, J. and Kleppe, A., 2003. *The object constraint language: getting your models ready for MDA*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.
Xiang, Z., Stoter, J.E., and Ai, T., 2008. Formalization and automatic interpretation of map requirements, workshop on international cartographic commission on generalisation and multi-representations, June, 2008, Montpellier, France [online]. Available from: http://aci.ign.fr/montpellier2008/papers/24_Zhang_et_al.pdf.

Appendix 1. Prototype of IMTOP

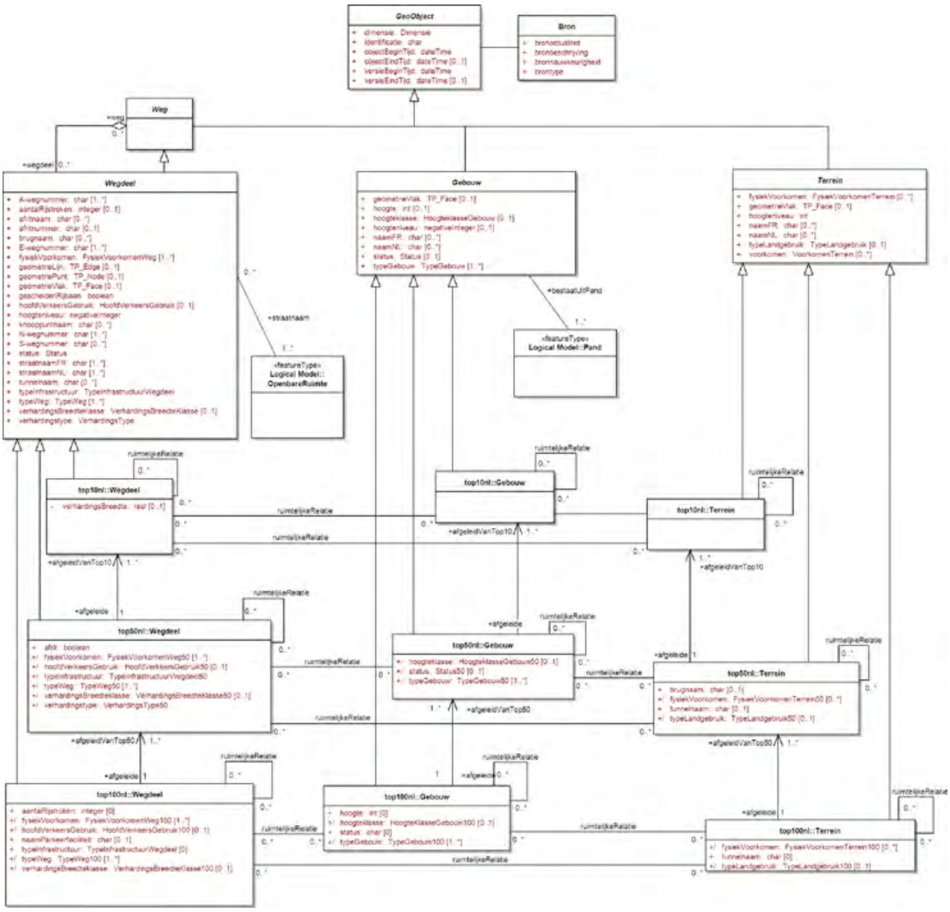

Figure A1. Prototype of IMTOP, shown for classes building (gebouw), land use (terrein) and road (wegdeel), represented at three sales: 1:10K, 1:50K and 1:100K. The 'ruimtelijkeRelatie' (spatial relationships) associations are added to specify spatial conditions between instances of the same class or of different classes within one scale (see Section 4.2.2).