Towards Cartographic Constraint Formalization for Quality Evaluation

Xiang Zhang, Tinghua Ai, Jantien Stoter and Jingzhong Li

Abstract This paper presents a first-order representation to formalize cartographic constraints for automated quality evaluation of multi-scale data. Formalizing constraints for cartographic applications is a challenging task. It requires precise definition of entities, spatial and semantic relationships for individuals, groups and classes of objects, and their (intra-/inter-scale) relationships. Also constraints defining the visual presentation of the same entities can be different depending on the scale and context. This paper categorizes and formalizes different types of information needed for the quality evaluation, based on which cartographic constraints are formalized. The formalism is demonstrated by applying it to group features such as networks and alignments, and finally to constraints of different levels of complexity. We show the potential of the proposed formalism and discuss possibilities for further development.

Keywords Formal map specifications · Multi-scale modeling · Automated evaluation of generalized maps

1 Introduction

Spatial data is being created and maintained at various scales to meet the increasing demand of geo-information. The increase of data volume and update speed has led to heterogeneous representations of geographic areas. This inevitably causes inconsistencies between multiple representations. In generalization domain,

X. Zhang (✉) · T. Ai · J. Li
School of Resource and Environmental Sciences, Wuhan University, Wuhan, China
e-mail: xiang.zhang@whu.edu.cn

J. Stoter
GISt, OTB, Delft University of Technology, Delft, the Netherlands

M. Buchroithner et al. (eds.), *Cartography from Pole to Pole*,
Lecture Notes in Geoinformation and Cartography, DOI: 10.1007/978-3-642-32618-9_7,

cartographic constraints are widely used to express user expectations concerning the quality of maps (including data and visual qualities) and hence are used to control and evaluate the quality of generalization.

Various models of constraints were proposed for map generalization and quality evaluation (e.g. Weibel and Dutton 1998; Burghardt et al. 2007). This research is a continuation of our previous work (Zhang et al. 2008). Also Touya et al. (2010) proposed a model of constraints for generalization which shares similar aspects. Note that, constraints for the evaluation differ (slightly) from those for generalization. First, constraint priority, the sequence in which constraints are evaluated, is not relevant for the evaluation of generalized results (Zhang 2012). Second, preferred action(s) are needed for modeling constraints for generalization (Burghardt et al. 2007; Stoter et al. 2009) but not for the automated evaluation. Third, corresponding relationships between objects of different scales must be explicitly expressed for the evaluation.

Besides, constraints were previously formulated in a more or less informal way, which brings about confusions in their interpretation, even for human experts (Burghardt et al. 2007; Zhang et al. 2008). Rather than proposing a new model of constraints, this paper presents a formalism that refines the existing models and provides a foundation for the machine-based interpretation of constraints. The ultimate goal is that, by interpreting the constraints, the machine 'knows' how to evaluate it, which high-level concepts, relations and contexts to detect, which algorithms/operations to use, etc. This formalism should also be useful for automated generalization. In the following, we firstly present an object-oriented model for relevant spatial entities and relationships (Sect. 2), where high-level concepts and relationships are exemplified. Section 3 formalizes cartographic constraints into first-order logic expressions. Section 4 discusses the expressive power of the formalism, implementation issues and possibilities for automated quality evaluation. More details and design principles are discussed in the doctoral thesis of Zhang (2012).

2 Spatial Entities and Relationships

This section formalizes some basic concepts pertinent to the formalization of cartographic constraints. First, we distinguish between three categories of entities: individuals (objects), universals (classes), and collections (groups). This is widely received in automated generalization (Ruas 2000; Stoter et al. 2009) and is also in line with findings in ontology research (Bittner et al. 2004; Mark et al. 2001). Then we give formal definitions of spatial entities in a multi-scale setting (Sect. 2.1) and describe the relationships between different entities (Sect. 2.2). Several examples of the formalism are shown in Sect. 2.3.

2.1 Entities, Objects and Groups

An *entity* can be either an object (feature) or a group of objects (Fig. 1). It is formally given by a 2-tuple:

$$\text{Entity} = \langle ID, Granu \rangle \tag{1}$$

where:

- *ID* is the identifier of the entity (object or group) being referred to;
- *Granu* $\in$ {*micro, meso, macro*} indicates its granularity.

Here granularity specifies the degree to which the objects are grouped. For example, a building object is a micro entity; objects grouped by a partition or into an alignment belong to a meso entity; a feature class of objects (e.g. all buildings in a dataset) is a macro entity.

Object and group of objects are two fundamental concepts derived from entity. An *object* is an individual which can be given by a triple:

$$\text{O} = \langle ID, CL, Geo \rangle \tag{2}$$

where:

- *CL* is the feature class, e.g. *Building* or *Road*;
- *Geo* is the geometry of the object, which is composed of: (i) geometry types *Geo type* $\in$ {*point, linear, areal*} and (ii) the spatial representation (coordinate pairs) *Geo. rep* = $\{p_1,\ldots, p_n\}$.

We identify groups as explicit entities with identifiers, because groups have structural properties (e.g. density, spatial distribution and other statistical properties) which should be preserved during the generalization. A *group* is a collection of objects, and can be given by a 4-tuple:

$$\text{Gr} = \langle ID, OC, R, Geo \rangle \tag{3}$$

where:

- $OC = \{O_1,\ldots, O_n\}$,n $\geq$ 0 is a collection that forms the group;

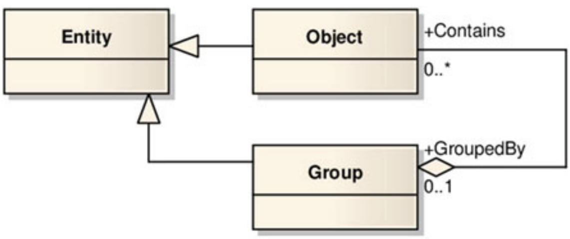

Fig. 1 UML diagram of entity, object, group and their relations (Zhang 2012)

- *R* is the relationship that holds among the objects in the collection;
- *Geo* is the representative geometry of the group.

This definition is flexible enough to cover various kinds of group entities. First, a group can be a collection where some relationship holds for its members (e.g. alignment), or between the members and some other objects (e.g. neighborhood and partition relationship). Second, the above definition also allows groups of objects from different feature classes that create 'higher-level' phenomena (e.g. a 'housing aggregate' composed of a house, a garden and an outbuilding). The group entity in the second sense is similar to the concept of composite object described in (van Smaalen 2003).

Finally, a *feature class* only consists of meta information and is given by $\mathrm{FeaCls} = \langle Cls, Geotype \rangle$, where $Cls \in \{Building, Road, \ldots\}$ and $Geotype \in \{point, linear, areal\}$.'*Geotype*' is different from '*Geo*' in the Object definition (*Def.* 2) which has type and geometry properties.

The semantic relationship describing that an object is an instance of a feature class can be defined as:

$$\begin{aligned} &InstanceOf(O_i, FeaCls_i) \\ &\qquad \equiv (O_i.CL = FeaCls_i.Cls) \\ &\qquad \wedge (O_i.Geo.type = FeaCls_i.Geotype). \end{aligned} \tag{4}$$

Now, we can express all instances of a feature class C with a specialized group:

$$\mathrm{Gr_C} = \langle \mathrm{ID}, \{\mathrm{O_i} | (\forall \mathrm{O_i})(\mathrm{InstancOf}(\mathrm{O_i}, \mathrm{C}))\}, \emptyset, \mathrm{Geo} \rangle \tag{5}$$

Here the object collection ($\mathrm{Gr_C}$. *OC*) is given by a set-builder notation, meaning that for all objects in the domain of discourse, anything that is an instance of class *C* is an element of $\mathrm{Gr_C}$. *OC*. Hence Gr_C is actually the extension of class *C*. In a specific application, Gr_C can be used to indicate all the instances of a class that exist in a particular dataset.

Note in *Def.* 5 that we set $\mathrm{Gr_C}$. $R = \varnothing$ since the relation is already defined in the set-builder notation of $\mathrm{Gr_C}$. *OC*. For meso groups, a non-empty relationship *R* (see *Def.* 3) should be explicitly specified to explain how the group members are related (see examples in Sect. 2.3).

We also use the following predicate syntax to denote that O_i is an instance of some class, where <class_name> is a placeholder (e.g. *Built_up_area*(O_i) means O_i is a built-up parcel):

$$< class\_name > (\mathrm{O_i}) \equiv \mathrm{InstanceOf}(\mathrm{O_i}, < class\_name >) \tag{6}$$

2.2 Spatial Relationships Between Entities

This section focuses on a set of relationships between individuals (i.e. spatial entities) that is relevant to the proposed evaluation. We further distinguish between the relationships and properties within the same scale (intra-scale relationships) and those of different scales (inter-scale relationships).

Intra-scale relationships between individuals in the geospatial domain **D** can be characterized in terms of qualitative and quantitative relationships. The relationships in topological space are usually qualitative, whereas the ones in metric space are usually quantitative in nature.

We choose to formalize spatial relationships using functions instead of predicates or mathematic relations. Because predicates or mathematic relations are mostly qualitative, so they cannot describe the continuous change of spatial properties over scale transitions. Besides, although quantitative relationships can be qualified, the qualification process can be cognitively plausible and context dependent (Worboys 2001). The functional specification of spatial relationships is given below:

$$\begin{aligned} f \quad &: \quad \mathbb{D}^{\mathrm{n}} \to \mathbb{R}^{\mathrm{n}} && (\text{Quantitative}) \\ &: \quad \mathbb{D}^{\mathrm{n}} \to \{\text{true}, \text{false}\} && (\text{Qualitative}) \\ (x_1, \ldots, x_n) &\mapsto f(x_1, \ldots, x_n) \end{aligned} \tag{7}$$

where $\mathbf{D}^{\mathrm{n}}$ is the domain of function f which takes $x_1,\ldots,x_{\mathrm{n}}$ as its input (x_i is a spatial entity); $\Re^{\mathrm{n}}$ is the output space of f. Although $\Re$, or more precisely $\Re^{+}$, is in most cases sufficient, $\Re^{\mathrm{n}}$ is specified to allow for a larger degree of flexibility. Ai et al. (2008) for instance presented a Fourier transform based shape descriptor, which yields a series of numbers to characterize shapes. This case can be formalized in a function: $FourierShape(x) = (c_1, \ldots, c_n)$.

This functional view provides a unified approach to the formalization of qualitative and quantitative relationships. Qualitative relationships in particular can be seen as functions with Boolean output. Examples of relationships and properties in a functional form are $Size(x_1) = 4\,\mathrm{mm}^2$, $Distance(x_1, x_2) = 3\,\mathrm{mm}$ and $Contain(x_2, x_3) = true$.

Contextual relationships are special intra-scale relationships. According to Mustière and Moulin (2002), contextual relationships can be divided into: (i) being part of a significant group (e.g. alignments), (ii) being located in a particular area, and (iii) being in relation to the 'same-level' entities.

The contextual relationship here only means 'being located in a particular area'. Being part of a significant group can be expressed by the *member-of* relationship between a group and its members (see *Def.* 3). The third type of relationships is formalized as normal intra-scale relationships (*Def.* 7). The concept of 'same-level' is defined by our division of entities into individuals, collections and universals. The contextual relationship that an object is located in a particular context can be defined as follows:

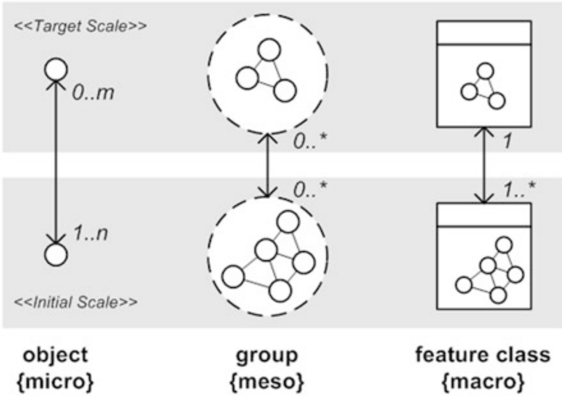

Fig. 2 Inter-scale relationships between objects, groups and classes (Zhang 2012)

$$\mathrm{InContext(x)} \equiv (\exists \mathrm{c})(\mathrm{Context(c)} \wedge \mathrm{Contain(c, x)}). \tag{8}$$

Our definition of spatial context requires that *c.CL* = *<Context>* and *c.Geo.-type* = *areal*. For example, in describing a building b_1 located in urban areas, we write *InUrban*(b_1), meaning that there exists an instance c_1 such that both *Urban*(c_1) (see *Def.* 6) and *Contain*(c_1, b_1) hold. Note that we omit the context instance *c* from the notation *InContext*(*x*), because we only concern what type of context x is located in but not which context instance. Uncertainty issues should be handled in measuring contextual relationships.

Inter-scale relationships. Spatial entities of different scales that represent the same real-world phenomena are related (linked) by inter-scale relationships (Bobzien et al. 2008). The inter-scale relationships between entities of different granularities are illustrated in Fig. 2.

The inter-scale relationships have varying multiplicities depending on the granularity of involved entities. For individual objects, 1-to-1 or 1-to-0 is possible if object was kept or removed by generalization; many-to-many (n-to-1 or n-to-m) relationships are caused by aggregation or typification. Alternatively, n-to-m relationship between micro-objects can be modeled by 1-to-1 relationship between meso-groups. For feature class level entities, 1-to-1 relationship is the common case, though sometimes several classes may be merged into one class (n-to-1).

Here, we decompose the many-to-many relationship into a set of 1-to-1 relationships. The same strategy was discussed in Bobzien et al. (2008). We call it *corresponding* relationship and define it as a binary relationship:

$$Corr\left(E_i^{Sp}, E_j^{Sq}\right) \begin{array}{ll} : \mathbb{D}^2 \to \{\mathrm{true}, \mathrm{false}\} & (\mathrm{Certain}) \\ : \mathbb{D}^2 \to [0, 1] & (\mathrm{Uncertain}) \end{array} \tag{9}$$

where *Sp* and *Sq* indicate the scales of the respective entities; *p* and *q* are scale variables. The corresponding relationships created by generalization are regarded as certain links; whereas those created by data matching are uncertain links (with matching probabilities).

2.3 Meso-Groups as Examples

In this section, examples are given on meso-level groups, namely networks and alignments, using the formalism described in Sects. 2.1 and 2.2.

Networks are characterized by elementary properties such as connectivity and collective ones such as spatial distributions and patterns (e.g. dendritic pattern). A network feature is given by a group entity with a connectivity relationship:

$$Gr_{net} = \langle ID, OC = \{o_1, \ldots, o_n\}, Connectivity(OC), Geo\rangle \tag{10}$$

where $o_i.Geo.type = linear$ is required; Geo is the representative geometry of the network (e.g. catchments of rivers). Usually, $Gr_{net}.Geo.type = areal$.

By introducing the concept of *path*, p, we can define the relationship connectivity as:

$$\begin{aligned} Connectivity\Big(OC\Big) \equiv \Big(\forall o_i, o_j\Big)\Big(\Big(o_i, o_j \in OC\Big) \\ \Leftrightarrow Touch\Big(o_i, o_j\Big) \\ \vee \Big(\exists p\Big)\Big(\Big(p \subseteq OC \wedge Touch\Big(o_i, p\Big) \wedge Touch\Big(o_j, p\Big)\Big)\Big) \end{aligned}$$

where $\Leftrightarrow$ is a bidirectional logical implication; $p = \{o_1,\ldots,o_m\}$, $m < n$ is a sequence of connected spatial features such that $(\forall o_i,o_{i+1})((o_i,o_{i+1} \in p) \Rightarrow Touch(o_i,o_{i+1}))$. This definition also allows for connected polygons.

More generally, we can define a binary relationship that hold between connected objects:

$$\mathrm{Connected}\big(o_i, o_j\big) \equiv (\exists p)\big(Touch(o_i, p) \wedge Touch\big(o_j, p\big) \vee Touch\big(o_i, o_j\big)\big). \tag{11}$$

Alignments are linear arrangements of individuals (e.g. buildings, ponds, islands). Such linear patterns should be kept in generalization (Christophe and Ruas 2002; Stoter et al. 2009). An alignment is given by a group entity with a relationship called *Aligned*:

$$Gr_{align} = \langle ID, \{o_1, \ldots, o_n\}, Aligned(o_1, \ldots, o_n), Geo\rangle \tag{12}$$

where *Geo* is usually represented by the skeleton of the alignment. This skeleton can be used to indicate the virtual (curve) line to which the group members align, so we require that $Geo.type = linear$. The spatial relationship, *Aligned*, that glues the group members together is based on a general notion of consistency and can be defined recursively:

$$\begin{aligned} &Aligned(o_1, o_2) = Consistent(\{o_1\}, o_2) \\ &Aligned(o_1, o_2, o_3) = Consistent(\{\{o_1\}, o_2\}, o_3) \\ &\cdots = \cdots \\ &Aligned(o_1, o_2, \ldots, o_n) = Consistent(\{\ldots\{\{o_1\}, o_2\}, \ldots\}, o_n) \end{aligned}$$

For the semantics of *Consistent* one is referred to Zhang et al. (2013). Major properties of alignments include spatial distribution and orientation. Alignments or clusters are regular spatial distributions, so we use homogeneity (Christophe and Ruas 2002; Zhang et al. 2013) to quantify the degree of regularity of alignments or clusters. It can be formally expressed as $Homo(align_i) : \mathbf{D} \rightarrow [0,1]$. Similarly, $Orientation(align_i) : \mathbf{D} \rightarrow \Re^{+}$ expresses the orientation of $align_i$. Parallel relationship between alignments, $Parallel(align_i, align_j) : \mathbf{D}^2 \rightarrow [0,1]$, is also useful if two alignments (especially curvilinear ones) are to be compared in terms of orientation.

3 Constraints as First-Order Expressions

In expressing cartographic constraints, several side notes should be considered. First, the domain of discourse $\mathbf{D}$ is subdivided into sub-domains $\mathbf{D}^{S1}, \ldots, \mathbf{D}^{Sn}$, distinguishing between entities at different scales. So o^{S1} and Gr^{S2} are an object and a group of different scales. Second, aliases are frequently used to simplify the expressions (e.g. $align_1$ is a Gr_{align}). Third, the conditions to be respected by constraints are formalized as conditional expressions, e.g., $Size(o_1) \geq \mu$ where μ is a user-defined parameter. In the following, we formalize a subset of constraints drawn from NMA specifications (Stoter et al. 2009) using a first-order language.

To begin with, a minimum dimension constraint (C1) saying that area of any target polygon should be larger than a certain threshold is formalized as:

$$(\mathrm{C1})\left(\forall o_i^T\right)\left(\left(o_i^T.Geo.type = areal\right) \Rightarrow Size\left(o_i^T\right) \geq \mu\right).$$

Note that, o_i^T, with T denoting target scale, is used to simplify the expression $(o_i)(o_i \in \mathbf{D}^T)$ in this context. Likewise, o_i^I denotes an object at the initial scale. The second constraint (C2) is a requirement saying that important initial buildings should be kept:

$$(C2)\left(\forall b_i^I\right)\left(ImportantBuilding\left(b_i^I\right) \Rightarrow \left(\exists b_i^T\right)\left(Corr\left(b_i^I, b_i^T\right)\right)\right)$$

where b_i is an alias of an instance of building class in this case; the type predicate (*Def.* 6) restricts the range of the quantifier to important buildings. C2 explicitly states that for every important building in the initial data, there exists at least one building in the target data that correspond to it.

The third constraint (C3) is a preservation constraint with added contexts (scopes). It requires that target shape should remain concave if the initial polygons are of high concavity:

$$\begin{aligned}(C3)\left(\forall a_i^I\right)&\left(\left(Concavity\left(a_i^I\right) \geq \mu\%\right) \wedge \left(\exists a_i^T\right)\left(Corr\left(a_i^I, a_i^T\right)\right)\right.\\ &\left.\Rightarrow Similar\left(Concavity\left(a_i^I\right), Concavity\left(a_i^I\right)\right)\right).\end{aligned}$$

Note that, the degree of concavity *Concavity* should be measured at an implementation level, and 'high concavity' is application dependent (e.g. $\mu = 95\ \%$). We use a predicate *Similar* here since its concrete form is unknown at this moment. Note that the scope that limits the polygons in the initial data states that C3 only considers the initial objects (i) that are highly concave in shape and (ii) that were kept in the target data.

The next example shows a constraint requiring that proximate and roughly parallel features (line–line or polygon–polygon) should become topologically adjacent (C4):

$$\begin{aligned}(C4)\left(\forall o_i^I, o_j^I\right)\Big(\Big(Distance\left(o_i^I, o_j^I\right) < \mu_1\Big) \wedge \Big(Parallel\left(o_i^I, o_j^I\right) \geq \mu_2\Big) \\ \wedge \left(\exists o_i^T, o_j^T\right)\Big(Corr\left(o_i^I, o_i^T\right) \wedge Corr\left(o_j^I, o_j^T\right) \\ \Rightarrow Adjacent\left(o_i^T, o_j^T\right)\Big)\Big)\end{aligned}$$

where μ_1, distance, and μ_2, degree of parallelism, are to be specified in certain applications. Similar to C3, C4 only considers the initial object pairs where the two objects are close and parallel to each other and where both objects were kept in the target data. This is an example where the topological relationship between objects should change. Next, we exemplify that topological relationship (connectivity) should be preserved (C5) on top of the relationship Connected (*Def.* 11):

$$\begin{aligned}(C5)\left(\forall o_i^I, o_j^I\right)\Big(Connected\left(o_i^I, o_j^I\right) \\ \wedge \left(\exists o_i^T, o_j^T\right)\Big(orr\left(o_i^I, o_i^T\right) \wedge Corr\left(o_j^I, o_j^T\right) \\ \Rightarrow Connected\left(o_i{}^T, o_j^T\right)\Big)\Big)\end{aligned}$$

where o can be either linear or areal features. C5 can be used to check whether the connectivity has been kept between any two objects or for network features. Consider the following variation of C5:

$$(C5')\left(\forall net_i^I\right)\left(\exists net_i^T\right)Corr\left(net_i^I, net_i^T\right)$$

where *net* is an alias of Gr_{net}. One may argue that the existence of a corresponding network in the target data automatically ensures the connectivity of the initial network. This is however not true as is shown in Fig. 3. In the illustration, the initial network was kept after generalization, but some river branches such as l_2 become disconnected from the initial network. Such a violation to connectivity can be identified by C5 but not C5′.

Finally, we formalize some constraints on alignments (al_i). The first one (C6) requires that alignment should be kept; the second one (C7) requires that for the target alignment its orientation should be similar to the initial one; the third one (C8) requires that the spatial distribution (regularity) of alignment members should be preserved or even enhanced.

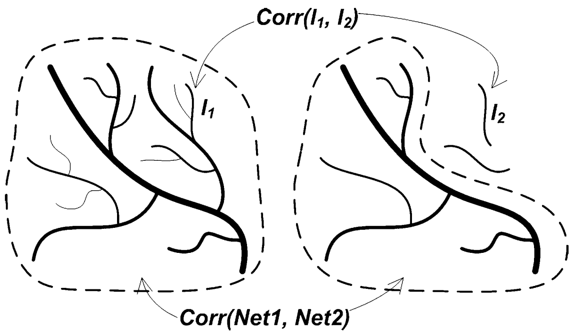

Fig. 3 Example of connectivity: initial network (*left*) and generalized network (*right*) (Zhang 2012)

$$(C6)(\forall al_i)\left(al_i \subset D^I\right) \Rightarrow \left(\exists al_j\right)\left(al_j \subset D^T\right) \wedge Corr\left(al_i, al_j\right)$$

$$(C7)\left(\forall al_i^I\right)\left(\exists al_i^T\right)\left(Corr\left(al_i^I, al_i^T\right) \Rightarrow \left(\left|Orientation\left(al_i^I\right) - Orientation\left(al_i^T\right)\right| \leq \theta\right)\right)$$

$$(C8)\left(\forall al_i^I\right)\left(\exists al_i^T\right)\left(Corr\left(al_i^I, al_i^T\right) \Rightarrow \left(Homo\left(al_i^T\right) \geq Homo\left(al_i^I\right)\right)\right).$$

4 Discussion

4.1 Expressiveness of the Formalism

The formalism described is powerful enough to express constraints with different combinations of *entity class*, *scope modifier*, *constrained property* and *condition* to be respected. A detailed account of the syntactic structure of constraints estimates that in theory nearly 1,000,000 kinds of constraints can be expressed (Zhang 2012). The expressive power also comes from the ability to specify semantic, conditional or contextual scopes to limit the entities to be considered by a constraint. For example a constraint 'the road leading to a building in a peninsula should be kept' can be formalized as:

$$\begin{aligned}(\mathrm{C9})\left(\forall r_i^I b_j^I\right)\left(LeadTo\left(r_i^I, b_j^I\right) \wedge InPenisula\left(b_j^I\right)\right) \\ \Rightarrow \left(\exists r_i^T, b_j^T\right)\left(Corr\left(r_i^I, r_i^T\right) \wedge Corr\left(b_j^I, b_j^T\right)\right).\end{aligned}$$

In this example, the range of quantification is specified through scope modifiers, i.e., the relationship *LeadTo* and the contextual relationship *InPenisula* (see *Def.* 8). This constraint states that for the initial roads and buildings that satisfy that buildings are located in peninsulas and roads lead to those buildings, there exist target objects that correspond to those initial roads and buildings (i.e. initial objects are kept).

4.2 Implementation and Use of the Formalism

This work provides a solid foundation for constraint formalization and is more geared towards a machine-based interpretation and reasoning of the constraint expressions, so that the meaning is clear to both human and computers. The proposed formalism is independent of implementation techniques or standards but can be translated into specific techniques. For example, the Object Constraint Language (OCL) (Warmer and Kleppe 2003) appears to have the potential to implement the presented formalism, since OCL roots in first-order predicate logic and it is able to express entities and relationships using UML constructs such as classes and methods. Research in geospatial domain has demonstrated the use of OCL and UML in modeling topological rules in spatial databases (Bejaoui et al. 2010) and specifications for MRDBs (Friis-Christensen et al. 2005; Stoter et al. 2011).

OCL is a more user-friendly language which requires much less mathematical understanding. To further bring down the barrier of use, we suggest designing GUIs according to the structure discussed in Zhang (2012) to enable a click-and-drag manner of constraint definition, which can then be translated into a machine readable format automatically.

4.3 Possibilities for Automated Quality Evaluation

It is interesting to see automated evaluation of generalized results in a logical framework, i.e., by interpreting the formal constraints and reasoning if they hold in data. This requires that all relevant entities, properties and relationships are stored in a database. Take constraint C9 for example, besides building and road instances, the objects that satisfy the *LeadTo*, *InPenisula* and *Corr* relationships (detected offline) should also be stored in relational tables. Because OCL constraints can be directly translated into SQL queries, C9 can be written in OCL in a way that buildings and roads that do not satisfy the constraints are returned.

However, due to the amount of information and the purpose-dependent nature of map generalization, not all information can be stored and some has to be enriched on-demand. In a fully formal approach, data enrichment operations could also be semantically annotated such that the logical unit can find the operations by semantic matching. In extreme cases high-level concepts such as peninsula can be defined by a semantic network or formal ontology, the machine can chain different operations to detect the concepts (e.g. Zhang et al. 2008). This vision is very challenging and requires quite a lot of research on the semantic aspect of geo-operations.

A disadvantage of using Boolean logic for automated evaluation is, however, that a constraint is always evaluated to be either violated or not. Soft evaluation (with violation degrees) is hence not possible. Given this shortcoming, the proposed formalism still provides a powerful way to precisely define cartographic constraints. In addition, automated evaluation in a logical framework is still effective in identifying database errors and preventing undesired insertions into the database.

Acknowledgments This research was supported by the National High-Tech Research and Development Plan of China (No. 2012AA12A404). We thank the anonymous reviewers for their remarks that improved the quality of this paper.

References

Ai T, Shuai Y, Li J (2008) The shape cognition and query supported by Fourier transform. In: Ruas A, Gold C, Cartwright W, Gartner G, Meng L, Peterson MP (eds) Headway in spatial data handling, LNGC, pp 39–54

Bejaoui L, Pinet F, Schneider M, Bédard Y (2010) OCL for formal modelling of topological constraints involving regions with broad boundaries. GeoInformatica 14(3):353–378

Bittner T, Donnelly M, Smith B (2004) Individuals, universals, collections: on the foundational relations of ontology. In: Proceedings of the 3rd international conference on formal ontology in information systems, pp 37–48

Bobzien M, Burghardt D, Petzold I, Neun M, Weibel R (2008) Multi-representation databases with explicitly modeled horizontal, vertical, and update relations. Cartogr Geogr Inf Sci 35(1):3–16

Burghardt D, Schmidt S, Stoter J (2007) Investigations on cartographic constraint formalisation. In: 10th ICA workshop of ICA commission on generalisation and multiple representation, Moscow

Christophe S, Ruas A (2002) Detecting building alignments for generalisation purposes. In: Richardson DE, Van Oosterom P (eds) Advances in spatial data handling. Springer, Heidelberg, pp 419–432

Friis-Christensen A, Jensen CS, Nytun JP, Skogan D (2005) A conceptual schema language for the management of multiple representations of geographic entities. Trans GIS 9(3):345–380

Mark DM, Skupin A, Smith B (2001) Features, objects, and other things: ontological distinctions in the geographic domain. In: Montello DR (ed) Spatial information theory: foundations of geographic information science. LNCS, pp 488–502

Mustière S, Moulin B (2002) What is spatial context in cartographic generalisation? In: Symposium on geospatial theory, processing and applications, volume 34-B4, pp 274–278

Ruas A (2000) The roles of meso objects for generalization. In: Proceedings of the 9th international symposium on spatial data handling, pp 3B50–3B63

Stoter J, Burghardt D, Duchêne C, Baella B, Bakker N, Blok C, Pla M, Regnauld N, Touya G, Schmid S (2009) Methodology for evaluating automated map generalization in commercial software. Comput Environ Urban Syst 33(5):311–324

Stoter J, Visser T, van Oosterom P, Quak W, Bakker N (2011) A semantic-rich multi-scale information model for topography. Int J Geogr Inf Sci 25(5):739–763

Touya G, Duchêne C, and Ruas A (2010) Collaborative generalisation: formalisation of generalisation knowledge to orchestrate different cartographic generalisation processes. In: Fabrikant S, Reichenbacher T, van Kreveld M, Schlieder C (eds) Geographic information science. LNCS, pp 264–278

van Smaalen J (2003) Automated aggregation of geographic objects: a new approach to the conceptual generalisation of geographic databases. PhD thesis, Wageningen University, Wageningen

Warmer J, Kleppe A (2003) The object constraint language: getting your models ready for MDA, 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Boston

Weibel R, Dutton G (1998) Constraint-based automated map generalization. In: Proceedings 8th international symposium on spatial data handling, pp 214–224

Worboys MF (2001) Nearness relations in environmental space. Int J Geogr Inf Sci 15(7):633–651

Zhang X (2012) Automated evaluation of generalized topographic maps. PhD thesis, Twente University, Enschede

Zhang X, Stoter J, Ai T (2008) Formalization and automatic interpretation of map requirements. In: 11th ICA workshop of ICA commission on generalisation and multiple representation, Montpellier

Zhang X, Ai T, Stoter J, Kraak MJ, Molenaar M (2013) Building pattern recognition in topographic data: examples on collinear and curvilinear alignments. GeoInformatica 17(1):1–33