Modelling geometric rules in object based models: an XML / GML approach.

Trevor Reeves[†‡], Dr Dan Cornford[†], Dr Michal Konecny[†], Dr Jeremy Ellis[‡]

[†]Knowledge Engineering Group, School of Engineering and Applied Science, Aston University, Birmingham B4 7ET, UK

[‡]Key Traffic Systems Ltd, Ardencroft Court, Ardens Grafton, Alcester, Warwickshire, B49 6DP, UK

Abstract

Most object based approaches to Geographical Information Systems (GIS) have concentrated on the representation of geometric properties of objects in terms of fixed geometry. In our road traffic marking application domain we have a requirement to represent the static locations of the road markings but also enforce the associated regulations which are typically geometric in nature. For example a give way line of a pedestrian crossing in the UK must be within 1100-3000 mm of the edge of the crossing pattern. In previous studies of the application of spatial rules (often called 'business logic') in GIS emphasis has been placed on the representation of topological constraints and data integrity checks. There is very little GIS literature that describes models for geometric rules, although there are some examples in the Computer Aided Design (CAD) literature. This paper introduces some of the ideas from so called variational CAD models to the GIS application domain, and extends these using a Geography Markup Language (GML) based representation. In our application we have an additional requirement; the geometric rules are often changed and vary from country to country so should be represented in a flexible manner. In this paper we describe an elegant solution to the representation of geometric rules, such as requiring lines to be offset from other objects. The method uses a feature-property model embraced in GML 3.1 and extends the possible relationships in feature collections to permit the application of pa-

rameterised geometric constraints to sub features. We show the parametric rule model we have developed and discuss the advantage of using simple parametric expressions in the rule base. We discuss the possibilities and limitations of our approach and relate our data model to GML 3.1.

Introduction

The use of object based modelling frameworks is well established in GIS (Worboys and Hearnshaw 1990). However, present implementations are largely based on static representations of the application domain model and do not address the relationship between different spatial objects, other than topological constraints. In the application for which we are developing the object based model it is necessary to impose and check constraints (which we will call rules) relating to the geometric relationship between features. For example we need to be able to constrain one straight line to be within a given orthogonal offset range of another line. An additional problem is that we require that the offset range might vary according to certain external parameters, and we show how this form of constraint can be included in the model. In this paper we show the approach we adopt to solving this problem in an object based framework. In particular we show the underlying object based model and show how this can be mapped to an XML schema, and how the schema can be used to store real features that we use in our application domain.

In the following section we review previous approaches to rule representation within GIS systems. We then describe the application domain, road marking modelling, which motivates the solution we have developed. We go on to illustrate the object based model we have created and show how rules, and in particular parameterised rules can be represented within this. To ensure that the rules are flexible and simple to update or extend we next illustrate their implementation using XML schema and show how this relates to the object based models. We look at how our approach relates to GML 3.1, and conclude with suggestions as to how the work might be extended.

Rule representation in spatial data

Methods for modelling and enforcing geometric constraints (rules) between spatial entities currently exist i.e. restrictions and relationships per-

taining to the positional attributes of geometric shapes. Since there is a discernible difference in the way these rules are handled in the CAD and GIS environments, both of these domains are reviewed.

GIS Approach

Work in the GIS field has previously focused on the semantic integrity of data sets, to ensure the logical correctness of geographical data. The uniqueness of integrity constraints for spatial data is identified in Cockcroft (1996), where *spatial integrity constraints* are introduced under the concept of 'business rules' for spatial application domains. In non-spatial application domains, business rules are identified to preserve the integrity of the logical data model; this is no different for spatial application domains, only that spatial business rules refer to the spatial properties of the data or enforce spatial relations between data.

The importance of spatial data integrity is addressed in Borges et al. (1999), where a method for specifying 'integrity rules' within the Object Modelling Technique – G (OMT- G), a geographic applications extension to OMT, at an early stage within the database design sequence is suggested (Borges 1997). It is suggested, as in Cockcroft (1998), that the integrity rules must be enforced at data entry and update; this ensures the integrity of any state of the database. The integrity rules or constraints are essentially enforced by ensuring that certain spatial relations are present between spatial entities within the database at any given time.

The modelling of topological relationships for complex spatial objects are described by Price et al. (2001), which builds on earlier work (Price et al. 2000). They show, at the conceptual level, how the different topological relationships can be defined as constraints to be imposed on spatial entities that comprise a higher composite spatial entity. That is, when modelling a spatial phenomenon that can be described as the composition of a number of other individual spatial phenomena, to ensure the integrity of the data, the composition relationships between the spatial parts and the spatial whole may be required to meet some topological relationship condition. An example of where these *part-whole* relationship constraints could be imposed would be to ensure that the total phone service coverage area contains or equals, the geometric union of the spatial extents of the individual phone service cells which comprise it.

Spatial integrity constraints are classified in Cockcroft (1997) as: *topological*, *semantic*, and *user*.

− The topological constraints imposed are based on the topological relationships; Contains, Does not contain, Contains Entire, Intersects, Entirely within, Is not within, Within, Is not entirely within, Does not intersect, Does not contain entire. The topological relationships employed by Price et al. (2001) are based on those specified by Egenhofer and Herring (1991) and Clementini and Felice (1994), and are given as; Boundary-Overlap, 4 Interior-Overlap, Mixed-Overlap, Disjoint-Separate, Disjoint-Interpenetrating, Contains, Equals, Inside.

− Semantic constraints are concerned with the meaning of geographical features, and apply to the properties of the geographical objects that need to be stored.

− User integrity constraints are much more specific to the application domain and are not necessarily based on semantics.

Different methods for realising the inclusion of integrity constraints in spatial databases exist, these methods include; constraints as queries, schema based approaches, object-oriented approaches and the business rule approach. Cockcroft (1998) and Cockcroft (2001) go on to specify the means for implementing a spatial business rules system; rules are stored in their own 'rule repository', separate from the data itself. The business rules, or integrity constraints are then enforced at data entry, and each time an update occurs to the data. The integrity rules are stored at the metadata level; that is they are defined for different types of geographical features, not individually for each instance. The system outlined in Cockcroft (2001) provides a means for the application domain modeller to define the integrity constraints on data types present in a spatial database.

CAD Approach

Parametric models within many CAD systems can be used to enforce geometric constraints between a number of different geometric objects. Parametrisation in CAD models is defined by Pratt (1998) as:

'*the association of named variables, or expressions involving named variables, with certain quantities in a model. In a shape model these quantities are usually dimensions.*'

Pierra et al. (1994) describe a 'parametric instance', i.e. an instance of a feature represented in a parametric model, to consist of a set of parameters (potentially either numeric or Boolean), a set of geometric items (points, curves etc), and for each one of these primitives, a function that is able to compute it from the parameter values and the other geometric items in the model.

These functions are called parametric functions and can contain four main constructs :

- Constraint-Based Definitions: these are typically spatial relationships which describe one geometric item as some form of constraint between other geometric items. Geometric constraints are used in many parametric CAD models to control the behaviour of shape elements in a design, included for the improvement of design functionality (Pratt 1998). The geometric constraints are in the form of explicit spatial relationships, for example a perpendicularity constraint between two planar faces, or a tangency constraint between a line and a circle (Pratt 1998).

- Numeric and Boolean Valued Expressions: these describe numerical or Boolean logic based relationships between parameters in the model and properties of the geometric items in the model.

- Grapho-Numeric Expressions: these allow geometric items within the model to be used as arguments to functions/operators within parametric functions; e.g. distanceBetween(point1, point2).

- Constraint-Based virtual definitions: e.g. projectionOf point1 onto line2.

The explicit geometric constraints included in the parametric functions are very different to the topological constraints that are imposed in much of the GIS literature to maintain spatial integrity.

The term variational is used to denote the type of model that exhibits both parametrisation and geometric constraints (Pratt 1998). Pratt (1998) and Pierra et al. (1994) outline the two methods that currently exist for the representation of variational models:

- Explicit Models: Parameters are associated with dimensional elements in the model, and constraints are explicitly specified between particular elements such as faces or edges.

- Implicit or History-Based Models: The primary representation of the model is in terms of the sequence of operations used to construct it. In this case there is no explicit information about the model shape at all; that information does not become available until after the specified operations have been performed, the result of this being an explicit model.

Our work shares some scope and motivation with the GIS approach to the application of 'business rules' in spatial data but the form and representation of our rules draw largely from rule representation within the CAD domain.

Road markings and rules: the application domain

The application domain is that of a Road Traffic design software to aid transport professionals to quickly and easily represent traffic features.

Traffic controls are seen every day on our roads, typically installed by an appropriate governing authority with the purpose of controlling the behaviour of traffic in a defined way. These typically fall into three categories:

- Road Markings (stop lines, double yellow lines etc)
- Traffic Lights
- Road Signs

Governing authorities throughout most countries in the world have developed strict and complete regulations governing the selection, location and physical appearance of traffic controls on their public highways. The designing of new highways or updating of existing highways must adhere to these strict regulations on the use of traffic controls.
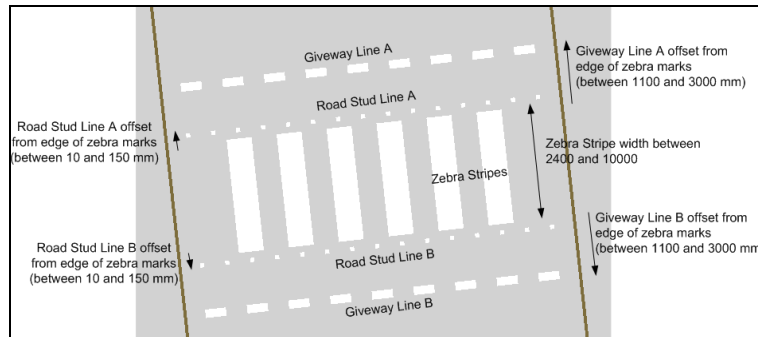
Fig. 1. UK Zebra Crossing Regulations (UK DOT, 2003)

Typically a set of traffic control regulations defines many types of traffic control; each traffic control type has a shape or geometry defined by means of one or more geometric shapes that must exist within prescribed geometric constraints. Instances of these traffic control types exist as discrete entities in the real world. Traffic control regulations can be viewed as a set of rules for constructing and placing instances of traffic controls in the real world. Initially our work has focused on the regulations pertaining to road markings within the UK and the USA although it is envisaged that this will be extended to other countries in the near future.
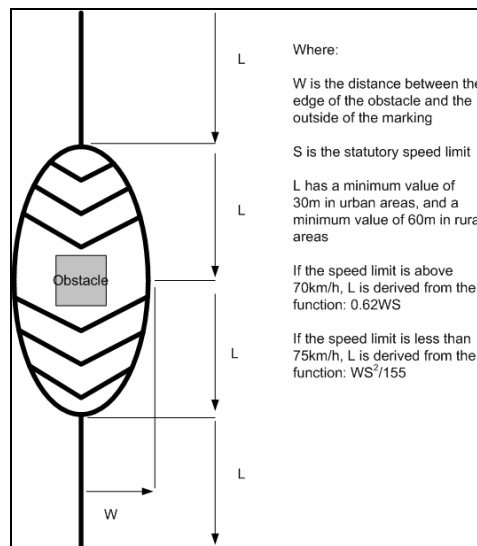

Fig. 2. USA Obstruction Marking (US DOT, 2003)

Figures 1 and 2 show examples of the constraints imposed by the traffic regulations on zebra crossings (UK) and obstruction markings (USA) respectively. These and all other Traffic Controls need to be designed by the Transport Professional easily using the software but at the same time the software should enforce the rules defined.

Object based approaches to modelling traffic features

Traditional GIS layer based vector approaches to modelling road markings would represent each component of the road marking as a point / line / polygon on a road markings layer. As such it would be very difficult to logically group the basic geometric components of individual traffic features, although this could be achieved using attribute information. A more natural representation is to use object based models. Object models represent the world in terms of features (or objects) rather than layers (Worboys and Hearnshaw 1990) and thus are more flexible and richer. In our approach we take advantage of the flexibility of modern object oriented modelling (Shalloway and Trott 2005) exploiting abstraction, inheritance and aggregation to provide a flexible, powerful model for road markings, which shares a number of features with GML 3.1. The high level overview of the model is given in Figure 3, with the geometry model in Figure 4.
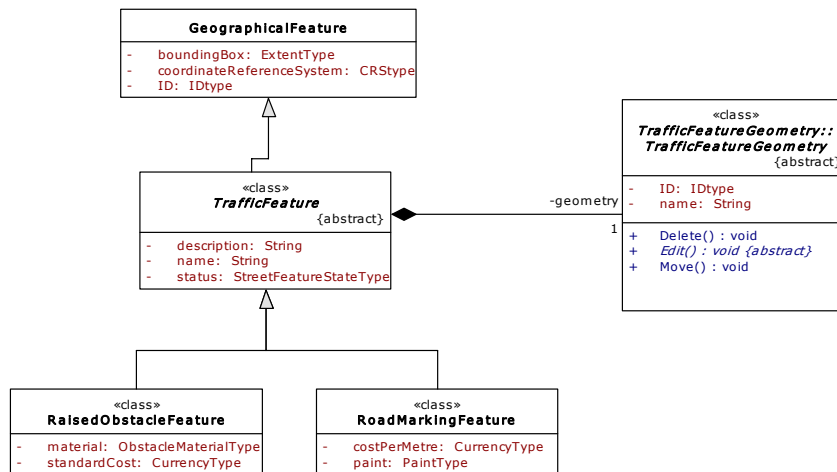

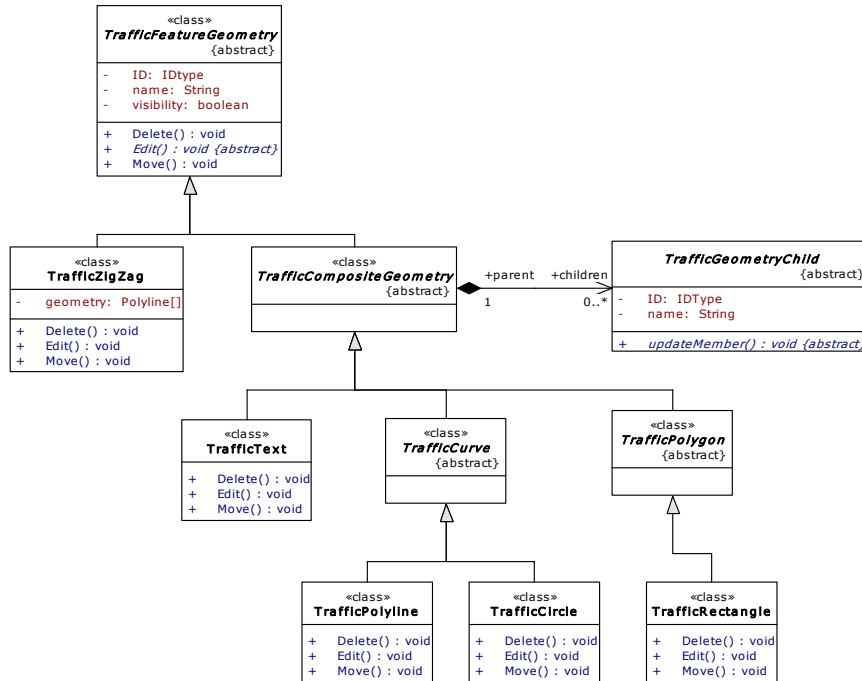**Fig. 3.** Traffic Feature Model overview

Fig. 4. Traffic Feature Geometry overview

Modelling rules between feature components

In order to model the rules/constraints between geometric elements within a feature's geometry, using a standard GML-like geometry model is insufficient. The model presented here takes its inspiration from many of the concepts presented earlier for variational CAD models.

Elements in the feature's geometry are described through constraint relationships; these are typically explicit spatial relationships that are used to determine the geometric state of one geometric element based on the geometry of another 'parent' geometry element. Figure 5 provides an overview of the UML model for the relationships between a geometry element and any 'child' geometries it may contain. The offset relationship is only included here for simplicity, though as suggested by Pratt (1998) relationships such as perpendicularity and tangency could be included. Geometry structures can now be viewed to take the form of recursive trees, where a geometry element contains potentially many 'child' geometries (whose re-

lationships are constrained by the child types), and these child geometries themselves can contain children.
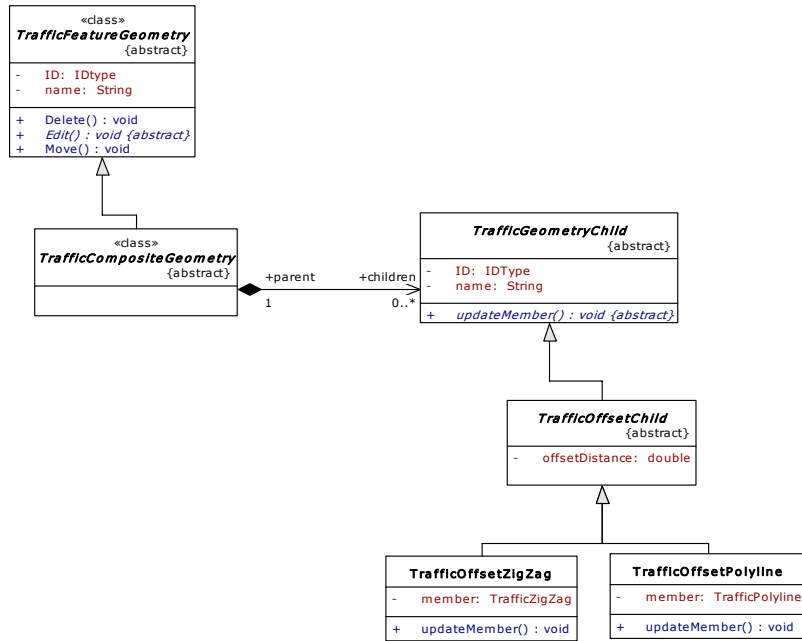

Fig. 5. Traffic Feature Geometry Child overview

This geometry model is essentially an *Explicit Variational Model* as described in Pratt (1998) and Pierra et al. (1994), though the definition and instantiation of these geometry structures through the use of the XML rule bases have more in common with the History-Based Models. The parametric nature of this model will be discussed in the next section.

Parameterised rules and simple expressions

To enable geometric elements and geometric constraints within geometry definitions to relate to each other in more complex and less structured ways than the constraint-based geometry model permits, a parameter model has been devised. The parameter mechanism allows the rule base designer to specify relationships between certain attributes of the geometry and geometry constraint (child) classes in terms of real number or Boolean-value based expressions. A parameter value is essentially defined as a function of a number of other parameters.
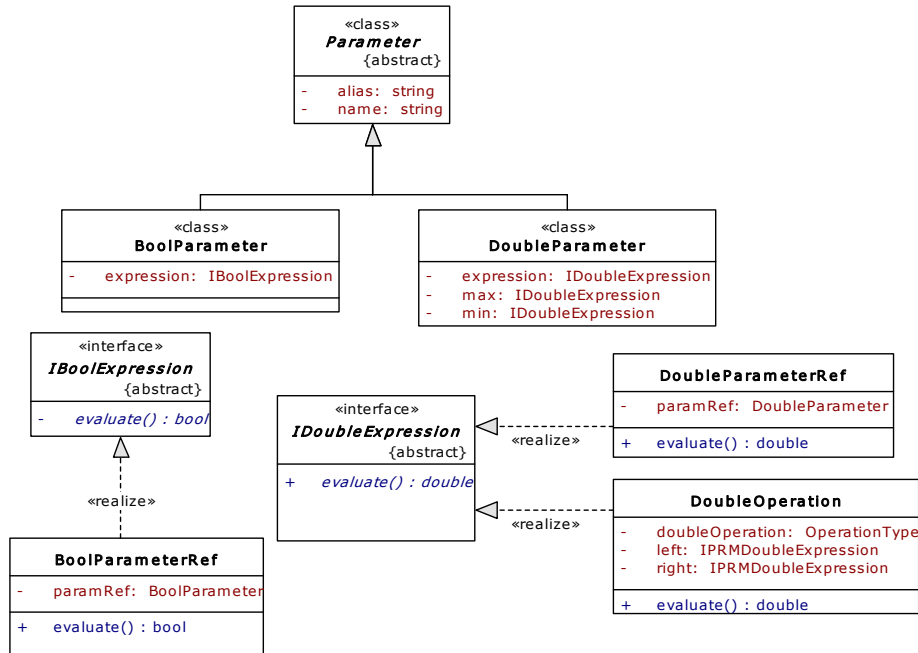
Fig. 6. Basic UML Model of Boolean and Double Parameters

A real number valued parameter (or double parameter), for example, consists of an expression, used to determine a real number value. There are also minimum and maximum bounds that this value is only permitted to lie between. The minimum and maximum bounds provide the means to enforce the upper and lower limits specified in the traffic regulations. These minimum and maximum bounds are themselves defined by expressions, providing increased flexibility for creating feature-geometry definitions.

Figure 6 shows the simplified UML model for parameters, the capability of the expressions can be extended by implementing the IDoubleExpression and IBoolExpression interfaces. When numeric and Boolean based attributes in the various geometry and geometric constraint classes are exposed as parameters they can be referenced by other parameters and have their values defined as functions of other parameters belonging to other elements in the model. Figure 7 shows how the distance attribute of the TrafficOffsetChild class can now be defined as a double parameter, and

hence the distance at which the member geometry is offset from its parent is derived through the evaluation of a function that is potentially dependent on other parameter values.

The following example illustrates how an expression might be used in the definition of a UK Zebra Crossing (see Figure 1). The distance at which a stud line is permitted to be offset from a line representing the centre of the crossing could be expressed as:

OffsetDistance = -($StripeWidth/2 + $StudOffset)

Where $StripeWidth and $StudOffset are references to other parameters within the same feature definition that represent other quantities of the feature or its geometry.
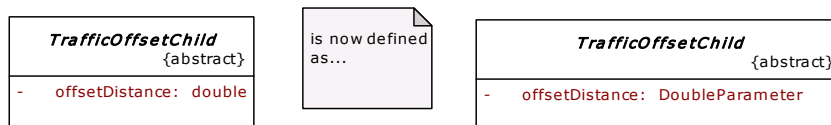

Fig. 7. Exposing attributes of the geometry and child types as parameters

Similarly, the Boolean-valued attribute 'visibility', which controls whether or not a geometric element within a feature's geometry is visible/included in the current instance can be exposed as a Boolean parameter:

Visibility = $IncludeRoadStuds

Example road marking features

Figure 8 shows a representation of an XML instance describing the structure of a zebra crossing. This encoding aims to encapsulate those constraints specified by the traffic regulations as seen in Figure 1. A number of 'custom' feature level parameters are defined for the road marking feature itself. One of these, 'StripeWidth', is used to model the width of the zebra stripes that occur along the centre of the road marking. The stripes themselves are not included in this definition for simplicity. The parameters 'StudOffset' and 'GivewayOffset' are used to hold the offset distances of the stud lines and giveway lines from the edge of the zebra stripes re-

spectively.   A single polyline is used to represent the centre of the zebra crossing, around which the other elements in its geometry are positioned.

Each geometry element has a visibility attribute, represented as a Boolean-valued parameter. The traffic regulations state that the inclusion of the road studs in Zebra Crossings are optional; this is realised in this encoding by the custom parameter 'IncludeRoadStuds'; the visibility parameters of both road stud lines are set equal to 'IncludeRoadStuds', ensuring that both road studs are either included or not included in the zebra crossing.

Element	Value / Type
sfr:RoadMarking	{name, alias, description, regulationAttributes?, customParameters?, Constru…
id	70E91102-58D0-4d90-8F76-B9A20C1E8336
sfr:name	Zebra Crossing
sfr:alias	Pedestrian crossing for the UK
sfr:description	Denotes where pedestrians can cross the road
sfr:customParameters	{_Parameter+}
sfr:DoubleParameter	{(name, alias, useExpressionDefault), (Expression)}
sfr:name	StripeWidth
sfr:alias	Width of zebra stripes
sfr:useExpressionDefault	true
sfr:Expression	{min, max, parameterExpression}
sfr:min	2400
sfr:max	10000
sfr:parameterExpression	3000
sfr:DoubleParameter	{(name, alias, useExpressionDefault), (Expression)}
sfr:DoubleParameter	{(name, alias, useExpressionDefault), (Expression)}
sfr:BoolParameter	{(name, alias, useExpressionDefault), (expression)}
sfr:name	IncludeRoadStuds
sfr:alias	Include Road Studs in the Zebra Crossing
sfr:useExpressionDefault	true
sfr:expression	TRUE
sfr:geometry	
sfr:Polyline	{((name, visibility?), (CurveStyle)), (GeometryChildren?)}
id	3346429A-9661-495f-B11D-E8D7F031BF7C
sfr:name	Centre construction line
sfr:visibility	FALSE
sfr:CurveStyle	{CurveStyle?}
sfr:GeometryChildren	{OffsetPolyline*}
sfr:OffsetPolyline	{(((name), (distance, side)), (startNodeOffset, endNodeOffset)), (member)}
id	2BBDDF10-C212-4da3-990B-628022ECB717
sfr:name	Left road stud line offset from edge of zebra marks
sfr:distance	{min, max, parameterExpression}
sfr:min	-($StripeWidth/2 +155)
sfr:max	-($StripeWidth/2 +10)
sfr:parameterExpression	-($StripeWidth/2 + $StudOffset)
sfr:member	{Polyline?}
sfr:Polyline	{((name, visibility?), (CurveStyle)), (GeometryChildren?)}
id	1C860D6C-8226-4024-B67E-95B1925DB161
sfr:name	Left Road Stud Line
sfr:visibility	$IncludeRoadStuds
sfr:CurveStyle	{CurveStyle?}
sfr:OffsetPolyline	{(((name), (distance, side)), (startNodeOffset, endNodeOffset)), (member)}
sfr:OffsetPolyline	{(((name), (distance, side)), (startNodeOffset, endNodeOffset)), (member)}
sfr:OffsetPolyline	{(((name), (distance, side)), (startNodeOffset, endNodeOffset)), (member)}

Fig. 8. A representation of the XML instance defining a Zebra Crossing, to encapsulate the constraints imposed by the UK Traffic Regulations

Figure 9 shows a properties dialog box–type representation of an instance of this zebra crossing. The panel on the left hand side displays the geometry elements along with the child association types that are used to constrain them. The panel on the right provides the means for a user to edit the properties of the geometry and child types. Here we can see access to the offset distance of the right road stud line – represented as a double parameter. The offset distances for the stud lines and give way lines for the Zebra Crossing are derived from feature-level parameters, and so these values will only be modified through the manipulation of the feature properties.
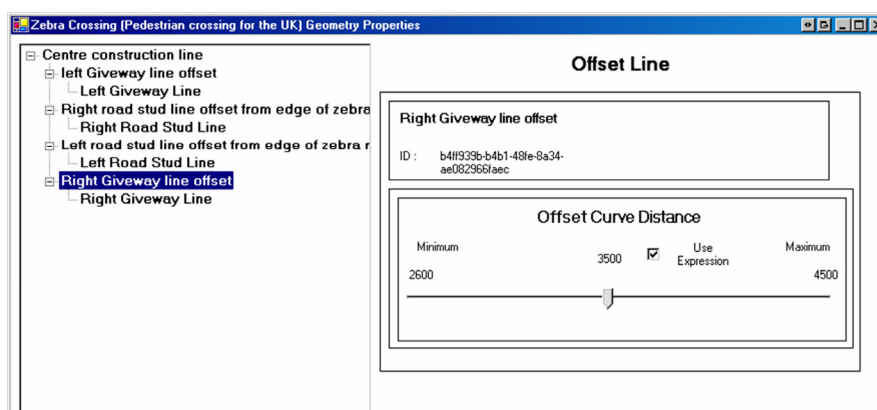

Fig. 9. The structure of the geometry for an instance of a UK Zebra Crossing as defined by the XML encoding shown in Figure 8

Links with Geography Markup Language

GML 3.1 is a developing set of standards for the encoding and transmission of spatially referenced data (Lake et al. 2004). GML was developed by the Open Geospatial Consortium[1] (OGC) and provides a rich set of XML based schema for describing spatial data, including geometry, topology, coordinate systems, coverages and grids, temporal data and observations. The central pattern used in GML is the feature (or object) – property model. The manner in which GML is used in applications is by extending the base abstract feature (or feature collection) model provided by the GML feature schema in a user defined application schema. This gives GML immense flexibility but also introduces its own semantic problems

---

[1] Open Geospatial Consortium: http://www.opengeospatial.org/

since each user can in theory develop their own application schema. The
key benefit of using GML is that we have an open standard for the trans-
mission of our data, which can be achieved using web services (Graham et
al. 2001), making genuine interoperability a real possibility.

In this work we make two contributions; we develop an application
schema for road marking features, but because the GML 3.1 model is not
rich enough to permit us to represent this model using the standard
GML3.1 geometry schema, we produce a novel geometry schema to per-
mit a range of constrained geometry types, such as an offset line. While
this goes somewhat against the GML recommended best practices (Lake et
al. 2004), it is the only plausible method to impose such business rules in
GML. In developing the schema for the representation of the road marking
features we have incorporated the GML feature – property model. The
benefit of the feature – property model is that the name of the property
conveys weak semantic meaning to that property in a manner similar to the
resource description framework, on which GML was initially based.

The schema constrain the creation of instances of the road marking fea-
tures, for example that shown in Figure 8. These instances maintain
knowledge of their own internal associations, however it is possible to re-
solve all parameterised components, that is evaluate all parameterised rules
to represent the feature in terms of static geometry, to produce a pure GML
3.1 compatible feature collection. At some point the resolution of static
geometry is necessary to display the features in a GIS or CAD environ-
ment in any case. Of course in doing this 'explode' or 'export' we lose the
flexible representation of the rules within the feature, however since very
few applications, other than those we are developing are likely to support
the use of these rules, this is not relevant. Evaluating the model to a pure
GML 3.1 representation does however offer the ability to communicate the
resulting road marking design across a web service to a whole range of
GML enabled clients, allowing easy communication with external clients
and the public.

Summary

In this paper we have reviewed the representation of geometric rules
within both GIS and CAD, with emphasis on the types of rules that are
relevant to the representation of Traffic Controls. We have shown how we
have extended an object based spatial model to permit the representation of

parameterised rules within a GIS context. This has united the CAD and GIS approaches to rule representation, and is very flexible. We have defined a set of schema that implement the UML represented object models that are closely tied to the developing GML format. In particular we are able to create a set of data driven object types from our rule instances, which provides great flexibility in modelling a wide range of traffic features and rules, without the need to change any source code. The instances are readily converted to GML for easy display across the web.

An interesting direction to take the work in the future would be to integrate the model more tightly with GML, by further abstracting the geometric rule representation to allow it to be included in a future GML specification, however at present it is not clear that there are a sufficient range of applications requiring this extra complexity to merit its inclusion in GML.

References

Borges K. (1997) Geographic Data Modelling - An Extension of the OMT Model for Geographic Applications. Master's thesis, Joao Pinheiro Foundation (in Portuguese).

Borges K, Laender A, and Davis Jr C (1999) Spatial Data Integrity Constraints in Object Oriented Geographic Data Modelling. In Proceedings of 7th ACM GIS, Kansas City, USA,

Clementini E, and Felice P (1994) A Model for Representing Topological Relationships Between Complex Geometric Features in Spatial Databases. Info. Sciences, pp 1 – 17.

Cockcroft S (1996) Towards the Automatic Enforcement of Integrity Rules in Spatial Database Systems. In Proceedings of the Spatial Information Research Centres' 8th Colloquium.

Cockcroft S (1997) A Taxonomy of Spatial Data Integrity Constraints. GeoInformatica, 1(4): pp 327 – 343.

Cockcroft S (1998) User Defined Spatial Business Rules: Storage, Management and Implementation - A Pipe Network Example. In Proceedings of the Spatial Information Research Centres' 10th Colloquium.

Cockcroft S (2001) Modelling Spatial Data Integrity Rules at the Metadata Level. In Proceedings of the Sixth International Conference on GeoComputation.

Egenhofer M, and Herring JR (1991) Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases. Technical Report pp 1 – 33, Detp. Of Surveying Engineering, University of Maine, Orono, ME.

Graham S, Simeonov S, Boubez T, Daniels G, Davis D, Nakamura Y, Neyama R (2001) Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI, Pearson Education, London.

Lake R, Burggraf D, Trninic M, Rae L (2004) Geography Mark-Up Language: Foundation for the Geo-Web, John Wiley and Sons, London.

Pierra JC, Potier G, Girard P (1994) Design and Exchange of Parametric Models for Parts Library. In Proceedings of the 27th International Sympossium on Advanced Transportation Applications, pp 397 – 404, Aachen, Germany.

Pratt M (1998) Extension of the Standard ISO 10303 (STEP) for the Exchange of Parametric and Variational CAD Models. In Proceedings of the Tenth International IFIP WG5.2/5.3 Conference.

Price N, Tryfona R, Jensen CS (2000) Modeling Part-Whole Relationships for Spatial Data. In Proceedings of the 8th ACM GIS, pp 1 – 8.

Price N, Tryfona R, Jensen C (2001) Modeling Topological Constraints in Spatial Part-whole Relationships. In Proceedings of the 20th International Conference on Conceptual Modeling, Yokohama, Japan.

Shalloway A, Trott JR (2005) Design Patterns Explained: A New Perspective on Object-Oriented Design, Addison-Wesley, London.

UK Department for Transport (D.O.T) (2003), *Traffic Sign Manual Chapter 5 – Road Markings*, The Stationary Office.

U.S. Department of Transportation (DOT) (2003) Federal Highway Administration (FHWA), *Manual on Uniform Traffic Control Devices for Streets and Highways* [Online]. Available: http://mutcd.fhwa.dot.gov/pdfs/2003/pdf-index.htm.

Worboys MF, Hearnshaw HM (1990) Object-Oriented Data Modelling for Spatial Databases*,* International Journal of Geographical Information Systems, 4, pp 369-383.