Modelling Spatial Data Integrity Rules at the Metadata Level

Dr Sophie Cockcroft
School of Commerce, University of Queensland
s.cockcroft@commerce.uq.edu.au

Abstract. Data quality in populated spatial databases has been of great concern over the last decade. Much work has gone into means of incorporating spatial "business rules" in GIS with the ultimate aim of reducing errors in data entry, or detecting them after they are made. This paper reviews means by which spatial integrity rules have been modeled in the GIS literature, and with the aid of a case study described one approach to modeling and using integrity rules in practice.

1. INTRODUCTION

The results of spatial data analysis depend on the quality of data in the populated database on which the analysis is done. Sometimes not enough is known about the data to be able to assess its quality. Often there are known problems relating to the quality of the data which cannot be automatically rectified. Modern GIS are often used for spatial reasoning. Errors that are not apparent when using a GIS for visualisation become more critical when used in this way. Many authors have commented on data quality issues in GIS. Thorough reviews of data quality problems were given by (Hunter, 1992) and (Kainz, 1995). More recently positional accuracy was explored by Goodchild and Hunter (1997). Other issues include attribute accuracy, logical completeness and consistency (Servigne et al., 2000). An accepted approach to amending errors is to develop data integrity rules and apply these either at data entry or after the event in order to "clean" datasets.

2. CLASSIFICATION OF INTEGRITY CONSTRAINTS

The issue of spatial data quality in relation to integrity constraints has received much recent attention [(Servigne et al., 2000), (Medeiros and Magalhaes, 1993), (Medeiros and Cilia, 1995), (Medeiros and Andrade, 1994), (Medeiros and Pires, 1994), (Cockcroft, 1997). Definitions and classifications of the various types of integrity constraints that pertain to spatial systems were given in (Cockcroft, 1997) and (Servigne et al., 2000). ***Topological constraints*** can be subdivided into those that apply to *structural errors*, *geometric errors.* Further, there are *topo-semantic* constraints (Servigne et al., 2000). These address errors that relating to the meaning behind topological objects; Servigne et al (Servigne et al., 2000) defined these as topo-semantic, a subset of topological. Cockcroft (1997) further subdivided them into *semantic constraints and user defined integrity constraints*. The former relate to the nature of objects – e.g. water is wet and walls are solid, therefore roads should not run through lakes or buildings. The latter to those of a more esoteric nature defined by humans e.g. a nuclear power station should not be within a certain radius of a power station. The latter type of constraints is often described in non-spatial literature as business rules. An interesting subset of Topo-semantic rules, which was not explicitly defined by servigne et al, is topological rules based on attribute values, for example; "A butterfly valve may not intersect a pipe if the diameter of the pipe is greater than 40'". The work of Servigne et al (2000) focuses on identifying and correcting mainly topological errors within a runtime environment. A key feature of their work is that errors in existing systems are detected and, where possible, corrected automatically.

3. TOPOLOGICAL MODEL

Binary topological relationships were rigorously defined by Egenhofer and Herring, (1990) they are briefly introduced here, because they form the basis for work in this area. In the binary topological model relations between two objects A and B, which can be points lines or polygons, are defined in terms of sixteen intersections. The interiors of A and B could intersect, the boundaries could intersect or the boundary of A could intersect with the interior of B or vice versa. This gives four possibilities and sixteen possible combinations. It should be noted that the types of constraints addressed in this work are only those based on binary topological relationships in vector based systems.

Relationship	$\delta\delta \; ^{oo} \; \delta o \; ^{o}\delta$	Relationship	$\delta\delta \; ^{oo} \; \delta o \; ^{o}\delta$
r_0	0 0 0 0	r_8	0 0 0 1
r_1	1 0 0 0	r_9	1 0 0 1
r_2	0 1 0 0	r_{10}	0 1 0 1
r_3	1 1 0 0	r_{11}	1 1 0 1
r_4	0 0 1 0	r_{12}	0 0 1 1
r_5	1 0 1 0	r_{13}	1 0 1 1
r_6	0 1 1 0	r_{14}	0 1 1 1
r_7	1 1 1 0	r_{15}	1 1 1 1

key:

$\delta\delta$ - Boundaries Intersect
^{oo} - Interiors Intersect
δ^{o} - Boundary of first intersects with interior of second

$^{o}\delta$ - Boundary of second intersects with interior of first

4. APPROACHES TO DEFINITION AND IMPLEMENTATION OF INTEGRITY CONSTRAINTS

4.1 Logical level

Hadzilacos and Tryfona (1992) gave a complete logical data model for geographical applications which included a thorough definition of constraints.

The constraint definition given below is from (Hadzilacos and Tryfona, 1992) and illustrates a constraint to find the ideal position for a recreational park i.e. near to a forest but not within it. The motivation for development of this language was not to improve data quality. The constraints in question were applied on existing data to constrain the data to a given set of user requirements. The notation r_1 - r_{16} refers to the sixteen combinations described above.

r_1: Common border no common interior
r_6: A is inside B with no common interior
r_7: A is inside B sharing common borders but they are not equal
r_{15}: A partially overlaps B

DEFINE OBJECT CLASS OBJ_PARK
CARTOGRAPHIC TYPE REGION
ON LAYER PARK
CONSTRAINT neigbouring_with_forests AND not_inside_forests

DEFINE OBJECT CLASS OBJ_PARK_SITES
CARTOGRAPHIC TYPE REGION
ON LAYER PARK

DEFINE CONSTRAINT neigbouring_with_forests ON OBJ_PARK_SITES AMONG OBJ_PARK_SITES, OBJ_FOREST AS

$\{r_1(p,f) \vee r_6(f1,p1) \vee r_7(f1,p1) \vee r_6(p1,f1) \vee r_7(p1,f1) \vee r_{15}(f1,p1) |\ P \in \text{OBJ\_PARK\_SITES} \wedge f1 \in \text{OBJ\_FOREST} \wedge P1 \in \text{OBJECT\_PARK\_SITES} \wedge \text{P1.PARK\_SITES=TRUE} \wedge \text{F1.FOREST\_ID=NULL}\}$

Hadzilacos et al described their data modeling approach, as automatable. They did not give details of how the user would interact with the system.

4.2 Implementation level

To define constraints in existing GIS it is necessary to have some knowledge of a proprietary scripting language. The example in Figure 1 is an extract from the Smallworld language Majik, which defines a trigger to control the behaviour of a road object. In this way topo-semantic user defined constraints can be applied to data entry.

```
    _method
road.road_centre_line_trigger(feature,op,geometry)
     ##
     ## method to update length field of road object
     ##
      # test if operation is :detach or centre_line is
unset,
     _if  (c << _self.centreline) _is _unset _or
        op = :detach
     _then
      # if so set length to unset
      _self.length << _unset
     _else
        # otherwise, get length of chain in database
units (i.e. mm)
      # and convert to km
      _self.length << c.line_length/1000000
     _endif
    _endmethod
```

Figure 1 Smallworld Magik code

This is not very accessible to end -users. The approach to constraint specification put forward by (Servigne et al., 2000) was to define rules in an english like fashion. The basic algorithm used was

1. Choose a first class of entities
2. Choose a second class of entities
3. Choose a relation among the list proposed
4. Define the specification.

The output would be a rule in the form

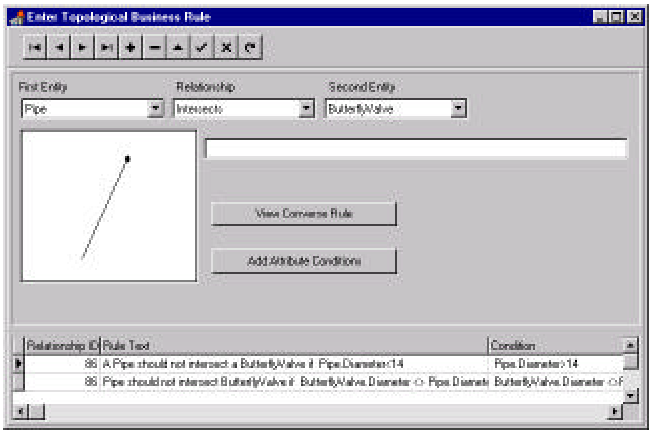
(Road, Inside, Building, Forbidden)

A unique feature of their constraint specification technique is that the user was also able to specify that a relationship should occur *at least n times, at most n times or exactly n times.*

(Servigne et al., 2000) suggested that the approach of Hadzilacos et al . (1996).was not suited to end users (i.e. domain experts with little training in computer science) since it required the formation of sentences in set notation. They offered an approach to simplify the specification of semantic rules for the user by means of a user interface with drop down lists. The interface approach put forward by (Servigne et al., 2000) has also been adopted here.

5. SPECIFYING RULES AT THE METADATA LEVEL

What differs about the work presented here is that rules can also be specified in terms of their attribute values, as described in the pipe and butterfly example above, and illustrated in Figure 2. Additionally, all entities, attribute and the rules thereon are stored in a metadatabase or repository. This repository is standalone but can be coupled with any GIS. The data concerning relationships and rules is stored in a structure illustrated by the entity relationship in Figure 4. It is updated via a graphical user interface as illustrated in Figure 2. The user interface design follows that of (Servigne et al., 2000), the logic of specifying the rules is intentionally hidden from the user. The full development and design of the repository has been described elsewhere (Cockcroft, 1999). It is a standalone software tool that integrated with GIS to manage rules and metadata. Figure 3 illustrates the processing of a rule. After it is stored in the repository, it is applied whenever the project to which it related is opened (see Figure 4)


Enter Topological Business Rule

First Entity: Pipe
Relationship: Intersects
Second Entity: ButterflyValve

Relationship ID	Rule Text	Condition
86	A Pipe should not intersect a ButterflyValve if Pipe.Diameter<14	Pipe.Diameter>14
86	Pipe should not intersect ButterflyValve if ButterflyValve.Diameter <> Pipe.Diamete	ButterflyValve.Diameter <>F

Figure 2 Topo-semantic rule entry

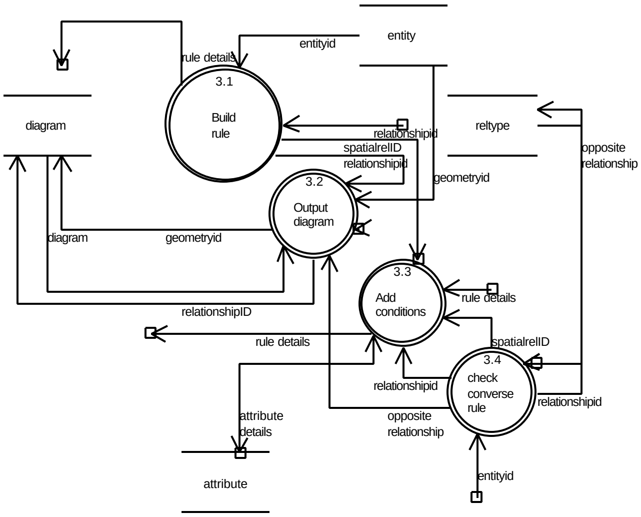


Figure 3 Processing of rule

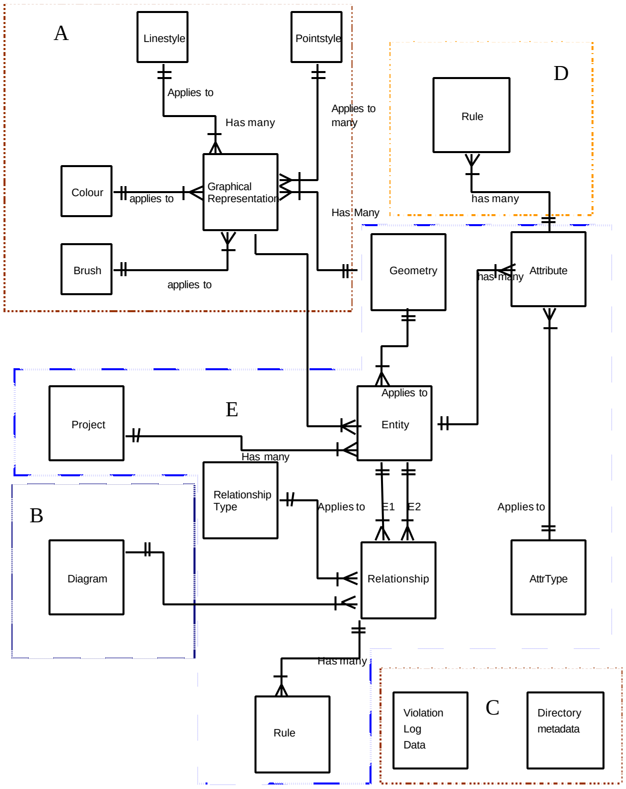


Figure 4 Entity relationship diagram of data stored in repository

With reference to section 3 the relationships handled in this prototype are

- Contains
- Does not contain
- Contains Entire
- Intersects
- Entirely Within
- Is not within
- Within
- Is not entirely within
- Does not intersect
- Does not contain entire

These spatial relations are those supplied by the GIS used in developing the demonstration prototype, they are extensible to support other GIS, but the redefinition of algorithms for detecting intersection is beyond the scope of this work. Rules are interpreted by parsing the rule, relationship, attribute relations.

Area A of the ERD in Figure 4 is concerned with how objects are rendered in the GIS. Area B concerns the diagram entity which is internal to the repository and stores the spatial relationship diagram as illustrated in the large white box in Figure 2. Area C depicts tables storing directory metadata returned to the repository from the GIS. Area D illustrates the conceptual model of how attribute rules are stored. Area E contains the core information for representing an entity and its relationships.

5.1 Case Study

The aim of this chapter is to illustrate how the repository described in section 2 manages a number of the constraints identified in the taxonomy described in (Cockcroft, 1997), and outlined in section2. Section 2.2 provides some background to the case study.

5.2 Stewart island case study

The following sections describe a scenario with respect to the placing of hunting areas in Stewart Island. Of particular interest in this case study is how the system handles polygon manipulation.

Stewart Island, in New Zealand, provides excellent opportunities for hunting whitetail deer, red deer, and possums (DoC, 1995). Permits are required for all hunting on Stewart Island and must be applied for in advance. For the purposes of hunting, the island is divided into a number of "blocks". Each hunting party may apply for only one block. Permits are issued for a maximum period of 10 days and bookings for blocks can be made up to 12 months in advance. Confirmation of booking is required 60 days prior to the hunters' arrival on the island. Failure to do this will result in cancellation of the booking. Permits can be posted, or collected from the New Zealand Department of Conservation (DoC) office in Halfmoon Bay prior to a visit. Most blocks are suitable for 6 hunters. Advice on camp locations is available from the DoC office. Permits for hunting on Maori land are required, and are
obtainable from Rakiura Maori Land Inc.

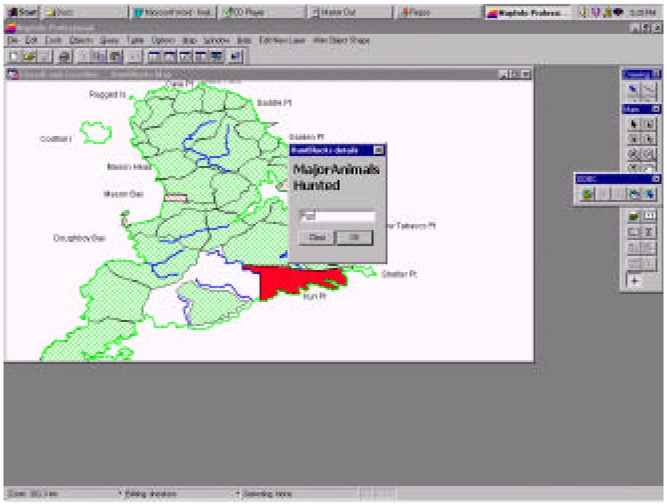
Hunting blocks are positioned in such a way that identifiable natural features border them. There are certain areas of Stewart Island that are neither under the control of the Department of Conservation nor are they available for hunting for other reasons. From time to time it would be useful for DoC to have a facility to alter the configuration of the hunting blocks. If they were stored in a system integrated with the repository described in this work, it is contended that such a process would be easier than making the alterations by hand. There are a number of constraints governing the positioning and use of hunting blocks:

1. Animals hunted on Stewart island will be one of the following: Whitetail Deer, Red Deer or Possums.
2. A hunting block may not intersect Maori Land unless a permit has been obtained from Rakiura Maori Land Inc.[1]
3. Hunting blocks, Closed Areas, Maori Land Huts and all other land-based objects must be entirely within a coastline.
4. A hunting block may not contain more than 6 hunters.

The first constraint can be implemented by means of domains. It is a *static semantic* constraint, being based on the reality of livestock on Stewart Island and not the users' decision. The second constraint is a topological constraint based on attribute values. Unlike the pipe diameter example, however, this is a constraint strictly from the user domain. In terms of the taxonomy given in chapter 2, it is a *user constraint*. The existence of constraint 3 restricts the objects that can be topographically connected to a coastline. It is a *static semantic constraint* in the format of a domain. Constraint 4 is an attribute constraint. Again, it is in the nature of a user constraint - the user has determined this to be the case, for safety or other reasons.

5.2.1. Animals hunted on Stewart island will be one of the following: Whitetail Deer, Red Deer or Possums

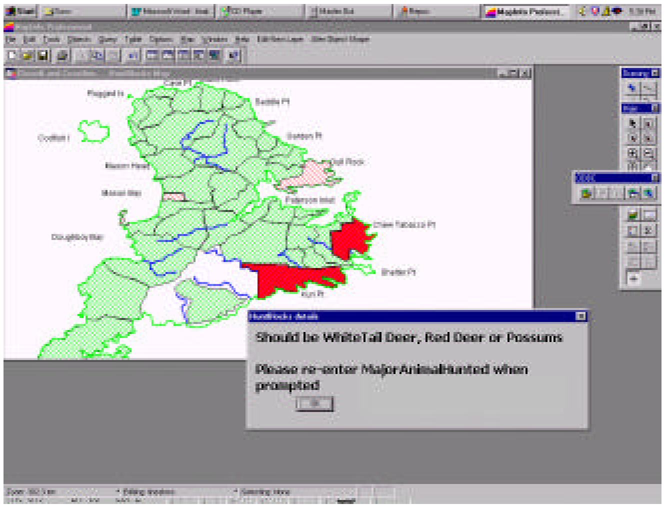
The results of incorrect data entry are shown in Figure 5 and Figure 6.



MajorAnimals Hunted

Pol

Figure 5 Entering an incorrect attribute for constraint 1



Should be WhiteTail Deer, Red Deer or Possums

Please re-enter MajorAnimalHunted when prompted

Figure 6 Resulting error message

[1] This rule differs somewhat from the reality of the situation in that permits are issued to hunters and do not apply to hunting blocks. However, it is designed to illustrate a spatial constraint based on attribute values and since hunters are not illustrated on the map, hunting blocks are used as a proxy

5.2.2. A hunting block may not intersect Maori land unless a permit has been obtained from Rakiura Maori Land Inc.

This section shows a hunting block extending into Maori land in Figure 7 and the consequences of this action in Figure 8

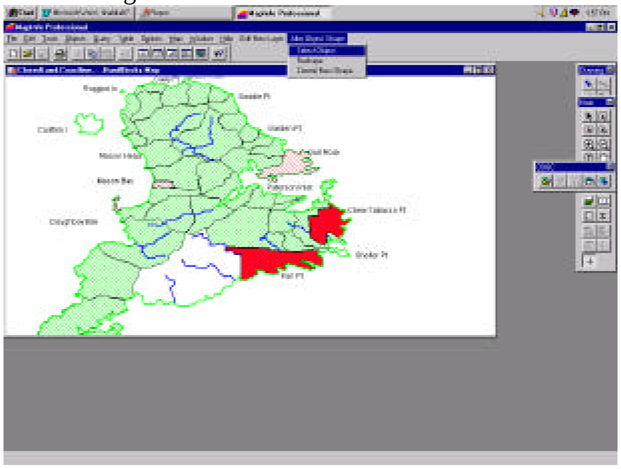
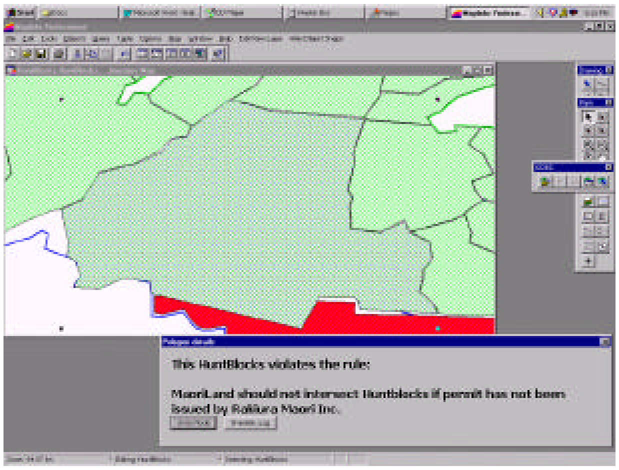


Figure 7 Selecting a hunting block in order to change its shape



This HuntBlocks violates the rule:

MaoriLand should not intersect Huntblocks if permit has not been issued by Rakiura Maori Inc.

Figure 8 Violation of rule concerning intersection with Maori land

5.2.3. Hunting blocks, closed areas, Maori land, huts and all other land-based objects must be entirely within a coastline

In order to illustrate what happens when a land-based object is placed outside a coastline, a hunting block is extended into the sea in the following example (see Figure 9). In this case, when presented with the dialog box the user would select Undo Node (Figure 10) to rollback the object to its former state.

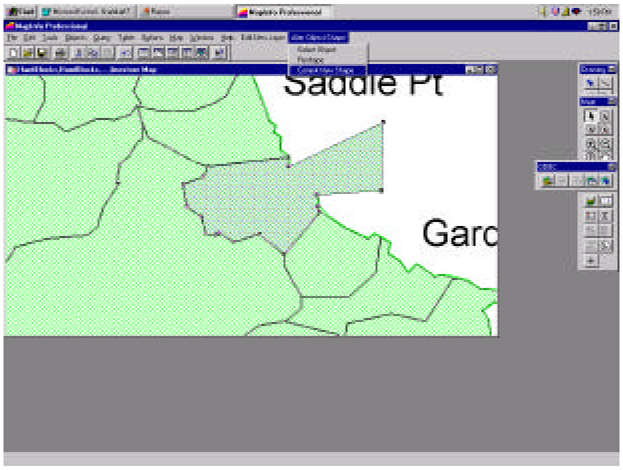
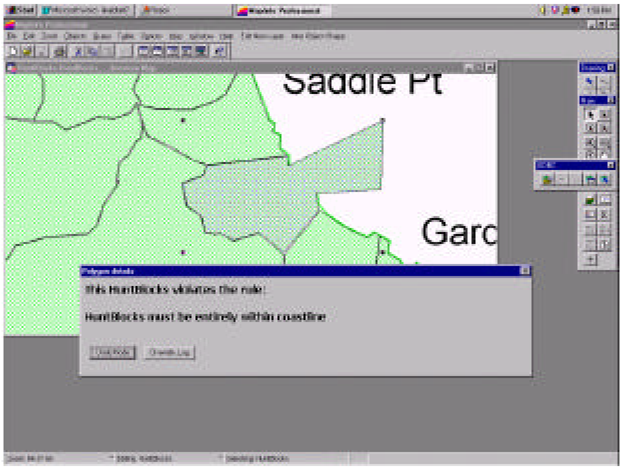


Figure 9 Shape of hunting block is changed



This HuntBlocks violates the rule:

HuntBlocks must be entirely within coastline

Figure 10 Error message generated from violation of constraint 3

5.2.4. A hunting block may not contain more than six hunters

This section shows, in Figure 11 and Figure 12, the error observed on entering a number of hunters greater than 6.

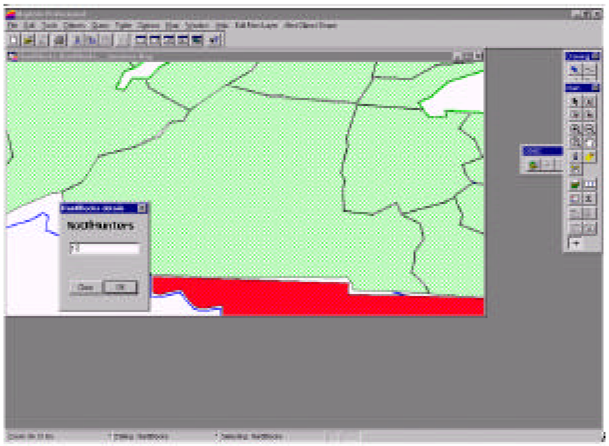
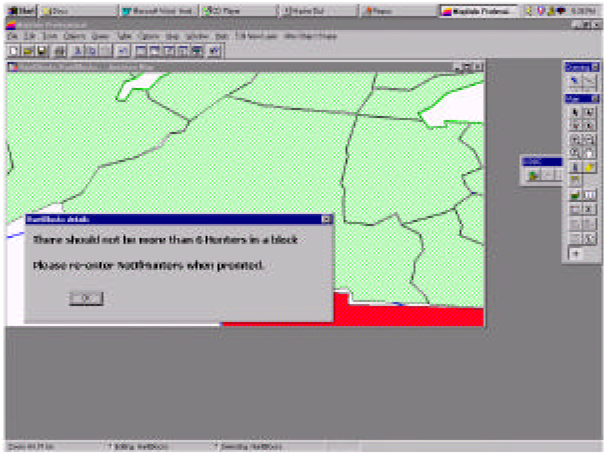


Figure 11 Entering an incorrect number of hunters



There should not be more than 6 Hunters in a block

Please re-enter NoOfHunters when prompted.

Figure 12 Resulting error message

6. CONCLUSION

This paper has briefly set in context the problem of defining topological constraints based on semantic information. One approach is illustrated by means of a case study.

REFERENCES

Cockcroft, S. (1997) A taxonomy of spatial data integrity constraints, *Geoinformatica,* 1**,** 327-343.

Cockcroft, S. (1999) "Spatial Data Management in an Integrated Software Engineering Environment" in *Information Science* Otago, Dunedin.

DoC (1995) "Stewart Island Hunting Information", Department of Conservation, Te Papa Atawhai, New Zealandq.

Egenhofer, M. J. and Herring, J. R. (1990) "A Mathematical Framework for the Definition of Topological Relationships" in *4th International Symposium on Spatial Data Handling*, Vol. 2 (Eds, Brassel, K. and Kishimoto, H.), IGU Commission on Spatial Data Hnadling, Zurich, 803-813.

Goodchild, M. and Hunter, G. (1997) A simple positional Accuracy Measure for Linear Features, *International Journal of Geographical Information Science,* 11**,** 299-306.

Hadzilacos, T. and Tryfona, N. (1992) A Model for Expressing Topological Integrity Constraints in Geographic Databases, *Lecture Notes in Computer Science,* 639**,** 252-268.

Hadzilacos, T. and Tryfona, N. (1996) Logical data modelling for geographic applications, *International Journal of Geographical Information Systems,* 10**,** 179-200.

Hunter, G. a. M. B. (1992) Understanding Error in Spatial Databases, *The Australian Surveyor,* 37**,** 108-119.

Kainz (1995) In *Elements of Spatial Data Quality*(Ed, Guptill) Elsevier, Oxford.

Medeiros, C. and Magalhaes (1993) "Rule application in GIS - a case study", UNICAMP.

Medeiros, C. and Pires, F. (1994) Databases for GIS, *ACM SIGMOD record,* 23**,** 107-115.

Medeiros, C. B. and Andrade, M. J. (1994) Implementing Integrity Control in Active Data Bases, *Journal of Systems and Software,* 27**,** 171-181.

Medeiros, C. B. and Cilia, M. (1995) "Maintenance of binary topological constraints through active databases" in *3rd ACM International Workshop on Advances in Geographic Information Systems*(Ed, Bergougnoux, P.), Baltimore, Maryland USA, 127-133.

Servigne, S., Ubeda, T., Puricelli, A. and Laurini, R. (2000) A Methodology for Spatial Consistency Improvement of Geographic Databases, *Geoinformatica,* 4**,** 7-34.