Project Number	**ESPRIT / LTR /24 939**
Project Title	**AGENT**
Deliverable Type	**Report**

Deliverable Number	**D A2**
Internal ID	**AG-98-08**
Contractual Date of Delivery	**01/09/98 (technical Annex)**
Actual Date of Delivery	**26/08/98**
Version	**2.0**
Workpackage/Task contributing	**A.2**
Authors	**Dept. of Geography, University of Zürich <UNI-ZH>**
Confidentiality	**Public**

Title of Deliverable	**Constraint Analysis**

Abstract	**Constraints are limitations to resources, space, time and potential solutions to problems that must be respected in order to achieve useful results. Constraints to map generalisation derive from controls to map production, as well as procedural considerations. After describing the roles of constraints, specific graphic and geographic constraints are itemised and illustrated. Connections between constraints, data , knowledge, process and agent modelling are described and explored in a using a variety of examples.**
Keyword List	**map generalization, constraints, feature class, map controls, measures, spatial conflicts, multi-agent systems**

THIS PAGE DELIBERATELY LEFT BLANK

Executive Summary

Abstract. A major reason for limited success in automating map generalization is the lack of frameworks which express the constraints under which various algorithms must operate. In this report, constraints are identified as concrete consequences of controlling factors to generalization, such as map purpose and scale, symbolism, media and data quality, as well as those of the map production process and its procedural logic. Chapter one summarises the key concepts that the analysis builds upon. In chapter two, after discussing the origins and roles of constraints, we propose a framework for applying analytic and constructive generalization tools that yield design solutions which satisfy the constraints. Chapter three begins an itemisation of constraints that map symbolism itself imposes, regardless of the phenomena or feature classes being portrayed, illustrating more than 30 graphic constraints. Itemising continues in chapter four, which systematically describes specific constraints that geographic entities can impose on generalisation; the themes discussed are boundaries, landuse and landcover, streets, roads, buildings, watercourses, waterbodies, digital terrain models and contour lines. Ways in which constraints can affect map generalisation (and map production in general) are discussed in chapter five, in terms of semantic modelling, process modelling and agent modelling. In a multi-agent system, constraints can serve to define goals of agents, the satisfaction of which is evaluated by means of measures that describe the current state of the map, usually locally, and often with reference to the ungeneralised version. Throughout the report, examples of measures and heuristics are given to illustrate ways in which a generalization system can determine where to generalise and can evaluate the quality of the results.

Discussion. Dealing with constraints in such a general way is difficult because of complexities in the semantic webs that describe how controls and constraints interact with one another and affect generalization tasks. We regard this as a «start-up cost» to creating a holistic generalization system rather than a permanent impediment. Because many maps contain common sets of features and symbologies, because relationships among feature classes tend to be predictable, and because the effects of scale and graphic limits are so universal, a specification of constraints for a given generalization problem is likely to be applicable to other kinds of maps with relatively minor modifications.

Relation to other Work Packages and Tasks. This report serves as input to several other tasks. Specifically, tasks A.1, «generalisation modelling in an agent paradigm,» A.3, «Geographic object modelling,» and B.1, «System modelling» have overlapping content and will be further integrated during the next phase of the project. In addition, tasks C.1, «Measures for agents» and C.2, «Measures for organisations» are strongly related by virtue of the necessity to quantify constraints, measure their strength and evaluate their satisfaction. Identification of key constraints also is a necessary input to task D.1, «Selection of basic algorithms» in that generalisation algorithms that fail to respect relevant constraints (and there are many such algorithms) are not helpful when pursuing the type of holistic solutions this project has mapped out for itself.

Table of contents

1 Introduction

1.1 Purpose

The AGENT *Technical Annex* (IGN 1996) describes the objectives of Task A2 as developing an «ontology of constraints and the way they are used in the modelling of agents and a description of how they should be used in the modelling of agents» and goes on to specify that constraints «define the conditions and limits of generalisation, and thus define under which conditions a particular operator should be activated» (p. 37). It should be noted that this objective requires not only articulation of constraints but also a framework for relating them to activities of agents. This requirement is a bit problematic, as it requires assumptions to be made about modelling agents in advance of tasks that are intended to do this. But we feel that meeting this objective is nonetheless possible. This document will generally follow the approach used by Weibel and Dutton (1998) in conceptualising the origins and roles of constraints to map generalisation:

> In the conceptual framework proposed by Beard (1991), a constraint is understood as a condition similar to the predicate of a rule. In contrast to a production rule, however, Beard's definition does not bind a constraint to a particular action. *We prefer to define a constraint to mean a limitation that reduces the number of acceptable solutions to a problem.* This notion is similar to the one used in constraint programming in computer science (van Hentenryck et al. 1996). However, in our usage constraints are not used directly as a foundation of a programming technique that focuses on operations such as constraint propagation, satisfaction, normalization, and optimization, but rather as a framework for designing a generalization application as a whole. Hence, *constraints in our case can be thought of as a design specification to which solutions should adhere*, helping to specify the nature of cartographic or database products and the techniques (algorithms, strategies, etc.) that are necessary to derive them.
>
> (Weibel and Dutton 1998: 2; emphasis added)

Some «design specifications» for maps are thematic (content), others are graphic (appearance); some describe a final product, others describe methods for achieving it. Extending the approach summarised above does not imply that software design considerations will be ignored. On the contrary, we feel it is extremely important to relate analysis of constraints to the means by which they will be implemented (i.e., object-oriented databases and multi-agent systems). Still, we intend to make as few assumptions as possible about system design while developing a viable framework for a design. Our approach to describing constraints will be both general and specific, in that it relates both to characteristics of map symbolism and of specific classes of geographic data. Following a discussion of terms and definitions, the origins, nature and roles of constraints are described in chapter 2. The symbolism-related aspects of controls are covered in chapter 3; these tend to be common to all maps, regardless of content. Tables and figures summarise and illustrate these constraints, indicating a few ways in which they might be implemented. chapter 4 then describes a set of phenomenon-specific constraints for the feature classes to be used in this project. Possible roles of constraints in generalisation system design are presented in chapter 5, covering semantic, process and agent modelling. These are intended to be heuristic, as they are preliminary explorations and other tasks are more directly focused on their specification. Chapter 6 very briefly summarises the overall findings of this task.

1.2 Terms and Concepts

This section attempts to differentiate concepts and to discuss terminology. All the above terms can have different senses: common sense, cartographic sense, computer science sense, and so on. The terminology choice is by necessity arbitrary, and is subject to refinement; if it has utility it will persist.

A map feature can be described as a database (DB) object or a geographic object (which may also be groups of objects with topologic, semantic or other relations). Such objects may not necessarily be objects in the object-oriented (OO) programming sense. Sometimes groups of related geographic objects are referred to as *phenomena*.

Certain terms related to data and process modelling in map generalisation will be encountered in this document. Some examples are:

- Conflict
- Constraint
- Controls
- Evaluation
- Heuristic
- Measure / measurement
- Phenomena
- Property
- Rule
- Strategy
- Tactic
- Treatment

Please refer to the glossary of terms at the end of this document for a full set of working definitions.

1.2.1 Controls

All professionally-produced maps have specifications that define their content, size, scale, colours, symbolism and usually procedures or rules to be observed at various stages of production, including generalisation. Cartographers generally call such considerations *controls* of map production and generalisation, as they act to specify parameters to the process. Controls are critical in specifying many of the *constraints* required for successful generalisation. However, many aspects of *holistic* generalisation are not specified, or are under-specified by controls, as we will see.

1.2.2 Constraints

The effect of constraints is to reduce the number of possible results of a process, while at the same time increasing the proportion of acceptable ones. The two basic categories of constraints are:

- *geographic* and *cartographic constraints* (arising from characteristics of data and map specifications)
- *process constraints* (arising from resource limitations and workflows)

These can interact in processing data, as illustrated by figure 1.1 which indicates:

- Possible treatments (or possible paths to results): the possible evolution of the DB (1a2d5, 1a2b4, 1b3c5).
- Possible states: any intermediate state of the DB during the process (1,2,3,4,5).
- Possible results: any state of the DB that satisfy all the constraints (4,5).

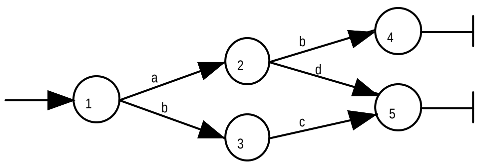


Fig. 1.1: A hypothetical network of constraints in a generalisation process

1.2.3 Key Aspects of Constraints

1 A constraint can be specified as something to *maintain* or something to *avoid*. Many constraints can be expressed either way, such as «maintain a certain separation between line symbols,» or «avoid overlapping line symbols.» In some cases, one is done while constructing symbolism, the other while evaluating it, but some algorithms might include both phases.

2 A constraint can be *absolute*, something to *optimise* or both. For example, it may be absolutely required to maintain an object's legibility or its topological type. Planimetric accuracy may also have an absolute limit. Within that limit positional accuracy can be optimised to be «as good as *controls* should specify if a specific constraint is absolute or not.

3 A constraint has an application *scope* or extent. e.g., all objects, all the objects of a class, a region, and may involve relations between objects or just individual objects. We distinguish between *spatial* scope and *contextual* scope.

4 A constraint can be *intrinsic* or *extrinsic*. Intrinsic constraints (such as maintaining legibility or avoiding overlaps) consider only one state of an object or a database. Extrinsic constraints (such as maintaining planimetric accuracy or characteristic shape)compare the state of an object with the initial or some other prior one.

5 A constraint can be *independent* or *contextual.* Independent constraints consider only one object, e.g., a building's area must exceed a minimum size; a river's tributary must have an order of 2 or more to be shown. Contextual constraints consider relations between objects, e.g., two buildings cannot occupy the same location; a river must stay inside a valley.

6 Constraints may operate differently at different scales of spatial analysis, i.e., *local*, *global*, *intermediate* (also termed *micro*, *macro* and *meso*), but their boundaries are somewhat fuzzy. Global constraints influence intermediate ones, which then may influence local ones, but not normally the reverse. For example, map scale and line weights, being global constraints, can trigger decisions to dilate polygons that would otherwise be eliminated. Dilating polygons, however, does not change map scale except for the enlarged objects, and changing a polygon's line weight to avoid dilating it is generally not an acceptable tactic.

7 *Violations* of constraints can be detected and evaluated by analysing micro-, meso- and macro-scale data using one or more *measures*.

1.2.4 Relations between constraints

The distinctions made above lead to different way to deal with constraints:

1 Several constraint can be independent or contextually related. If they affect one another, we can order their relations by *prioritising* them, or (as relations between constraints are not always transitive or capable of being ordered linearly) model them as a *semantic network*. In general, it is possible to assign relative importance's (priorities) to constraints, but these can (and should) be altered in specific contexts.

2 It is important to differentiate the *priority* and the *severity* of constraints. Severity is defined as the degree to which a constraint is violated, using some metric that measures yield when applied to map data or simulated symbolism. The constraint's priority is how important it is, usually in relation to other constraints (e.g., a ranking). A constraint violation can be quite severe (a building is very small) but still have a low priority level (because in that situation, imperceptibility is less important than the fact that building density is very high). Severity and priority, however, may not always be independent; in many cases the severity of a constraint influences its priority, but not always.

3 As soon as several constraints are allowed to operate at once, a *constraint management system* is needed to take into account relations among constraints, identifying their dependencies, and organising processes to reflect constraint severities, priorities and scopes.

1.2.5 Heuristics and Domain Knowledge

This refers to information from experience, knowledge of cartography, algorithms and strategies that may help to propose and eliminate alternative solutions, in order to guide the generalisation process and improve the quality of the results.

Examples:

1 Decomposing problems in macro-operations, e.g., follow feature class priorities to treat the road first, then the buildings along it.

2 Ordering processes so as to make major changes before minor ones.

3 Knowing that the parameters of some algorithm are always in a given range, and/or are usually derivable from certain measurements.

4 Knowing that a given algorithm is not appropriate for certain kinds of objects.

Many of the «rules-of-thumb» employed by cartographers are heuristics, and have proven to be difficult to formalise, often because there are a variety of exceptions, many of which are difficult to characterise (often because they are graphic rather than verbal). But such domain knowledge is an important source of constraints, as is evident in discussions of geographic themes in section 4.

1.2.6 Rules

The analysis of constraints presented below may involve *rules*, although the development of rules for generalisation is not properly a part of this task. Where the term is used, it is in the sense of knowledge based systems (KBS), that is, as a consequence of following or violating a law. Thus a rule in a KBS may derive from constraints, and often results in triggering an action (or a chain of actions).

Definition:

- A production, i.e., IF condition THEN conclusion

Types of rules:

- Characterisation rule: (ex: IF description THEN conflict).
- Treatment rule (ex: IF conflict THEN operation).
- Meta-rule (rule to decide on what rules to use, in what order).

Rules will not be further elaborated in this task report.

1.2.7 Conflicts

Conflicts arise due to constraint violations (geo / carto / process). However, not all violations represent conflicts (e.g., failing to close a polygon may not cause a conflict even though it is a topological violation). In general, conflicts:

- usually involve several map objects (but an object can conflict with itself);
- between objects do not occur in a database or at source scale (assumption);
- can stem from side-effects of generalisation/symbolisation operations; and
- have specifics and severities that can be described by measures.

Conflicts generally have a predictable lifecycle:

- initial : Conflicts of symbolism drawn to scale (before changing scale and generalising)
- intermediate : Conflicts created by too much data and/or generalisation operations
- final : No conflicts or minimal acceptable ones (after generalising)

Conflicts that might exist between objects in a database are typically of a logical nature, such as topological inconsistencies or duplicate identifiers. We assume that our database is "clean". In map generalisation, on the other hand, the vast majority of conflicts are physical, *spatial consequences of reducing map scale*. The greater the degree of scale change, the more cluttered with ink an ungeneralised map will be, and this signals the extents of conflicts. This is the main reason why it has proven so difficult to design generalisation software that works well across more than two or three degrees of scale reduction (at least for topographic maps).

1.2.8 Evaluation

Evaluation is the process of quantifying and qualifying properties of representations of phenomena, and usually involves three steps:

- *Measurement*: computation of raw parameters
- *Characterisation*: Interpretation of parameters e.g., classification, weighting
- *Evaluation*: Judgement or interpretation of a characterisation with respect to criteria given by constraints and/or goals.

Figure 1.2 illustrates these distinctions.

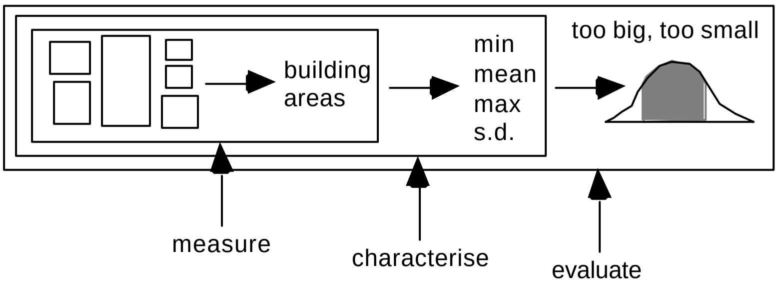


Fig. 1.2: Measurement leads to characterisation in order to evaluate phenomena

1.2.9 Measure / Measurement

A measure is a procedure for computing measurements, which are the basis for evaluating characteristics of phenomena and assessing the need for and the success of generalisation. Measurement can be understood at four conceptual levels:

Operational concept: Distance

Mathematical Concept: Hausdorff

Algorithm: Approximation of Hausdorff between two lines

Value (measurement): 3.2

Another Example:

Operational concept: Topological consistency

Mathematical Concept: Planar graph theory

Algorithm: node and chain cycling

Value (measurement): FALSE

Mathematical measures may be exact or approximate, according to the algorithm chosen to implement them. Results of measurements such as scalars or Booleans may only provide information about one aspect of an operational concept (example 1), or indicate an overall result without describing particular problems (example 2). Measurements may thus need to be compared or combined, and distributions of them may need to be looked at (which of course involves additional measures, i.e., statistical inference).

1.2.10 Properties / Character

A *property* is an intrinsic attribute or aspect of an object that may be the basis for constraints that bind it. Most of these are definitional or «built-in,» and thus are not optional such as tabular attributes might be (but some may never be formalised). Properties can be thought of as specifications of semantics of classes of phenomena. For example, a map series or a database might specify various properties, such as:

- *Geometric*: Most GIS objects possess geometries that may be simple or compound. At some level these are decomposed into point objects, line objects and polygon objects. Geometries for this project's purposes are assumed to be embedded in Cartesian 2-dimensional space.

- *Graphic*: Roads and railroads are symbolised on top of river symbols; Boundary line symbols overprint on all others and never conflict with them; Railroad tracks do not have sharp corners
- *Topological*: Polygons have positive areas, holes in polygons have negative areas; Polylines do not intersect except at endpoints; Boundary networks partition map space completely
- *Structural/Semantic*: Bridges are associated with crossings; dams, waterfalls and gauges are associated with rivers; A road has a certain type of sinuosity
- *Gestalt*: The general distribution of land use, settlements, roads, and rivers is rooted in the region's geomorphology, climate and culture. Human perception of such patterns in the landscape involve Gestalts («a form or configuration having properties that cannot be derived by the summation of its component parts»: Random House Webster's College Dictionary)

  Cultural: The look of a map is a product of the culture that made it and the culture being mapped. Thus both the character of phenomena and that of their portayal are culturally determined and not subject to universal standards of content or aesthetics. This implies that a generalisation system must reflect cultural biases in its constraints.

Taken together, properties form what is sometimes called the *character* of map objects and geographic phenomena. As feature character should be preserved when generalising, properties often serve as bases for constraints, but other constraints derive from *procedural knowledge* and *controls*, as the following chapter describes.

1.2.11 Treatment

We define a *treatment* as a group of operations (of any kind) performed on an object or on a set of objects to generalise them. Treatments can employ many different approaches (procedural modelling, OO modelling, agent modelling, knowledge-based systems (KBS), neural networks, decision trees, problem solving with state-transitions, etc.). Furthermore, solutions can be created in different ways (deductively, empirically, machine learning, by chance…). In this project, object modelling and agent modelling are the primary mechanisms, but other tools will also be brought into play.

2 Origins and Nature of Constraints to Map Generalisation

In order to understand the role of generalization constraints, it is essential to analyse how they interact with the other components of the overall process, which can be conceived of as having the following (partially ordered) components:

- Controls
- Constraints
- Strategies
- Tactics
- Assessment Tools
- Transformation Tools

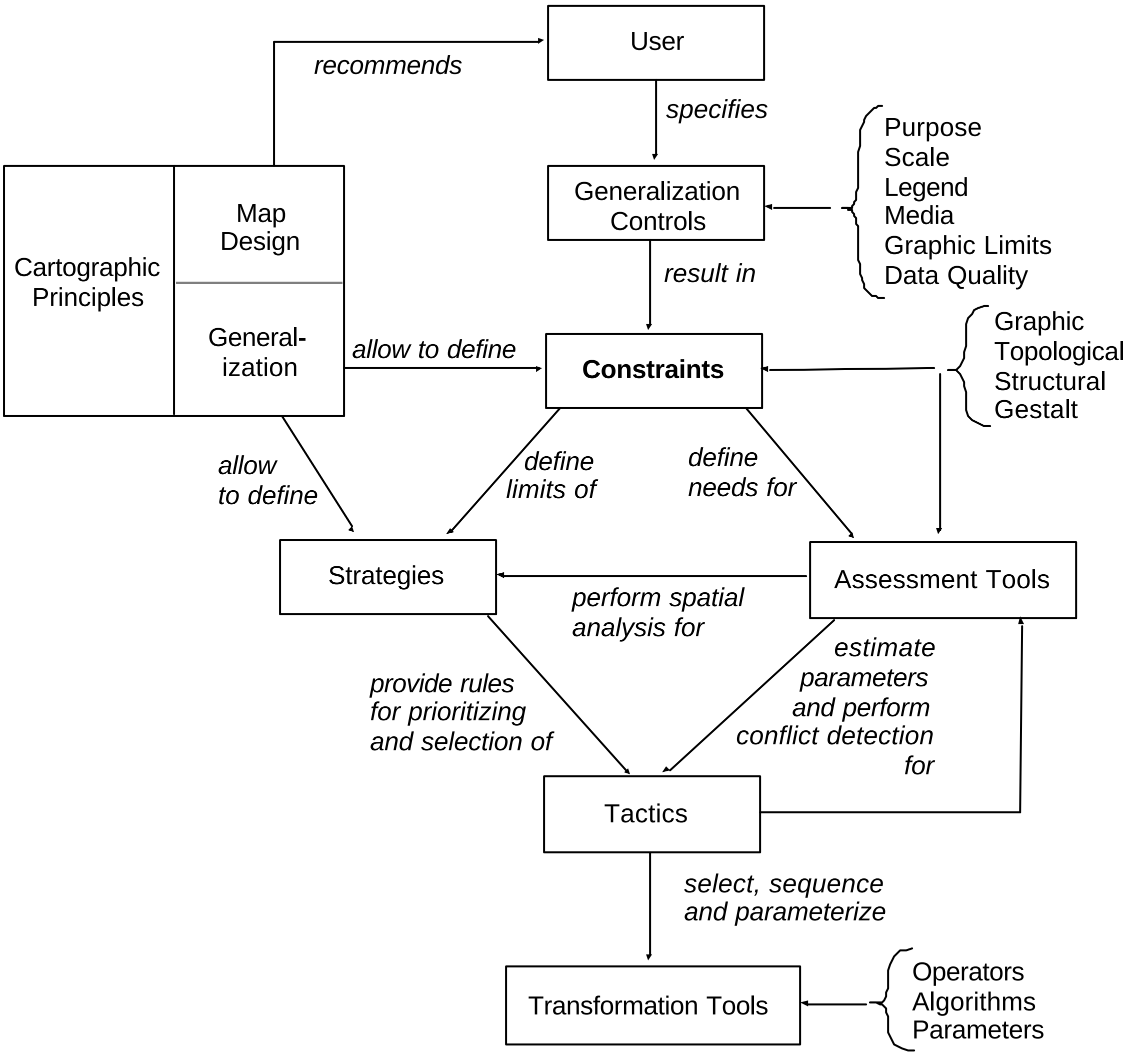


Fig. 2.1: Framework of constraints in map generalization

These elements are initially influenced by 'external factors', including a *user* (e.g., cartographer) of a cartographic database or map, who specifies overall design decisions (*controls*), which are in turn elaborated by design principles that embody the rules of generalization and good cartographic practice. Figure 2.1 shows how these various components interact and indicates the role of constraints. Note that this figure depicts the dependencies among components rather than an actual workflow. Although most workflows would involve a similar top-down process, they would give greater emphasis to sequencing, evaluation and reiteration of processing steps (Brazile 1998, Ruas and Plazanet 1996). The elements of this framework all have particular roles and ways in which they interact:

- Generalization *controls* specify the nature of a map to be produced and of the data and media to be used. They represent 'invariants' of the generalization process, the consequences of which are defined externally by designers of map production systems. Section 3 explains generalization controls in more detail.
- *Constraints* are derived from cartographic principles and define the criteria and conditions under which generalization must operate. There are different types of constraints (*graphic*, *topological*, *structural*, *Gestalt*; cf. sec. 2.2). Constraints may further be distinguished by their *scope* (cf. sec. 2.3). For a given map or database product, constraints prioritise and parameterise the generalisation controls (such as map purpose). Priority of constraints is rarely static, instead expressing contextual conditions, the mutability of feature classes and flexibility in applying procedures (cf. secs. 2.3, 2.4).
- *Strategies* define workflow at a global level, including establishing defaults, prioritising feature classes, sequencing operators and defining operational regions. They derive from constraints and from analysis of map data.
- *Tactics* control the application and sometimes the selection of appropriate *tools* in specific situations. They are more data-driven than strategies, and may be able to override default parameters set previously.
- *Assessment tools* are used for structure recognition, conflict detection and evaluating outputs from transformation tools. They operationalise constraints, and help to trigger generalization operations by analysing source data to identify spatial conflicts that may arise at target scale (as task reports C1 and C2 should describe), in order to determine if and where transformation tools have satisfied constraints.
- *Transformation tools* (operators, algorithms, parameters) are invoked when assessment tools detect conflicts within or between features, and are sequenced by strategies and tactics. Generalization *algorithms* transform map data in either the spatial or the attribute domains. *Operators* group these algorithms into functional classes (e.g. simplification, smoothing, aggregation, etc.). *Parameters* guide or modify the operation of algorithms at run time (refer to forthcoming reports for tasks D1 and D2).

2.1 Controls

Many – but not all – constraints in map generalisation in one way or another originate as elements of *controls to generalisation*. We identify several categories of controls that are important in this regard:

- *Map Purpose*. Traditionally this encompassed whether a map was topographic, navigational or thematic, and whether it was part of a series or an atlas. With GIS these categories have become both blurred and specialised, and more emphasis placed on building databases rather than just maps. We presume that map purpose largely dictates content and symbolism, especially in production environments that rely on formal map specifications. Map purpose defines what subset of database information will be used, also indicating feature class priorities, thus begins to define the operative generalisation constraints.
- *Map Scale*. Whether directly specified or derived from other factors, scale is the most constraining control, and responses to it involve the most decision-making. Scale-related specifications may at times dictate changes in symbolism in addition to triggering simplification, displacement, amalgamation and other transformations of map features.
- *Map Legend* (i.e. the symbol set). Symbology drives generalisation because it is map symbols that conflict rather than the features (data) they represent. Assessment tools must be aware of symbol dimensions and priorities in order to effectively detect and resolve spatial conflicts. Naturally, symbolism needs to be appropriate to the scale and purpose of the target map; this illustrates that controls can constrain one another.
- Output Medium. Physical limitations of the output medium must be respected. These include spatial resolution and colour/grayscale resolution, which in turn limit line weight, symbol size, type size and style, and the range of tones, colours and patterns. Modern hardcopy devices do not impose severe limitations, but interactive displays still do, especially for maps designed for display across networks, where display device characteristics may be highly variable and cannot be known in advance of generalisation.
- *Graphic Limits* (i.e., symbol coarseness). Minimum line weights, polygon and point symbol sizes can be specified, and usually will be larger than what are physically possible, depending on the map's purpose. In concert with map scale and output resolution, this determines the amount of detail that a map can contain.
- *Source Data Quality*. The accuracy of map data may affect the need for and behaviour of some operators (e.g., displacement), and its fineness and density of detail can affect others (e.g., simplification). GIS data

quality can vary by feature class and region; if this is documented (e.g., via metadata) its effects can be controlled for (Brazile 1998).

- *Topological Relations.* These are properties of adjacency, connectivity and containment implied by the configurations of nodes, edges and faces, and can be specified for individual features both within and across feature classes. We view them as controls because they are based on graph-theoretical invariants that must be satisfied. Line/polygon topology partitions space in ways that can be altered by generalisation transformations and still satisfy such invariants.

Controls imply certain constraints, although rarely in a one-to-one way. For example, *map scale* drives generalisation by multiplying feature densities and limit the perceptibility and legibility of symbols. Several controls may combine to define some constraints, such as legibility, which in addition to map scale is controlled by *output medium*, *graphic limits* and *data quality*. *Map purpose* also controls legibility in the sense of imposing certain standards and content requirements that may constrain generalisation choices; therefore it influences other controls, such as legends and graphic limits. Because a well-specified map will denote all feature classes to be used as well as indicate which are most or least important, what accuracies are required and possibly what attributes are symbolised, map purpose controls important aspects of the translation of data base objects to graphic objects.

2.2 Constraints

Constraints implement generalization controls. As such they must respond to the requirements for maintenance of perceptibility, logical consistency, and structural integrity specified by the controls. Constraints governing map generalization can be classified into the following categories:

- graphic
- topological
- structural (or semantic)
- Gestalt
- Procedural

The above categories slightly revise the earlier ones in Weibel (1997), reflecting the fact that the previous paper focused on a more specific generalization problem. The classes of constraints are described below. These are used in the ontology of individual constraints presented in chapters 3 and 4. Also see Weibel (1997) for a comprehensive description of constraints for the specific problem of simplifying and generalising polygonal maps (e.g., political boundaries, soil maps).

Graphic constraints arise from featrure and symbol geometry, and specify basic size and proximity (i.e., area and distance) properties and are mainly dictated by graphic limits and the shgapes and sizes of features. Examples for individual features include minimal size, minimal width, and minimal length of features. If multiple features are involved, graphic constraints define minimal separability and help to enforce proximity relations. Bader and Weibel (1997) describe algorithms capable of observing size and proximity constraints for polygonal map generalization.

Topological constraints ensure that basic topological relationships (connectivity, adjacency, containment) between features are maintained. For individual features self-intersecting lines and polygon boundaries should be avoided. When multiple features are involved, spatial transformations should not alter the topological relationships of the remaining features, even when these only indirectly represent the original features. For instance, when links are deleted from a road network, the connectivity of the resulting ones should be maintained, and objects of other feature classes alongside roads should not change sides of roads or overlap with them. This is illustrated below.

Structural constraints define criteria that describe both *spatial* and *semantic structure* and inter-dependencies. Spatial structure on the level of individual features relates to shape (i.e., internal structure of features) and its preservation (sinuosity of lines, convexity/concavity of areas). At this level of complexity, structure is difficult to distinguish from simple geometry and topology. For groups of features (cf. Regnauld 1997), spatial structural constraints define the preservation of alignment (e.g., buildings parallel to road, buildings arranged in rows); feature clusters (e.g., villages or ponds); size relationships between objects of one or several feature classes; feature density (features per unit area); spatial distribution of point or small area features (e.g., uniform, clustered); and directional arrangement (e.g., structure of a river network – dendritic, radial, etc.). Semantic and structural constraints relate to the maintenance of logical class relationships in reclassifications of feature classes or attribute categories, or to the preservation of the logical context (e.g., a building that falls into a lake is usually out of context; likewise, rivers on hilltops make no sense).

In contrast to graphic and topological constraints, structural constraints describe higher order concepts and involve less explicit relationships. For instance, when generalising a river network, topological constraints dictate that connectivity between individual stream segments be maintained and interior links should not be deleted. Structural constraints apply to transformations of entire networks. To this end, they may make use of topological measures such as the *b* index, node accessibility or topological ordering schemes (Strahler, Horton, Shreve), yet can also use non-topological metrics, such as volume rate of flow or impedance. In any case, an *ensemble* of relations is examined.

It is obvious, then, that structural constraints require enhanced data models to represent both spatial structure (e.g., triangulations, Voronoi, hierarchical tessellations, etc.) and semantic structure (e.g., by enriching attributes and metadata, and by using object-oriented modelling with multiple inheritance), as spatial and logical structure cannot always be inferred by computational techniques. Commercial GIS's are only just beginning to offer these degrees of modelling capability.

Gestalt constraints relate to aesthetics and complex perceptual aspects. They can largely be equated with those principles of map generalization that necessitate cartographic license and intuitive design decisions on behalf of the cartographer. Examples include the preservation of line character through line caricature, maintenance of a characteristic distribution when new objects are formed through typification and aggregation, enforcement of visual balance and an even degree of generalization across the map, or the interplay of features belonging to different classes. While most such constraints are graphic in nature, they are enforced more by strategies than through tactical decisions.

Some constraints are harder to define than others. Due to their aesthetic, intuitive and global nature, Gestalt constraints obviously are most resistant to formal specification. Topological constraints are easiest to define, and often yield Boolean results. Graphic or geometric constraints can usually be formally specified, but they frequently involve thresholds which depend on generalization controls (e.g., legend or graphic limits). Among structural constraints, those relating to individual features (e.g., line sinuosity) are easier to specify and implement than those which affect groups of features.

The above classification is useful for our purposes, but other schemes would have been equally valid, of course. The main objective is specifying all necessary constraints that define a particular problem and organising them as consistently as possible. For example, in our framework constraints relating to 'shape' are called *structural* constraints, while other authors may classify them as *graphic* or *geometric*. ‘Shape’ can express internal structure at the level of individual features, and may also be exhibited by groups of features, such as by the hull of a cluster of buildings. Clearly, ‘shape’ is an ill-defined concept which is not easily expressed by scalar geometric measures (e.g., ellipsoidal eccentricity of the earth), even for characterising individual objects. More complex (and useful) measures, however, tend to become objects in their own right (such as convex hulls or harmonic equations describing geoids).

2.3 Scope of Constraints

Constraints can be characterised in various ways, such as local vs. global, or hard vs. soft. Instead of using such dichotomies, we adopt the following categories:

- Spatial Scope: Neighbourhoods within which constraints are in effect
- Contextual Scope: Conditions under which constraints may be enforced

In terms of spatial scope, we can distinguish the following cases (Weibel 1977):

1. Within a feature; e.g., regimes of differing line complexity
2. Between features; e.g., clusters of buildings, groups of islands
3. Between feature classes; e.g., roads aligned with rivers, farms

4. Within regions; types of partitionings that may be used include:
   - Blocks enclosed by streets and roads
   - Buffers around high priority features
   - Delaunay triangulations
   - Voronoi regions
   - Administrative units
   - Arbitrary grids

The concept of region used here is a flexible one; it can denote a natural grouping of map features (such as a village), a partitioning based on a feature class (typically transport routes, but several classes may participate in forming partitions), or arbitrary/abstract spatial units (such as a land survey grid or a tessellation of map space). The latter type may have no semantic relation to map data, but can still be useful in designing a divide-and-conquer strategy (Ruas 1995). In general, regions can help to limit search space for contextual generalization operations (e.g. displacement or aggregation), whether one feature class or several are involved.

The contextual scope of constraints pertains to the circumstances under which they apply, and the degree to which they may be enforced. For a given constraint, rules may be defined that specify:

1. what its priority is relative to other constraints
2. under what particular circumstances, if any, it can be relaxed
3. what the relaxation involves parametrically and algorithmically

For instance, *feature density* (as measured by ink-to-paper area ratios in local neighborhoods) is a structural constraint, but is difficult to maintain. As symbols take up more space with generalization, it is bound to grow, even with feature elimination. However, it should stay within certain limits to avoid visual clutter or unnatural size relations. The balance or distribution of feature densities across a map is, on the other hand, a Gestalt constraint, as gross changes in or redefinition of patterns of density across a map are perceived as different, inappropriately.

To illustrate priority, avoiding imperceptible detail may be a stronger constraint than maintaining a faithful representation, at least for certain feature classes. This might involve, for example, choosing one simplification operator over another because it removes small crenulations better, even though this increases shape distortion. Likewise, symbology-based constraints (e.g., line weight) normally of high precedence, may in fact be relaxed under certain conditions or at critical scales (as when dual carriageway symbols are replaced by narrower ones or when parcel boundaries are omitted).

Certain feature properties, especially line topology, although highly constrained can still be generalised under certain conditions. For instance, road connectivity can be modified when pruning cul-de-sacs or replacing highway interchanges with simpler connections, removing ramps and resymbolizing overpasses. In general, however, as connectivity can easily be disrupted in generalising networks, it is necessary to specify priorities and other measures for evaluating pruning effects to ensure that deleting links does not adversely distort the shape of networks.

Enforcing constraints can cause conflicts (particularly between feature classes) which need to be foreseen and handled properly. For example, as figure 2.2 illustrates, minimum size or separation criteria may indicate that a group of small lakes (a) be coalesced (b); however, this could cause the resulting lake feature to intersect a road running through the area, violating another constraint. Different constraint priorities can lead to different solutions: lakes below the minimum area threshold can be eliminated; the lakes can be fused and the road routed around the resulting features; lakes on either side of the road can be fused into separate features, avoiding the road, the solution shown in (c). The strategy may be controlled by map purpose, from which the relative mutability of hydrographic and transportation features can be derived.

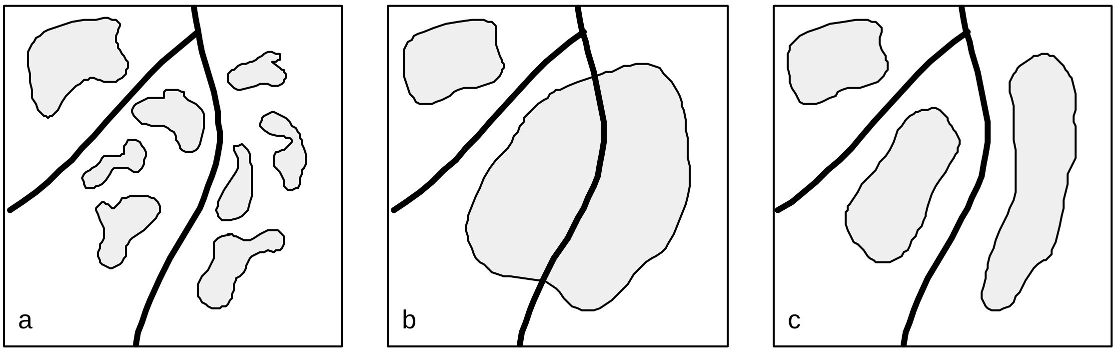


Fig. 2.2: Resolution of two conflicting constraints

In this example, the spatial and contextual scopes of constraints were both salient. The initial violation was of a spatial nature, but the choice of a solution needed to respect context (e.g., that roads ought to be displaced less than waterbodies, and that lakes can be aggregated under certain conditions.

2.4 Constraints and Generalization Tools

The maintenance of constraints are eventually enforced by generalization tools. Assessment tools and generalization tools are applied in a feedback loop and controlled by strategies and tactics: Assessment tools are used for conflict detection (identifying whether a constraint is violated and to what degree); the execution of transformation tools is triggered when a conflict is detected; and the success of these transformations is subsequently evaluated again by assessment tools. As mentioned above, closer examination of strategies and tactics is beyond the scope of this paper (but see Brazile 1998). The influence of constraints and assessment and transformation tools, however, will be further analyzed here.

Figure 2.1 describes the relationships between the various components of our framework at *run time*. However, constraints are also useful at *design time*, that is, for the design and development of generalization tools. The difference is that at run time, constraints are parameterised with a specific set of user-defined generalization controls, while at design time, they merely act as design specifications. To use a simple example, at run time a linear constraint for the sides of area objects may be specified as 0.35 mm, while at design time the constraint simply declares that there must be a user-definable parameter for minimal width and a mechanism to enforce it. To design a suitable mechanism that works properly at run time, a method is needed for conflict detection (i.e., violation of that particular constraint) as well as another one to resolve the conflict by spatial and/or attribute transformations. Note that assessment tools used to evaluate results tend to be identical to those needed for conflict detection.

We believe that careful specification of constraints leads to the development of better algorithms as they can be designed to respond to explicit requirements rather than as isolated tools (such as many existing line simplification algorithms). The (graphic) size and proximity constraints specified in Weibel (1997) to polygonal map generalization, made it possible to subsequently develop algorithms that went beyond simple non-contextual line simplification methods (Bader and Weibel 1997) by considering structural characteristics specific to polygonal subdivisions. For assessment tools, minimal size was implemented by simple area calculation, and minimal width was translated to a tool based on a 'rolling ball' principle which detects critical regions by identifying self-intersections of buffers constructed around polygon boundaries. Overlapping areas were directly used in several transformation tools for enlargement of narrow polygons, aggregation of disjoint polygons and displacement of polygons, making use of triangulated data structures. The tool kit also included an algorithm to contract polygons which are too small or otherwise insignificant to their skeletons, where this would be appropriate (e.g., to turn lakes into streams).

Linear – representing linear features (rivers, roads, railways, utilities) or outlines of areal features (boundaries, shorelines)

- *Areal* – representing areal features such as lakes, buildings, districts, or land use by polygons

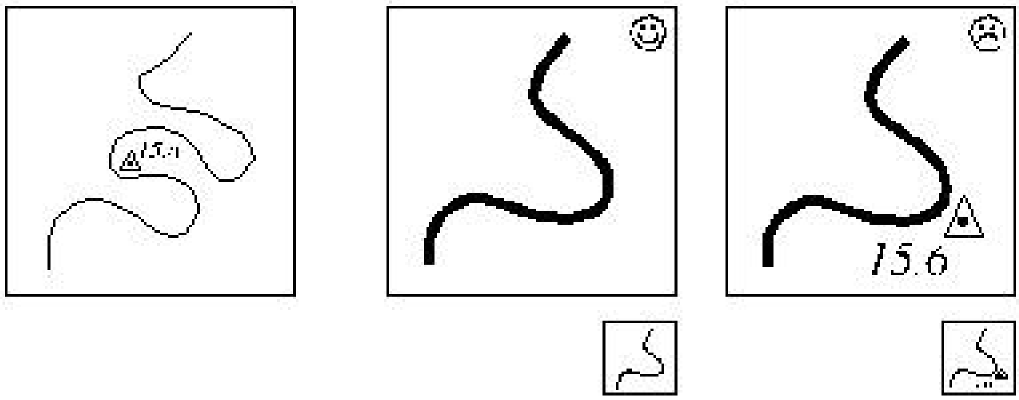
The most important cartographic constraints are listed below using these categories. Each constraint has other characteristics (such as whether it preserves properties or avoids problems) which are summarised in tables 3.1, 3.2 and 3.3, that also include examples of measures and algorithms to detect and eliminate constraint violations. Figures accompanying constraint descriptions serve to illustrate each one, usually showing a source map (left), which is then generalised incorrectly (ignoring the constraint being illustrated), and then correctly (observing the constraint). This should help readers to understand differences between constraints, some of which may at first appear to be rather similar (and in many cases are closely related).

Following the exposition of the nature of symbolism constraints, some parameters for «graphic limits» (minimal sizes and separations)are provided, based on guidelines from the Swiss Society of Cartography. Different map producers may prefer different values for certain limits, but in any case these are critical controls that drive many generalisation operators, and must always be specified.

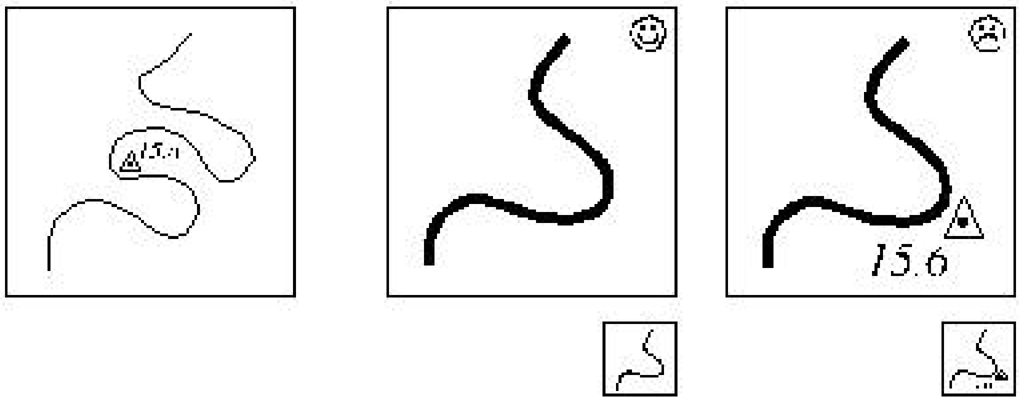
Likewise, positional accuracy limits must be specified for all features on a map, or separately for each feature class (although some mapping agencies only specify them for readily identifiable locations, rather than for all features). These limits will be in ground units and therefore must be translated to display scale. They indicate how much distortion and displacement of features is permissible, and thus serve as a bounding constraint for many map generalisation operations.

3.1 Iconic Symbols

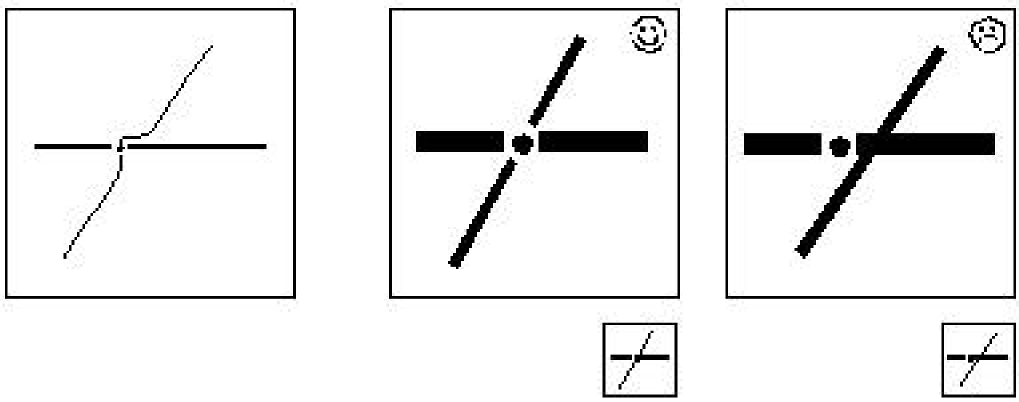
Feature types which may be portrayed as iconic symbols include geodetic control points, buildings of interest, landmarks, or transportation features (e.g. motorway interchanges), many of which have polygon geometry at source scale. At small target scales, these features can no longer be displayed to scale and are therefore 'iconised'. (collapsed to icons). The principle constraints for iconic representation are listed below, each accompanied by a thumbnail sketch that illustrates it:



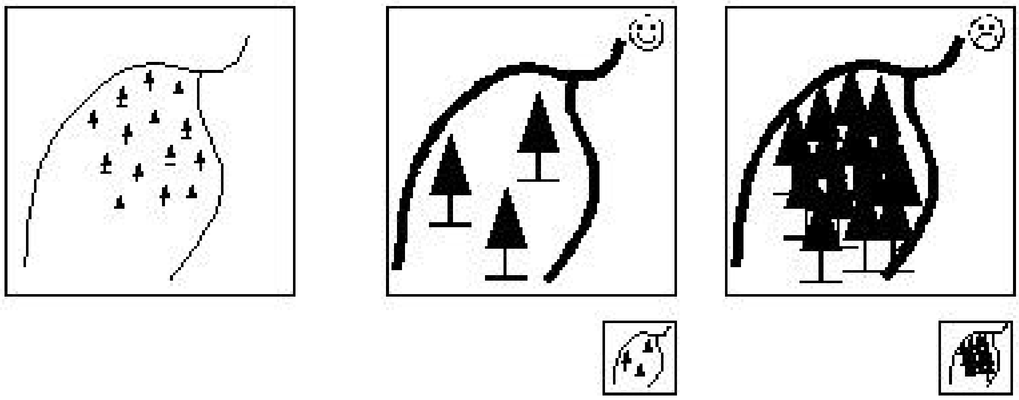
I1. *Geodetic accuracy* geodetic control points may be omitted but should not be displaced



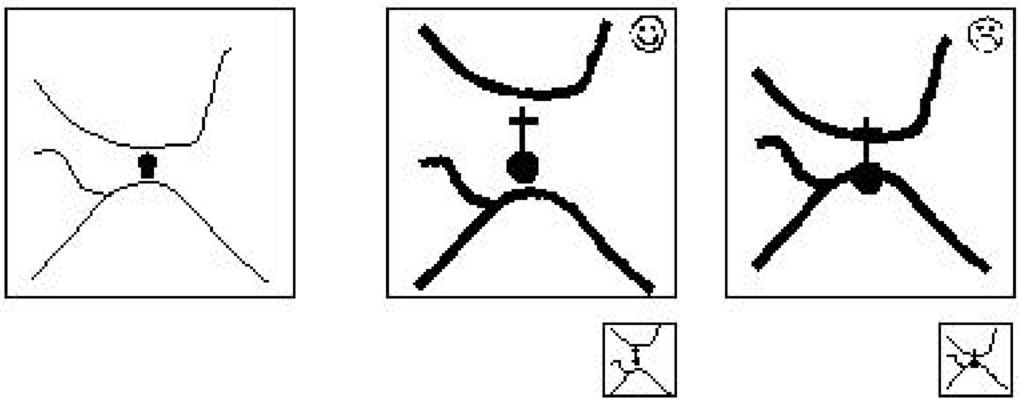
I2. *Positional fidelity* symbols should be as close as possible to features they depict



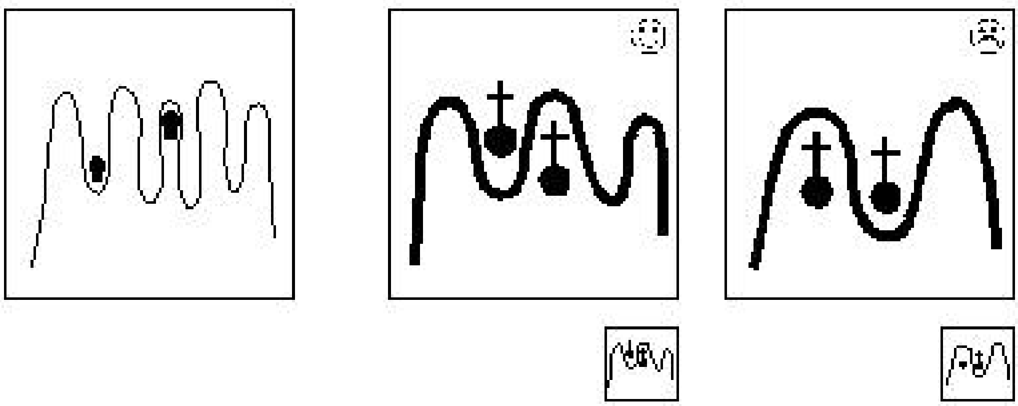
I3. *Shared positions* if iconic symbols share positions with other features (e.g., a junction symbol and a node of a road network), the equivalence relation should be maintained



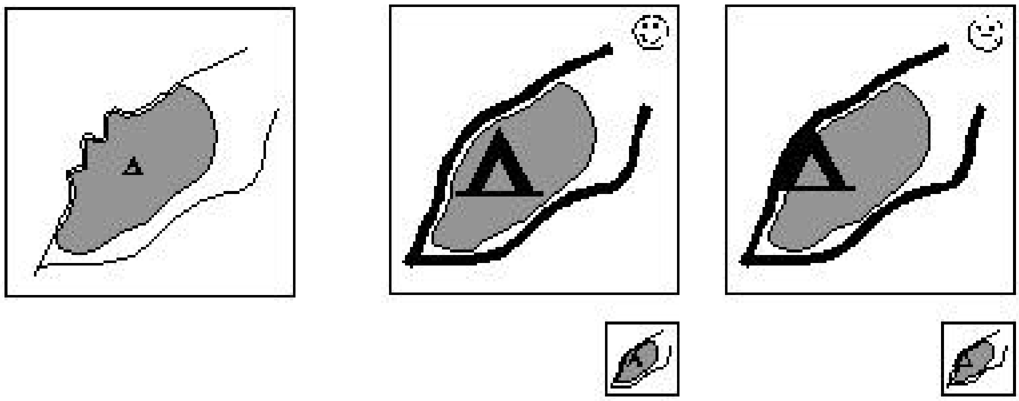
I4. *Separability* prevent iconic symbols from interfering with each other (overprinting)



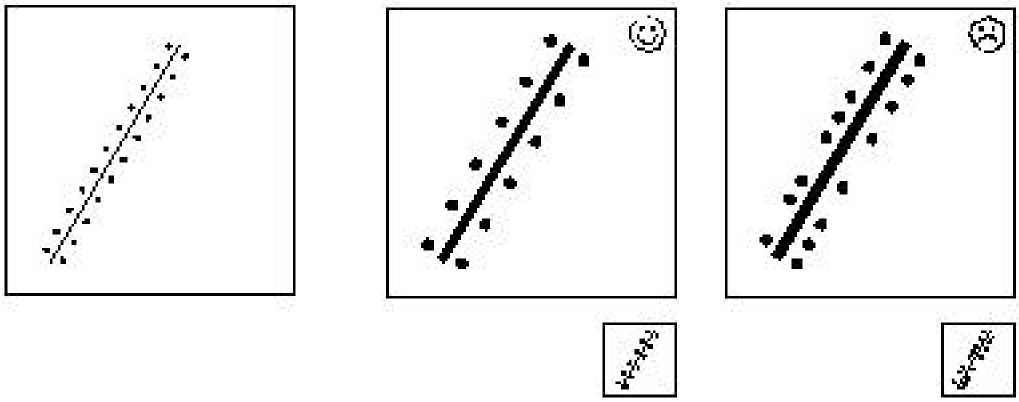
I5. *Masking avoidance* avoid overprinting of line symbols or small area features (e.g., buildings) unless it is an explicitly represented logical relation



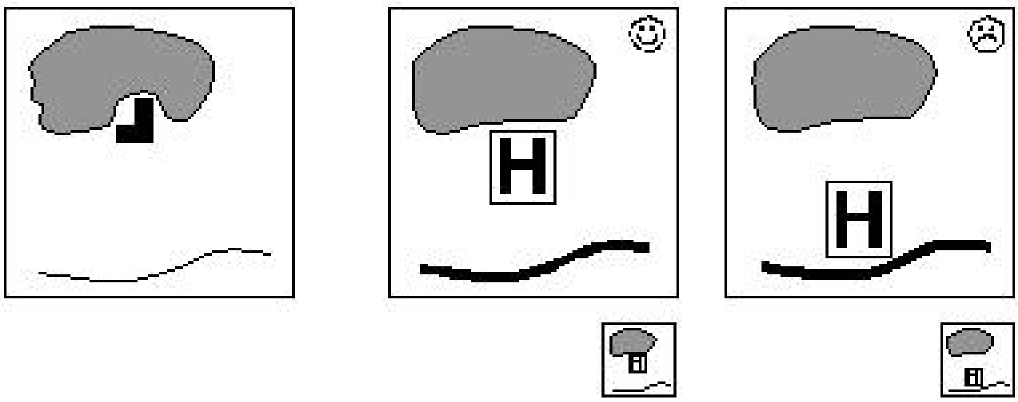
I6. *Local topology* keep iconic symbols on the same side of their neighboring features



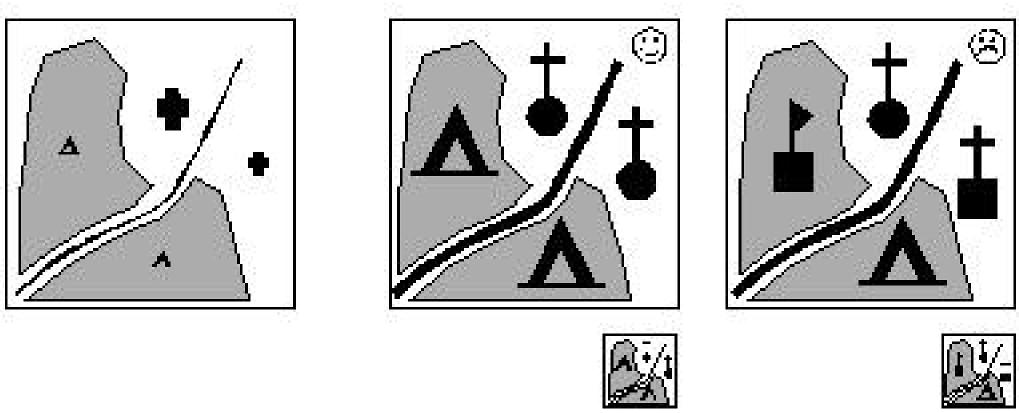
I7. *Displacement* displace non-geodetic icons when their neighboring features move



I8. *Alignment* preserve particular alignments and arrangements of point symbols



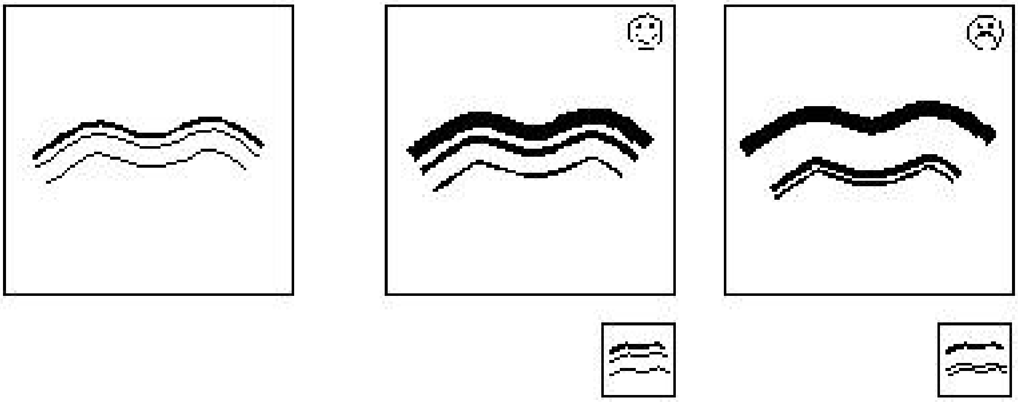
I9. *Proximity* maintain the (relative) proximity relationships to neighboring point, line and area symbols



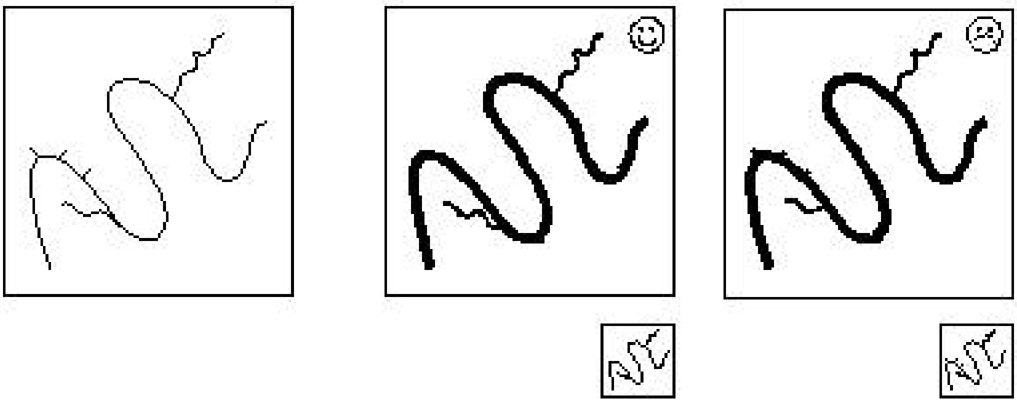
I10. *Equal treatment* ensure equal treatment (selection, displacement, etc.) across all partitions of the map

3.2 Linear Symbols

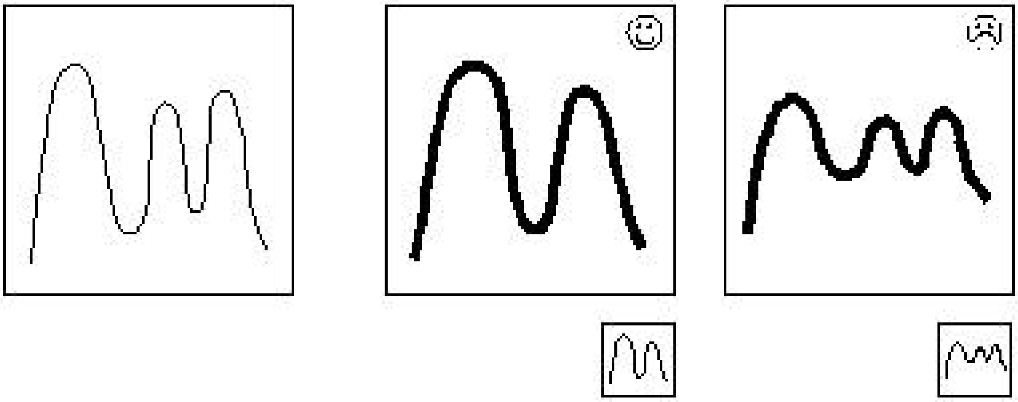
Linear symbols are used to display not only linear feature types such as rivers, streams, roads, trails and railroads, but also polygon boundaries (when they are drawn). Thus, they share some constraints with areal symbols. They share the common requirement of positional fidelity with iconic symbols. However, such shared constraints tend to specialise to various degrees.



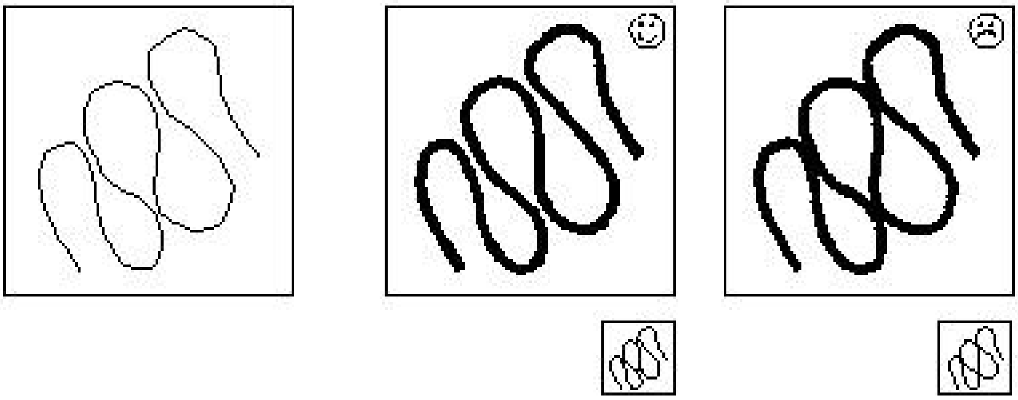
L1. *Positional fidelity* linear features (or portions thereof) should be displaced within class-specific limits



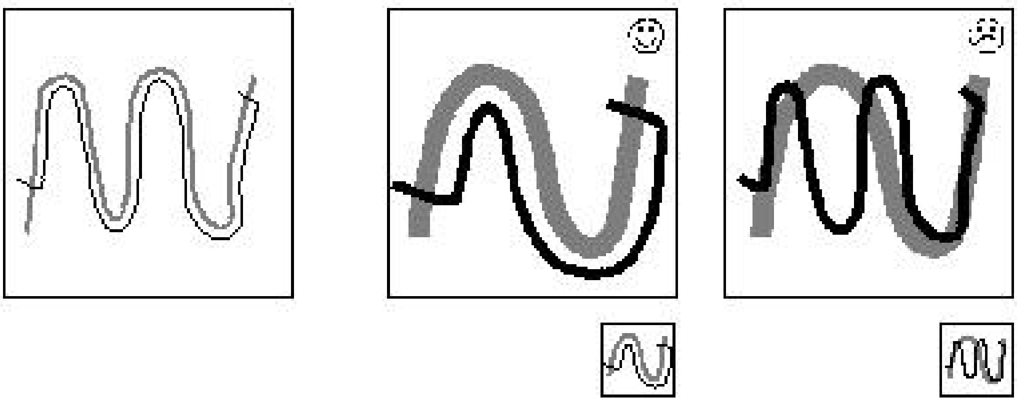
L2. *Perceptibility* remove imperceptible line segments (which are too short) and crenulations (which coalesce); or alternatively exaggerate them. Make sure that locations of junctions between lines remain distinct



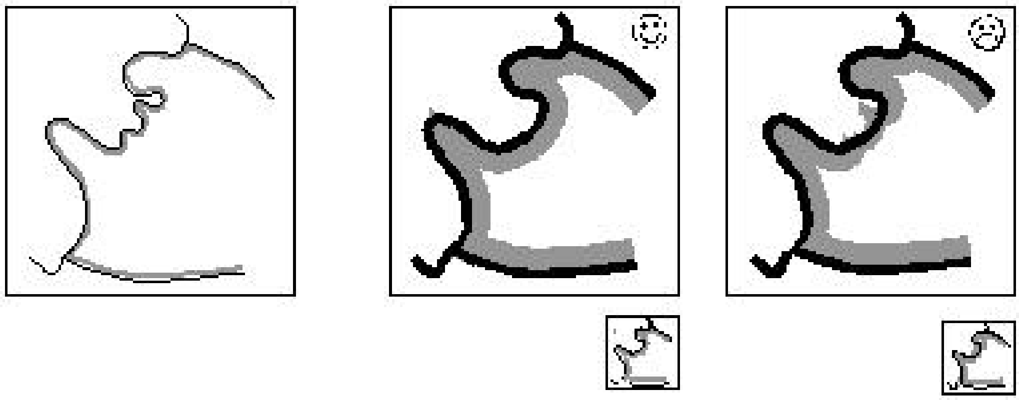
L3. *Shape distortion* avoid gross changes in sinuosity and shape (i.e., minimise shape distortion), unless required by typification or extreme scale change



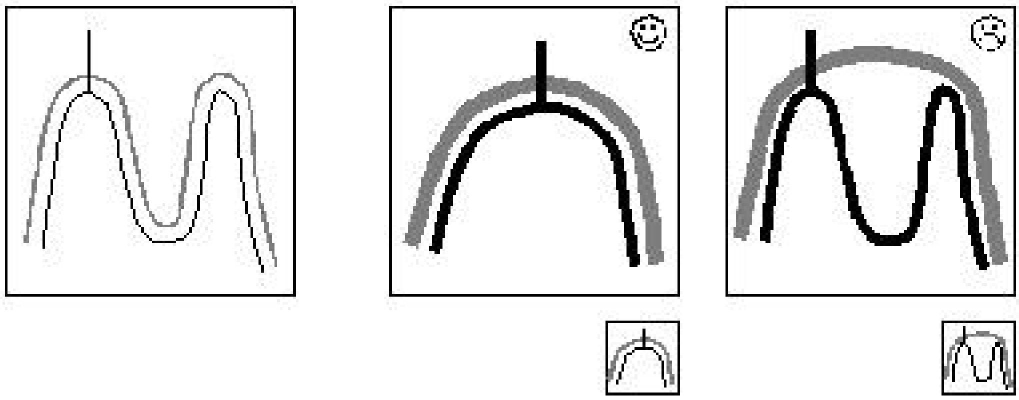
L4. *Self-intersection* linear features must not touch or cross themselves (except at end nodes)



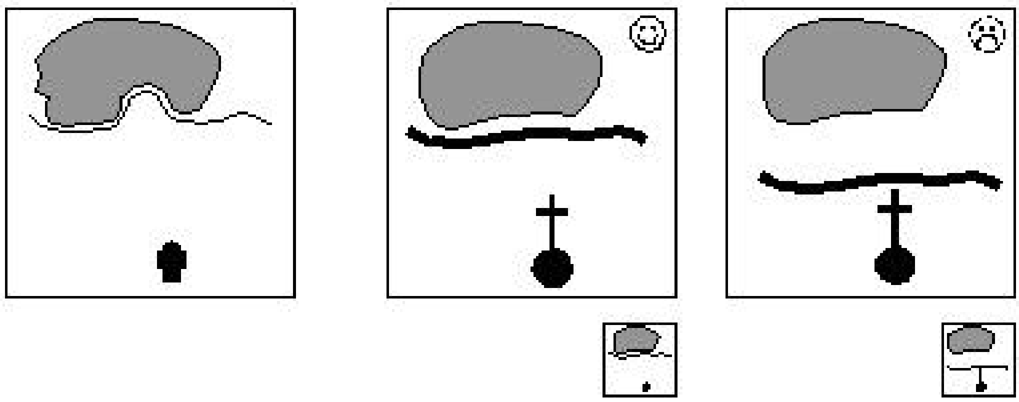
L5. *Other-intersection* linear features must not intersect other ones if they did not do so previously



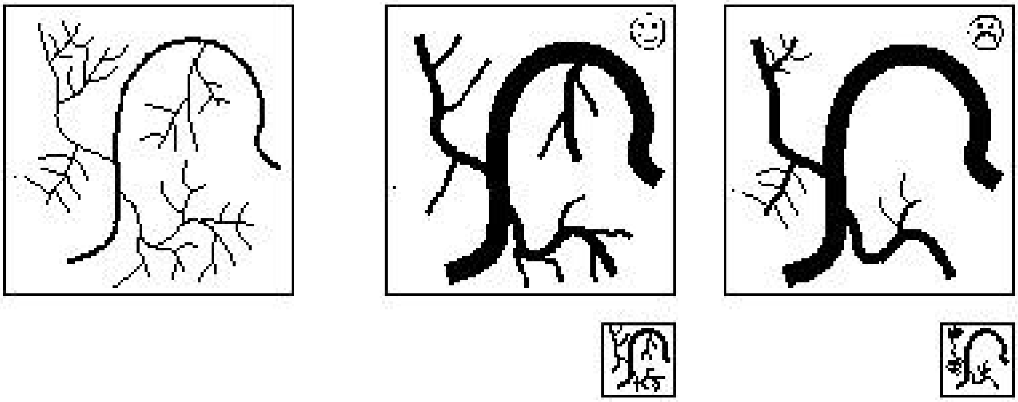
L6. *Shared lines* preserve shared line primitives with other FCs (e.g., where a river and a political boundary share a common geometry)



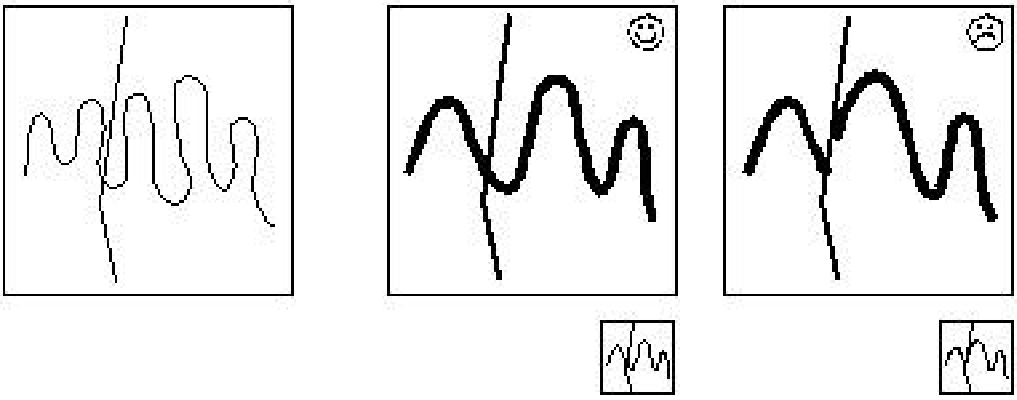
L7. *Alignment* preserve alignments between retained linear features



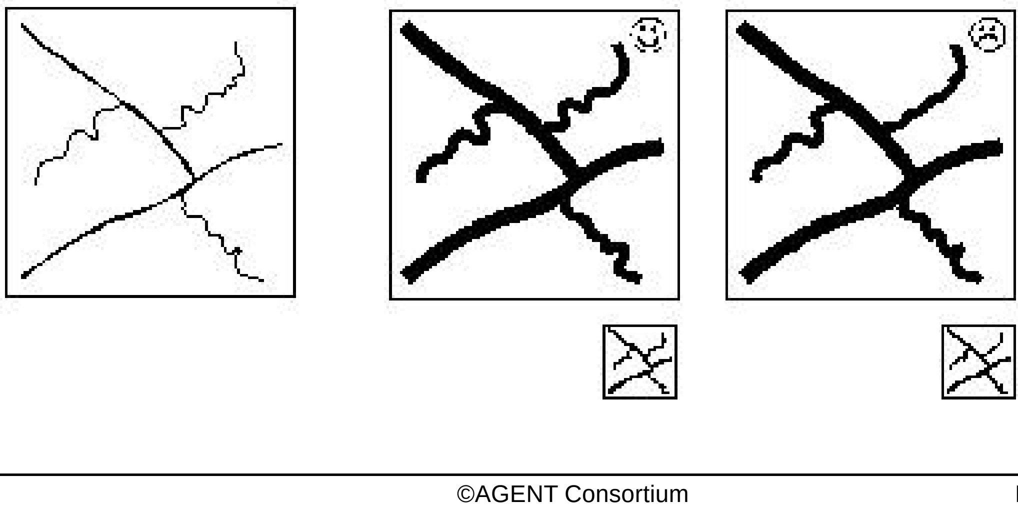
L8. *Proximity* maintain relative distances between linear and other features



L9. *Elimination order* eliminate linear elements according to their importance with respect to map purpose



L10. *Partitioning* maintain 1st and 2nd order continuity of lines and area outlines across partition boundaries

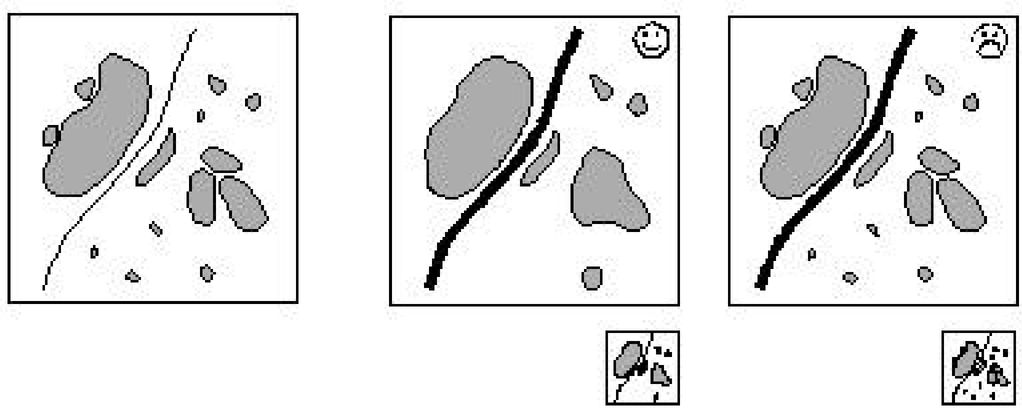


L11.	*Equal treatment*	ensure equal treatment (selection, displacement, etc.) across all partitions of the map

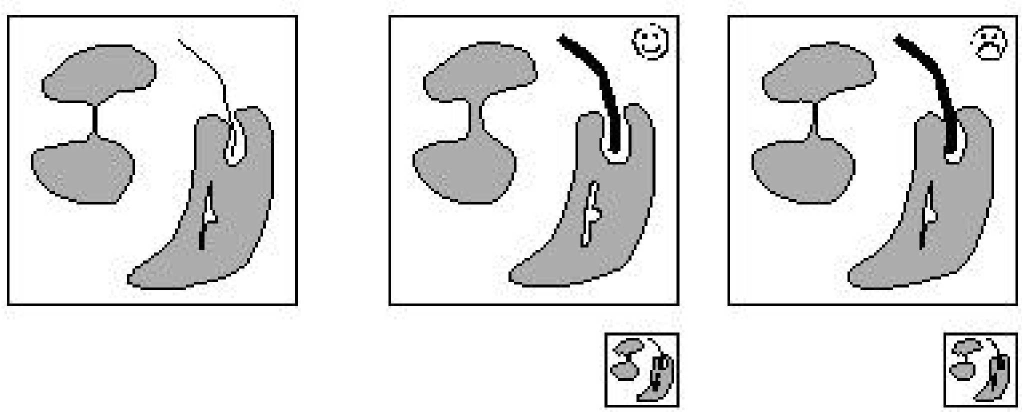
3.3 Areal Symbols

Polygonal areas can be rendered with or without outlines and filled with collars and patterns or left empty. They may serve as in foreground or background graphics. They may be disjoint or tile together, and be connected to linear symbols or not. Despite all these variations, certain constraints generally apply:

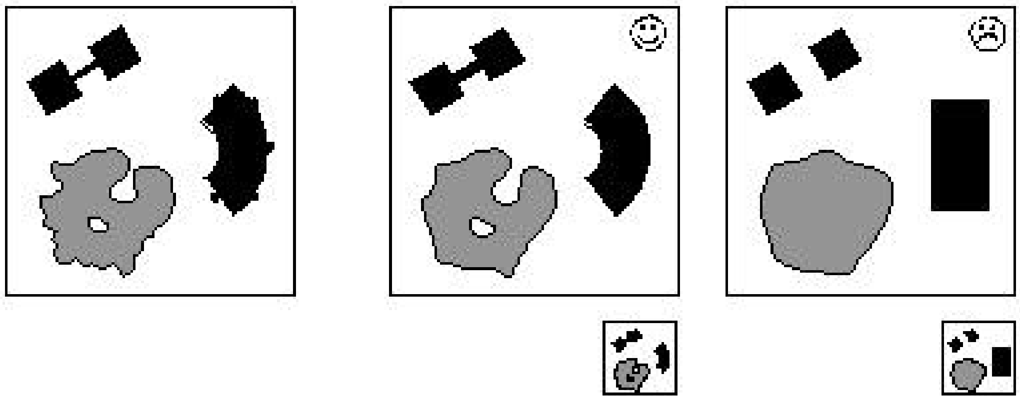
A0. *Line constraints* All line constraints are inherited when outlines of polygons are drawn.



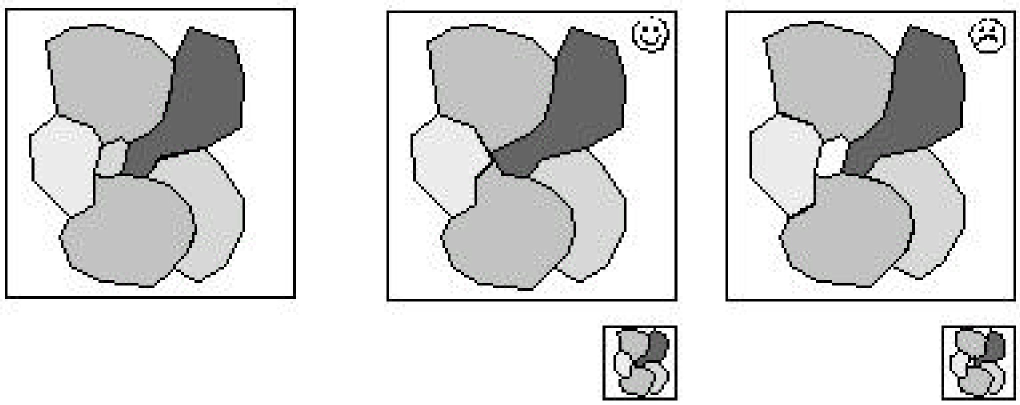
A1. *Minimum size* if a polygon would be too small, delete, enlarge or merge it



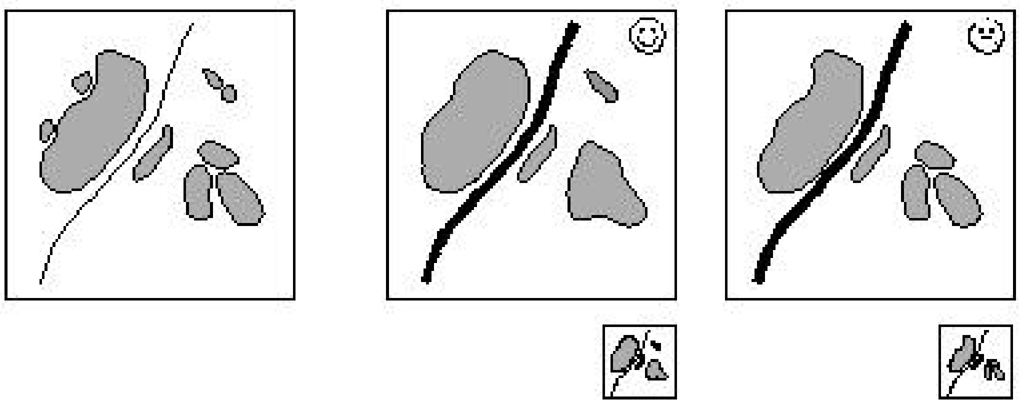
A2. *Minimum width* if part of a polygon would be too narrow, enlarge it



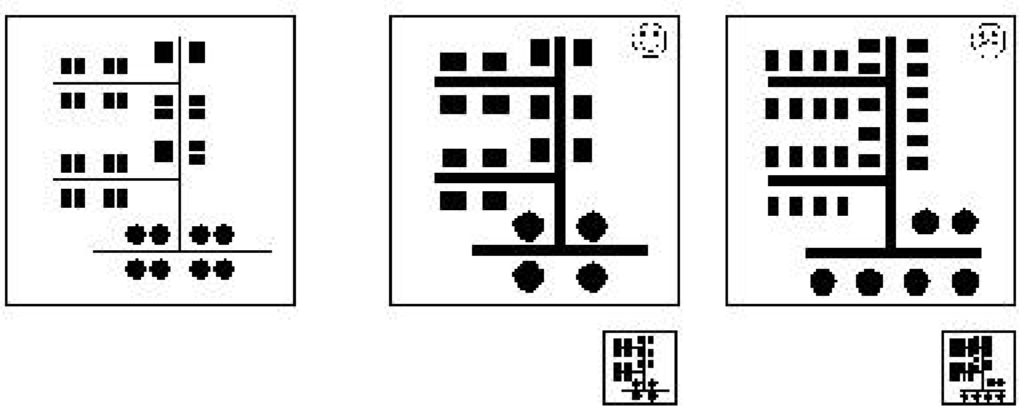
A3. *Shape distortion* — maintain the particular shape and possibly topology of small area features (e.g. buildings)



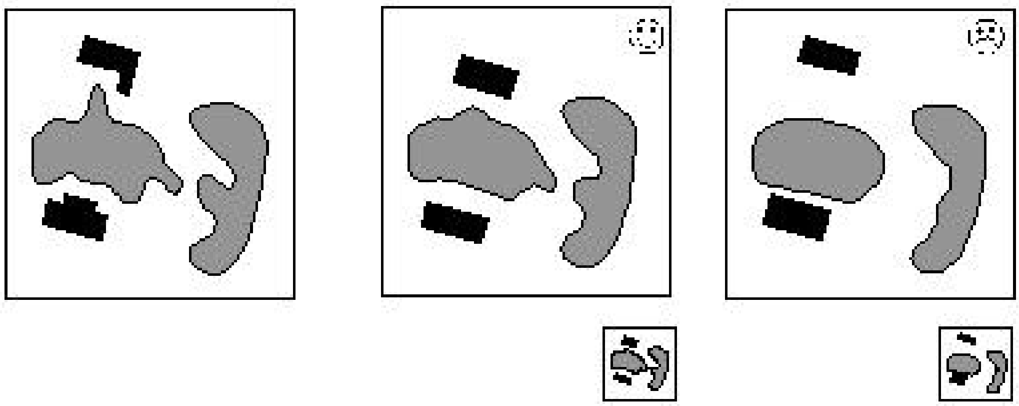
A4. *Collapsability* — in planar subdivisions, elimination should collapse polygons to lines that meet at central points, and the area of the eliminated polygon should be distributed among the neighboring polygons



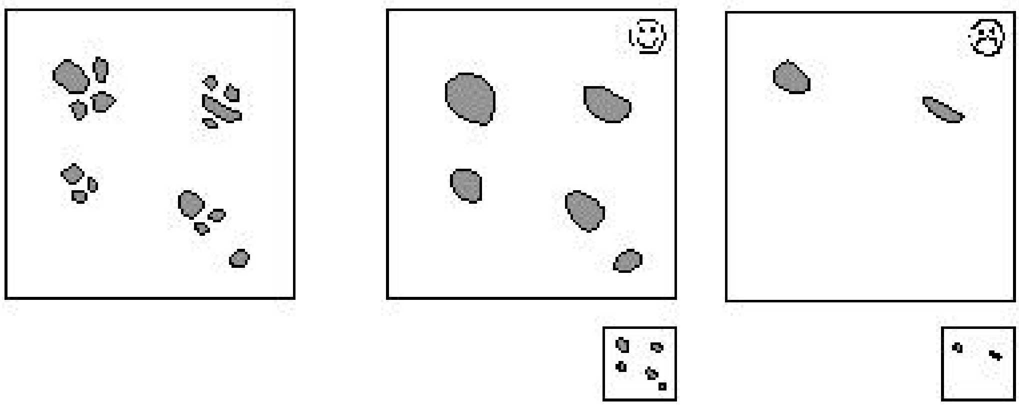
A5. *Aggregability* — disjoint areal features can be eliminated and/or aggregated (if close and similar enough)



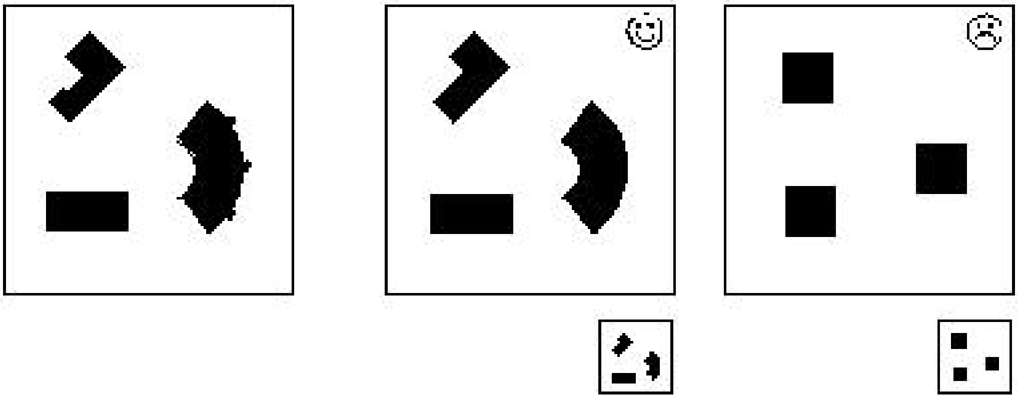
A6. *Alignment* — preserve alignment and pattern within groups of disjoint areal symbols (e.g. buildings) when they are aggregated or typified



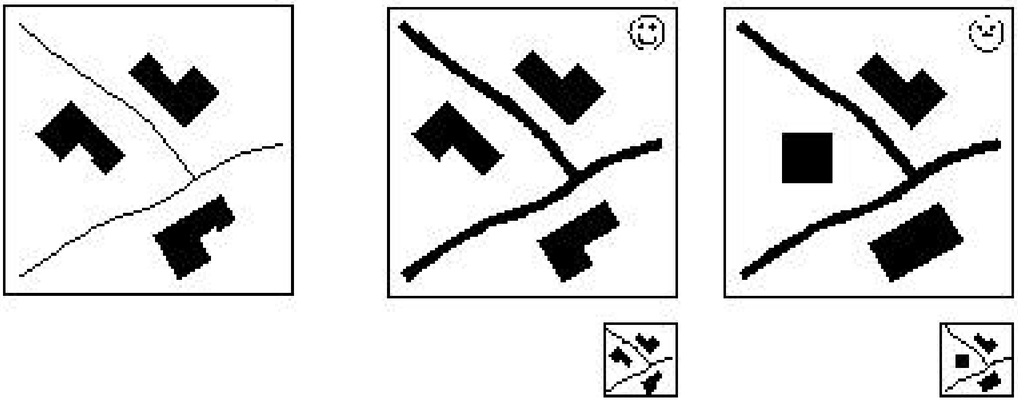
A7. *Proximity* — maintain relative distances between disjoint area features



A8. *Area distribution* — the global distribution of polygon area values should change as little as possible as the result of generalisation



A9. *Visual balance* Avoid gross changes in shape, unless required by typification or extreme scale change



A10. *Equal treatment* ensure equal treatment (selection, displacement, etc.) across all partitions of the map

Summary of Symbol Constraints

Table 3.1: Constraints on ICONIC SYMBOLS

NUM	CONSTRAINT NAME	TYPE	I/E	POSSIBLE MEASURES	ACTIONS TO ENFORCE
I1	Geodetic accuracy	Gx	e	epsilon circles	give priority to control points
I2	Positional fidelity	Gx	e	distance from "label points"	eliminate icons if too dense
I3	Shared positions	GTa	e	geometric matching	keep track of "peer features"
I4	Separability	Gx	i	compare bounding boxes	displacement, elimination
I5	Masking avoidance	Ga	i	buffers and bounding boxes	displacement, elimination
I6	Local topology	Tl	e	point-in-polygon, triangulation	displace together
I7	Displacement	Gx	e	buffering, triangulation	displace together
I8	Alignment	Gx	e	triangulation with angle computation	constrain angular changes
I9	Proximity	Ga	e	distances to neighbors	constrain distance changes
I10	Equal treatment	Gžx	i/e	histograms, densities, avg proximities	consistent application of constraints

Table 3.2: Constraints on LINEAR FEATURE SYMBOLS (generic polylines)

NUM	CONSTRAINT NAME	TYPE	I/E	POSSIBLE MEASURES	ACTIONS TO ENFORCE
L1	Positional Fidelity	Gc	e	vector/area displacement	derive and keep displacement limits
L2	Perceptibility	Gc	i	segment size statistics	simplification, weeding
L3	Shape distortion	Gc	e	total angularity, area displacement	maintain characteristic/critical points
L4	Self-intersection	Tc	i	planarity tests with epsilons	"smart" line simplification
L5	Other-Intersection	Tc	e	planarity tests with epsilons	iterating "smart" line simplification
L6	Shared lines	GSTcp	e	geometric matching of objects	minimize duplicated lines
L7	Alignment	Sc	e	constrained triangulations	limit "elasticity" of parallel lines
L8	Proximity	Ga	e	distances to neighbors	constrain distance changes
L9	Elimination order	STc	e	label chains w degree of importance	prune according to attributes
L10	Partitioning	GSc	e	identify bisected features	maintain connectivity & curvature
L11	Equal treatment	Gžc	i/e	histograms, densities, dimensions	consistent application of constraints

Table 3.3: Constraints on AREA FEATURE SYMBOLS (generic polygons)

NUM	CONSTRAINT NAME	TYPE	I/E	POSSIBLE MEASURES	ACTIONS TO ENFORCE
A1	Minimum size	Gp	i	area at scale	dilate, eliminate, collapse, typify
A2	Minimum width	Gp	i	buffers, triangulation, Voronoi	dilate, eliminate, collapse, typify
A3	Shape distortion	Gc	e	total angularity, vector displacement	segmentation, simplification
A4	Collapsability	Gp	i/e	area, shape, importance	polygon collapse
A5	Aggregability	Ga	e	avg distances, importance	local amalgamation
A6	Arrangement	Sp	e	avg distances, directions	displacement, typefication
A7	Proximity	Sp	e	distances to neighbors	constrained displacement
A8	Area distribution	žp	e	histograms, entropy	consistent application of constraints
A9	Visual balance	žp	e	size histograms, densities, entropy	consistent application of constraints
A10	Equal treatment	žcp	e	histograms, densities, avg proximities	consistent application of constraints

Legend:

G = graphic; S = structural; T = topological; = Gestalt

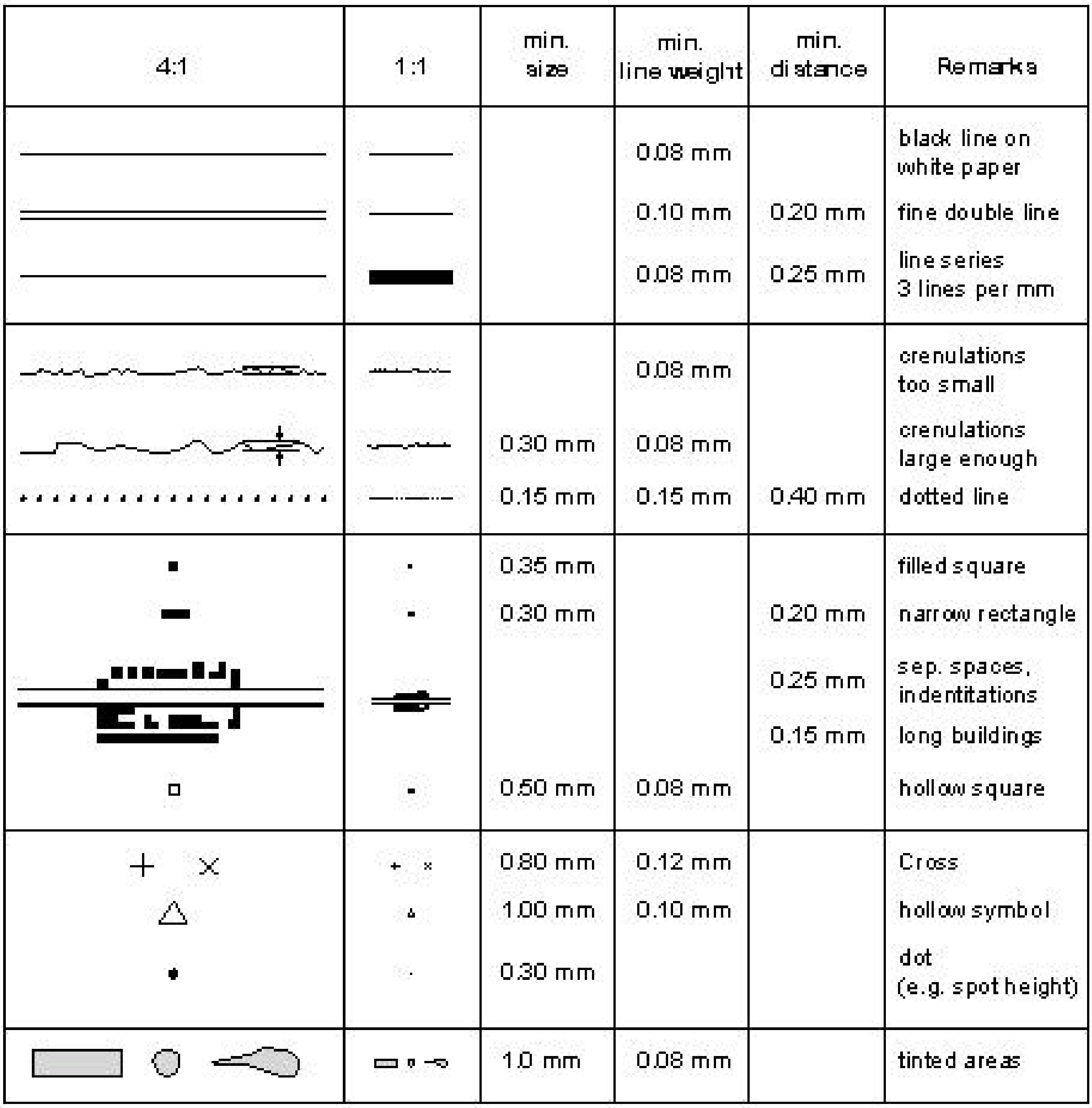
c = chain; p = polygon; x = point; a = any_type

i = intrinsic; e = extrinsic

3.4 Graphic Limits

The smallest line weights, spot sizes and symbol separations allowed to be displayed on a map are given by what we term *graphic limits*. These minima reflect the display media and technology that are used as well as perceptual criteria, and thus will be larger for interactive maps than for most paper maps. As figure 3.1 illustrates, the graphic limits of symbols and spaces between symbols should be chosen to be significantly larger than the associated limits of visual perceptibility.

The values given below are valid for standalone black symbols on white paper. For symbols in light shades of gray or in colour these values must be considerably increased. Strong competition with neighboring map elements also calls for increased values for graphic limits.



4:1	1:1	min. size	min. line weight	min. distance	Remarks
			0.08 mm		black line on white paper
			0.10 mm	0.20 mm	fine double line
			0.08 mm	0.25 mm	line series 3 lines per mm
			0.08 mm		crenulations too small
		0.30 mm	0.08 mm		crenulations large enough
		0.15 mm	0.15 mm	0.40 mm	dotted line
		0.35 mm			filled square
		0.30 mm		0.20 mm	narrow rectangle
				0.25 mm	sep. spaces, indentitations
				0.15 mm	long buildings
		0.50 mm	0.08 mm		hollow square
		0.80 mm	0.12 mm		Cross
		1.00 mm	0.10 mm		hollow symbol
		0.30 mm			dot (e.g. spot height)
		1.0 mm	0.08 mm		tinted areas

Fig. 3.1: Examples of graphic limits for printed maps from Swiss cartographic guidelines

Contributions of graphics to this chapter by Beat Peter, GIUZ, are greatly appreciated.

4 Geographic (Theme-based) Constraints

The project has identified certain classes of geographic features, data for which may be used in system prototypes. These are:

- Political and Administrative Boundaries
- Land Use and Land Cover
- Transportation (roads, trails, railways) and Buildings
- Waterbodies (area features)
- Watercourses (linear features)
- Terrain Features (included principally to help in constraining other features)

One goal of compiling feature-specific constraints is to see which constraints operate across feature classes. Many of these are described in section 3, which deals with cartographic aspects of constraints. Here we examine properties of geographic phenomena in order to specify constraints that enable their uniqueness to be identified, measured and hopefully preserved in the course of applying generalisation methods. However, certain characteristics (for example sinuosity) are common to more than one theme (such as rivers, roads and boundaries following them), and thus appear in more than one theme listed below. Whenever a constraint re-occurs, its original description will be referenced rather than repeated. To make referring to constraints easier, each one is assigned a unique number. The ordering of these numbers is *not significant*, however, as they are merely labels. The numbering scheme has two levels; the first denotes the contextual scope of the constraint, the second the specific constraint. There are five scopes identified, as follows:

1.1–1.x Constraints *internal* to a feature class
2.1–2.x Constraints imposed *on* other feature classes
3.1–3.x Constraints imposed *by* other feature classes
4.1–4.x Constraints imposed *on* partitions
5.1–5.x Constraints imposed *by* partitions

By partitions we mean areal subsets of map space that are treated as units, usually to constrain the space within which interactions among map objects must be modelled. Partitions can be formed from linear features (e.g., roads defining blocks), areal features (e.g., administrative units) or arbitrary grids. For purposes of defining partition-related constraints it does not matter which of these approaches is used. Before describing specific theme constraints, some additional general terms and assumptions concerning modelling of features may be helpful. These may not apply to all feature classes, and may be modified in actual system prototypes, but they are rather common assumptions that are not normally violated.

1. The word *chain* will be used to refer to segments of boundaries that exist between topological nodes. *Arc* is a synonym.
2. Similarly, the term *polygon* will be used to denote the encoded form of areal features. Polygons are formed from one or more chains, and chains are assumed not to be duplicated for shared boundaries. A polygon can be simple (isolated) or adjacent to others. Polygons may contain enclaves («lakes») or exclaves («islands») which themselves are polygons.
3. In general, we assume it is possible for objects in different feature classes to share geometry, usually via referencing common chains. Whether or not this practice is followed has strong implications for how certain constraints are enforced.
4. Partitions are assumed to be polygons, and it must be possible to identify all features lying (entirely or partially) within a partition, as well as to identify which partitions are neighbours of each other.
5. Each constraint is labelled according to what *type* of constraint it is (see sec. 2) Geometric (*G*), Topological (*T*), Structural (*S*) or Gestalt ( ), whether it principally constrains points (*x*), chains

(*c*), polygons (*p*) or any type (*a*) and whether the constraint is intrinsic (*i*) or extrinsic (*e*). *f* is used for the special case of field representations for digital terrain models (DTMs) The symbol (*R*) – for *reflexive* – is appended in the case of binary relations that can be symmetric (i.e., can be imposed by other feature classes as well as on other classes).

4.1 Constraints for Boundary Themes

Theme: BOUNDARIES (Polygon Features, constructed by chains)

Sub-classes: PROPERTY; POLITICAL; ADMINISTRATIVE

Author: Geoffrey Dutton and Robert Weibel, GIUZ

Version: 19 March 1998; 2 July 1998

Specific Properties

1. Boundaries have a dual nature, being lines that represent regions (finite areas). While it is conceivable that boundaries might be encoded that are not part of complete regions, we do not consider this exception here. Boundaries that run off the map can be completed by clipping them to the map extents, and only need be treated as polygons for certain analytic purposes.
2. Some boundary sub-classes may be hierarchically related, such as municipal and provincial boundaries. While a complete topological integration may be possible across their sub-classes, there are often exceptions (e.g., areas that do not nest properly or missing pieces or territory). We therefore assume that sub-classes will not be integrated except to the extent that shared boundaries are explicitly defined in the database. That is, sub-levels are represented separately from larger ones that contain them.
3. Parcel boundaries (property lines) are a legitimate species of boundaries, but as they will not be treated in this project, their specific constraints will not be elaborated. This is because parcels are not normally represented on medium-scale topographic maps.

1. Internal Constraints for Boundaries

1.1. *Maintain polygon closure (Tpi).* It is assumed that boundaries denote closed regions. Hence, a boundary chain cannot occur in isolation either before or following generalisation.

1.2. *Consistently eliminate polygon chains (Tpi).* Because boundaries represent polygons, if one boundary chain is eliminated, all chains (not only references to them) for that polygon should also be eliminated.

1.3. *Eliminate isolated polygons by simple deletion (Tpi).* If a disjoint polygon such as a small enclave is eliminated, the boundary simply vanishes.

1.4. *Eliminate non-isolated polygons by interpolation (Tpi).* If a small connected polygon is eliminated, all junctions where its boundary met others must be collapsed to one point and new edges formed that radiate from it.

1.5. `Preserve original line character` (*Sce*). The inherent character expressing the generating process of a boundary should be preserved. For instance, the angular shape boundaries based on surveyed lines should be maintained; likewise, where boundaries coincide with a river they should remain smooth.

2. Constraints imposed by Boundaries on other classes

2.1. *Preserve shared chain primitives with other FCs (Tc(R)).* If a (portion of a) boundary is (defined as) coincident with another boundary or line representing another feature class, the other line(s) should reflect changes made to the geometry of the boundary due to generalising.

2.2. *Preserve containment relations (Tp).* If a feature is contained by a boundary's polygon prior to generalization, it should remain contained afterwards if it still exists. However, this could trigger a conflict with the contained feature's displacement constraint, should its absolute position need to be retained.

2.3. *Preserve proximity to boundaries* (*Gxp*). Point and small isolated polygon features lying arbitrarily close to a boundary (e.g., border crossings) may or may not be regarded as conflicting with it, depending on whether overprinting is permitted (because boundaries are not physical linear objects). In any case, such features should remain near to the generalised boundaries, subject to displacement limits.

2.4. *Preserve alignment to boundaries* (*Gce(R)*). Neighboring polygons of the same boundary class are by definition aligned to (share) a common boundary (constraint 2.1). Other feature classes (e.g., roads, rivers, land use) may be *roughly aligned* with boundaries and their symbology should reflect this relationship after generalization. This is a relaxation of constraint 2.1; in such cases it is assumed that different feature classes do <u>not</u> share chains.

2.5. `Use of boundaries for positional control` (*Sae*). If boundary polygons are used to partition a map (for divide-and-conquer processing) constraints 2.1-2.4 may be interpreted more stringently. In such cases, boundaries can serve as surrogates for directly maintaining relations of features that lie close to or on them. By maintaining relative positions to a common boundary, such features can maintain positions relative to one another, even though there is no analytic structure (e.g., Delaunay) that associates them directly.

3. Constraints imposed on Boundaries <u>by</u> other classes

3.1. *Follow shared higher priority vertices (Tci(R)).* Boundaries should not be displaced from points that define them. If important boundary vertices (such as road intersections or survey corners) are individually displaced, the boundary should follow the displacement to remain coincident with such points.

4. Constraints imposed <u>on/by</u> partitions

See constraint 2.5 for remarks on situations where boundaries *define* partitions. Using political/administrative units to partition a map may often be useful, as membership in such units may be already coded for many features. That is, such units already tessellate space, and may be useful for map processing at some scales.

4.1. *Ensure consistent treatment in all partitions (cpe(R)).* Actions taken within partitions to modify boundary geometry (e.g., simplification, aggregation) must be globally consistent within each boundary sub-class, both within and between partitions.

4.2. *Preserve chain continuity across partition edges (Tce(R)).* Boundaries that cross partitions must be generalised consistently such that all boundaries maintain first- and (if necessary) second-order continuity between adjacent partitions.

4.2 Constraints for Landuse/Landcover Themes

Theme: LAND USE/COVER (Polygon features, constructed by chains)

Sub-classes: URBAN, AGRICULTURAL, RECREATIONAL

Authors: Beat Peter, Geoffrey Dutton, GIUZ

Version: 20 July 1998

Specific Properties

1. Refer to Specific Properties for *boundaries* for data modelling concepts.
2. "*patch*" and "*zone*" are used interchangeably to refer to a polygon.

3. "Land*cover*" refers to the predominant material or vegetation across a region. "Land*use*" categorises the primary land-related activities across a region, whether physically observable or not.
4. Zones are phenomena-defined, and therefore based on attributes (classified categories of land use and land cover). We assume that all zones are defined in a single dataset, not just those that represent a specific category. This means that an attribute (or at least an identifier) is required to determine a zone’s category.
5. Some land classification schemes are hierarchical, allowing classes and their associated geometry to specialise. While this can be useful for generalisation, we make no assumptions about whether or how this is done, and thus would treat each such level of detail as an independent dataset or class of objects (i.e. as multiple representations).
6. On some maps, landuse or landcover is defined everywhere to form a complete mosaic of polygons. On topographic maps landuse/landcover is depicted only in certain areas. The source data may need to be conditioned to allow the second type of depiction if it is a mosaic, and certain constraints will operate differently in each case.
7. In most cases, landuse patches have a low or even lowest priority compared to the other feature classes, and a low potential for conflict as well, because landuse symbolisation is usually an area tint that underlies all other symbolism.
8. Landuse/landcover as a secondary feature class is particularly affected by relations with other feature classes such as shared primitives (see I3 and L6 in chapter 3) or proximity relations (see I8 and L6). If roads, rivers etc. need to be displaced or reconfigured, landuse/landcover patches may have to be a) displaced accordingly or b) resized if conflicts with other objects of the named feature classes arise (e.g. a forest patch between two roads). If the same polylines that define the roads also define the edges of patches, this constraint is simple (but requires the capability to represent shared line primitives). If not, the procedure needs to be governed by rules similar to those for placing boundary lines, moving them to coincide with roads when they appear to be constrained by them.
9. On topographic maps, landuse/landcover patches provide important visual information of the general structure of a landscape. Depending on the map purpose, this information should be retained in the target map. Several possibilities exist to prevent landuse/ landcover classes (which consist mainly of small polygons) from being reduced dramatically in number of patches and total area on the target map. They are discussed in the next section under items 1.2 to 1.6.

1. Internal Constraints

1.1 The entire set of *boundary* internal constraints generally applies to landuse and landcover, but different criteria or measures may be applied when evaluating them.

1.2 *Respect importance of small polygons (Gpi).* (Isolated) zones with an area below a threshold could either be deleted or enlarged, depending on the importance of a patch in a region (partition) and its context (important land*use*).

1.3 *Allow fusion (Spce)*. Groups of (small) zones of the same category that share boundaries can be aggregated to a single, larger object representing the group at a smaller scale, possibly by deleting chains separating such zones and rebuilding polygons.

1.4 *Allow amalgamation (Spe)*. Sufficiently small, nearby (but disjoint) patches can be coalesced into larger ones of the same category (disjoint case of 1.4). This extends the shape of the large patch usually to cover the small ones.

1.5 *Allow Typification (Gpe)*. Groups of small patches of the same category can be replaced by fewer but larger ones of that category to retain landuse or landcover characteristics without amalgamation. This only applies when polygons are isolated or included objects, not a complete mosaic.

1.6 *Preserve corner character (Gce)*. For some landuses e.g. agriculture, their boundaries may turn corners at right or other characteristic angles that should be retained. Patch orientation, which is in many such cases is often orthogonal to contour-lines, may also be constrained.

2 Constraints imposed by Landuse/Landcover on other classes

2.1 *Use landuse to modify other classes' constraints (Spi)*. In a few cases (e.g. military installations or restricted access areas) patches might be used as «treatment partitions» to impose limitations for operations (such as differential displacement or special iconification rules) on other feature classes. This implies that certain attributes or attribute values be consulted to identify patch priorities that differ from those for the feature class as a whole.

3. Constraints imposed on Landuse/Landcover by other classes

3.1 *Respect topographic determinants (Sci)*. Land use areas may sometimes be constrained by topography. For example, military areas and national parks are often structured by natural geographic entities, e.g. ridges or valleys. If topography is generalised, this can influence shapes of such areal units. When a DTM is not being used, such area patches could be simplified directly. In either case, some structural knowledge is helpful to identify such situations and link specific zone boundaries to constraining natural and cultural features in the source database.

3.2 *Respect intervening features (Spi)*. Other feature classes with high priority may prevent landuse/landcover patches from being aggregated. For example, the built-up area of a town may extend between fields of crops. If the shape of the settlement is to be preserved, it will act as a constraint to prevent the amalgamation of fields (which ought not to underlie buildings). However, it may often be the case that the settlement's "fingers" would be too narrow at target scale, and must be simplified, allowing the agricultural areas to be combined.

4. Constraints imposed on/by partitions

4.1 Certain exceptions can be foreseen that can cause the same phenomena to be generalised differently in different locations (which may be defined via partitions or otherwise). Depending on the regional context of a landuse/landcover patch, differing priorities and constraints may apply. Internal constraint parameters governing elimination, typification or amalgamation, for instance, could be different for landuse/landcover categories in urban than they would be in mountainous regions. For example, the minimum area criterion for woodland might also be different in and near urban areas than in rural areas. Also, forest patches might be typified differently in natural woodlands than in plantations. Such subtleties could be modelled by attribute codes that summarise relevant characteristics (nominated and weighted by the user or architect) to describe a set of contextual constraints the zones can impose. Landuse/landcover polygons can also be used to delimit partitions that require modified treatment due to their special nature. This is a *strategy* rather than a constraint, but it also implements semantic knowledge,

---

4.3 Constraints for Road Network Themes

Theme: ROADS (Line feature organised as a network)

Authors: Sebastien Mustière, Anne Ruas IGN; Geoffrey Dutton, GIUZ

Version: 24 August 1998

The constraints below are specified for inter-urban roads. Urban ones will be treated somewhat, but not entirely, differently; in urban areas streets and buildings are more inter-related and thus constrain each other more strongly.

Recall from ch. 1 that *intrinsic* constraints consider only one state of the database; e.g., insure legibility. *Extrinsic* constraints compare a state with a reference; e.g., preserve planimetric accuracy.

In certain circumstances, some of the following constraints can be ignored, and their relations can change, depending on map controls.

Specific Properties

1. Geometrically, roads are represented as *polylines* which connect only at endpoints.
2. Topologically, roads are *networks* (cyclic and/or acyclic graphs) that may or may not be directed. The direction of segments, if specified, is generally not significant (in contrast to hydrographic networks which have unidirectional flows).
3. Roadways may be *divided*, and may be portrayed as dual carriageways on some maps, but sometimes this is represented in databases only as attributes (number-of-lanes, median-strip).
4. *Routes* are collections of roads that possess the same name along throughout; often routes need to be externally identified and imposed on lower-level linear objects. The *classification* of roadways (e.g. A-roads, B-roads, M-roads) generally indicates their volume of traffic.
5. The *intrinsic importance* of roads varies, and this information is usually provided as attributes; the importance of certain roadways for generalisation purposes can be relative or contextual, and thus may need to be computed analytically. Also, as map scale decreases roadways often increase in importance relative to other feature classes (are more visually dominant).
6. The places and ways in which roads *intersect* frequently are important locations and their character needs to be preserved when generalising.

Utilisation of roads on maps

Constraints imposed are directly imposed by the roles roads have *qua* phenomena. On a map roads

1. are used for travel, to navigate from one point to another;
2. are fixed landmarks in the landscape: we use roads to know where we are and to locate other features;
3. are used to estimate travel distance and time;
4. are used for access: what road do I take to reach this point? What area is accessible by this road network?
5. reflect terrain relief: if roads are sinuous, one can assume that there are steep slopes;
6. are used for network analysis, and this helps to establish which roads can be eliminated in order to generalise. However, sometimes roads are more important than such analyses may imply (e.g., the only access to a tourist attraction or military base).

1 Constraints on a road

Constraints 1.1 - 1.3 are intrinsic;

1.1. *Forbid self intersections* (i.e., maintain line topology): It is independent of final symbology. (*Tci)*

1.2. *The geometry must reflect the road's character* can be independent of final symbology. (*GSci*)

1.3. *The graphic symbols must be legible,* should suppress high-frequency detail, avoid coalescing with themselves or other roads, and communicate the significance of their routes. (*Gai)*

Constraints 1.4 - 1.10 are extrinsic and work at several levels of analysis (local, intermediate, local).

1.4. Preserve geometric accuracy (Gce).

1.5. Preserve orientation (*Sce*).

1.6. Preserve local shapes (*Sce*).

1.7. Preserve global shapes (*Sce*).

1.8. Preserve ratios of bends size (*Sce*). This means exaggerating consistently.

1.9. Preserve ratios of homogeneous sections (homogeneous = sinuous, straight…) (*Sce*)

1.10. Preserve important bends and bends series; they are landmarks and can denote navigational hazards (*Sce*).

2 Constraints on neighbouring roads

Constraints 2.1 - 2.6 are intrinsic, and may work at several levels of analysis

2.1. *Forbid intersections between two lines without node* (*Tci*). It is independent of final symbology.

2.2. *respect minimum distance between two lines* (*Gci*).

2.3. *avoid coalescence between two lines* (*Sci*).

2.4. *Respect minimum size between cross-roads* (*Gci*).

2.5. *Respect minimum size of dead-end* (*Sci*).

2.6. *Respect minimum size of areas delimited by roads* (*GScpi*; areas defined by the lines). This is not a constraint directly on the roads, but these areas may not be features in the database.

Constraints 2.7 - 2.14 are extrinsic, and may work at several levels of analysis

2.7. *Preserve alignment at a cross-road* (*Sce*): this is more important for straight feature than for sinuous one.

2.8. *Preserve relative orientations* (*Sce*): this includes preservation of parallelism and perpendicularity.

2.9. *Preserve line length ratios* (*Sce*).

2.10. *Preserve and keep consistent local pattern shape* (circle shape, cycles, round-about) (*Sce*).

2.11. *Preserve as well as possible and always keep consistent the given topology* (*Tce*)

i. *Avoid disconnecting two roads previously connected by a node* (*Tce*). This is a stronger constraint than ii, below.

ii. *Avoid removing an arc previously connecting two nodes* (*Tce*).

2.12. *Avoid cross-road coalescence due to symbolisation* (*Tce*)

i. The perceived location of a cross-road should be consistent with its location in the DB

ii. The perceived topology around a cross-road should be consistent with the DB one.

iii. The perceived shape and orientation of lines around a cross-road should be consistent with the DB one.

2.13. *Preserve geometric accuracy of cross-roads* (*Gxe*; nodes defined by cross-roads). Usually, we put more strong constraint on the location of cross-roads than on the location of the whole line, because the cross-roads are more remarkable points on the ground.

2.14. *Preserve leap-over relations (above/under)* (*STce*). If overpasses are modelled as features, this constraint is required for consistency between several classes.

3 Constraints on the whole network

3.1 *Preserve the number of connected sub-graphs of the network* (*STce*).

3.2 *Preserve relative accessibility between different origins and destinations*. (*GTce*).

3.3 *Preserve access area of the network* (*STcpe*).

3.4 *Keep most important roads* (*STce*). The importance of a road depends on its attributes, its implicit semantic or touristic importance. Several types of importance can exist.

3.5 *Keep roads density distribution relative to road importance* (*S ce*). Importance must be understood like in constraint 3.4 (and not only on national classification).

3.6 *Preserve network patterns* (*STce*): ex: star-like, tree-like, rectangular grid, reticulated web.

4 Constraints of consistency between several classes (contextual)

Note: depending on the feature priorities, a constraint can be imposed on or by the other classes, thus these are treated in one section rather than two.

4.1 *preserve road position in thalwegs* (*Scpe*): if the vertex of the bend is on the bottom of a thalweg, it should remain.

4.2 *preserve relative position to contour lines* (*Sce*): if the road climb before generalisation, it must climb after generalisation

4.3 *constraints of A-II between two roads apply also with other features (river, railway)* (*STce*).

4.4 *preserve consistency of the position and type of leap-over features (bridge/tunnel) when moving the intersection location* (*STce*).

4.5 *preserve relation with orographic features* (*Sce*): if the road round the hill by the south, it should be kept.

4.6 *preserve access to significant features* (*STce*): some features shall remain accessible by the network Link to them can be deleted if these features are deleted.

4.7 *preserve landmarks* (*Sce*): some objects shall remain because they help as guides in the network. For example, towns at dead-ends and intersections are more important than towns along a road

4.8 *features should remain on the same side of the road (and on the same areas defined by the roads)* (*STca*).

4.9 *the relative proximity and orientation of features (like houses) to the road shall be kept* (*GSae*). In particular, relative proximity should not be reversed.

____

4.4 Constraints for Street and Building Themes

Theme: BUILDINGS and STREETS (built-up areas)

Sub-classes: RESIDENTIAL; COMMERCIAL; INSTITUTIONAL; INDUSTRIAL; STREETS, ALLEYS, ROADS, HIGHWAYS

Author: Anne Ruas, IGN; Geoffrey Dutton, GIUZ

Version: 24 July 1998

Preliminary Remarks

1. Due to the intimate relationship between local streets and buildings (particularly in built-up areas), and due to the general futility of treating each in isolation, these themes are discussed together; This section is thus a deliberate departure from the other discussions in this chapter.
2. Constraints are defined at different levels: *micro*, *meso*, *macro*;

   "macro" characterises regional or overall map space; related to scope of semantic constraints

   "meso" characterises a neighborhood treated as a unit on a map, usually involving several to dozens of features of various types; related to scope of structural constraints

   "micro" characterises a neighborhood defined by a feature, part of one or several that can be treated in isolation; related to scope of geometric and graphic constraints
3. The terms *Semantic* and *Structural* usually can be used interchangably, although structural knowledge is only one aspect of semantics and also involves geometric relationships
4. References to chapter 3 (i.e. cartographic symbolism) is described as Cross Reference: CR

Specific Properties

1. Topologically, urban streets have the same constraints as roads do (see sec. 3 above), although motorway-class roads are usually treated differently than streets in urban areas. Also, the prevalence of dead-ends is frequently higher in urban situations than in roads generally.
2. Streets are used to access buildings, hence should not be eliminated to leave buildings isolated
3. Likewise, eliminating all buildings along a street may call for the street to be eliminated.
4. The territory enclosed by a block is generally used as a unit of analysis at the meso level, specifically to estimate building density and constrain building displacement.
5. As map scale decreases, streets tend to dominate buildings in urban areas until at some scale (perhaps > 1:100,000) almost all buildings are replaced by area tints in topographic maps.

1. Micro constraints on buildings: Polygons

1.1 *Accuracy* (*Gpe*) CR: A0 (L1)

i. Buildings should stay close to their initial position. If particular geodetic points rely on them, this constraint is enforced

1.2 *Orientation* t(*Gpe*) CR

i. A building should preserve its initial main orientation, with respect to adjoining buildings and streets, and absolutely if closely neighboring features are not transformed.

1.3 *Shape* (*Gpe*) CR: A0 (L3); A3

i. Buildings should preserve their main *orthogonality*
ii. Buildings should maintain their *elongation* to be recognisable

iii. If buildings have *unusual shapes* such as curves, these shapes should be maintained

1.4 *Granularity* (*Gpi/e*) CR: A0 (L2)

i. Internal outline segments should be of perceptible dimensions (see 3.4). GPI

1.5 *Size* (*Gpi*) CR A1; A2

i. Building size should exceed the minimum area limit (see 3.4). A1

ii. Building width should be large enough to avoid visual confusion A2

1.6 *Functionality* (*Spi*) CR

i. Important buildings (according to map purposes) should be maintained

ii. Isolated buildings can be important buildings as reference objects

2. Meso Constraints on Buildings (i.e. set of buildings within an area)

2.1 *Topology* (*Tpe*) CR A0(L4, L5)

i. Existing connectivity, adjacent and inclusion relationships must be maintained if buildings are maintained, (subject to aggregation).

2.2 *Orientation* (*Gpe*)

i. Near, neighboring buildings should try to preserve their relative main orientation

2.3 *Proximity/repartition* (*Gpi/e*) CR A6-7-8

i. Smallest distance between two polygons must be greater than separation distance (*Gpi*) CR A7

ii. Relative distance between two polygons should be maintained (*Gpe*) CR A7

iii. Relative distances between a set of generalised buildings should present the same clustering effect as was initially measured (*Spe*) CR A6

iv. Specific patterns and alignments (e.g., along curves) should be maintained (*Spe*) CR A8

v. No specific distribution should be created if it does not exist initially (e.g., if buildings are not organised along a grid, they should not be after generalisation) (*Spe*) CR A8

2.4 *Size distribution* (**pe*) CR A8

i. Size dilation should not disturb relative size order of buildings whenever their semantic meaning is the same (*Spe*)

ii. Two building which have nearly the same sizes can have equal final size (*Gpe*)

2.5 *Shape distribution* (*G/Spe*)

i. Within a group of buildings, if some are removed, the typified subset of buildings should preserve buildings which have remarkable shapes (remarkable inside the group); this implies choosing different forms of simplification or typification at meso scales.

2.6 *Density* (*Spi/e*) CR A9/10

i. Groups of buildings should not have too great density (normally measured by the ink/paper area ratio for simulated symbolism at output scale) (*Spe*) CR A0(l2)

ii. A group of buildings having too great density can be aggregated or re-symbolised as a built-up area if it corresponds to the spatial pattern of that portion of a town

2.7 *Semantics* (*S/ pe*)

i. Two buildings can be aggregated together if they do not visually have contrasting semantic meanings. (*Spe*)

ii. The distribution of different sub-classes of buildings within an area can change but proportion should try to be maintained, unless if it is required by map specifications. Some indication of semantic exceptions should be maintained (e.g. a commercial area within a neighbourhood of detached houses). (*S/ pe*)

3. Macro constraints on buildings

3.1 *Quantity constraints*

i. According to map purposes, the area occupied by specific buildings can be maintained (for statistical study).

3.2 *Density distribution*

i. The ordering of density values among meso objects prior to and after generalisation should not greatly change. (*pe*) CR A9

ii. Differentiation of densities should be still perceptible after generalisation (*pe*) CR A9

3.3 *Shape of districts* (*Gpe*)

i. Globally, districts should have the same kind of shapes (elongation, convexity) as they initially did even if their sizes increase (e.g., resulting from street removal).

3.4 *Size of districts* (*pe*) CR A9

i. Globally, districts should maintain a difference of sizes according to initial conditions. That is, aggregation should not alter the relative sizes out of proportion.

3.5 *Semantic distribution* (*Spe*) CR A9

i. If a town has a specific distribution of semantic districts, it should be maintained (town centre, detached houses areas, commercial area, long-building area,…).

4. Meso-constraints Street-buildings

4.1 *Topology* (*Tpe*) CR A0(L4, L5)

i. Existing inclusion relationships between a building and a street partition must be maintained (i.e., buildings must not move across streets).

4.2 *Orientation* (*Gcpe*)

i. Close street and buildings should try to preserve their relative main orientation

4.3 *Proximity/repartition* (*Gcpi/e*) CR A6-7-8

i. Relative distance between a street and a building must be greater than separation distance unless specifically authorised. It may occur in maps in which maintaining their visual contrast is important. (*Gpi*) CR A7

ii. Relative distance between a street and a building must be as close as it was initially (*Gpe*) CR A7

4.4 *Functional dependency* (*Spe*)

i. Buildings should be removed/aggregated if their streets are eliminated, *but*

ii. Generalisation of street networks should allow for access to ‘important’ buildings.

5. Meso-constraints on Streets

Constraints on (urban/suburban) streets themselves are the same as road constraints, except for:

5.1 *Network Shape* (*GSce*)

i. If a specific shape such as a ring road exists, it should be preserved

5.2 *Network density/repartition* (*STce*)

i. Network density should conform to the initial pattern. The generalised network should be more dense where it was initially.

5.3 *Functional constraint* (*STce*)

i. A generalised network should allow to access to the city centre(s)

ii. A generalised network should allow to avoid the city centre(s) (if possible)

iii. A generalised network should allow an easy access to main semantic roads

---

4.5 Constraints for Watercourse Themes

Theme: WATERCOURSES (linear hydrographic features)

Sub-classes: RIVERS; STREAMS; CANALS

Author: Robert Weibel, GIUZ

Version: 1 July 1998

Specific Properties

1. Watercourses are mainly made up of lines which are represented as chains. Depending on scale, they may also be delineated as areas; in that case, the watercourse is represented by double lines or by a polygon.
2. Natural watercourses usually exhibit a current, but the presence of a current or even of water is not required for their portrayal.
3. Natural watercourses are constrained by topography and follow the shape of the terrain surface.
4. Natural watercourses usually form tree-like networks. Braided streams and rivers are exceptions from the tree structure; they include cycles.
5. Natural watercourses exhibit particular shape patterns as a result of fluid dynamics (meandering, braiding) and as a result of existing geology and geomorphological processes (dendritic, radial, or parallel network structure).
6. Natural watercourses may be perennial (exist during the entire year) or intermittent (exist only when sufficient water is available).
7. Man-made watercourses -- particularly canals -- exhibit more regular pattern because they are often engineered to minimise distance, vertical drop and curvature. They are also not equally constrained by topography as their natural counterparts: vertical drops can be overcome by locks and canals can even pass through tunnels.
8. Where watercourses are modelled as lines, their display is subject to the cartographic constraints for linear symbols.
9. Whenever watercourses are represented as areas, their display is equivalent to waterbodies.

1. Internal Constraints

1.1 *Maintain fluvial line character (ce).* The inherent character expressing the generating process of a watercourse should be preserved. For instance, the gentle sinuosity of a meandering river, the chaotic pattern of a braided stream, and the designed shapes of canals should all be maintained.

1.2 *Control partial collapse (ci).* Collapse of double line symbols used to display watercourses is possible wherever the double lines are too close at target scale (partial collapse), but should not only be controlled by distance but also by semantics. Dammed stretches of a watercourse or sections which are very wide or important for navigation can be exaggerated in width to avoid collapse. Otherwise, collapse should be done consistently to avoid creating the impression of lakes embedded in a watercourse.

1.3 *Maintain connectivity of hydrographic network (Tce).* The connectivity of the network of watercourses must at all scales be maintained. In a natural network, the flow logic must be maintained (that is, the network can only be pruned from the leaves inwards). In a network of canals, the logic of connectivity is based on traffic flow. This assumes source data are properly topologically connected.

1.4 *Maintain the spatial pattern of the hydrographic network (ce).* The specific structure and pattern of a hydrographic network (e.g., dendritic, radial) should be maintained. No part of the network should be excessively pruned in favour of another.

1.5 *Maintain the hierarchy of main rivers and tributaries (ce).* The hierarchy of main rivers and smaller tributaries should be clearly preserved. Exterior links of main rivers should not be eliminated, while tributaries can be pruned. Using stream ordering such as Horton Order can be a useful heuristic in pruning tributaries.

2. Constraints imposed <u>on</u> other classes

2.1 *Intersect contours at point of maximal curvature (Gxci(R)).* Watercourses should remain located in valley bottoms. Hence, (natural) watercourses should intersect contours at the point of locally maximal curvature. Contours need to be adjusted if the constraint is violated.

2.2 *Maintain descendant flow logic (Gci(R)).* Streams and rivers should follow paths through contour lines which logically result in a monotonically descending profile (i.e. water must flow downhill). If necessary, contour lines must be displaced or the elevations of the DTM adjusted.

2.3 *Maintain relative positions (Tae).* Left-of and right-of relationships and other relative positional relationships must be preserved. For instance, a house located on the left bank of a river should not be moved to the right bank.

2.4 *Maintain bridge symbolisation (Tce(R)).* When ground transportation routes (roads, railroads) cross watercourses a bridge is required. Hence, if a transportation route crosses a watercourse, the appropriate symbolisation for a bridge should be used. Conversely, if a transportation route did not previously cross a watercourse, this disjunctive relationship should also be maintained.

3. Constraints imposed <u>by</u> other classes

3.1 *Compensate for lake removal (Tcpe).* If a lake in a hydrographic network is removed the drainage network connectivity must be re-established to compensate for the gap opened by the now missing waterbody.

3.2 *Compensate for contour removal (Tce).* If contour indentations marking the course of a stream are no longer shown, the stream itself should be omitted.

4.6 Constraints for Waterbody Themes

Theme: WATERBODIES (areal hydrographic features)

Sub-classes: LAKES; PONDS; RESERVOIRS; PERENNIAL; INTERMITTENT

Author: Robert Weibel, GIUZ

Version: 1 July 1998

Specific Properties

1. Waterbodies denote closed surfaces of water which are represented by polygons made up of chains.
2. Waterbodies show no or only slight current (standing water).
3. Natural waterbodies are usually connected to networks of linear hydrographic features (streams, rivers). Usually, there are one or multiple input streams from the upstream drainage basin and a single outlet.
4. In exceptional situations, no intakes and outlets exist for natural or man-made waterbodies (e.g., when the intake or outlet is sub-aquatic, that is, through the bottom of the waterbody). Such exceptions must be especially coded.
5. Natural waterbodies may be perennial (exist during the entire year) or intermittent (exist only when sufficient water is available). Their coding and symbolism should reflect this.
6. Small man-made waterbodies (e.g., fire or retention reservoirs) often exhibit more regular pattern (e.g., they may have a circular or rectangular shape).
7. The shoreline of waterbodies follows the contour of the lake surface’s elevation on the surrounding topography.
8. Waterbodies inherit all cartographic constraints for line symbols and areal symbols.

1. Internal Constraints

1.1 *Maintain original shoreline character (ce).* The inherent character of the shoreline of waterbodies should be maintained. For natural waterbodies, the shape is usually an irregular undulation reflected in the surrounding contour lines. Artificial waterbodies exhibit designed shapes that can be easily parameterised.

1.2 *Maintain original area shape (pe).* Apart from the inherent character of the shoreline, the overall shape of waterbodies must by maintained. The holistic shape impression of the area polygon becomes increasingly important with decreasing scale. Man-made alterations such as dams, jetties, piers and marinas should keep their character as well as possible.

1.3 *Consistently select waterbodies (Sci).* Multiple criteria -- size, depth, importance to water supply, importance to tourism or other economic factors) -- may be applied to select the waterbodies for display at a smaller scale. These selection criteria must be consistently applied.

2. Constraints imposed on other classes

2.1 *Maintain consistency with elevation contours (Gce).* The shoreline of waterbodies must follow the elevation contour corresponding to the elevation of the waterbody surface (no intersections created). Any modifications of the shape of the waterbody must therefore be propagated to the surrounding elevation contours (including optional depth contours).

2.2 *Maintain charge and discharge logic (Tcpe).* Waterbodies (e.g., lakes) must have at least one inflowing watercourse that charges water and one outflowing watercourse that discharges the waterbody, unless they did not have such in source data. This constrains pruning of hydrographic networks.

2.3 *Maintain bridge symbolisation (Tcpe).* When ground transportation routes (roads, railroads) cross waterbodies a bridge is required. Hence, if a transportation route crosses a waterbody, the appropriate symbolisation for a bridge should be used. Conversely, if a transportation route did not previously cross a waterbody, this disjunctive relationship should also be maintained.

2.4 *Maintain relative positions (GTcpe).* Buildings and iconic symbols along a shoreline should preserve their relative orientation to it after generalisation. They should be displaced along with the shoreline and not end up in the water. This requires special attention when groups of lakes having adjacent buildings (e.g., resort communities) are amalgamated.

3. Constraints imposed by other classes

3.1 *Make room for transportation (GTcpe).* A road or railway following a steep lake shore should be displaced towards the lake, inducing a shape modification of the lake (and possibly of depth contours).

4.7 Constraints for Digital Terrain Model Themes

Theme: DIGITAL TERRAIN MODEL (TOPOGRAPHY)

Sub-classes: GRID DTM, TIN, SURFACE PATCH DTM

Author: Robert Weibel, GIUZ

Version: 14 July 1998

NOTE: This section is included for reference and future implementation. It is not anticipated that DTM data will be utilised in the prototype system.

Specific Properties

In many ways, terrain (or topography) represents a special case among the FCs of a topographic map. Hence, a number of general observations should be made about the special nature of this FC class before going into details about the constraints controlling its generalization.

1. Topography imposes a number of important physical constraints on processes and activities placed in the environment.
2. With *hydrography*, there is a very direct mutual relationship:
   - Terrain dictates the flow of water.
   - Hydrography (fluvial erosion) shapes terrain.
3. With other feature classes, there exists partial coincidence:
   - Political and administrative *boundaries* may follow ridges (or rivers = drainage channels)
   - *Forests* are generally more frequent on north facing slopes (on the northern hemisphere).
   - *Roads* must minimise slope (leading to hairpins); may follow contours (e.g., along lake shores); will tend to have the apex of bends coincident with the apex of contours of convex and concave topographic shapes.
   - *Railroads* must minimise slope under even more stringent constraints than roads (smaller gradient possible, larger minimal curve radius). As a consequence, tunnels (e.g., 'spiral tunnels') and topographic cuts and dams are more frequent for railroads.
   - *Settlements* tend to be concentrated on the more gentle slopes and preferably on south facing slopes. Individual buildings may be placed on steeper slopes (but these tend to be special buildings, such as mountain cabins).
4. *Priority*: Despite its importance as a physical variable, terrain is of *secondary importance* during the resolution of spatial conflicts in topographic mapping. For example, in a situation where a river, a road, and a railroad all pass through a narrow gorge the three linear features have to be enlarged considerably to remain clearly visible and separable at smaller scales, and hence the terrain representation (contours, hillshading) has to make room.
5. *Internal representation*: Terrain is internally represented by a so-called *digital terrain model (DTM)*. It is the only topographic theme that must be internally represented as a (continuous) field, usually by means of data structures such as regular grids or TINs (triangulated irregular networks).
6. *Graphical representation* (i.e., graphical representation) is usually accomplished by contours lines or hillshading.
7. Because of the differences in representation, DIGITAL TERRAIN MODEL and CONTOUR LINES are treated separately as two feature classes.

1. Internal Constraints

Note: Internal constraints on the DTM are only of relevance for terrain generalization. That, however, is not the primary focus of the project.

1.1 *Maintain continuous field property (Gfi)*. The DTM should remain a valid representation of a continuous field after generalization.

1.2 *Maintain complete coverage (Gfi)*. The continuous field represented by the DTM should remain defined over the entire study area; no holes should be introduced. Exception: If 'dead areas' have been previously defined, they should be maintained as holes.

1.3 *Maintain monotonous descent property (Gfi/e)*. The terrain surface should be monotonously descending; no spurious local pits should be created. Exceptions: karstic terrain or natural or artificial depressions.

1.4 *Stay within vertical bounds (Gfe)*. The global vertical extent (min/max of elevations) should not be altered during generalization.

1.5 *Preserve local extremes (Gfe)*. Similar to the global vertical extent, the (significant) local minima and maxima should not be altered by more than a specified elevation threshold.

1.6 *Preserve salient features (Sfe).* Salient terrain features – identified analytically or defined during database (DTM) population – should be maintained. The shape of salient features, however, may change.

1.7 *Preserve convex/concave shapes (Sfe).* Relative shares of concave and convex terrain shapes ('valleys' and 'mountains') should not change. E.g., the terrain surface should not be 'flattened' by filling valleys.

1.8 *Maintain surface discontinuities (Sfe).* If there are sharp incisions or ridges, they should be maintained as discontinuities and should not be planed away.

2. Constraints imposed <u>on</u> other classes

As mentioned above, terrain ranks rather low in terms of its importance compared to other feature classes. It is therefore a rather 'reactive' feature class in operations such as displacement.

As a consequence of this, we are listing constraints between terrain and other classes under heading C.

This varies of course with scale and map purpose: On large scale maps, topography will be modified to a lesser degree by other feature classes, while it will largely have to compensate the modifications of other classes at smaller scales.

3. Constraints imposed <u>by</u> other classes

3.1 *Minimise displacement of spot heights (Gfxi).* At the x/y locations of spot heights, the elevation of the DTM surface should not change by more than a specified threshold. Modifications to the DTM elevation will cause spot heights to be displaced laterally. (Note: This constraint only holds for large to medium scales, where spot heights are displayed.)

3.2 *Maintain flatness of lakes (Gfpe).* The DTM surface must be flat (level) within the boundary of lakes or the sea surface.

3.3 *Preserve fluvial continuity (Gfcpe(R)).* All drainage paths over the DTM surface must be able to reach the boundary of the DTM (no artificial pits). That is, it must be possible to extract a complete drainage graph. Exception: Karstic terrain.

3.4 *Preserve fluvial coincidence (Tfce(R)).* The drainage network which can be extracted from the generalised DTM should follow the hydrographic network as far as possible.

3.5 *Preserve boundary coincidence (Tfce(R)).* Where boundaries (political, administrative, other) coincide with ridges or drainage channels, this relationship must not be destroyed as a consequence of the generalization process.

3.6 *Maintain road footprint (Tfce).* The cut and fill of the footprint of roads should be maintained at large scales (high resolutions).

3.7 *Maintain topographic positions (Tfae(R).* Features of other classes should not change their topographic position. For example, if a house is on a south facing slope, it should not be on a north facing slope following generalization.

4. Constraints imposed <u>on/by</u> partitions

4.1 *Preserve surface continuity (Gfe).* Along both sides of a partition boundary the DTM surface must find a continuation. Continuous differentiability of the DTM surface must be maintained.

4.2 *Form geomorphological regions (Sfe).* For terrain generalization (not the primary focus of this project), partitions should be the result of a geomorphometric classification, and hence have a meaning as 'physiographic' regions.

4.3 *Adapt to geomorphological regimes (Sfe).* Provided the regions represent different geomorphological regimes/types, generalization rules should adapt to the local regime.

4.8 Constraints for Contour Line Themes

Theme: CONTOUR LINES (TOPOGRAPHY)

Sub-classes: none

Author: Robert Weibel and Mats Bader, GIUZ

Version: 14 July 1998

NOTE: This section is included for reference and future implementation. It is not anticipated that contour line data will be utilised in the prototype system.

Specific Properties

1. See the description in the section on DIGITAL TERRAIN MODEL.
2. Contour lines are a particular method of graphical representation of the topographic surface (which is itself modelled by a digital terrain model). Contour lines, as isolines of equal elevation, depict the topography in a quantitative way.
3. Contour lines are most meaningful at large to medium scales (1:500 to 1:100,000). At small scales (< 1:100,000), contours can merely render topography in a qualitative way.
4. Two quality levels are often distinguished for contour drawings:
   - *Engineering quality* drawings are required to be clearly legible, logical, and consistent. However, utmost graphical refinement is not required. Hence, contour labelling may not be aligned with the contours, and a contour may intersect with itself (which is mathematically possible at saddle points).
   - *Cartographic quality*. contours require further cartographic refinement (contour labelling; contours must not intersect). This is the standard for topographic maps.

1. Internal Constraints

Note: Internal constraints on the DTM are only of relevance for terrain generalization. That, however, is not the primary focus of the project.

1.1 *Avoid self-intersection (Tci).* Self-intersecting contours are only tolerable in engineering quality drawings.

1.2 *Maintain closure (Tci).* Contour lines should either be closed or bounded by the map sheet (or DTM) boundary. Exceptions: Intermediate contours which are defined to be open; partial suppression of contours in rocky or steep areas.

1.3 *Maintain completeness (Sci).* For a given contour interval, all elevations must be represented by contours. Exceptions: Partial suppression of contours in rocky or steep areas; intermediate contours will increase the number of contours.

1.4 *Eliminate small contours (Gci).* Contour lines that are shorter than a specified length threshold will be suppressed entirely (rather than enlarged).

1.5 *Avoid congested contours Gci).* Neighboring contours should not touch each other (happens on steep slopes). Possible remedial actions: (slight) displacement, partial suppression (blanking), replacement by alternative display method (e.g., rock engraving).

1.6 *Maintain geomorphological line character (Wce).* Individual contour lines must represent the geomorphological structure of the underlying terrain. Hence, contours tend to be jagged in rocky or karstic terrain, or smooth in rolling terrain.

1.7 *Geomorphological shape representation property (Wce(R).* Geomorphological structures and landforms (e.g., a ravine, a ridge) should be maintained and extended across neighboring contours. Adequate representation of landforms is the primary objective of contour displays.

2. Constraints imposed <u>on</u> other classes

As mentioned above, terrain ranks rather low in terms of its importance compared to other feature classes. It is therefore a rather 'reactive' feature class in operations such as displacement.

As a consequence of this, we are listing constraints between terrain and other classes under heading 3.

This varies of course with scale and map purpose: On large scale maps, topography will be modified to a lesser degree by other feature classes, while it will largely have to compensate the modifications of other classes at smaller scales.

3. Constraints imposed by other classes

3.1 *Minimise spot height displacement (Gxce(R)).* Spot heights should only marginally be displaced laterally as a result of the modification of contour lines. The new position of spot heights should be consistent with their height. (Note: This constraint only holds for large to medium scales, where spot heights are displayed.)

3.2 *Maintain spot height containment (Txce).* Spot heights should always be contained within the contour interval that their elevation falls into. (Note: This constraint only holds for large to medium scales, where spot heights are displayed.)

3.3 *Intersect streams at contour apex (Gce(R)).* Contours should intersect streams at a local apex of the contours (locally highest concave curvature).

3.4 *Intersect roads at right angle (Gce(R)).* Where the line width of roads is wide enough (especially when the road is displayed as a double line), contours should intersect at a right angle with roads. Roads or trails displayed at a narrow line width can be intersected by contours at an arbitrary angle.

3.5 *Do not intersect shorelines (Tcpe).* Shorelines of lakes, ponds, or the sea (a special case of a contour line) must not be intersected.

4. Constraints imposed on partitions

4.1 *Adapt to geomorphological regimes (Wce).* Provided the regions represent different geomorphological regimes/types, generalization rules should adapt to the local regime.

5 Constraints and Modelling

In the context of this project, the term *modelling* refers to three activities, which are clearly very intimately related:

1. *Data/knowledge Modelling* – Abstracting and organising phenomena and relationships into elements that can be represented in a computer by abstract data types.
2. *Process Modelling* – Formalising methods used to define, characterise and solve problems, specifying how they relate and interact.
3. *Agent Modelling* – designing and organising agents based on models of phenomena (objects) and cartographic operations (algorithms and workflows).

Mostly this document addresses data modelling concerns, as most of the constraints to generalisation that we have studied relate to spatial data or representations of it. However, as Mackaness (1995) describes, both manual and automated generalisation involve a number of procedural constraints as well. Furthermore, as agents are essentially processes, it is quite necessary to describe (model) process constraints in order to achieve the objectives of this project. The ideas this report presents with regard to agent modelling are intended to be heuristic and are thus preliminary and far from definitive. Please refer to the reports for tasks A1 and B1 for details on agent architecture.

5.1 Modelling Data to Support Generalization Constraints

Geographic data, specifically vector cartographic data, can be modelled in a number of valid ways, each having their own strengths and weaknesses. For the purposes of this project, however, the useful range of representations are constrained, as the GOTHIC database is object-oriented, and also capable of modelling topological relations dynamically. Certain base classes there will thus be defined, and from these a hierarchy of classes will be specialised to represent various map features, along with refinements of methods declared for the base classes. Alongside of the map objects will be agents, possibly assuming the identity of map objects, or controlling areas of map space or representing constraints. Constraints themselves might also be represented as objects, if they are not instantiated as agents. Many constraints are in fact simply parameters (e.g., minimum areas, widths, and separation distances, line weights and symbol diameters) that are user-specified for classes of features or their graphic representations, as chapters 3 and 4 have itemised.

However, there are still certain constraints that are not easily represented in an O-O context, but which are potentially useful in generalisation. These involve relationships among features and provide semantic information about their behavior and roles. For example, certain features such as buildings and access roads, parks and monuments, bridges and rivers, etc., tend to have logical relationships, such that one should not be eliminated or moved without regard to another. We call such paired objects *peer features*, and believe it might be helpful to analyse map data to identify them, then insert references into objects to let them point to one another. In composing maps, peers can be quickly identified, and operations on them can then be harmonised. In the following sub-section, the nature and use of semantic and structural information is discussed, and a framework developed for incorporating at least certain types of such information into a spatial database. We will describe how certain geometric and topological patterns can be detected via pre-processing map data, and the results of such analyses stored for use in map production, streamlining the process. But other useful information can also be drawn from outside sources or user knowledge.

5.1.1 Modelling Structural and Semantic Knowledge

We call descriptions of the organisation of map features in space *structural knowledge*, which is a type of *semantic knowledge* (semantics need not have a spatial component). Structural and semantic knowledge of map features takes a variety of forms and levels of meaning. It can range from simple facts concerning individual features (such as a building being a historic monument, or a run of a river being a main branch or a tributary) to facts that relate pairs of features (such as indicating a road that leads to a campground or a lake which is the water supply for a town), to complexes of facts that relate larger groups of features (such as the linear features such as rivers, highways and railroads; and alignments of building footprints characterised by their spacing and orientations. Knowledge obtainable through such analyses minimum spanning tree and shape statistics relating a set of lakes or a Delaunay network itemising proximities of a set of buildings and roads). As the above examples indicate, structure and semantics may be based on user knowledge of the landscape and roles of features in it, coded as feature attributes, or via interpretations of measures and structures made on sets of map objects, frequently ones that are in immediate proximity. These interpretations are, of course, constrained by cultural attitudes toward maps; a given geometric arrangement or topologic pattern may signal a diversity of phenomena, depending on the landscape and the mapmaker.

Discovering structure. Some kinds of structural knowledge can only be discovered *post hoc,* often derived via geometric or topological analyses of map objects. Examples include sinuosity analysis of rivers and roads, leading to their logical segmentation according to local shape characteristics; identification of parallel stretches of should be considered as a proxy for information that could be gathered in the field and attached to map objects. Semantic knowledge of *networks* often falls in this category: the importance of routes and road segments can be modelled by simulating driving behaviour (Morisset and Ruas 1997), or taken from route classifications, but the true importance of roads can only be understood by measurements of actual use, such as number of lanes, traffic volumes or average speeds, data that many transportation agencies compile. It is not necessarily easy, however, to join transportation agency data to one's road objects given the different naming, route identification and even physical representations of road networks that may exist between two sources of data.. This is a simple example of the *conflation* problem that causes difficulties in handling *multiple representations* of map data (Cobb et al. 1998); it illustrates the limitations to sharing semantic knowledge, which is almost always contextual (bound to the nature, purpose and procedures of the institutions that create and use representations of it).

Even if semantic/structural information is easily obtainable, it is only meaningful if it serves the user's purposes. Using another network example, a river network might be classified automatically (based on topological analysis and measures such as stream segment lengths), or segments could be classified using their actual names. Yet another approach would be to import stream gauge statistics of volume-rate-of-flow data for each segment, and classify them accordingly. But none of these approaches will be sufficient if a map's purpose is, for example, to describe to boating and fishing parties which streams are navigable, hazardous or otherwise notable. Temporal semantics may also be needed, such as the seasonality of streams or the likelihood of flash flooding. A road map, on the other hand, would have more need for stream names than for any of their flow statistics.

Describing semantics. We can, then, classify structural and semantic knowledge of maps into several levels of increasing complexity:

I. Individual features

1. Function in the landscape (natural, cultural, prominent, ordinary, ...)
   – based principally on semantic knowledge;

B. Role/importance in a map (junction, landmark, edge, ...)
   – based principally on geometric and semantic knowledge;

C. Distinguishing characteristics (sinuosity, rectilinearity, elongation, ...)
   – based principally on structural and semantic knowledge;

D. Associations (via containment, proximity, similarity, ...)with other features
   – based principally on geometric and structural knowledge;

II. Networked features

A. Function in the landscape (main route, tributary, access road, bottleneck , ...)

B. Role/importance in a map (topological connection, container, ...)

C. Distinguishing characteristics (directionality, degree, capacity, name, ...)

D. Associations (via junction, containment, proximity, parallelism, ...)with other features

III. Grouped features

A. Function in the landscape (catchment, village, neighbourhood, strip development , ...)

B. Role/importance in a map (characterisation, differentiation, treatment, ...)

C. Distinguishing characteristics (density, granularity, uniformity, uniqueness, ...)

D. Associations (via proximity, connectivity, similarity, ...)with other groups of features

Note that any given feature can have semantics in any or all categories, as it may have a variety of roles, depending on its context and on map purpose. Each type of network (rivers, railroads, roads, paths) often is described in isolation, but invariably has crossings with others that could be explicitly modelled. This is necessary, for example, in order to make the structural semantics of *bridges* clear. In some databases (e.g., TIGER), groupings of features may be pre-defined (such as the city, district, postcode and block a building lies within) on the basis of identifiers or street addresses. In many databases such information is lacking or may be insufficient for generalisation. Then it is necessary to *discover* features that can affect one another through generalisation. This is usually done by *partitioning*, then imposing *proximity structures* on the geometry of features to identify nearest neighbours and compute inter-object distances that might cause constraint violations (Ruas 1998).

Semantics of map partitions. When a structure such as a Delaunay triangulation is computed to discover proximity relationships, the information derived must somehow be made available to the map objects it includes. How this is best done mainly depends on the following considerations:

1. stability of the map database (frequency of updates);
2. whether Delaunay structures are stored as persistent objects;
3. the time and space complexity costs of (re-)computing such structures;
4. ways in which structural information are linked to or incorporated into data schemata.

As for (1), we will assume that the database is at least stable enough to avoid having to regularly recompute proximity relationships, as long as (2) allows them to be archived in the database. The overhead involving recomputation of structures (3) may be tolerable or not, depending on the number of them and the frequency with which they are built and accessed. Thus, the fourth consideration may be fairly important in deciding upon strategies for handling structural information, but relatively little is known about how to do this optimally.

One way to encode Delaunay data might be to provide map objects with lists of their neighbours, as determined by triangulating them. The vertices and edges connecting each object and the ones immediately adjacent to it , along with the ID of the neighbours, could be copied to lists that could be ordered by proximity. For isolated objects such as buildings, it is almost impossible to possess more than six such neighbours (Saalfeld 1998). Roads, and other linear objects, however, could have a much larger number of adjacent buildings connected to them, depending on the size and density of the partition being triangulated. However, all that would have to be stored for roads is a reference to each building that is linked to it, either in their order along the road or ordered by proximity. A road object would then obtain the geometric properties of it associated proximity links by asking the buildings to provide them.

The type of data that a building might store to relate it to its neighboring buildings is illustrated in table 5.1. It identifies the neighboring object IDs, the (minimum) ground distance separating them, the angle of the line connecting their centroids, and the endpoints of the line forming the shortest path between them. Buildings would have similar records to relate them to positions along roads (or rivers or railroads) that they were closest to as well,

BLDG_ID	NEIGHBR	PROX_M	DIR_ANG	X!	Y1	X2	Y2
1003	1007	2.7	21	212.3	845.2	212.9	841.3
	1023	3.8	-23.7	211.7	842.1	209.8	840.3
...							
	1015	10.2	-48.4	212..0	842.5	214..6	839.1

Table 5.1: Example of proximity records relating buildings in a partition

There are other properties of objects in partitions beyond those that refer to single objects or pairs of objects. These structural properties include densities (which may be measured in several ways and can be characterised by means, min/max and moments), proximities (likewise), shapes (elongation, area-perimeter measures, etc.) and more, as will be detailed in AGENT work package C. Some of these measures may be attributed to specific objects, but whether they can be or not, it is still necessary to provide summary statistics for entire partitions (or clusters, however objects are grouped for analysis). To store such information (which can be expensive to compute), objects that represent the groups of features will need to be instantiated. These objects might be the Delaunay structures themselves, or organised in other ways. They could also be built into agents.

Partitioning map features can be used to control their treatment (i.e., rendering) in subtle ways. By this we mean that *constraints to treatments may be imposed by land use, land cover or other types of zones*, based either on their principle classification or on other attributes they may possess. For example, the size or class thresholds for selecting roads, buildings and point features may be different in a park, preserve or military installation than they are in other places. Land use and zoning boundaries can be overlaid on other features to accomplish this, and can also be a way to classify buildings as industrial or institutional, for example (if such attributes do not already exist for buildings). Whether this is useful depends mostly on the map purpose, and whether buildings need to be distinguished for treatment based on functional criteria. This, then, is an application of map partitioning, one that is not aimed at supporting divide-and-conquer strategies, but in reifying cartographic knowledge. It can be particularly useful in making maps that have well-defined regions (e.g., urban, suburban, rural) where different treatments (hence the salience and ordering of constraints) may be necessary. The themes *boundaries* and *land use/land cover* thus provide ready-made partitioning schemes that can be used as appropriate to modify the types of treatment other features receive.

5.2 Modelling Constraints of Generalization Processing

The overall process of digital map generalization has some inherent procedural constraints that need to be respected. Many of these are related to the sequencing of operations, which can be done in various ways. While it is premature to prescribe a particular process for the project to follow, a rough scenario is given below to illustrate some of the decision points and possible dependencies to be considered:

1. Controls are specified for a map (particularly its purpose and scale), dictating:
   a. The feature classes to be included and their relative importances
   b. The legend, or symbolism set to be used for features at the given scale
   c. Graphic limits, which specify thresholds of legibility, also constrained by (f)
   d. Default selection/elimination criteria for features within each class
   e. A map window to which features must be clipped
   f. The spatial and spectral resolution of the display surface
2. Spatial analysis methods are selected for detecting constraint violations (but are not yet invoked). Methods described below include:
   a. Allometric modelling (e.g., formulations of the Radical Law)
   b. Attribute analysis (to identify features not qualified for retention)
   c. Buffer, triangulation and other analyses (to detect overlapping symbols)
   d. Feature grouping (identifying neighbours in clusters of features)
   e. Partitioning plans (strategies for divide-and-conquer)
3. Allometric modelling (which applies dimensional analysis to predict influences of size or scale on shape) is used to estimate the numbers of features of each class that are expected at target scale; the degree of reduction this indicates can be used in determining the need for and number of partitions, as well as the potential severity of competition for map space. For descriptions of allometric analysis, see (Töpfer, F. and Pillewizer, W. 1966) and (Naroll and von Bertalanffy 1956).

4. Attribute analysis can identify specific features that are important to retain and possibly exaggerate, either generically (e.g., by areal extent) or specifically (e.g., by codes indicating importance as a landmark). Likewise, it can identify features that are minor enough to drop. Constraints may exist between feature classes, for example:
   i. Important buildings can require access roads to be present, however minor.
   ii. Roads classified as relatively unimportant will be dropped, and cause buildings along them to be eliminated, unless (a) prevents eliminating a road.
   iii. Water bodies below minimum size may need to be enlarged rather than dropped if a related feature next to them (e.g., a factory) is retained.

   Such decisions may be taken on the basis of attribute values, if manual coding or pre-processing has been done to identify closely associated or «peer» features, or that otherwise finds exceptions to default elimination decisions.
5. Area features that are below minimum size but identified as needing to be retained are exaggerated or typified to the necessary degree. If done later, their potential conflicts will need to be analyzed a second time.
6. Buffer analysis can next identify overlaps or proximity problems among the remaining selected features, using «simulated density» (scale-enlarge symbolism drawn in an off-screen or on-screen image buffer). Places where features touch themselves can be recorded for later use in choosing or guiding simplification algorithms. Note that display resolution can and should affect such simulations.
7. Partitioning and grouping methods (e.g., Delaunay triangulations, Voronoi regions, minimum spanning trees) are invoked to identify which proximity problems are to be treated together by identifying natural groups of features.
8. The severity of geometric and topological conflicts in each such group (more than one of which may exist in a partition) are evaluated by diagnostic measures. The nature and comparability of these measures are critical to orchestrating remedial solutions to the conflicts. Structural constraints, such as sinuosity regimes, parallel orientations or regularity of centroid distributions, also need to be identified at this time.
9. Tactics for sequencing transformations must be decided. Two main approaches, not mutually exclusive, can be identified:
   i. Apply transformations to the most important feature class first, then to the next, etc. «Importance» may vary with map purpose; generally the more important a feature is, the less it will be displaced and the more it may constrain transformations of less important features.
   ii. Apply transformations in order of decreasing effect on map geometry and topology (in order not to change things more times than necessary); for example, simplify before smoothing, amalgamate before displacing, and eliminate before typifying.
   iii. Evaluations of generalization transformation results may be made in between applying operators, or not until a sequence of operators is completed. This may depend on which operators are to be invoked; for example, simplification may be followed directly by smoothing.
10. As (9) indicates, iterations between transformation and analysis may continue until solutions converge to acceptable states. However, sufficient halting criteria are necessary to avoid inconsequential changes and infinite loops.

A complete analysis of procedural constraints – if indeed this is possible – would surely identify a great many decision points. It would also have to identify what types of data and criteria were needed to make these decisions. But it would not really have to specify how the control, analysis and transformation should be packaged (the specific capabilities of algorithms, agents, etc.). However, the analysis could identify certain critical decision points that are likely to be problematic, and at which users might be required to intervene. These might be strategic or tactical; «little» decisions (like whether to displace or amalgamate two buildings) may have as difficult as «big» ones (such as what features should be allocated to the same cluster or partition for analysis).

5.3 Constraints as Goals for Agents

One characteristic of certain types of agents is that they have *goals* (which are sometimes described as *desires*). It seems generally possible for an agent to develop its own goals, to prioritise goals and to modify them, but *for our application many goals will be externally defined and imposed upon agents*. Achieving goals – or making as much progress toward them as may be possible – will translate into reaching acceptable generalisation solutions. One therefore should be able to formulate goals for agents from identified *constraints to generalisation*. At least sometimes rather direct mappings between constraints and goals can be found, and in the process various key parameters and types of data that are involved identified.

For example, a goal might be for a feature to keep its relative proximity relations with its neighbours after generalising. Whether agents are modelled as map features or as something else, this goal can still be approached if current distances to neighbours and minimum allowable distances between features on the target map are known. Maintaining the latter distance criterion is also a goal, and is based on properties of symbolism plus rules for special cases, if any. Keeping inter-feature distances balanced and large enough requires satisfying constraints, internalised as goals. A focus on goals is useful because, in the words of Kinny et al. (1996: 2.2)

> ... goals, as compared to behaviours or plans, are more stable in any application domain. Correctly identifying goals leads to a more robust system design, where changes in behaviours can be accommodated as new ways of achieving the same goal. In other words, a goal-oriented analysis results in more stable, robust and modular designs.

Following this approach leads one to ask certain questions about agent architecture:

1. ***How to map constraints to goals?***
2. ***Are there constraints which should not be goals?***
3. ***Are there goals which do not represent constraints?***
4. ***Do different types of agents share any common goals?***
5. ***How can achievement of goals be assessed?***
6. ***What agents should be given which goals?***

These questions are closely related. Answering them may require analyses of what Weibel and Dutton (1998) identify as «scope of constraints,» the conditions and neighbourhoods within which constraints operate and are valid. Constraints may have an order of precedence, but this may vary situationally. When a constraint's salience increases, goals may shift, at least temporarily or for that situation. Satisfaction of an important goal (such as maintaining topological consistency) may need to be put off or even sacrificed to serve a lower-priority one (such as avoiding clutter), as when railroad track symbols are deleted within the boundaries of an urban area. «Avoiding clutter» is a general goal that can have different interpretations and can be operationalised in context-dependent ways. «Maintaining topological consistency» is a goal that has fewer interpretations and its satisfaction is easier to measure. Trading off between these goals, when necessary, presumably would be accomplished by executing alternative *plans* («strategies» in chapter 2). One can describe sets of goals, contingencies and plans without having to specify which particular agents will eventually handle them, but having done that it should be easier to describe the beliefs, capabilities and intentions required, leading to a full specification of agents themselves.

Almost all the constraints identified in chapters 3 and 4 are of two types: they represent criteria that should be *preserved* or that should be *avoided*. But it is often possible to recast one in terms of the other, if this makes implementation more tractable. Agents should have freedom to interpret constraints in either positive or negative terms.

5.3.1 Mapping Constraints to Goals

To expand a constraint into a goal is to describe *how* things are affected by controls in addition to *what* these things are. Goals may be either qualitative or quantitative, but ultimately measures need to be designed to describe what needs to be done for each goal adopted. Table 5.2 provides some simple qualitative examples of turning some graphic qualities specified by controls into constraints expressed as goals. Most of them would have to be modified to address specific feature classes, as would their associated measures of severity and success. Furthermore, to adequately depict their interrelationships would require a more complicated structure than a table, such as a semantic net, as controls can affect multiple goals and goals constrain one another.

5.3.2 Constraints Which Might Not Be Goals[1]

Some cartographers are of the opinion that map generalisation is difficult to formalise because as it is part of map design, «the devil is in the details.» Indeed, there are many cartographic (or simply graphic) subtleties that map designers and drafters should understand and attend to. These include masking line symbols at overpasses, aligning dashes in dashed lines with bends and intersections, avoiding superposition of symbols that have very similar colours, and simplifying streams slightly more when they are aligned to political boundaries. The quality of a map is dependent on the sum of hundreds of minute details such as these, and NMAs therefore often take great pains to specify their employees how to treat them, usually by issuing style sheets and manuals. Most such rules are indeed graphic constraints, but many of them are not closely related to generalisation *per se*. Ignoring them in the context of AGENT is not only possible, but probably necessary from a practical point of view. However, it is not easy to draw a line between necessary and unnecessary constraints, and maps resulting from our work will no doubt have many minor shortcomings as a result.

An example of such a borderline constraint is a stream junction occurring close to a bridge that crosses the river downstream of the junction. Even though the road or bridge symbolism will mask that of the streams, it is considered good practice to displace the stream junction upstream to avoid conflict with the crossing, and possibly giving the impression that there are two bridges rather than one. If such constraints were to be handled, a goal would need to be specified, along the lines of «Do not allow junctions in hydrographic networks to occupy the same space as transportation features.» The action to meet the goal would be specified as a strategy, such as «Displace such hydrographic nodes at least 2 mm in an upstream direction, propagating the displacement to the streamlines (and possibly to nearby features as well, such as buildings and roads).»

At the other end of the scale are constraints that potentially affect everything on the map. A good example is whether or not the map is to be generated in colour or in black-and-white. Ideally, once this is specified, all decisions that the constraint can affect would use appropriate criteria to determine legibility. But as this is really a map design problem as well as a generalisation problem, solutions may be under-specified unless a great deal of human effort is made to trace out their implications in advance. So, a decision should probably made up front whether the system should be able to handle this constraint. If it is to be ignored, users must be made aware that maps produced in monochrome may contain flaws not found in colour maps.

[1] Contributions from Frank Brazile (GIUZ) to the content of this section are highly appreciated.

Table 5.2	EXAMPLES OF RELATIONSHIPS AMONG CONTROLS, CONSTRAINTS AND GOALS NOTE: Goals need to be specified more concretely for each particular feature class	
CONTROLS	**GRAPHIC ASPECTS**	**CONSTRAINTS / GOALS**
Map Purpose	Class Priorities	Place Themes according to Importance
	Legibility	Maintain Appropriate Symbol Size & Separation
	Complexity	Portray Neccesary Information and Detail
Map Scale	Legibility	Avoid overprinted symbols
	Feature Density	Avoid Clutter, consistently
	Perceptibility	Include only Visible Details
	Proximity	Limit Symbol Proximity (incl. own Vertices)
Map Legend	Point Symbols	Determine Repertoire and size limits
	Line Symbols	Determine Repertoire and width limits
	Area Symbols	Determine Repertoire and size, width, length limits
	Color	Maintain Contrasts and Harmony
Ouput Medium	Color palette	Determine Repertoire of hues and values
	Resolution	Determine minimum Line Weights, Pattern Detail, etc.
Graphic Limits	Spot Sizes	Enforce minimum Diameters
	Line Weights	Enforce minimum Widths
	Pattern Types	Ensure Patterns are discernable and distinct
Data Quality	Spatial Resolution	Adjust Resolution to Scale
	Spatial Accuracy	Allow Movement within Error Limits
	Attribute accuracy	Allow logical collapse of Categories
	Geometric Shape	Allow distortion of Shape appropriate to Class
Topology	Nodes	Avoid spurious Nodes
		Embed Point Features in Faces
	Edges	Lines must terminate at Nodes
		Edges cannot cross each other
	Faces	Faces must close
		Faces may be aggregated
		Avoid subdividing faces

5.3.3 Goals Which Do Not Represent Generalisation Constraints

Agents may have goals that are not based on generalisation constraints. One obvious example is for an agent to always reach a terminal state. Another might be to synchronise certain activities with other agents (thereby answering question 4 above in the affirmative, but perhaps trivially). But these could also be regarded as elements of *plans*, not goals. As specifications for agents, environments, interactions and organisations are built up, some task-specific goals may be identified which are not derived from constraints. Here we will not concern ourselves with this possibility, given that the goal of this report is a detailed specification of constraints to generalisation and how they might be systematically abstracted and applied.

5.3.4 Identifying Common Goals

The methodology proposed by Kinny et al. (1996) for specifying MAS architectures operates in two phases: (1) deriving an *agent model*, and (2) deriving an *interaction model*, both of which they claim to be independent of the BDI (beliefs-desires-intentions) architecture which they prefer to use. In their view, an agent model is like an OOP class hierarchy (the static aspect), while an interaction model focuses on responsibilities, services and communications (the dynamic aspect). The paper then focuses on defining agents using the BDI paradigm, involving specification of a *belief model*, a *goal model* and a *plan model.* The goal model «describes the goals an agent may possibly adopt, and the events to which it can respond.» The methodology for developing the models «begins from the services provided by an agent and the associated events and interactions. These define its purpose, and determine the top-level goals that the agent must be able to achieve. Analysis of the goals and their further breakdown into sub-goals leads naturally to the identification of different means, i.e., plans, by which a goal can be achieved.» (Kinny et al. 1996: 2.2)

In this approach the services provided by agents (their specific functionality) in large part determines the goals they will adopt. To the extent that agents provide similar services they will possess similar goals, which we postulate to be internalisations of constraints. The fewer goals exist in the system of agents, the less conflict among them is likely to occur, and the easier and faster it should be to satisfy goals. Therefore it would seem to be better to specify a small number of common goals than a large number of individual goals whenever possible. Even though we do not have a clear idea of how agents will be defined, we can still identify some of the *services* that they must perform. Although many types of services may be needed, the most obvious candidates for services are *generalisation operators*, e.g.:

• Selection	- Choosing themes and feature classes to include in a map
• Elimination	- De-selecting specific features based on attributes, etc.
• Simplification	- Removing excessive detail from features
• Smoothing	- Softening the appearance of caricatured lines
• Aggregation	- Clustering point features to become areal features
• Amalgamation	- Clustering small areal features to become larger ones
• Merging	- Joining nearby linear features into a smaller set of lines
• Displacement	- Movement of features away from one another
• Typification	- Altering geometry to standardise shapes or patterns
• Collapse	- Reducing the dimensionality of a feature
...	(mostly from McMaster and Shea 1992)

In McMaster and Shea's framework, these operators are seen as working together in order to satisfy certain *philosophical objectives* (although it is not very clear what philosophy has to do with them). The largest group of these objectives is called *theoretical elements*, which are itemised as follows:

1. reducing complexity
2. maintaining spatial accuracy
3. maintaining attribute accuracy
4. maintaining aesthetic quality
5. maintaining a logical hierarchy, and
6. consistently applying generalisation rules. (McMaster and Shea 1992: 28)

These elements can be conceived of as top-level goals for agents, although they might not be the *only* ones that may be needed, and their formulation could probably be made more explicit. Because they are so general, we will continue to call them *objectives* so as to distinguish them from agents' *goals*, which must be more concretely specified to be useful. Looking at the first five of these objectives in relation to operators, we find some commonalties, as table 5.3 shows:

Table 5.3	Some Objectives Served by Generalization Operators								
	GENERALIZATION OPERATORS (services)								
GENERALIZATION OBJECTIVES	**Selec-tion**	**Elimin-ation**	**Simpli-fication**	**Col-lapse**	**Displa-cement**	**Smoo-thing**	**Aggre-gation**	**Amalga-mation**	**Merg-ing**
Reduce/Maintain Graphic Complexity	•	•	•	•	o	o	•	•	•
Maintain/Standardize Spatial Accuracy		•	±	±	±	±	o	o	o
Maintain/Standardize Attribute Accuracy	o	o					o	o	o
Maintain/Standardize Aesthetic Quality	•	•	•	o	•	•	•	•	•
Reduce/Maintain Attribute Hierarchy	o	o					•	•	•

Legend: • = Strong Effect; o = Weak Effect; ± = May Help or Inhibit; blank = Not Related

This informal exercise indicates that it should be possible to design agents to perform map generalisation that share important goals. It also shows the major application-domain services that agents might supply seem to organise into three groups:

A. Choosing Features (Selection, Elimination)

B. Transforming Individual Features (Simplification, Smoothing, Collapse, Displacement)

C. Combining Features (Aggregation, Amalgamation, Merging)

Selection and Elimination both weed objects from the target map, usually prior to invoking other operators, but also can come into play to resolve problems that other operators cannot handle satisfactorily. Simplification and Smoothing often are invoked together to identify and maintain characteristic shapes. Aggregation, Amalgamation and Merging unite neighboring features of the same class into fewer map objects. Displacement and Collapse solve problems of crowding by giving objects more breathing space, as Selection and Elimination also do, less analytically but with more drastic effect. Typification is not included in table 5.3 because it can be performed in different ways, some of which are purely geometric transformations and others which may perform elimination or some form of combination. For example, substituting rectangles for buildings, regularising arrangements of buildings and reducing the number of bends in a road are all different kinds of typification, and each relates to generalisation objectives somewhat differently.

Notice, however, that there is very little about this set or grouping of operators that addresses *holistic* generalisation concerns, beyond the displacement and combining of neighboring features. McMaster and Shea's sixth objective, «consistently applying generalisation rules,» (omitted from table 5.3) is essentially holistic, but none of the conventional operators directly address it.[2] One could argue that consistent use of generalisation operators and parameters will usually result in a self-consistent map, but in complex cases or when scale transitions are large, and *especially* in an multi-agent environment, inconsistencies will almost certainly appear unless they are actively prevented from occurring or later reconciled. Agent goals must be developed that formalise, refine and instantiate this objective.

5.3.5 Assessing Achievement of Goals

In the framework proposed here, most goals represent criteria derived from constraints to generalisation. For an agent to realise its goals, it may or may not have to act. Whether it acts or not depends on if

[2] At least as they are commonly implemented in the vector domain. In the raster domain operators such as simplification and smoothing are more likely to be applied consistently (although their results may not always be regarded as appropriate).

problems are detected in its realm of competence. Such diagnoses can be performed by an agent itself or for it by other agents, which then inform the one responsible for handling the problems. In general, these diagnostics are called *measures*, as they are (normally) quantitative assessments of spatial relations which measure geometric, topological and other properties. They can qualify a map as a whole, regions within a map, feature classes, individual features, groups of features and even portions of features.

Likewise, once agents act to invoke generalisation operators in response to problems described by measures, it is usually necessary to reassess the situation to determine if the solutions are satisfactory. More often than not, the same measures would be used following generalisation as were employed to trigger it. But some measures are quite local (such as detecting symbolism overlaps, characterising curvatures or assessing strengths of colour contrasts), while others are more global (such as assessing changes in feature densities or checking whether separately-processed partitions of a map fit together properly). Note that identifying mismatches between adjacent partitions probably requires different measures than were used to define the partitions.

Assessing goal achievement in large measure amounts to determining whether constraints have been satisfied. Three types of results can be anticipated:

A. A constraint is/is not satisfied

B. Certain related constraints are satisfied, others are not

C. A constraint may be partially satisfied

As goals can be made hierarchical (such that one can *override* or *subsume* others), certain constraints may end up being ignored, because more important ones take precedence. A partially-satisfied constraint (for example, a local disruption of line topology) may be acceptable if it occurs for an overriding reason (e.g., to eliminate crowding) and its consequences, if any, are dealt with. In such cases, we speak of constraints being *relaxed*. When this occurs, it should be a deliberate decision rather than inadvertent failure. Therefore, for every constraint which can play a part in a solution must have associated measures that indicate its satisfaction; some of these measures may be truth values, others may be nominal, ordinal, interval or ratio. Measures may also have dimensions (units of measure) such as meters or can be dimensionless quantities, such as probabilities or counts.

Constraints, goals and measures: An example. One basic goal of simplification and elimination is to *avoid imperceptible detail*. The general constraint this goal supports is that of *legibility*, and the scope of the constraint is typically all features in a class (rather than involving combinations of FCs or contextual regions). Determining where the constraint is violated and achieving the goal then involves finding and deleting line segments and areal objects that are too small to be visually resolved, given one's scale, medium, purpose, symbolism and display parameters. These controls may, of course, combine to produce different criteria for imperceptibility in different feature classes. The measures that detect constraint violations (thus determining which features have legibility problems) would likely also be used to determine the effectiveness of the operators that were used to handle them.

What measures or criteria would make good indicators? There are two general cases, linear features and areal features. Linear feature measures can of course be used to characterise boundaries of polygons, but may not always be useful in such contexts. Some examples of measures relating to imperceptibility are:

1. Linear Features (e.g., rivers, roads)

   A. Minimum, maximum, mean, median segment lengths

   B. Distributions (histograms) of segment lengths

   C. Shape complexity measures (curvatures, entropies, fractal dimensions...)

   D. Distributions (histograms) of shape complexity measures

2. Areal Features (e.g., lakes, paved areas, buildings)

   A. Minimum, maximum, mean, median segment lengths

B. Distributions (histograms) of segment lengths

C. Minimum, maximum, mean, median polygon areas

D. Distributions (histograms) of polygon areas

E. Elongation or eccentricity (medial axes, area/perimeter measures...)

Note that areal features can be imperceptible overall even if the segments forming them all are long enough to be perceptible (because of being too narrow). This is where measures such as 2E are useful.

Linear features can be too complex, causing certain details to coalesce at some scales. Examples include river meanders and winding roads, bends of which contract or overlap, and thus become imperceptible. Although measures 1C and 1D given above can detect such situations, they represent a rather different aspect of the perceptibility problem. In fact these are violations of a topological constraint, that of *self-intersection* (when line weight is taken into account). To some degree, getting rid of imperceptible segments may help these situations too, but more sophisticated operators are frequently needed.

One might consider all such measures as characterising special cases of a general phenomenon, which can be termed *proximity*. In the context of imperceptibility, it refers to *nearness to self*. The various types of proximity include nearness to neighbouring vertices, to neighbouring inflections, and to opposite sides of a polygon. Almost all the relevant measures are based on concepts of distance, which can be metric (Euclidean, Hausdorff, etc.) or topological. This generality should not be surprising given the imperatives associated with scale change on graphic media. It implies that the goal of *avoiding imperceptible detail* is a highly general one that can be expressed in many related forms and applies to every feature class. In short, it is a goal to which many agents should be able to relate.

6 Putting Constraints to Work

Generalisation is difficult to automate because it involves transformations to an entire map that must make sense geometrically, topologically and structurally, considered at micro, meso and macro levels. This report has articulated a basis for formalising such transformations using *constraints*. Many constraints flow from *controls* to map production, including properties of data, symbolism and protocols used in producing maps. Other constraints are those of process, in which the effects of earlier treatment decisions and activities influence later ones. How constraints are implemented may strongly depend on the expressivity of data models (i.e., whether and how objects, themes, topology, structure, attributes and time are portrayed in databases). This, however, is beyond the scope of this report, as is the architecture of agent systems that will perform decision-making for generalisation.

Having put forward a taxonometric analysis of constraints to map generalisation, having given examples of specific graphical, thematic and procedural constraints and having begun to formalise them, we are convinced that such a perspective can lead to more robust and holistic solutions to generalisation problems that to date have eluded automation. The analyses presented in this report cover many types and aspects of constraints, across a range of potential implementation complexity. Even intrinsic constraints that involve only a single object (e.g., *avoid self-intersection*) can present challenges to applications. Those that involve extrinsic comparisons (across database states) and backtracking (rolling back unsatisfactory interim solutions) require efficient database access and update mechanisms. When several objects or themes constrain one another, obtaining effective and efficient results will depend on how well data are modelled. For generalisation to succeed, the spatial and semantic qualities and relations of objects need to be expressed.

This is not an easy task for object-oriented techniques to undertake, even when multiple inheritance is possible. While objects can communicate and have elaborate behaviours, they are difficult to organize to act and react in concert. Designing and organising *agents* that seek to satisfy constraints can help to reconcile transformations of map objects within and across feature classes. Creating such extra channels through which spatial, functional and semantic connections between objects can be modelled will overcome the insularity of passive objects (i.e., map features) that float in a void.

The first stage of implementing constraints can be quite reactive; agents can make map objects aware of their neighbours and instruct them to push and pull one another in a tug-of-war in order to give each sufficient breathing space. As the severity of constraint violations increase (as it must in crowded areas when scale is further reduced), a larger repertoire of diagnostics and transformations (such as elimination and typification) must be appealed to, and decisions about where they should be used must be made. Not only the choice of *transformations* (as tasks D1, D2 and D3 will address), but also the order in which they are performed must be decided. As there is rarely a unique or perfect solution in such situations, these choices can result in different selections and configurations of features, the utility of which must be evaluated. This is why *measures* (to be formalised by tasks C1 and C2) are important: they characterize the state of specific aspects of a map, allowing solution states to be compared to initial states and to each other. They also provide feedback about whether constraints have been satisfied, and if not how close to satisfaction they are.

A simple example of relationships between constraints and measures is described below. It is intended to be illustrative rather than prescriptive, and deals primarily with a specific category of constraints and measures (topology). While topological laws (such as requiring all line intersections to occur at nodes) must be respected, *this does not require topological structure of a map to remain static*. Networks must in fact be pruned in the course of generalising them, and doing this changes their "connective shape". If, for example, agents are constrained to minimize changes in shapes of road networks, they need criteria to guide their work. Eliminating links leads to eliminating nodes that are connected to onle one or two others; dropping links tends to increase topological distance (the minimum number of links that must be traversed to move from one node to another), while eliminating nodes tends to decrease it.

Many measures of network shape have been developed. To give an example, a simple one is *average degree*, defined as twice the total number of links divided by the total number of nodes (each node "owns" only half the links incident to it). The higher this quantity, the more densely-connected a network is. Pruning a network can change this parameter, and agents may strive to keep it as constant as possible (thus using it in an extrinsic manner). Thus *average degree* can serve as both a *measure* and a *constraint.*

Because *average degree* is a global parameter, it is hard to use as a tactical decision-making criterion. That is, looking at its value really doesn't help when deciding whether to eliminate a particular link (nodes can be eliminated automatically following link deletion). It is probably best used extrinsically, to compare the current database network state to a previous one, or to evaluate a set of alternative prunings in order to choose the one for which *average degree* differs least from its initial value.

Choosing an optimal pruning according to this measure can be difficult for two reasons:

1. The solution space (possible different prunings) can be very large
2. There is no test of significance for differences in *average degree*

The first problem exists because global solutions must be evaluated; the parameter does not provide guidance in developing them, only in evaluating them. The second problem means that small changes in *average degree* may or may not indicate a significant shift from the initial state. One way to restrict the solution space is to rank *links* according to some measure of their importance, and use this parameter as a guide to eliminating them. For example, for each link one could compute *its total degree* (*TD*) as the sum of the degrees (number of incident links) of each of its two end nodes, minus one (because the link is counted twice). Links with the smallest values of *total degree* would be eliminated first in the course of pruning the network. That is, for each link, compute *TD* as

,

where *j* sums the *n* incident links for each end node, summed over the link's two nodes. Global values of *TD* can be accumulated for the whole network, and as it is pruned this sum is adjusted. The strategy of maximizing the global average of *total degree* can be made a goal of agents. This would tend to eliminate the more poorly-connected roads before the better-connected ones, which is a common generalisation strategy. Links with *TD* equal to 1 are isolated segments, which normally should not occur in road themes. Links with *TD* values of 2 are dead ends, and those with *TD* of 3 are topologically redundant. The tactic of eliminating the lowest-valued links first can help in guiding pruning, but it is insufficient, as many links will have equal *total degree* values, with the modal value being 5 (or possibly 7 in urban areas). Thus the problem is still underconstrained, such that additional constraints should be brought into play.

One way to incorporate other constraints would be to consult certain attribute values for road links, if available. These might include class of road, number of lanes, width, length, volume of traffic, etc; there is no reason why all criteria for eliminating links must be topological, nor do they need to have integer values. This can be done by normalizing scalar link attributes and using them as multiplicative weights for *TD*. Elimination can then proceed as before, using the weighted values.

One might think that eliminating links with low values of *TD* would cause the *average degree* of the network to increase. Note that it decreases when any link is eliminated. Whenever a node is eliminated (because it is a cul-de-sac of degree 1 or it becomes a simple junction of degree 2), *average degree* can increase. However, in a well-connected network more links than nodes will be eliminated and therefore eliminating poorly-connected links first will have the least effect on *average degree*.

Yet a further constraint can be added in order to override the above procedure where necessary. Certain minor road links that would normally be considered good candidates for deletion may in fact have special importance; for example, they may be the only route that crosses a river or a mountain range, or provide the only access to a town, a park or a significant building. As such information is rarely directly given in a database, it must be inserted, either manually or through some sort of multi-theme analysis, in order to be available to agents that perform network pruning. While agents could be built that perform such analyses on the fly, this might be an inefficient approach, given that such constraints are so easy to store as attributes of links and other objects they involve (and in a *Gothic* database would probably be specified as *references*).

The simple examples of constraints and measures described above are intended to be heuristic rather than definitive. They serve to point out how constraints may be quantified and some roles they can play in achieving generalisation solutions. Relationships between constraints and measures will be studied in task C1 and subsequently in C2, This will lead, in tasks D1-D3, to the incorporation of constraints within transformation algorithms. The design of data models (in tasks A4 and A5) and of agents and their organization (tasks B1 a,nd B2) will also be performed against a backdrop of constraints and the measures needed to formalise and evaluate them. In this way

References

Bader, M. and Weibel, R. (1997): Detecting and Resolving Size and Proximity Conflicts in the Generalization of Polygonal Maps. *Proceedings 18th International Cartographic Conference,* Stockholm (S), 1525-1532.

Beard, M.K. (1991): *Constraints on Rule Formation*. In: Buttenfield, B.P. and McMaster, R.B. (eds.). Map Generalization: Making Rules for Knowledge Representation. London: Longman, 121-135.

Brazile, F. (1998): A Generalization machine that incorporates quality assessment. *Proc. SDH98* (8th Int. Symp. on Spatial Data Handling, Vancouver B.C., July 1998), 308-320.

Cobb, M.A., Chung, M.J., Foley, H., Petry, F.E. and Shaw, K.B. (1998): A rule-based approach for conflation of attribute data,. *GeoInformatica* 2:1, 7-36.

Van Hentenryck, P. et al. (1996): *Strategic Directions in Constraint Programming*. ACM Computing Surveys, 28(4): 701-726.

Institut Géographique National France (1996). *Agent: Automated Generalisation New Technology Technical Annex*. ESPRIT Project 24 939 LTR Reactive Project. 68 p.

Kinny, D., Georgeff, M. and Rao, A. (1996). *A methodology and modelling technique for systems of BDI agents*. Proc. 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World. Lecture Notes in Artificial Intelligence, vol. 1038: 890,

Mackaness, W.A. (1995). A Constraint Based Approach to Human Computer Interaction in Automated Cartography. *17th Int. Cartographic Conference,* Barcelona (E): 1423-1432.

McMaster, R.B. and Shea, K.S. (1992): *Generalization in Digital Cartography*. Washington D.C.: Association of American Geographers, 134 p.

Morisset and Ruas 1(997): Simulation and agent modelling for road selection in generalisation. *Proceedings 18th Int. Cartographic Conference,* Stockholm (S), v. 3: 1376-1380.

Naroll, R.S. and von Bertalanffy, L. (1956): The principle of allometry in biology and the social sciences. *General Systems Yearbook*, v. 1. 76-89.

Regnauld, N. (1997): Recognition of building clusters for generalization. In: Kraak, M.J. and Molenaar, M. (eds.): *Advances in GIS Research II* (7th International Symposium on Spatial Data Handling). London: Taylor & Francis, 185-198.

Ruas, A, (1995): Multiple Representations and Generalization. *Course notes, Scandinavian Cartography Seminar,* August 1995, 37 pgs.

Ruas, A. (1998): O-O-constraint modelling to automate urban generalisation process. *Proc SDH98* (8th Int. Symp. on Spatial Data Handling, Vancouver B.C., July 1998), 225-235.

Ruas, A. and Plazanet, C. (1997): Strategies for Automated Map Generalization. In: Kraak, M.J. and Molenaar, M. (eds.): *Advances in GIS Research II* (7th Int. Symposium on Spatial Data Handling). London: Taylor & Francis, 319-336.

Ruas, A. and Mackaness, W.A. (1997): Strategies for Urban Map Generalisation. *Proceedings 18th Int. Cartographic Conference,* Stockholm (S), v. 3: 1387-1394.

Saalfeld, A. (1998): Sorting spatial data, for sampling and other geographic applications. *GeoInformatica* 2:1, 7-36.

Töpfer, F. and Pillewizer, W. (1966): The Principle of Selection. *The Cartographic Journal* **3**: 10-16.

Töpfer, F. (1974): *Kartographische Generalisierung.* Leipzig: VEB H. Haack,336 p.

Weibel, R. (1997): *A Typology of Constraints to Line Simplification*. In: Kraak, M.J. and Molenaar, M. (eds.): Advances in GIS Research II (7th International Symposium on Spatial Data Handling). London: Taylor & Francis, 533-546.

Weibel, R. and Dutton, G. (1998): Constraint-based Automated Map Generalization. *Proc SDH98* (8th Int. Symp. on Spatial Data Handling, Vancouver B.C., July 1998), 214-224.

Ware, J.M. and Jones, C.B. (1997): A Spatial Model for Detecting (and Resolving) Conflict Caused by Scale Reduction. In: Kraak, M.J. and Molenaar, M. (eds.): *Advances in GIS Research II* (7th Int. Symposium on Spatial Data Handling). London: Taylor & Francis, 547-558.

Appendix A

Glossary of Terms (Working Definitions)

Constraint: A limitation reducing the number of acceptable solutions to a problem

Control: An external physical, technical, logical or institutional precondition that may participate in creating one or more constraints

Cultural: (re constraint); aspects of map semiotics that reflect the historical development of the portrayed landscape and the conceits and assumptions of the mapmaker

Feature: An object identified on a map, consisting of one or more geometric element. Features can share map space and geometric data, and have topological and other relationships

Feature Class: An abstract category used in defining the nature and scope of related geographic phenomena for mapping; also sometimes called a *layer*.

Gestalt (re constraint); a form or configuration having properties that cannot be derived by the summation of its component parts

Goal: A specific positive or negative criterion that can cause agents to act on their environment or interact with other agents to solve problems

Gothic: Laser-Scan's object-oriented database technology on which a variety of different software applications have been developed. Besides being object-oriented, the Gothic database is versioned, topologically structured and spatially indexed

Graphic: (re constraint); characterizing size, shape, proximity and detail of map features as they are rendered

Link: A *segment* or *polyline* that connects to other *segments* or *polylines* at either one or two *nodes*; an edge in a graph

Macro: Characterizing regional or overall map space; related to scope of semantic constraints

Meso: Characterizing a neighborhood treated as a unit on a map, usually involving several to dozens of features of various types; related to scope of structural constraints

Micro: Characterizing a neighborhood defined by a feature, part of one or several that can be treated in isolation; related to scope of geometric and graphic constraints

Network: A set of *links* that connect only via *nodes*

Node: The topological role of a *vertex* in a graph

Point: A location in (2-dimensional) space that represents an object

Polyline: A linear sequence of *vertices* that intersects itself nowhere

Polygon: A linear sequence of *vertices* that forms a simple loop

*** Property:** An attribute that characterises some aspect or behavior of an object

Partition: A data-defined polygonal area or a cell of an arbitrary tessellation, used for divide-and-conquer processing; a linear map element that is used to create a partition

Region: A portion of the earth's surface that can be considered as a coherent whole due to its physiographic, environmental or cultural properties

*** Segment:** Two *vertices* and a straight-line connection between them

Semantic: (re constraint); logical and/or semiotic relationships between symbols and that which they denote

Structural: (re constraint); characterizing shapes of features and spatial arrangements among them that can express both geometric and semantic relationships

Topological: (re constraint); properties of geometric forms, such as adjacency, containment and connectivity, that are invariant under transformations such as bending and stretching

Vertex: Any member of an ordered sequence of *points*

***** Has different, specific technical meaning in Laser-Scan Lexicon