The second GML prototype of the new TOP10vector object model

Drs. M.E. de Vries, drs. T.P.M. Tijssen, drs. J.E. Stoter,
drs. C.W. Quak and prof. dr.ir. P.J.M. van Oosterom

GISt Report No. 12, Part 1, Main text
Delft, December 2002

Summary

This report describes the second GML prototype for the Dutch TOP10NL topographic data 1:10,000. The prototype was built by the TU Delft, section GIS-Technology as part of a larger project 'Object Orientation TOP10Vector', that was started in 2000 by the Dutch Topographic Service (TDN). The report opens with some background information about the TOP10Vector project. It then gives an overview of the changes with reference to the first GML prototype (published in 2001). These changes were the result of discussions with the other participants in the project: ITC Enschede, CGI Wageningen and the Topographic Service itself. The report describes implications of changes in the data model for the conversion process (from MicroStation Design files via FME and Oracle Spatial to GML). It then discusses the main characteristics of the second GML prototype, with a focus on the choices that were made in implementing the data model. Some of these choices may need reconsideration (e.g. mixed geometries). The report concludes with an overview of issues that are still open and also points at the new version of GML (GML 3.0), which offers a number of new possibilities (metadata, temporal aspects, more geometry types, use of topology).

The prototype consists of test datasets plus two application schemas, one for the data itself and one for the metadata. These two application schemas (.xsd documents) are included in an appendix. Other more technical information related to the production of the test datasets and the conversion into GML can be found in a separate document (Part 2 of this publication). The test datasets can be obtained from the following site: http://kartoweb.itc.nl/top10nl/TOP10NL_eng/index2.htm.

ISBN: 90-77029-01-X
ISSN: 1569-0245

Section GIS technology, Faculty of Civil Engineering and Geosciences
Thijsseweg 11, 2629 JA Delft, the Netherlands
Tel: +31 (0)15-278 4548; Fax +31 (0)15-278 2745;
E-mail: gist@tudelft.nl

Contents

Chapter 1

Introduction

The Dutch Topographic Service (TDN) is the supplier of TOP10vector, a digital vector product with topographical information of the Netherlands territory at a scale of 1:10,000. This report describes the improvements made by Delft University of Technology (TUD) to the first GML prototype of the new, so called, TOP10NL (object oriented TOP10vector) [8]. The result is the second GML prototype of TOP10NL. Besides the GML documents also ESRI Shapefile variants of the prototype are produced in order to give more potential users an impression of the contents of the new TOP10NL, as visualization tools for GML are currently not widespread.

1.1 Prototype history

The second prototype of TOP10NL is used in the public evaluation of the new product in the first half of 2002. Besides the development of this second prototype, the involvement of the TUD in this phase of the project consisted of consultancy and support related to the creation of the demo CD and TOP10NL website. The first prototype of the TOP10NL was evaluated within the project internally by the Centre for Geo Information of Wageningen University (CGI) [22]. This evaluation indicated several aspects of the first prototype which should be improved. CGI also produced the initial overview of user requirements and prioritization [10], on the basis of which the International Institute for Aerospace Survey and Earth Sciences Enschede (ITC) developed an initial conceptual model for TOP10 data [12]. An application XML Schema definition as implementation of the new TOP10NL model and GML documents with real-world data were created by TUD [8, 9, 21].

Important characteristics of the new conceptual model are: unique object-identifiers, a partitioning of the surface as the basis for geometry (exceptions occur in case of overlap, e.g. road segments in tunnels or road segments on bridges), 2.5D objects with 3D coordinates, possibility of complex features (an aggregation of road segments into one - or more - 'named' roads) and the incorporation of metadata and temporal data for each object instance (versioning). The last characteristic opens the way for 'change only' updates distributed to user organizations, but was not implemented in the first prototype of TOP10NL (see also [2]).

Both the first and the second prototype of TOP10NL are implemented using GML 2.0 [15]. GML 2.0 was accepted as an implementation specification by the OpenGIS Consortium in April 2001. The rationale behind the choice of GML is the fact that it is based on the world-wide accepted XML standard and that a rapidly increasing number of tools is available to generate, check and interpret XML/GML [21]. During this project we also encountered a drawback of the very dynamic (Internet) world: according to the new base XML schemas, the OpenGIS GML 2.0 (XML) schemas were not valid anymore. We first had to adjust the GML 2.0 XML schemas before we could continue the development of TOP10NL XML schemas based on GML. This enabled the formal checking of the GML TOP10vector prototypes against the XML Schemas.

As indicated above, both GML and Shapefiles were produced. However, it is important to realize that there are a number of drawbacks related to the Shapefile prototype: objects with two geometries are split into two entities, attribute names are limited to 10 characters, attribute data types are often 'string' instead of 'date' (translation of datetimes from the database to Shapefiles is limited) or 'number' (value for 'unknown', 'onbekend' in Dutch, is missing), and another method for relating metadata to the objects. For this reason only the GML version should be seen as true representatives of the new TOP10NL.

1.2 Project background

The Dutch Topographic Service (TDN) is currently improving their products and production environment. The production environment will be based on object technology [3] [20]. The strategy to improve the products is described in [13] and the first step is to re-engineer the Digital Landscape Model of the TOP10vector product [19]. The design and development of the GML prototype described in the previous report [8] are part of this re-engineering process.

Another project related to the project which resulted in the (second) GML prototype of TOP10NL concerns the visualization efforts. This includes the development of a digital cartographic model (DKM) accompanying the current digital landscape model (DLM) implemented in GML. Further, this project includes investigation into viewing with a special purpose viewing tool based on a GML parser, but also converting the GML data to SVG (and then use a standard SVG-viewer). TDN has asked the ITC and the TUD to assist with this project. During the writing of this report, the TUD activities in the DKM TOP10NL project were not yet finished.

A second project related to the GML prototype of TOP10NL is the development of an object oriented DLM of more medium and smaller scale data. It has to be investigated how much of the TOP10NL schema's can be re-used, whether it is more sensible to extend the current TOP10NL model to include also the other scales within the same model or that for every scale a different model is needed (based on shared common parts in the model). During the writing of this report, the DLM smaller scales project was in an initial phase.

1.3 Overview of the report

An overview of the (model) changes in the second GML prototype of TOP10NL with reference to the first prototype is given in Chapter 2. Most changes have implications for the data model, for the implementation in GML and for the resulting data sets.

The process to create the second GML prototype is quite similar to the creation of the first prototype: it starts with the creation of sample data sets by the Dutch Topographic Service in accordance with the new requirements. Then the data is loaded into an Oracle spatial database and after many intermediate operations and manipulations GML documents and Shapefiles are produced. More details about to the data conversion process are supplied in Chapter 3.

In Chapter 4 important concepts and terminology of XML and GML are introduced. Chapter 5 describes the application schema and implementation in GML of the new TOP10NL. In contrast to the first prototype where two versions of the application schema were available, a strict and a non-strict version, now only the strict version has been developed.

This report is concluded in Chapter 6 with an evaluation of the second GML prototype of TOP10NL.

Chapter 2

Model changes

In this chapter an overview of the most important changes in the GML prototype of TOP10NL is presented. Three types of changes are described: 1. changes related to temporal aspects, 2. changes related to geometry and 3. other changes.

2.1 Temporal changes in the GML prototype

Temporal data model The data model of the first GML prototype of TOP10NL contained a simple (but sufficient) temporal model consisting of two time stamps: begin and end time ('begindatum' and 'einddatum' in Dutch). In the overall project is was however decided to use a more complicated temporal model including the following attributes for every topographic object:

1. 'ontstaan_uit': Dutch for 'originated from', which refers to a list, of unknown length, with the object_ids of the parents.
2. 'object_begindatum': Dutch for 'object begin date', the date of the first time the object is stored in the database of the TDN.
3. 'versienummer': Dutch for 'version number', the sequence number of the current version of the object (with the same object_id).
4. 'versie_begindatum': Dutch for 'version begin date', the start date of the current version of the object, equal to 'object_begindatum' if 'versienummer' is 1.
5. 'versie_einddatum': Dutch for 'version end date', the end date of the current version of the object. Two situations may be the case: 1. There is a successor version of this object (which has the same object_id, and the same value for 'versie_begindatum' as this version has for 'versie_einddatum') or 2. This object has no successor version, if it is a terrain object (forming a partitioning of the domain with the other objects) the space will be taken over by one or more new objects having this object in its 'ontstaan_uit' list (and having the same value for 'versie_begindatum' as the deleted object has for 'versie_einddatum').

Temporal data TDN created a number of objects with updated data for the new prototype. The challenge was to include this 'history' in the proper way. First of all

the original data set does not have unique object_ids, so before generating history TUD first assigned object_ids to all objects. A spatial temporal model was created by filling in the proper dates (editing done by TDN) and making sure that different versions of the same object really use the same object_id. The famous question is: 'In which cases does a change in an object result in just a new version of the same object, in which cases does it mean the deletion of the current object followed by the creation of a new one?' In this project is was decided that changes related to thematic attributes will result in new versions of the object and changes in the geometry will result in a new object.

Datasets with history Once a proper spatial temporal model is created and actually populated in the Oracle spatial database, different types of temporal datasets can be created:

1. A dataset corresponding to a specific date. The dataset represents the topographic situation at that moment in time. All objects in the dataset are current at the specified date, this date is within the time interval 'versie_begindatum' - 'versie_einddatum' of the objects included in the dataset.
2. A dataset containing only the changes between two specified dates. (Versions of) objects that do not change in this time interval, that are deleted before the interval starts or that are created after the interval ends are not included in the dataset.
3. A dataset containing everything in the database, including all old versions and deleted objects. Note that an object can change several times, resulting in many versions and/or deleted/created objects. In the datasets used in the prototype this is not the case.

2.2 Geometry changes in the GML prototype

Object classes with two geometries All subclasses derived from the class 'infrastructure' (road, railroad and water) can have two geometry representations. This should not be confused with multi-geometries (a multi-point, a multi-polyline, a multi-polygon or even more general a geometry-collection) used in modeling some of the other entities in the GML prototype; e.g. many of the different types of regions may consist of more than one polygon. The first geometry attribute of an 'infrastructure' object is now always a polygon, and the second is either a point or a polyline depending on the role of the object in the infrastructure network (junction or connection). An alternative would have been to have only one geometry attribute of type 'geometryProperty' in which all attribute values are possible. This was considered bad data modeling as it would not make explicitly clear which role the component plays: area representation or network representation. Logically it follows that the first geometry attribute type is 'polygonProperty', but for the second geometry attribute the 'geometryProperty' type was chosen. The first decision is obvious, but the second decision could also be labeled bad data modeling as the second geometry may either contain a 'point' or a 'polyline' (or even other values), which are not enforced by the data model. However, in order to avoid having two

subclasses (like road junction and road connection) for every infrastructure class is was decided not to apply strict geometry typing for the second geometry attribute.

Relating two geometries to one object instance Having a model which expects every infrastructure object to have two geometries is one thing, creating a GML prototype in which this is actually the case is another thing as the two geometries are not related in the separate source datasets. After loading the data into the database a 'smart' matching process had to be implemented. A 'smart' matching process is needed, because normally the centerline of a connection segment extends to the junction (point). The part of the centerline around the junction is not contained in the polygon of the connection, but in the polygon of the junction (to which it should not be linked as it should only be linked to the proper connection segment polygon).

Skip third dimension if not really used As the current dataset does not contain elevation data (z-coordinates), only pairs of xy-values can be found in the prototype. It was decided not to include the default value of '0' in the documents, because this can be confused with a real z-value of '0'. The XML schema of the current prototype supports both coordinate pairs and coordinate triplets.

Buildings in a separate layer In the first prototype the 'gebouw' (Dutch for 'building) objects were part of the spatial objects (in Dutch 'RuimtelijkeObjecten'), which together formed a partitioning of the domain. For the second GML prototype it was decided to exclude the building objects from the partitioning, at least in less densely built-up areas. It was considered to be a drawback that terrain objects covered by groups of buildings contained many holes (places where the buildings are located). The class terrain is extended with an additional classification code 'bebouwd gebied' (Dutch for built-up area), which represents the buildings (generalized to building blocks) in densely built-up areas. Separate buildings (located in the suburbs and countrysite) are in the 'gebouw' object class, the terrain underneath these buildings does not contain holes anymore.

2.3 Other changes in the GML prototype

Expansion of enumeration types During the creation of the GML documents several new values, which were not errors according to TDN, were discovered. After adding the new values to the corresponding enumeration types, it was possible to validate the GML documents again.

Exclusion of attributes with a null value In the first GML prototype, attributes without a value (so not 'Onbekend' or 'Overig', because these are considered as valid string values) were represented in the GML document with an empty element tag. It was decided to remove these empty elements from the corresponding object instances. So only objects having a proper value for an attribute, will show this attribute in the GML document.

Less 'string' type attributes In the first GML prototype many attributes were typed as 'string' (due to the automated conversion from the TDN source data to Oracle

Spatial via FME). In the second GML prototype we tried to apply more strict data typing of the attributes and 'string' attributes were converted to 'number' or 'date' data types. Again several errors were discovered this way, again showing the value of more strict data typing.

Metadata as separate entity The metadata attributes in the first prototype were all included in the object instances as attributes. This caused combinations of certain metadata attributes (data obtained on the same day, using the same equipment, resulting in the same quality) to be duplicated many times. In the second GML prototype of TOP10NL it was decided to introduce a separate metadata entity (object class), which has a metadata ID and contains all metadata attributes. Now only the distinct combinations are stored (under one and the same metadata ID) and the actual topographic object refers to this metadata object by means of the ID.

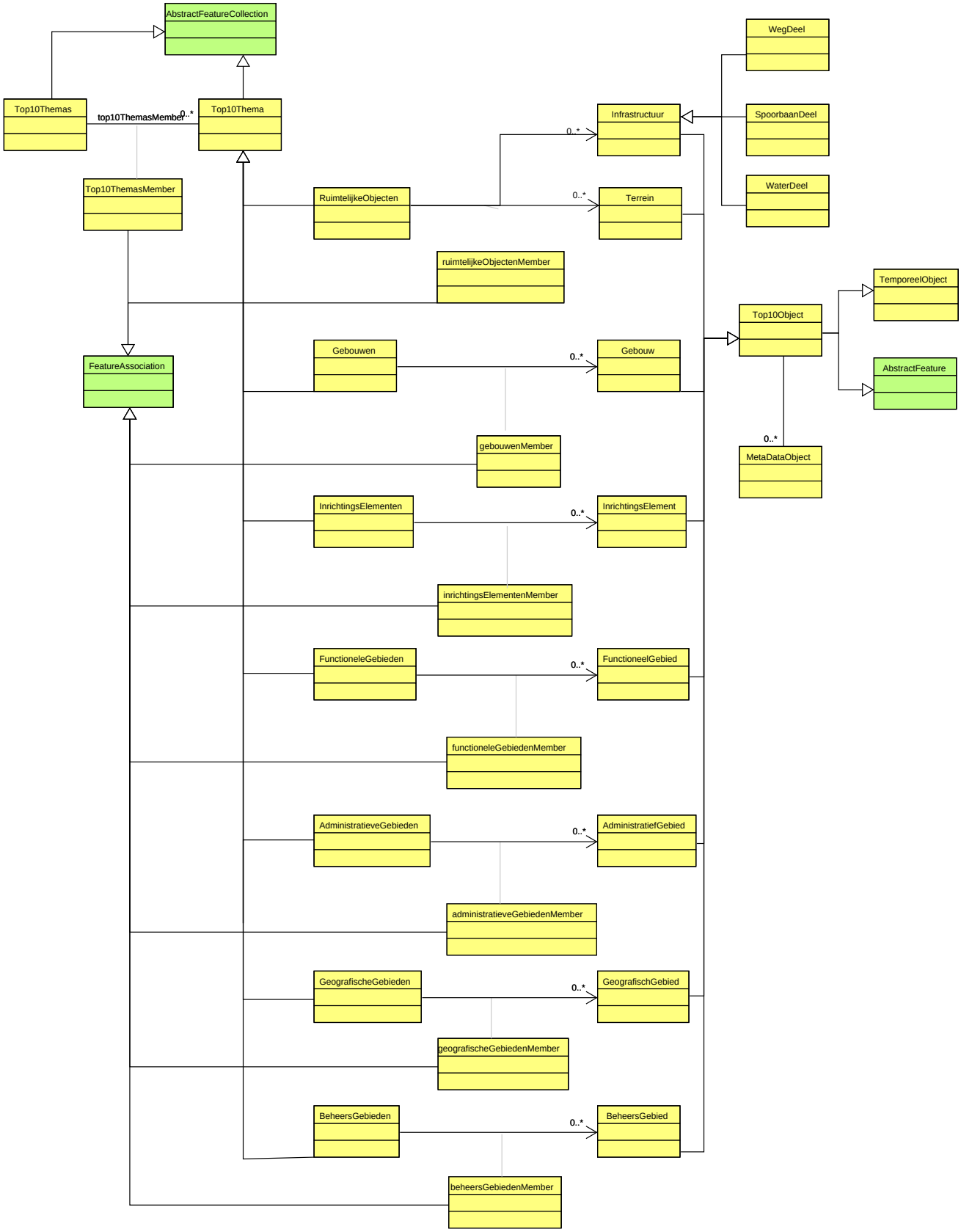

Figure 2.1: UML model of TOP10NL

Chapter 3

Data Conversion

3.1 Conversion process

The intention before starting with the second prototype was to create a simplified and streamlined conversion process. The outcome, influenced by the model changes and many other decisions during the project, was an even more complex and prototype specific process. The structure of the process has not really changed from the first prototype, so only the changes will be described in this chapter.

The final conversion process is shown in figure 3.1. The major differences with the previous process are the matching (and later joining) of geometries, the separate treatment of metadata, and the production of ESRI Shapefiles with the same content (as much as possible) as the GML documents.

Arnhem, Gouda and Tiel are used as test areas for this prototype. TDN produced more than 40 Design files as a starting point for the prototype. Most of these contained original data, some contained updates to the original data. The updates are included to illustrate the use of history in the new TOP10NL. For the 'infrastructure' entities, in addition to the files with areas, separate Design files with center lines (for connections) and points (for junctions) were produced. In the DBMS the original data and updates are stored in the same table, in the intermediate Shapefiles the updates are in separate files. Because only one type of geometry can be stored in a Shapefile the number of Shapefiles that make up the dataset for a test area can become substantial (up to 16 Shapefiles). For the final GML documents various datasets are created as selections of specific moments in time, with and without history, and with updates only.

3.2 FME processing

FME is used more often than in the first prototype, but this mostly means more of the same. A new element is the production of a dBase file with metadata, to be used in combination with the Shapefiles. For this a new FME translation had to be made. Also the secondary geometries of roads, waterways and railways required a new FME transformation module (because FME can only process one geometry per feature at a time). A curious finding was that the current version of FME available at the time

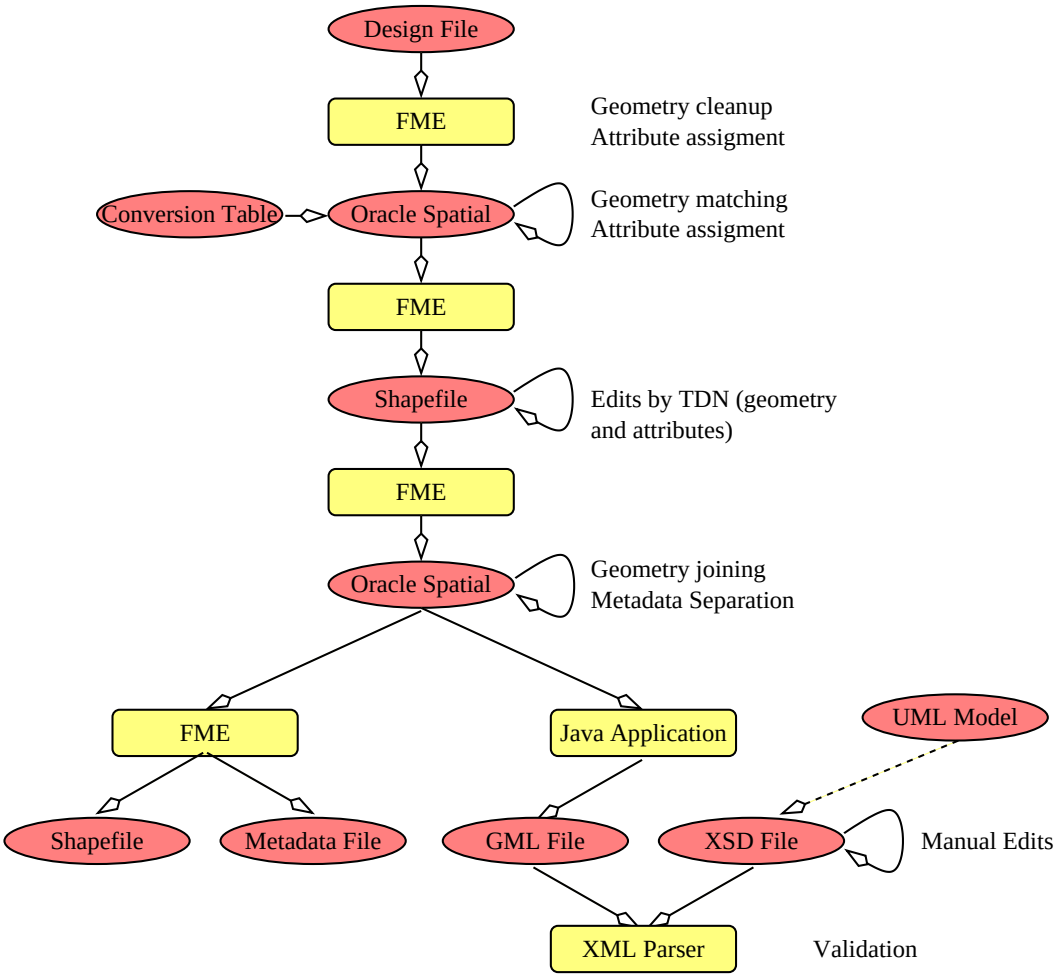

Figure 3.1: Conversion process

of the second prototype (2002) did not process geometry very well. The removal of 'island connectors' in the Design files resulted in erroneous geometry. So for the geometry processing an earlier version (2000) had to be used. The FME transformation modules can be found in Appendix B.

3.3 Oracle processing

New elements in the Oracle processing of TOP10NL data are the automatic matching of geometries and the separate treatment of metadata. Omitted from the second prototype is the erasing of buildings from the underlying terrain. Due to the increased complexity of the conversion process and the data itself (e.g. multiple geometry representations) more errors were present in the datasets prepared by TDN. To detect and correct these, even more pre- and postprocessing Oracle scripts were required than in the first prototype (examples of these can be found in Appendix C.1 and C.5). Oracle scripts are also used to find all domain values (to be used later to constrain the allowed values of attributes, see 5.4).

The matching and joining of geometries belonging to the same TOP10 object constituted

a substantial effort, this warrants some extra attention. The point and area geometries, or line and area geometries, for infrastructure objects were received in separate Design files. At this point no relationship exists between the geometries of an object. The purpose of the matching process is to find corresponding geometries (e.g. the road area and road centerline of the same stretch of road) automatically. In places where there is no overlap with other objects this is relatively easy, but at junctions, fly-overs etc. it will be difficult or even impossible. It was decided to only use geometry for the matching, no other attributes were involved. Using additional attributes (e.g the type of road segment: connection, junction or parking area) can improve matching, but it was estimated that the other attributes were not reliable enough to have a positive effect. Also a 'prudent' (as opposed to 'optimistic' or 'aggressive') matching strategy was adopted, a match is only finalized between geometries if a one to one relationship is established. This strategy was prompted by the conviction that it is simpler to add missing matches than to detect and correct erroneous matches. The matching result is stored in a temporary attribute, matched geometries remain as separate instances of the same object in the dataset. Matched geometries point to each other, special values are used to indicate uncertain matches or no match at all (see Appendix C.2).

The overall success rate was between 90% and 95% for the various datasets. Logically, automatic matching was less successful at and near junctions and fly-overs (see figure 3.2). Data errors are responsible for roughly half the missing matches. These matches were added (and a few incorrect matches corrected) by TDN in the Shapefile editing phase about halfway through the conversion process.

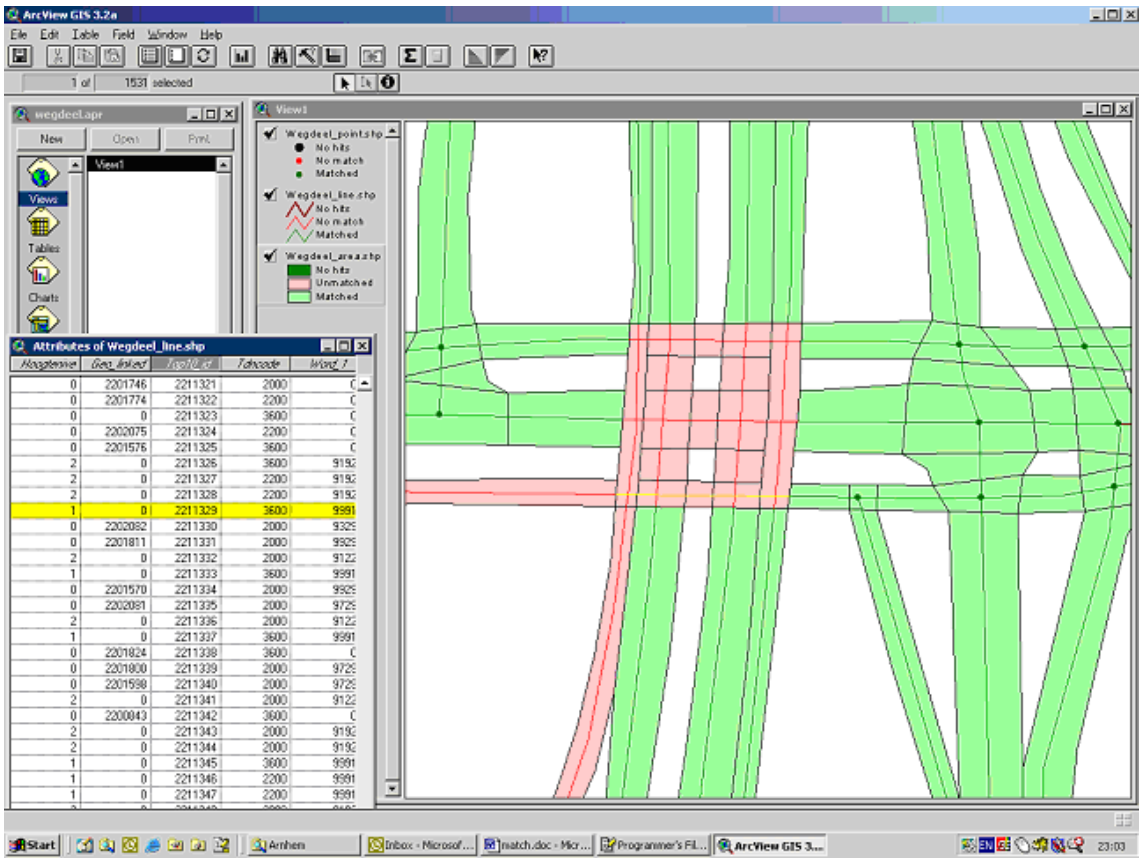

Figure 3.2: Example of automatic matching result

After another FME conversion to store the corrected Shapefiles back into the Oracle database, the geometries that are representations of the same feature are joined into a single object (Appendix C.3).

Up to this point in the conversion process metadata is still attached to the individual objects. To avoid repeating again and again identical metadata in the final documents it was decided to store the metadata separately. A small Oracle program is used to collect all 'unique' metadata records in a separate table, and a link is created in individual objects to point to the correct metadata (Appendix C.4). In the final steps of the conversion process the metadata table is converted into an XML document (to accompany the GML documents) and a dBase file (to be used with the Shapefiles).

3.4 Java application

The creation of the GML documents from the Oracle database is virtually unchanged in the new prototype. There have been some minor changes to print the GML format in accordance with the new schema. Other changes are the specification of a character encoding (in this case ISO-8859-1) to properly handle strings like 'geëlektrificeerd', and the proper formatting of attributes with multiple values (e.g. 'ontstaan_uit' and 'wegnummer'). In this prototype the Java application also takes care of the encoding of 'offending' XML characters by replacing them with their proper XML escape sequence (replace `'>'` with `'&gt'`, etc.).

Finally, the Java application issues the SQL statements to make the 'temporal' selections for the various GML datasets (a script to check the correctness of these selections can be found in Appendix C.6). In total 4 datasets were produced. The name of the GML file and the header of the document were used to indicate the temporal content of the dataset. The statements to select the appropriate subsets are ('wegdelen' as used as example entity):

```
-- complete dataset including all history:
-----------------------------------------
     select * from wegdelen;

-- dataset at januari 1 2002:
-----------------------------
     select * from wegdelen where
        versie_begindatum < to_date('2002-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS') and
       (versie_einddatum is null or
        versie_einddatum >= to_date('2002-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS'));

-- dataset at april 1 2002:
---------------------------
     select * from wegdelen where
        versie_begindatum < to_date('2002-04-01 00:00:00','YYYY-MM-DD HH24:MI:SS') and
       (versie_einddatum is null or
        versie_einddatum >= to_date('2002-04-01 00:00:00','YYYY-MM-DD HH24:MI:SS'));

-- only changes between januari 1 2002 and april 1 2002:
--------------------------------------------------------
      select * from wegdelen where
        (versie_begindatum >= to_date('2002-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS') and
         versie_begindatum <  to_date('2002-04-01 00:00:00','YYYY-MM-DD HH24:MI:SS')) or
        (versie_einddatum >  to_date('2002-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS') and
         versie_einddatum <= to_date('2002-04-01 00:00:00','YYYY-MM-DD HH24:MI:SS'));
```

Chapter 4

GML, XML and XML Schema: some terminology

GML has XML as its technical format and uses XML Schema as data definition language, i.e. as technique to describe the structure of the GML data. For that reason we will first introduce some XML terminology and than focus on GML. Readers who are familiar with XML and XML Schema can of course skip the first two sections and go directly to section 4.3.

4.1 XML

Like HTML XML is based on the 'markup' principle of content between begin and end tags. This makes XML documents - to a certain extent - self-describing. XML can easily be parsed by (standard) software, but is also readable and understandable for human eyes. XML is meant for the storage and exchange of structured data. XML is not 'free text', but on the other hand it is not so rigid as e.g. comma delimited data export files with their fixed number of columns and lack of flexibility (for a discussion of the merits of XML see [1]).

The basic unit of an XML document is called 'element'. An element has a name (or 'tag') and content (between the start and end tag). This content can be simple (a text string, a numeric value) or complex (nested 'child' elements). See example 1.

```
<Catalog xmlns="http://www.gdmc.nl/examples"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.gdmc.nl/examples catalog.xsd">

  <CatalogItem>
    <Book isbn="0-09-917331-X">
      <author>Feynman, Richard P. </author>
      <title>Surely You're Joking, Mr. Feynman</title>
      <pages>350</pages>
    </Book>
  </CatalogItem>

</Catalog>
```

Example 1:

An element can also have one or more 'attributes' directly after the element name. This 'attribute' in the XML syntax should not be confused with what is called an attribute in database terminology. An attribute in a database table (or in other terminology: the property of an object) can be implemented in XML as an XML-style attribute, but it will most of the time be implemented as a nested 'child' element of the object element. So in example 1 the 'title' property of the object 'book' is a child element of the 'book' element. Rule of thumb in this matter is that unique identifiers are implemented as XML attribute, while other object properties are implemented as child elements [4].

4.2 XML Schema

The structure of an XML document is described in a schema document, that can either be a DTD schema (=Document Type Description) or a schema written in the XML Schema language.

Both XML Schema and DTD have the same purpose: to describe and prescribe the structure of XML data documents. But, unlike DTD, the XML Schema language has XML as its technical format. This means that an XML Schema document (a document with extension .xsd) can be processed with the same tools as an XML document. XML Schema also has more possibilities than DTD to incorporate constraints in the data model [4]. From GML 2.0 onwards the GML specification of the OpenGIS Consortium uses XML Schema as data definition language.

An XML Schema schema (in the rest of this report also called 'XML Schema document', 'application schema', 'XML schema' or 'schema') contains two basic types of information: element (or element group) declarations and type definitions (complex types and/or simple types). The element declarations are meant to 'proclaim' the names of the elements that can appear in the XML document, while the type definitions are used to specify the nesting of elements (the XML 'tree') and the sequence in which they will appear. Example 2 shows the element declaration and the type definition for the 'book' example in the previous section.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://www.gdmc.nl/examples"
        xmlns:ex="http://www.gdmc.nl/examples"
        elementFormDefault="qualified" >

<element name="Catalog" type="ex:CatalogType"/>
<element name="CatalogItem" type="ex:CatalogItemType"/>
<element name="Book" type="ex:BookType"/>
<element name="CD" type="ex:CDType"/>

<complexType name="CatalogType">
  <sequence>
    <element ref="ex:CatalogItem" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
```

```
</complexType>

<complexType name="CatalogItemType">
  <choice>
    <element ref="ex:Book" minOccurs="1" maxOccurs="1"/>
    <element ref="ex:CD" minOccurs="1" maxOccurs="1"/>
  </choice>
</complexType>

<complexType name="BookType">
  <sequence>
    <element name="author" type="string" minOccurs="1" maxOccurs="unbounded"/>
    <element name="title" type="string"/>
    <element name="pages" type="integer"/>
  </sequence>
  <attribute name="isbn"/>
</complexType>

<complexType name="CDType">
  <sequence>
    <element name="artist" type="string" minOccurs="1" maxOccurs="unbounded"/>
    <element name="title" type="string"/>
    <element name="minutes" type="integer"/>
    <element name="tracks" type="integer"/>
    <element name="music" type="ex:KindOfMusic"/>
  </sequence>
</complexType>

</schema>
```

Example 2:

Apart from these so-called 'complex types' there are also 'simple types', for the data type of individual elements. In this way it is possible to restrict the possible values that can occur for that element (domain constraint). See example 3.

```
<simpleType name="KindOfMusic">
    <restriction base="string">
       <enumeration value="hard rock"/>
       <enumeration value="country"/>
       <enumeration value="classical"/>
       <enumeration value="musac"/>
       <enumeration value="rap"/>
    </restriction>
</simpleType>
```

Example 3:

One of the advantages of XML Schema as data definition language is the possibility to derive a type from another (super)type when defining simple and complex types. When

there is a class hierarchy of supertypes and subtypes in the conceptual data model, this inheritance structure can - to a certain extent - also be used in the XML Schema design, with the exception of multiple inheritance, which is not possible in XML Schema. An advantage of this inheritance mechanism is that basic supertypes can be defined once, and after that can be extended and reused in other parts of the same schema or in other schemas. This is also the way the GML specification works, as we will see in section 4.4. There are two ways to derive a subtype from a supertype: by 'extension' or by 'restriction'. When we discuss the TOP10NL GML prototype in Chapter 5 we will see some examples of both.

Element declarations in a schema document have a direct influence on the XML data documents that are based on that schema, because with the element declarations the names of the elements (the actual tags in the XML document) are specified. The 'complex type' definitions also have a direct influence, because they specify the hierarchical structure of the document: the nesting of elements (parent-child). There are also parts of the XML schema however that only play a role at design time, or when validating the data (validation is the test to see if the XML document is in accordance with the constraints in the XML Schema document). One characteristic that primarily plays a role at design time, is the inheritance structure of supertypes and derived subtypes. It makes the data model easier to maintain, but is not reflected as such in the XML data.

4.3 GML terminology

GML stands for Geography Markup Language and is one of the many specifications proposed by the OpenGIS Consortium (OGC) [16]. Purpose of the GML specification is to offer a core data model for geographic data in order to facilitate the exchange of data between different GIS, CAD or spatial database systems. This basic data model is written down in a number of XML Schema documents, and in addition the specification formulates some normative rules how to reuse and extend the basic GML data types.

GML documents have XML as their technical format. This is an important asset, because of the advantages of XML as an open, vendor-neutral format for data exchange that is also very suitable for use on the Internet. More importantly GML documents have a data structure that complies with a certain data model for geographic (or spatial) data. This conceptual model, called the Simple Features Model [14], forms the basis for the data model of GML. Part of the GML data model (the geometry part) is shown in Figure 4.1.

The current version of the GML specification (2.1.2) recognizes the following geometry types: Point, LineString, LinearRing, Polygon, MultiPoint, MultiLineString, MultiPolygon and Box (for specifying the bounding box). In the next version, GML 3.0, there are many more geometry types, also Bezier curve, Bspline, Circle, etc. [17].

Apart from these geometry classes, the OGC GML data model also contains classes for feature types (the object types that contain the spatial and non-spatial information), feature collections and for associations between classes (association or property types).

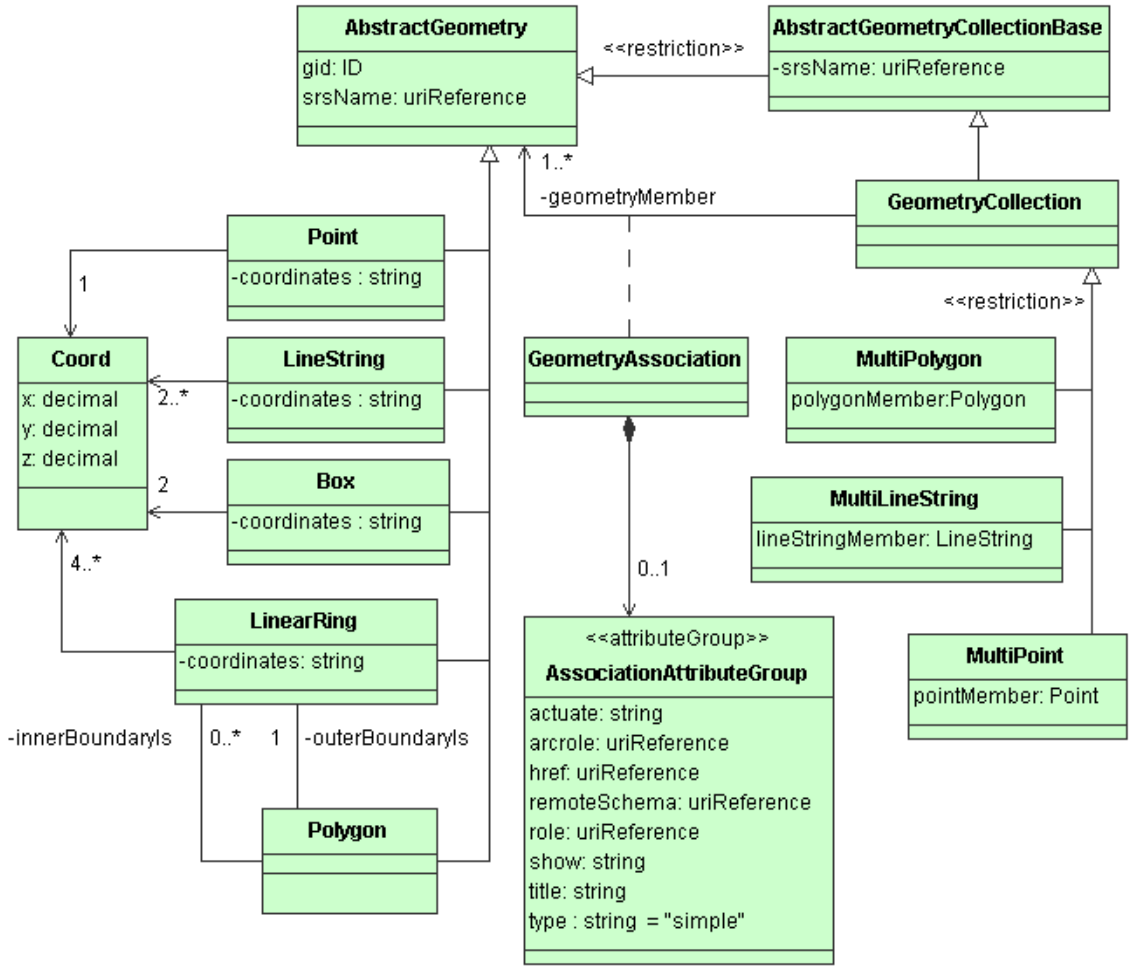

Figure 4.1: GML 2.1 Geometry model

Common to all GML implementations is a basic hierarchy:

```
FeatureCollection
    featureMember
        Feature
            nonSpatialProperty
            nonSpatialProperty
            ...
            geometryProperty
                Geometry (Polygon, LineString, Point etc.)
            geometryProperty
                Geometry (Polygon, LineString, Point etc.)
            ...
            nonSpatialProperty
            ...
```

A feature can have zero or more geometries. This is an important difference with e.g. ESRI Shapefiles, where only one geometry type per Shapefile is possible (point, line string or polygon). In other words, a GML feature instance like 'Road id=123' can have both a polygon geometry and a polyline (centerline) or point geometry (intersection nodes).

In XML jargon GML is called an XML 'vocabulary', a specific set of element names and element types to be used in a certain domain, in this case for geo-data exchange. GML is only a framework however, it is not a ready-to-use exchange format. An organization or information community will have to design its own GML implementation (or 'application'), in the form of one or more XML Schema documents.

The GML specification of the OGC offers a starting point for schema developers by providing a set of XML Schema documents with element names and types that can be considered as a kind of repository: organizations can take element names and types out of this repository and use them in their own XML schema when defining the data structure for their GML implementation. Many element names and types can be used directly, as they are (with the gml: namespace as a prefix), others will be used more indirectly, by deriving subtypes first. In the current version, GML 2.1.2, there are three .xsd files in the specification: feature.xsd, geometry.xsd and xlinks.xsd. When these schemas are imported in the organization's own schema, the names and types in the specifications' XML schemas can be used in the organization's schema. See Fragment 4.1 .

```
<!-- import constructs from the GML Feature and Geometry schemas -->
<import namespace="http://www.opengis.net/gml" schemaLocation="feature.xsd"/>
```

Fragment 4.1: XML import statement

It is only necessary to import 'feature.xsd' in one's own schema, because in feature.xsd there is an include of 'geometry.xsd' and in geometry.xsd there is an import of 'xlinks.xsd'. The difference between an `<import>` and an `<include>` has to do with the 'target' namespace of a schema: when the namespace of the schema that is included is the same, then `<include>` is used, when the namespace is different, then `<import>` must be used.

4.4 Deriving subtypes

The possibility of deriving subtypes from supertypes described in section 4.2 plays an important role in the mechanism used when creating GML data models. First of all, some of the classes (other names: object types or feature types) that are defined in the GML specification can only be used after they have been subtyped. When a type can not be used directly (i.e. can not be 'instantiated' as an object) but can only be used after it is subtyped, this type is defined as an 'abstract' type. Both the GML 'feature type' (the class that plays the central role in the GML class hierarchy), and the GML 'feature collection type' are abstract types, that can never be used directly in a GML application schema. So, what user organizations always have to do is to define their own feature types. This is done by deriving subtypes of the abstract type gml:AbstractFeatureType. Also a root level feature collection must be defined (an XML document always has one 'root' element), by creating a subtype of gml:AbstractFeatureCollectionType. (Note: In GML 3.0 there is now a gml:FeatureCollection element that can be used as ready-to-use name, so it is no longer necessary to define one's own feature collection elements in the application schema.)

The second way inheritance plays a role when constructing a GML schema is when for

one reason or another it is necessary to define your own geometry types. A reason can be that the geometry types in the GML specification do not provide enough functionality for the specific business data model. In the case of the Ordnance Survey (OS) Master Map GML [18], a user defined geometry type was created in the OS schema to be able to have polygons stored as lines in a wheel-topology structure. See Fragment 4.2

```
<osgb:topographicMember>
  <osgb:TopographicArea fid='osgb1000000334399185'>
    <osgb:featureCode>10111</osgb:featureCode>
    <osgb:version>2</osgb:version>
    <osgb:versionDate>2001-11-07</osgb:versionDate>
    <osgb:theme>Land</osgb:theme>
    <osgb:calculatedAreaValue>2361.775704</osgb:calculatedAreaValue>
    <osgb:changeHistory>
      <osgb:changeDate>1999-09-07</osgb:changeDate>
      <osgb:reasonForChange>New</osgb:reasonForChange>
    </osgb:changeHistory>
    <osgb:descriptiveGroup>Natural Environment</osgb:descriptiveGroup>
    <osgb:descriptiveTerm>Coniferous Trees</osgb:descriptiveTerm>
    <osgb:make>Natural</osgb:make>
    <osgb:physicalLevel>50</osgb:physicalLevel>
    <osgb:polygon>
      <osgb:outerBoundaryIs>
        <osgb:Ring orientation='anticlockwise'>
          <osgb:ringMember orientation='-' xlink:href='#osgb1000000334398476'/>
          <osgb:ringMember xlink:href='#osgb1000000334399067'/>
          <osgb:ringMember xlink:href='#osgb1000000334400712'/>
          <osgb:ringMember orientation='-' xlink:href='#osgb1000000334399063'/>
        </osgb:Ring>
      </osgb:outerBoundaryIs>
    </osgb:polygon>
  </osgb:TopographicArea>
</osgb:topographicMember>
```

Fragment 4.2:

So, for the geometry elements and geometry complexTypes users can decide to use the 'default' ones offered in the specification, but they can also choose to define their own. This also holds for the geometry property elements. In this case, the GML specification stipulates the following rules:

- all user defined geometry property types must (directly or indirectly) be subtyped from gml:GeometryPropertyType
- all user defined geometry types must (directly or indirectly) be subtyped from gml:AbstractGeometryType or gml:GeometryCollectionType

4.5 GML flavours

So, while the GML specification offers the basic structure plus a large number of (abstract) element types to be used in a GML implementation, one GML will differ from another in a number of ways.

First of all, every GML implementation will have its own element names (the actual tags in the GML document) for feature collections and features.

Secondly the hierarchy of feature collections and features might be simple (one root collection containing only features) or complex (the root collection contains other feature collections containing other feature collections). As we will see in Chapter 5 the TOP10NL GML prototype has one root collection and seven feature collections under the root collection.

Thirdly the names of the geometry elements might be standard (gml:Polygon, gml:Point etc.) or user defined (for example tgr:TigerPolygon). This is because user organizations can create their own geometry types by extending the gml:AbstractGeometryType (see above). The same goes for the geometry property elements (the 'association' elements between the feature elements and the geometry elements).

Many choices are made when constructing a GML schema. All choices have their advantages and disadvantages. Some of the choices have already been mentioned :

- use of feature collections (beside the root level feature collection)
- standard geometry types and elements or user defined ones

Other issues are:

- store polygons as polygons or have topology using XLink references to boundaries
- define many feature types (and use feature names for classification) or just a few (and use property values for classification)

Compared to the Ordnance Survey Master Map GML the TOP10NL prototype uses more feature collections (the Master Map GML only has one feature collection). Compared to the GML prototype of the American Census Bureau however the TOP10NL prototype has a lot less feature types [6]. And while both the Ordnance Survey and the Census Bureau have user defined geometry types, the TOP10NL only uses the standard geometry from the GML specification.

Chapter 5

Characteristics of the TOP10NL GML prototype

We will first give a short description of the XML 'tree' in the TOP10NL GML data documents. After that we will make some comments on the application schema (the XML Schema document tdn_strict2.1.xsd) and on the choices made in the design of the prototype.

5.1 GML documents: elements and XML 'tree'

Fragment 5.1 shows the basic structure of the TOP10NL GML prototype. Apart from the root level feature collection ('Top10Themas'), there are seven other feature collections:

- RuimtelijkeObjecten
- Gebouwen
- InrichtingsElementen
- FunctioneleGebieden
- AdministratieveGebieden
- BeheersGebieden
- GeografischeGebieden.

The 'RuimtelijkeObjecten' feature collection contains four feature types: SpoorbaanDeel, WegDeel, WaterDeel en Terrein. These four types are in one collection because together they form a partitioning of the space, without 'holes' and almost without overlap (the exception being tunnels, bridges, fly-overs and aqueducts). The other six feature collections are more homogeneous, they only contain one feature type each.

In GML there always has to be an instance of an association type to connect a feature collection and a feature, so we see also a number of association elements in the

TOP10NL GML tree: 'top10ThemasMember', 'ruimtelijkeObjectenMember', 'gebouwenMember' etc. An alternative would have been to use the standard 'gml:featureMember' element for this purpose. In the first GML prototype there were two versions: one with gml:featureMember, the other with user defined association types (the 'strict' version). For the second GML prototype it was decided only to develop a strict version, with user defined featureAssociation types. Advantage of this solution is that validating software can check whether or not a feature instance is in the 'right' collection. With the use of the default gml:featureMember this is not possible.

```
<?xml version="1.0" encoding="iso-8859-1"?>

<tdn:Top10Themas >
  <tdn:top10ThemasMember>
    <tdn:RuimtelijkeObjecten>

      <tdn:ruimtelijkeObjectenMember>
        <tdn:SpoorbaanDeel> ... </tdn:SpoorbaanDeel>
      </tdn:ruimtelijkeObjectenMember>

      <tdn:ruimtelijkeObjectenMember>
        <tdn:WegDeel> ... </tdn:WegDeel>
      </tdn:ruimtelijkeObjectenMember>

      <tdn:ruimtelijkeObjectenMember>
        <tdn:WaterDeel> ... </tdn:WaterDeel>
      </tdn:ruimtelijkeObjectenMember>

      <tdn:ruimtelijkeObjectenMember>
        <tdn:Terrein> ... </tdn:Terrein>
      </tdn:ruimtelijkeObjectenMember>

    </tdn:RuimtelijkeObjecten>
  </tdn:top10ThemasMember>

  <tdn:top10ThemasMember>
    <tdn:Gebouwen>

      <tdn:gebouwenMember>
        <tdn:Gebouw> ... </tdn:Gebouw>
      </tdn:gebouwenMember>

    </tdn:Gebouwen>
  </tdn:top10ThemasMember>

  <tdn:top10ThemasMember>
    <tdn:InrichtingsElementen>

      <tdn:inrichtingsElementenMember>
        <tdn:InrichtingsElement> ... </tdn:InrichtingsElement>
      </tdn:inrichtingsElementenMember>

    </tdn:InrichtingsElementen>
  </tdn:top10ThemasMember>

  <tdn:top10ThemasMember>
```

```
    <tdn:FunctioneleGebieden>

      <tdn:functioneleGebiedenMember>
        <tdn:FunctioneelGebied> ... </tdn:FunctioneelGebied>
      </tdn:functioneleGebiedenMember>

    </tdn:FunctioneleGebieden>
  </tdn:top10ThemasMember>

  <tdn:top10ThemasMember>
    <tdn:AdministratieveGebieden>

      <tdn:administratieveGebiedenMember>
        <tdn:AdministratiefGebied> ... </tdn:AdministratiefGebied>
      </tdn:administratieveGebiedenMember>

    </tdn:AdministratieveGebieden>
  </tdn:top10ThemasMember>

  <tdn:top10ThemasMember>
    <tdn:BeheersGebieden>

      <tdn:beheersGebiedenMember>
        <tdn:BeheersGebied> ... </tdn:BeheersGebied>
      </tdn:beheersGebiedenMember>

    </tdn:BeheersGebieden>
  </tdn:top10ThemasMember>

  <tdn:top10ThemasMember>
    <tdn:GeografischeGebieden>

      <tdn:geografischeGebiedenMember>
        <tdn:GeografischGebied> ... </tdn:GeografischGebied>
      </tdn:geografischeGebiedenMember>

    </tdn:GeografischeGebieden>
  </tdn:top10ThemasMember>

</tdn:Top10Themas>
```

Fragment 5.1: Hierarchy of feature collections and features

Fragment 5.1 only shows the hierarchy of feature collections and features in the TOP10NL prototype. In Appendix A.3 there is an example GML document that shows more detail.

For the modeling of the geometry in the TOP10NL GML it was decided only to use the standard GML geometry types. In version 2.1.2 these are: gml:Point, gml:LineString, gml:LinearRing, gml:Polygon, gml:Box, and the multi-geometry types gml:MultiPoint, gml:MultiLineString and gml:MultiPolygon.

There were also no user defined geometry property types introduced (the 'association' elements between a feature and its geometry). The two geometry property element names

used were: gml:polygonProperty and gml:geometryProperty. The use of gml:geometryProperty needs some explanation, because it is a very 'loose' type: it can contain all kinds of geometry, from gml:Point to gml:MultiPolygon. The prime reason to use gml:geometryProperty in the case of tdn:WegDeel, tdn:SpoorbaanDeel and tdn:WaterDeel was the fact that these infrastructure feature types have a polygon geometry (for the area) as their first geometry plus either a point or a line geometry as their second geometry (for either intersection node or centerline).

```
<tdn:WegDeel fid="TOP10.102502">
  <tdn:top10_id>2100309</tdn:top10_id>
  <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000017"/>
  <tdn:object_begindatum>2001-12-11T11:38:15+02:00</tdn:object_begindatum>
  <tdn:versienummer>1</tdn:versienummer>
  <tdn:versie_begindatum>2001-12-11T11:38:15+02:00</tdn:versie_begindatum>
  <tdn:dimensie>2D</tdn:dimensie>
  <tdn:tdncode>3103</tdn:tdncode>
  <tdn:type>Kruising</tdn:type>
  <tdn:toegankelijkheid>Openbaar</tdn:toegankelijkheid>
  <tdn:status>In gebruik</tdn:status>
  <gml:polygonProperty>
    <gml:Polygon srsName="EPSG:28992">
      <gml:outerBoundaryIs>
<gml:LinearRing>
  <gml:coordinates>
    158126.201,433638.278 158137.348,433631.177 158144.191,433645.798
    158132.739,433653.038 158126.201,433638.278
  </gml:coordinates>
</gml:LinearRing>
      </gml:outerBoundaryIs>
    </gml:Polygon>
  </gml:polygonProperty>
  <gml:geometryProperty>
    <gml:Point srsName="EPSG:28992">
      <gml:coordinates>
158134.856,433641.758
      </gml:coordinates>
    </gml:Point>
  </gml:geometryProperty>
  <tdn:wegtype>Regionale weg</tdn:wegtype>
  <tdn:hoofdverkeersgebruik>Gemengd verkeer</tdn:hoofdverkeersgebruik>
  <tdn:fysiek_voorkomen>Overig</tdn:fysiek_voorkomen>
  <tdn:kruisingstype>Overig</tdn:kruisingstype>
  <tdn:verhardingsbreedteklasse>&gt;7m</tdn:verhardingsbreedteklasse>
  <tdn:verhardingsbreedte>Onbekend</tdn:verhardingsbreedte>
  <tdn:verhardingstype>Verhard</tdn:verhardingstype>
  <tdn:verhardingsmateriaal>Onbekend</tdn:verhardingsmateriaal>
  <tdn:aantal_rijstroken>Onbekend</tdn:aantal_rijstroken>
  <tdn:rijrichting>Tweerichting</tdn:rijrichting>
  <tdn:hoogteniveau>0</tdn:hoogteniveau>
  <tdn:straatnaam>Voor de Kijkuit</tdn:straatnaam>
  <tdn:straatnaam>Kijkuit</tdn:straatnaam>
  <tdn:straatnaam>Bevrijdingslaan</tdn:straatnaam>
  <tdn:straatnaam>Hovenierslaantje</tdn:straatnaam>
  <tdn:wegnummer>Onbekend</tdn:wegnummer>
</tdn:WegDeel>
```

Fragment 5.2: Feature instance of WegDeel

Also in the case of the 'InrichtingsElement' feature type the generic element name 'gml:geometryProperty' was used because an InrichtingsElement object can have either a point or a line geometry. In the case of the four 'gebieden' feature types (administrative and other kinds of areas) gml:geometryProperty was used because there could be both polygon and multi-polygon geometries, and in the case of the 'Gebouw' feature type yet another mix is possible: either a polygon or a point.

In all these cases of heterogeneous geometries, only the gml:geometryProperty element can be used (or a user defined type that inherits from gml:GeometryPropertyType). The use of this generic data type has its drawback however: it is not possible to enforce e.g. the constraint that the second geometry of a road is always either a line or a point but never a polygon (the first geometry is already a polygon, see the example of WegDeel). If and when gml:geometryProperty is to be used is therefore one of the issues that needs to be discussed for the final version of the TOP10NL GML data model. (At a higher conceptual level, the modeling of the center-lines and intersection-nodes in the case of infrastructure objects may also need reconsideration.)

5.2 Data model: UML and XML Schema

When developing the TOP10NL GML prototype we built upon the conceptual data model for the TOP10Vector data developed in an earlier stage of the project (see Chapter 1). The inheritance structure of types and subtypes in the UML diagrams was followed closely in the design of the TOP10NL GML application schema (for UML and GML see [11]).

In the conceptual model there is a Top10Object class, with the common properties of all the relevant object types. In the XML application schema this is also the case, see Fragment 5.3.

```
<!-- ============================================================
       Type definition of Top10ObjectType (= inherited and extended by
                                            all TOP10 feature types)
  ============================================================== -->
  <complexType name="Top10ObjectType" abstract="true">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="top10_id" type="integer"/>
               <element ref="tdn:bronRef"/>
               <group ref="tdn:Temporeel"/>
               <element name="dimensie" type="tdn:dimensie"/>
               <element name="tdncode" type="integer"/>
            </sequence>
      </extension>
    </complexContent>
  </complexType>

  <group name ="Temporeel">
```

```
      <sequence>
            <element name="ontstaan_uit" type="integer" minOccurs="0" maxOccurs="unbounded" />
            <element name="object_begindatum" type="dateTime" />
            <element name="versienummer" type="integer" />
            <element name="versie_begindatum" type="dateTime" />
            <element name="versie_einddatum" type="dateTime" minOccurs="0"  />
      </sequence>
  </group>

<!-- ================================================================
       Type definition of InfrastructuurType (= inherited and extended by
       SpoorbaanDeelType, WegDeelType and WaterDeelType)
  ================================================================ -->
  <complexType name="InfrastructuurType" abstract="true">
    <complexContent>
      <extension base="tdn:Top10ObjectType">
            <sequence>
               <element name="type" type="tdn:typeInfra"/>
               <element name="toegankelijkheid" type="tdn:toegankelijkheid"/>
               <element name="status" type="tdn:status"/>
               <element ref="gml:polygonProperty"/>
               <element ref="gml:geometryProperty"/>
            </sequence>
      </extension>
    </complexContent>
  </complexType>


  <element name="WaterDeel" type="tdn:WaterDeelType"
    substitutionGroup="tdn:_RuimtelijkeObjectenFeature"/>
   <complexType name="WaterDeelType">
      <complexContent>
         <extension base="tdn:InfrastructuurType">
            <sequence>
               <element name="watertype" type="tdn:watertype"/>
               <element name="breedteklasse" type="string"/>
               <element name="breedte" type="tdn:numeriekOnb" minOccurs="0"/>
               <element name="hoofdafwatering" type="tdn:hoofdafwatering"/>
               <element name="zoutgehalte" type="tdn:zoutgehalte"/>
               <element name="fysiek_voorkomen" type="tdn:fysiek_voorkomenWater"/>
               <element name="gebruik" type="tdn:gebruik"/>
               <element name="stroomrichting" type="tdn:stroomrichting"/>
               <element name="hoogteniveau" type="integer" minOccurs="0"/>
               <element name="naam" type="string" minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
  </complexType>
```

Fragment 5.3:

As has been mentioned in Chapter 4, the technique of having (abstract) supertypes and subtypes that inherit properties from the supertype plays a role at design time: in the

UML diagrams and in the XML Schema document. It also plays a role when the GML document is validated (compared with its schema), but it is not directly reflected in the GML data document itself: for example, in the GML XML tree there is no Top10Object element tag. Advantage of using types like 'Top10ObjectType' is, that it makes the link between the conceptual model and the GML application schema more clear. It also makes the schema itself easier to maintain: when a property has to be added to all feature types, this only has to be changed in one place in the XML Schema document.

One aspect of the UML class diagrams could not be copied one-to-one into the XML Schema model: in UML multiple inheritance is possible, but in XML Schema a type can only inherit properties from one supertype at the same time. This is for example the case with Top10Object and TemporeelObject (for the version data) (see Figure 2.1). In XML Schema Top10Object cannot have both TemporeelObject and AbstractFeatureType as supertype. When multiple inheritance like this occurred, we used the 'group' construct (see Fragment 5.3, group 'Temporeel'). In this way the properties that in the UML class model are inherited from a super class can be recognized more easily in the XML Schema application schema as having something in common.

5.3 Metadata on object level, use of XLink

One aspect of XML that has not been mentioned yet is the possibility to point at elements in stead of directly 'contain' them as child elements. In this way it is possible to create links to data somewhere else in the same document or even in another document and/or on another server. Two specifications of the W3Consortium are relevant here: XLink [23] and XPointer [24].

In the TOP10NL prototype GML this mechanism is used to have the metadata for each object stored not as child elements of the object, but in a separate object called 'Bron' (=Source). Primary reason is to avoid redundancy, because many topographic objects will have the same metadata. Also these metadata properties are interdependent, e.g. with 'brontype' (method of capture) and 'bronnauwkeurigheid' (precision). In relational databases there would be a separate table because of this, with an n : 1 relationship between topographic object and metadata table.

In Fragment 5.4 we see an example: the WaterDeel object has its metadata in a separate document 'metadata.xml' and what is behind the '#' is the unique ID that can be used by parser software to find the object that contains the metadata properties for this particular WaterDeel instance.

```
<tdn:WaterDeel fid="TOP10.101224">
  <tdn:top10_id>6100003</tdn:top10_id>
  <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000017"/>
```

Fragment 5.4:

Although XLink/XPointer has great potential, there are at this moment only a few parsers (and browsers) that support it automatically (without the need for software developers to write additional code to follow the links). Within GML however it plays an important

role: it is also a way to implement topology, where Nodes and Edges are stored only once, and Faces consist of references to Edges etc. The Ordnance Survey Master Map GML also uses XLink, in their wheel-topology polygon type. (Also in GML 3.0 topology is implemented by using Xlink references.)

5.4 Constraints on domain values

The XML Schema language offers several possibilities to specify constraints on the values that are allowed for a property. The technique used is to define a data type as subtype of e.g. the 'string' data type and then restrict the possible values. For the TOP10NL prototype we used both `<enumeration>` and `<pattern>`.

a) enumeration

```
<simpleType name="fysiek_voorkomenWater">
    <restriction base="string">
       <enumeration value="In sluis"/>
       <enumeration value="Op aquaduct"/>
       <enumeration value="In duiker"/>
       <enumeration value="In grondduiker"/>
       <enumeration value="Overig"/>
    </restriction>
</simpleType>
```

b) pattern

With the `<pattern value="...">` construct it is possible to specify allowed values using format strings comparable to regular expressions.

```
<simpleType name="numeriekOnb">
   <restriction base="string">
      <pattern value="[0-9]*|Onbekend"/>
   </restriction>
</simpleType>
```

5.5 Cardinality of properties

An important constraint in a data model is the cardinality (or multiplicity) of a class, i.e. the minimum and maximum number of occurrences that is allowed for an object or property. In XML Schema the multiplicity of an element is specified by the attributes 'minOccurs' and 'maxOccurs'. The default for both is '1'.

In the TOP10NL GML prototype some of the non-spatial feature properties have a maxOccurs='unbounded', while a minOccurs of 0 is also possible.

a. maxOccurs='unbounded'

Because GML is XML, it is possible to have multiple occurrences of an object property (a list of values for an attribute as part of one object instance). This is not possible in e.g. the relational database model where attribute values are atomic and repeating groups are implemented in separate tables as part of the 'normalization' process (e.g. [7]).

The 'ontstaan_uit' ('originated_from') property is an example: an object can have zero or more ancestor-objects from which it is created, e.g. in case of the merging of these objects into one new object:

```
<tdn:WegDeel fid="TOP10.150044">
  <tdn:top10_id>2105003</tdn:top10_id>
  <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000019"/>
  <tdn:ontstaan_uit>6100004</tdn:ontstaan_uit>
  <tdn:ontstaan_uit>2100012</tdn:ontstaan_uit>
  <tdn:ontstaan_uit>5100212</tdn:ontstaan_uit>
  <tdn:object_begindatum>2002-02-22T14:38:44+02:00</tdn:object_begindatum>
   ...
```

Or, another example, a road can have more than one 'wegnummer' (road number):

```
   ...
  <tdn:straatnaam>Onbekend</tdn:straatnaam>
  <tdn:wegnummer>A12</tdn:wegnummer>
  <tdn:wegnummer>E25</tdn:wegnummer>
  <tdn:wegnummer>E30</tdn:wegnummer>
</tdn:WegDeel>
```

b. minOccurs='0': empty element or no element

In XML it is possible to leave out a (child) element that has a null value. Instead of having an 'empty' element (a begin and end tag but no content in between), the element is skipped alltogether. Because according to the conceptual data model some properties could have a cardinality of 0, we had to make a choice: either to have empty elements in that case, e.g. `<tdn:versie_einddatum></tdn:versie_einddatum>`, or skip these elements.

In the case of 'normal', non-spatial properties we chose to skip empty elements. The main reason was that properties with a numeric data type (type = 'integer') could not be validated by some parsers when they had no value (were empty). The validating software expected a zero in that case. Also when there was a user defined enumeration simpleType this caused problems, because the combination of having a list of allowed values and also allow for empty elements is difficult to implement with enumeration types in XML Schema.

So, in the schema for the TOP10NL prototype there is a minOccurs="0" for a number of elements, and in the GML data documents properties with a null value (that would become empty elements) were left out (see 5.5).

```
<tdn:WegDeel fid="TOP10.101784">
  <tdn:top10_id>2110084</tdn:top10_id>
  <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000017"/>
  <tdn:object_begindatum>2001-12-11T11:38:15+02:00</tdn:object_begindatum>
  <tdn:versienummer>1</tdn:versienummer>
  <tdn:versie_begindatum>2001-12-11T11:38:15+02:00</tdn:versie_begindatum>
  <tdn:dimensie>2D</tdn:dimensie>
```

```
<tdn:tdncode>3530</tdn:tdncode>
        ...
```

Fragment 5.5: Leave out empty elements (tdn:ontstaan_uit, tdn:versie_einddatum)

In the case of the spatial properties on the other hand empty elements were not taken out. So, in case of gml:polygonProperty and gml:geometryProperty, it is possible to have empty elements. This inconsistency is one of the issues that need to be discussed and decided upon for the final version of the TOP10NL GML data model.

Chapter 6

Evaluation and Conclusion

Although the TOP10NL GML prototype complies with the GML specification (it is valid GML), there are some aspects of the data structure that at the moment could pose a problem for existing standard software. Much depends on the software and database environment of the 'receiving' organization (native data storage format, software techniques used to process the GML). One of the issues could be: having more than one geometry type per feature (e.g. a Road with both a polygon geometry and a line or point geometry). Also having multiple occurrences in the case of 'straatnaam' (streetname) or 'wegnummer' (road number) (a list of values for this property) could be an issue.

In this Chapter we will go into strategies to parse GML data. We will also report on the 2nd GML Relay, held in December 2002. At the end of this Chapter there is an overall conclusion that summarizes the report as a whole.

6.1 GML parsing software

Reading and processing GML data can follow three strategies. The simplest way is to write data specific software, tailored to a specific GML data model. This is the way most software for the Ordnance Survey Master Map GML was built.

The second possibility is to write software that reads and analyzes the XML Schema document (*.xsd) first, before the actual GML data is processed (see e.g. [5]). In that way user defined geometry types can be discovered and 'understood' by the software, under the condition that these user defined geometry types are subtypes of the standard GML types defined in the specification. (An additional condition would be that this subtyping is as precise as possible: a user defined polygon type should be subtyped from gml:Polygon or gml:MultiPolygon, and not from gml:AbstractGeometryType.) Especially when there is more than one XML Schema document involved (as in the Ordnance Survey Master Map GML), the software for analyzing the XML Schema can become complicated. On the other hand, for 'human eyes' analyzing the data model written down in several .xsd documents can also be time consuming.

Somewhere in between is the third way: import or visualization software that is based on certain assumptions about the data, most importantly the assumption that there are no user defined geometry types.

Because the TOP10NL GML prototype does not have user defined geometry elements but only uses the standard GML names for point, polygon etc., it is in principle possible to process the TOP10NL GML data without knowledge of the TOP10NL XML Schema (the third strategy, see above). A possible exception has to do with the minOccurs='0' issue mentioned in section 5.5. When importing software expects a fixed number of child elements for a feature instance, reading only the GML data and not the XML application schema could result in errors or in loss of information (e.g. when the first feature instance is used to establish the data structure of all the features of that feature type).

6.2 Visualization

For the visualization of GML data of course more (semantic) information is needed (styling, classification, stack order ('hoogteniveau') in case of tunnels, bridges etc.), but this is inherent to the GML conceptual model: only the data is included, not the presentation aspects. The Dutch Topographic Service has started a new project for the cartographic aspects of the new TOP10NL product. This follow-up project about the cartographic presentation of GML will be carried out in cooperation with ITC Enschede.

6.3 GML Relay

On December 13th, 2002 the Netherlands Society for Earth Observation and Geo-Informatics (the KvAG) organized its 2nd GML Relay in Emmen, at the office of the Dutch Topographic Service. Purpose of the event was to show that interoperability between different software products based on the exchange of GML data documents works. As input data for the Relay one of the TOP10NL prototype GML documents was used (area of Tiel). The Relay was therefore also a good opportunity to test whether or not the TOP10NL GML prototype (data and schema) could be successfully imported and used (viewing, editing and exporting back into GML) in existing geo-software environments. Seven companies accepted the invitation to show the GML capabilities of their products: Intergraph (GeoMedia), eXQte (reseller of FME), Bentley (MicroStation), ESRI (ArcGIS), Snowflake (GO Loader), Oracle (Oracle Spatial) and Laser-Scan (Radius Topology). Some of these companies teamed up and combined software to import and export the GML data: eXQte and Bentley used FME in combination with MicroStation, Snowflake and Oracle used GO Loader in combination with Oracle Spatial. Laser-Scan used a combination of GO Loader, Oracle Spatial, it's own Radius Topology and Intergraph's GeoMedia.

This 2nd Relay (the 1st Relay was in 2001) was successful for two reasons: not only were seven companies present, but - and that was the real test - they all succeeded in reading the TOP10NL GML data into their respective software environments. Intergraph, Bentley and Laser-Scan performed a few edits (creation of new buildings, removal of part of a river, update of a boundary) on the imported GML features.

The data also had to be exported again into valid GML documents. When analysing the export files the following observations can be made:

- For visualization it is important that the bounding box (the minimal extent) is also provided in the GML document. Some of the export files did not contain bounding box

coordinates (lower left, upper right), in another case the extent was not correct.

- The same goes for the reference to the Spatial Reference System used. For the Dutch 'RijksDriehoeksmeeting' this is EPSG:28992. Some of the export files did not contain this srs-information. The export from Oracle had the reference, but not in the EPSG notation (but in the internal Oracle notation (SDO:90112)).

- In some cases there was loss of information somewhere in the import or export process: in the original TOP10NL start document it is possible to have multiple occurrences of a streetname, or of 'originated-from' (see section 5.5). These multiple occurrences where not there in the export file(s).

- The original TOP10NL start document contained unique 'fids' for each feature. In some export files these fids had disappeared, or they were not unique (or zero).

- Some products could handle the 'object oriented' nature of the basic GML conceptual model better than others. In the GML output exported from GeoMedia and Oracle the data structure of the TOP10NL GML was left intact: e.g. each Road (WegDeel) not only has a polygon geometry, but also either a line or a point geometry.

Despite of these issues, as a whole the 2nd GML Relay proved that support for GML in geo-software has much increased since 2001. The most important difference between the various software products seems to be support for more than one geometry type per feature object or not. This difference can of course disappear in future releases of the (beta) software that was used at the Relay. Another conclusion is, that the more 'advanced' characteristics of the TOP10NL GML prototype could more easily be handled by import/export software that uses the XML schema document to 'understand' the data structure.

6.4 Conclusion

In this report we described the second GML prototype for the new TOP10NL topographic data 1:10,000. In Chapter 2 we discussed changes with respect to the first prototype that were a result of changes in the conceptual data model. Some of these changes also influenced the conversion process (Chapter 3). In Chapter 4 we gave a short introduction to XML and GML terminology. In Chapter 5 we focussed on the main characteristics of the prototype and did so by showing fragments from both the XML application schema and the GML data document(s). Sometimes implementation choices had to be made, e.g. having empty elements in case of a null value or skipping these elements. We mentioned some other issues, that were more directly related to the conceptual data model, e.g. the fact that the gml:geometryProperty element sometimes contains a point geometry, sometimes a line geometry. One characteristic of the prototype has not explicitly been mentioned yet, but could be part of the discussion about the final GML model: the use of feature collections ('RuimtelijkeObjecten', 'Gebouwen') as containers for the actual features ('WegDeel', 'Gebouw'). When these collections would have metadata or other information at 'collection' level, this is a necessary way to construct the GML. In the case of this prototype however no extra information is stored at the collection level, so one could decide for a GML model with no feature collections except the 'root' collection.

When the prototype was built the GML specification was still at version 2.0. The new

GML 3.0 version offers many new possibilities: there are new geometry types (Bezier curve, Bspline, Circle etc.), topology is introduced, and there is more in the specification about metadata and temporal aspects.

The OpenGIS GML specification is still in a flux. Tests and discussions in working groups will probably result in changes. Because of the extensible nature of GML (based on XML and XML Schema) this will not harm the development of products like the new TOP10NL 1:10.000, but - to the contrary - only create new possibilities to define a GML model that can capture the new conceptual model of TOP10NL.

Appendix A

Schemas and GML example

A.1 tdn_strict2.1.xsd

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- File: tdn_strict2.0.xsd  -->
<schema targetNamespace="http://www.gdmc.nl/tdn"
        xmlns:tdn="http://www.gdmc.nl/tdn"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:gml="http://www.opengis.net/gml"
        xmlns="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified"
        version="1.0">

  <annotation>
    <appinfo>tdn_strict.xsd v2.1</appinfo>
    <documentation xml:lang="en">
      GML prototype schema for the Dutch Topographic Service 1:10.000 data.
      This 'strict' version has:
      - simple (enumeration) types to restrict the possible values of attribute data
      - feature association types to restrict the membership of feature collections
      Changes between v2.0 and v2.1:
      - tdn:BronRefType is no longer a subtype of gml:FeatureAssociationType
      - tdn:Bron does no longer have a reference to a substitutionGroup
      - there is a 'choice' in the root feature collection tdn:Top10ThemasType
    </documentation>
  </annotation>

  <!-- import constructs from the GML Feature and Geometry schemas -->
  <import namespace="http://www.opengis.net/gml" schemaLocation="feature.xsd"/>

  <!-- ============================================================
       Declarations for Top10Themas (= root element)
  ============================================================== -->
  <element name="Top10Themas" type="tdn:Top10ThemasType" substitutionGroup="gml:_FeatureCollection"/>
  <complexType name="Top10ThemasType">
      <complexContent>
        <extension base="gml:AbstractFeatureCollectionType">
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="tdn:top10ThemasMember"/>
```

```
          <element ref="tdn:Bron"/>
        </choice>
      </extension>
    </complexContent>
</complexType>

<element name="top10ThemasMember" type="tdn:Top10ThemasMemberType"/>
<complexType name="Top10ThemasMemberType">
  <complexContent>
    <restriction base="gml:FeatureAssociationType">
      <sequence minOccurs="0">
        <element ref="tdn:_Top10ThemasFeature"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>

<element name="_Top10ThemasFeature" type="gml:AbstractFeatureType" abstract="true"
  substitutionGroup="gml:_Feature"/>

<!-- ============================================================
     Type definition of Top10ThemaType (= inherited and extended by
                                          all Top10 collections)
============================================================== -->
<complexType name="Top10ThemaType" abstract="true">
  <complexContent>
    <restriction base="gml:AbstractFeatureCollectionType">
      <sequence>
          <element ref="gml:description" minOccurs="0"/>
          <element ref="gml:name" minOccurs="0"/>
          <element ref="gml:boundedBy"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>

<!-- ============================================================
     Type definition of Top10ObjectType (= inherited and extended by
                                           all Top10 feature types)
============================================================== -->
<complexType name="Top10ObjectType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
          <sequence>
             <element name="top10_id" type="integer"/>
             <element ref="tdn:bronRef"/>
             <group ref="tdn:Temporeel"/>
             <element name="dimensie" type="tdn:dimensie"/>
             <element name="tdncode" type="integer"/>
          </sequence>
    </extension>
  </complexContent>
</complexType>

<group name ="Temporeel">
```

```
    <sequence>
          <element name="ontstaan_uit" type="integer" minOccurs="0" maxOccurs="unbounded" />
          <element name="object_begindatum" type="dateTime" />
          <element name="versienummer" type="integer" />
          <element name="versie_begindatum" type="dateTime" />
          <element name="versie_einddatum" type="dateTime" minOccurs="0"  />
    </sequence>
</group>

<element name="bronRef" type="tdn:BronRefType" />
<complexType name="BronRefType">
  <!--complexContent>
    <restriction base="gml:FeatureAssociationType"-->
      <sequence minOccurs="0" maxOccurs="1">
        <element ref="tdn:Bron"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <!--/restriction>
  </complexContent-->
</complexType>

<!--element name="Bron" type="tdn:BronType" substitutionGroup="tdn:_Top10ThemasFeature"/-->
<element name="Bron" type="tdn:BronType"/>
<complexType name="BronType">
      <sequence>
         <element name="bron_id" type="integer"/>
         <element name="brontype" type="tdn:brontype"/>
         <element name="bronbeschrijving" type="string"/>
         <element name="bronactualiteit" type="string"/>
         <element name="bronnauwkeurigheid" type="string"/>
      </sequence>
      <attribute name="id" type="ID"/>
</complexType>


<!-- ============================================================
     Type definition of InfrastructuurType (= inherited and extended by
     SpoorbaanDeelType, WegDeelType and WaterDeelType)
============================================================== -->
<complexType name="InfrastructuurType" abstract="true">
  <complexContent>
    <extension base="tdn:Top10ObjectType">
          <sequence>
             <element name="type" type="tdn:typeInfra"/>
             <element name="toegankelijkheid" type="tdn:toegankelijkheid"/>
             <element name="status" type="tdn:status"/>
             <element ref="gml:polygonProperty"/>
             <element ref="gml:geometryProperty"/>
          </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- ============================================================
     Declarations for RuimtelijkeObjecten (= one of the themes)
============================================================== -->
```

```
<element name="RuimtelijkeObjecten" type="tdn:RuimtelijkeObjectenType"
  substitutionGroup="tdn:_Top10ThemasFeature"/>
<complexType name="RuimtelijkeObjectenType">
    <complexContent>
       <extension base="tdn:Top10ThemaType">
         <sequence minOccurs="0" maxOccurs="unbounded">
          <element ref="tdn:ruimtelijkeObjectenMember"/>
         </sequence>
       </extension>
    </complexContent>
</complexType>

<element name="ruimtelijkeObjectenMember" type="tdn:RuimtelijkeObjectenMemberType"/>
<complexType name="RuimtelijkeObjectenMemberType">
  <annotation>
  </annotation>
  <complexContent>
    <restriction base="gml:FeatureAssociationType">
      <sequence minOccurs="0">
        <element ref="tdn:_RuimtelijkeObjectenFeature"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>

<element name="_RuimtelijkeObjectenFeature" type="gml:AbstractFeatureType" abstract="true"
  substitutionGroup="gml:_Feature"/>

<element name="SpoorbaanDeel" type="tdn:SpoorbaanDeelType"
  substitutionGroup="tdn:_RuimtelijkeObjectenFeature"/>
<complexType name="SpoorbaanDeelType">
    <complexContent>
       <extension base="tdn:InfrastructuurType">
          <sequence>
             <element name="verkeersgebruik" type="tdn:verkeersgebruik"/>
             <element name="fysiek_voorkomen" type="tdn:fysiek_voorkomenSpoor"/>
             <element name="spoorbreedte" type="tdn:spoorbreedte"/>
             <element name="aantal_sporen" type="integer"/>
             <element name="functie" type="tdn:functieSpoor" minOccurs="0"/>
             <element name="elektrificatie" type="tdn:elektrificatie"/>
             <element name="hoogteniveau" type="integer" minOccurs="0"/>
             <element name="naam" type="string" minOccurs="0"/>
          </sequence>
       </extension>
    </complexContent>
</complexType>

<element name="WegDeel" type="tdn:WegDeelType"
  substitutionGroup="tdn:_RuimtelijkeObjectenFeature"/>
<complexType name="WegDeelType">
    <complexContent>
       <extension base="tdn:InfrastructuurType">
          <sequence>
             <element name="wegtype" type="tdn:wegtype"/>
             <element name="hoofdverkeersgebruik" type="tdn:hoofdverkeersgebruik"/>
```

```
            <element name="fysiek_voorkomen" type="tdn:fysiek_voorkomenWeg"/>
            <element name="kruisingstype" type="tdn:kruisingstype"/>
            <element name="verhardingsbreedteklasse" type="string"/>
            <element name="verhardingsbreedte" type="tdn:numeriekOnb"/>
            <element name="verhardingstype" type="tdn:verhardingstype"/>
            <element name="verhardingsmateriaal" type="tdn:verhardingsmateriaal"/>
            <element name="aantal_rijstroken" type="tdn:numeriekOnb"/>
            <element name="rijrichting" type="tdn:rijrichting"/>
            <element name="hoogteniveau" type="integer" minOccurs="0"/>
            <element name="straatnaam" type="string" minOccurs="0" maxOccurs="unbounded"/>
            <element name="wegnummer" type="string" minOccurs="0" maxOccurs="unbounded"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>

<element name="WaterDeel" type="tdn:WaterDeelType"
  substitutionGroup="tdn:_RuimtelijkeObjectenFeature"/>
 <complexType name="WaterDeelType">
   <complexContent>
      <extension base="tdn:InfrastructuurType">
         <sequence>
            <element name="watertype" type="tdn:watertype"/>
            <element name="breedteklasse" type="string"/>
            <element name="breedte" type="tdn:numeriekOnb" minOccurs="0"/>
            <element name="hoofdafwatering" type="tdn:hoofdafwatering"/>
            <element name="zoutgehalte" type="tdn:zoutgehalte"/>
            <element name="fysiek_voorkomen" type="tdn:fysiek_voorkomenWater"/>
            <element name="gebruik" type="tdn:gebruik"/>
            <element name="stroomrichting" type="tdn:stroomrichting"/>
            <element name="hoogteniveau" type="integer" minOccurs="0"/>
            <element name="naam" type="string" minOccurs="0" maxOccurs="unbounded"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>

<element name="Terrein" type="tdn:TerreinType"
  substitutionGroup="tdn:_RuimtelijkeObjectenFeature"/>
<complexType name="TerreinType">
   <complexContent>
      <extension base="tdn:Top10ObjectType">
         <sequence>
            <element name="landgebruik" type="tdn:landgebruik"/>
            <element name="fysiek_voorkomen" type="tdn:fysiek_voorkomenTerrein"/>
            <element name="toegankelijkheid" type="tdn:toegankelijkheid"/>
            <element name="voorkomen" type="tdn:voorkomen" minOccurs="0"/>
            <element ref="gml:polygonProperty"/>
            <element name="hoogteniveau" type="integer" minOccurs="0"/>
            <element name="naam" type="string" minOccurs="0"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>

<!-- ============================================================
```

```
     Declarations for Gebouw (= one of the themes)
 ========================================================== -->
<element name="Gebouwen" type="tdn:GebouwenType"
  substitutionGroup="tdn:_Top10ThemasFeature"/>
<complexType name="GebouwenType">
    <complexContent>
       <extension base="tdn:Top10ThemaType">
         <sequence minOccurs="0" maxOccurs="unbounded">
          <element ref="tdn:gebouwenMember"/>
         </sequence>
       </extension>
    </complexContent>
</complexType>

<element name="gebouwenMember" type="tdn:GebouwenMemberType"/>
<complexType name="GebouwenMemberType">
  <annotation>
  </annotation>
  <complexContent>
    <restriction base="gml:FeatureAssociationType">
      <sequence minOccurs="0">
        <element ref="tdn:Gebouw"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>

<element name="Gebouw" type="tdn:GebouwType"
  substitutionGroup="gml:_Feature"/>
<complexType name="GebouwType">
    <complexContent>
       <extension base="tdn:Top10ObjectType">
          <sequence>
             <element name="type" type="tdn:typeGebouw"/>
             <element name="functie" type="tdn:functieGebouw"/>
             <element name="hoogteklasse" type="tdn:hoogteklasse"/>
             <element name="hoogte" type="tdn:numeriekOnb"/>
             <element name="status" type="tdn:status"/>
             <element ref="gml:geometryProperty"/>
             <element name="hoogteniveau" type="integer" minOccurs="0"/>
             <element name="naam" type="string" minOccurs="0"/>
          </sequence>
       </extension>
    </complexContent>
</complexType>

<!-- ============================================================
     Declarations for InrichtingsElementen (= one of the themes)
 ========================================================== -->
<element name="InrichtingsElementen" type="tdn:InrichtingsElementenType"
  substitutionGroup="tdn:_Top10ThemasFeature"/>
<complexType name="InrichtingsElementenType">
    <complexContent>
       <extension base="tdn:Top10ThemaType">
         <sequence minOccurs="0" maxOccurs="unbounded">
```

```
          <element ref="tdn:inrichtingsElementenMember"/>
        </sequence>
      </extension>
    </complexContent>
</complexType>


<element name="inrichtingsElementenMember" type="tdn:InrichtingsElementenMemberType"/>
<complexType name="InrichtingsElementenMemberType">
  <annotation>
  </annotation>
  <complexContent>
    <restriction base="gml:FeatureAssociationType">
      <sequence minOccurs="0">
        <element ref="tdn:InrichtingsElement"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>

<element name="InrichtingsElement" type="tdn:InrichtingsElementType"
  substitutionGroup="gml:_Feature"/>
<complexType name="InrichtingsElementType">
    <complexContent>
       <extension base="tdn:Top10ObjectType">
          <sequence>
             <element name="type" type="tdn:typeInrichtingsElement"/>
             <element name="functie" type="string" minOccurs="0"/>
             <element name="hoogte" type="tdn:numeriekOnb"/>
             <element name="status" type="tdn:status"/>
             <element ref="gml:geometryProperty"/>
             <element name="hoogteniveau" type="integer" minOccurs="0"/>
             <element name="naam" type="string" minOccurs="0"/>
             <element name="nummer" type="tdn:numeriekOnb" minOccurs="0"/>
          </sequence>
       </extension>
    </complexContent>
</complexType>

<!-- ============================================================
     Declarations for FunctioneleGebieden (= one of the themes)
============================================================== -->
<element name="FunctioneleGebieden" type="tdn:FunctioneleGebiedenType"
  substitutionGroup="tdn:_Top10ThemasFeature"/>
<complexType name="FunctioneleGebiedenType">
    <complexContent>
       <extension base="tdn:Top10ThemaType">
         <sequence minOccurs="0" maxOccurs="unbounded">
          <element ref="tdn:functioneleGebiedenMember"/>
         </sequence>
       </extension>
    </complexContent>
</complexType>

<element name="functioneleGebiedenMember" type="tdn:FunctioneleGebiedenMemberType"/>
```

```
<complexType name="FunctioneleGebiedenMemberType">
  <annotation>
  </annotation>
  <complexContent>
    <restriction base="gml:FeatureAssociationType">
      <sequence minOccurs="0">
        <element ref="tdn:FunctioneelGebied"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>

<element name="FunctioneelGebied" type="tdn:FunctioneelGebiedType"
  substitutionGroup="gml:_Feature"/>
<complexType name="FunctioneelGebiedType">
    <complexContent>
       <extension base="tdn:Top10ObjectType">
          <sequence>
             <element name="type" type="tdn:typeFuncGebied"/>
             <element ref="gml:geometryProperty"/>
             <element name="naam" type="string"/>
          </sequence>
       </extension>
    </complexContent>
</complexType>

<!-- ============================================================
     Declarations for Administratieve Gebieden (= one of the themes)
============================================================= -->
<element name="AdministratieveGebieden" type="tdn:AdministratieveGebiedenType"
  substitutionGroup="tdn:_Top10ThemasFeature"/>
<complexType name="AdministratieveGebiedenType">
    <complexContent>
       <extension base="tdn:Top10ThemaType">
         <sequence minOccurs="0" maxOccurs="unbounded">
          <element ref="tdn:administratieveGebiedenMember"/>
         </sequence>
       </extension>
    </complexContent>
</complexType>

<element name="administratieveGebiedenMember" type="tdn:AdministratieveGebiedenMemberType" />
<complexType name="AdministratieveGebiedenMemberType">
  <complexContent>
    <restriction base="gml:FeatureAssociationType">
      <sequence minOccurs="0">
        <element ref="tdn:AdministratiefGebied"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>

<element name="AdministratiefGebied" type="tdn:AdministratiefGebiedType"
  substitutionGroup="gml:_Feature"/>
```

```
<complexType name="AdministratiefGebiedType">
    <complexContent>
       <extension base="tdn:Top10ObjectType">
          <sequence>
             <element name="type" type="tdn:typeAdminGebied"/>
             <element ref="gml:geometryProperty"/>
             <element name="naam" type="string"/>
          </sequence>
       </extension>
    </complexContent>
</complexType>

<!-- ============================================================
     Declarations for BeheersGebieden (= one of the themes)
============================================================== -->
<element name="BeheersGebieden" type="tdn:BeheersGebiedenType"
  substitutionGroup="tdn:_Top10ThemasFeature"/>
<complexType name="BeheersGebiedenType">
    <complexContent>
       <extension base="tdn:Top10ThemaType">
         <sequence minOccurs="0" maxOccurs="unbounded">
          <element ref="tdn:beheersGebiedenMember"/>
         </sequence>
       </extension>
    </complexContent>
</complexType>

<element name="beheersGebiedenMember" type="tdn:BeheersGebiedenMemberType" />

<complexType name="BeheersGebiedenMemberType">
  <complexContent>
    <restriction base="gml:FeatureAssociationType">
      <sequence minOccurs="0">
        <element ref="tdn:BeheersGebied"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>

<element name="BeheersGebied" type="tdn:BeheersGebiedType"
  substitutionGroup="gml:_Feature"/>
<complexType name="BeheersGebiedType">
    <complexContent>
       <extension base="tdn:Top10ObjectType">
          <sequence>
             <element name="type" type="tdn:typeBeheersGebied"/>
             <element ref="gml:geometryProperty"/>
             <element name="naam" type="string"/>
          </sequence>
       </extension>
    </complexContent>
</complexType>

<!-- ============================================================
     Declarations for GeografischeGebieden (= one of the themes)
```

```
================================================================== -->
<element name="GeografischeGebieden" type="tdn:GeografischeGebiedenType"
  substitutionGroup="tdn:_Top10ThemasFeature"/>
<complexType name="GeografischeGebiedenType">
    <complexContent>
       <extension base="tdn:Top10ThemaType">
         <sequence minOccurs="0" maxOccurs="unbounded">
          <element ref="tdn:geografischeGebiedenMember"/>
         </sequence>
       </extension>
    </complexContent>
</complexType>

<element name="geografischeGebiedenMember" type="tdn:GeografischeGebiedenMemberType"/>
<complexType name="GeografischeGebiedenMemberType">
  <annotation>
  </annotation>
  <complexContent>
    <restriction base="gml:FeatureAssociationType">
      <sequence minOccurs="0">
        <element ref="tdn:GeografischGebied"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>

<element name="GeografischGebied" type="tdn:GeografischGebiedType"
  substitutionGroup="gml:_Feature"/>
<complexType name="GeografischGebiedType">
    <complexContent>
       <extension base="tdn:Top10ObjectType">
          <sequence>
             <element name="type" type="tdn:typeGeografGebied"/>
             <element ref="gml:geometryProperty"/>
             <element name="naam" type="string"/>
          </sequence>
       </extension>
    </complexContent>
</complexType>

<!-- ============================================================
     Attribute type definitions (permitted values)
================================================================== -->
<simpleType name="brontype">
    <restriction base="string">
      <enumeration value="Luchtfoto"/>
      <enumeration value="Kaart"/>
      <enumeration value="RD"/>
      <enumeration value="GBKN"/>
      <enumeration value="Overig"/>
      <enumeration value="Onbekend"/>
      <enumeration value="Kadaster"/>
      <enumeration value="Gemeente"/>
      <enumeration value="Genterpreteerde luchtfoto"/>
      <enumeration value="Terreinverkenning"/>
```

```
    </restriction>
</simpleType>

<simpleType name="dimensie">
    <restriction base="string">
      <enumeration value="2D"/>
      <enumeration value="3D"/>
    </restriction>
</simpleType>

<simpleType name="breedteklasse">
    <restriction base="string">
      <enumeration value="&lt; 1m"/>
      <enumeration value="1-3 m"/>
      <enumeration value="3-6 m"/>
      <enumeration value="6-12 m"/>
      <enumeration value="12-20 m"/>
      <enumeration value="20-50 m"/>
      <enumeration value="50-150 m"/>
      <enumeration value="150-300"/>
      <enumeration value="300-450 m"/>
      <enumeration value="&gt; 450m"/>
   </restriction>
</simpleType>

<simpleType name="elektrificatie">
    <restriction base="string">
      <enumeration value="Gelektrificeerd"/>
      <enumeration value="Niet gelektrificeerd"/>
    </restriction>
</simpleType>

<simpleType name="functieGebouw">
    <restriction base="string">
      <enumeration value="Gemeentehuis"/>
      <enumeration value="Politiebureau"/>
      <enumeration value="Postkantoor"/>
      <enumeration value="Religieus gebouw"/>
      <enumeration value="Hospitaal"/>
      <enumeration value="Station"/>
      <enumeration value="Watertoren"/>
      <enumeration value="Vuurtoren"/>
      <enumeration value="Lichttoren"/>
      <enumeration value="Zendtoren"/>
      <enumeration value="Windmolen"/>
      <enumeration value="Watermolen"/>
      <enumeration value="Gemaal"/>
      <enumeration value="Dok"/>
      <enumeration value="Kas"/>
      <enumeration value="Opslagtank"/>
      <enumeration value="Overig"/>
      <enumeration value="Crematorium"/>
      <enumeration value="Manege"/>
      <enumeration value="Kapel"/>
      <enumeration value="Radarpost"/>
    </restriction>
```

```
</simpleType>

<simpleType name="functieInrichtingsElement">
    <restriction base="string">
      <enumeration value="Geluidswerend"/>
      <enumeration value="Scheidend"/>
      <enumeration value="Overig"/>
    </restriction>
</simpleType>

<simpleType name="functieSpoor">
    <restriction base="string">
      <enumeration value="Gemengd gebruik"/>
      <enumeration value="Alleen personenvervoer"/>
      <enumeration value="Alleen goederenvervoer"/>
    </restriction>
</simpleType>

<simpleType name="fysiek_voorkomenSpoor">
    <restriction base="string">
      <enumeration value="Op vast deel van brug"/>
      <enumeration value="Op beweegbaar deel van brug"/>
      <enumeration value="Op verhoogd vlak"/>
      <enumeration value="Op verlaagd vlak"/>
      <enumeration value="Overkluisd"/>
      <enumeration value="In tunnel"/>
      <enumeration value="Op weg"/>
      <enumeration value="Overig"/>
    </restriction>
</simpleType>

<simpleType name="fysiek_voorkomenTerrein">
    <restriction base="string">
      <enumeration value="Op talud"/>
      <enumeration value="Op verhoogd vlak"/>
      <enumeration value="Op verlaagd vlak"/>
      <enumeration value="Overkluisd"/>
      <enumeration value="In tunnel"/>
      <enumeration value="Op brug"/>
      <enumeration value="Overig"/>
      <enumeration value="Op vast deel van brug"/>
    </restriction>
</simpleType>

<simpleType name="fysiek_voorkomenWater">
    <restriction base="string">
      <enumeration value="In sluis"/>
      <enumeration value="Op aquaduct"/>
      <enumeration value="In duiker"/>
      <enumeration value="In grondduiker"/>
      <enumeration value="Overig"/>
    </restriction>
</simpleType>

<simpleType name="fysiek_voorkomenWeg">
    <restriction base="string">
```

```
      <enumeration value="Op vast deel van brug"/>
      <enumeration value="Op beweegbaar deel van brug"/>
      <enumeration value="Op talud"/>
      <enumeration value="Op verhoogd vlak"/>
      <enumeration value="Op verlaagd vlak"/>
      <enumeration value="Overkluisd"/>
      <enumeration value="In tunnel"/>
      <enumeration value="Overig"/>
    </restriction>
</simpleType>

<simpleType name="gebruik">
    <restriction base="string">
      <enumeration value="Viskwekerij"/>
      <enumeration value="Waterzuivering"/>
      <enumeration value="Zwembad"/>
      <enumeration value="Overig"/>
    </restriction>
</simpleType>

<simpleType name="hoofdafwatering">
    <restriction base="string">
      <enumeration value="Ja"/>
      <enumeration value="Nee"/>
    </restriction>
</simpleType>

<simpleType name="hoofdverkeersgebruik">
    <restriction base="string">
      <enumeration value="Snelverkeer"/>
      <enumeration value="Gemengd verkeer"/>
      <enumeration value="Busverkeer"/>
      <enumeration value="Langzaam verkeer"/>
      <enumeration value="Fietsers/bromfietsers"/>
      <enumeration value="Voetgangers"/>
      <enumeration value="Parkeren"/>
      <enumeration value="Overig"/>
    </restriction>
</simpleType>

<simpleType name="hoogteklasse">
    <restriction base="string">
      <enumeration value="Laagbouw"/>
      <enumeration value="Hoogbouw"/>
    </restriction>
</simpleType>

<simpleType name="kruisingstype">
    <restriction base="string">
      <enumeration value="Deel van rotonde"/>
      <enumeration value="Deel van verkeersknooppunt"/>
      <enumeration value="Overig"/>
    </restriction>
</simpleType>

<simpleType name="landgebruik">
```

```
    <restriction base="string">
      <enumeration value="Loofbos"/>
      <enumeration value="Naaldbos"/>
      <enumeration value="Gemengd bos"/>
      <enumeration value="Griend"/>
      <enumeration value="Populieren"/>
      <enumeration value="Heide"/>
      <enumeration value="Zand"/>
      <enumeration value="Akkerland"/>
      <enumeration value="Grasland"/>
      <enumeration value="Boomgaard"/>
      <enumeration value="Fruitkwekerij"/>
      <enumeration value="Boomkwekerij"/>
      <enumeration value="Dodenakker"/>
      <enumeration value="Plaveisel, basaltblokken"/>
      <enumeration value="Overig"/>
      <enumeration value="Bebouwd gebied"/>
      <enumeration value="Kassen"/>
    </restriction>
</simpleType>

<simpleType name="numeriekOnb">
    <restriction base="string">
      <pattern value="[0-9]*|Onbekend"/>
    </restriction>
</simpleType>

<simpleType name="rijrichting">
    <restriction base="string">
      <enumeration value="Eenrichting"/>
      <enumeration value="Tweerichting"/>
    </restriction>
</simpleType>

<simpleType name="spoorbreedte">
    <restriction base="string">
      <enumeration value="Normaalspoor"/>
      <enumeration value="Smalspoor"/>
    </restriction>
</simpleType>

<simpleType name="status">
    <restriction base="string">
      <enumeration value="In ontwerp"/>
      <enumeration value="In aanleg"/>
      <enumeration value="In gebruik"/>
      <enumeration value="Niet meer in gebruik"/>
      <enumeration value="Onbekend"/>
    </restriction>
</simpleType>

<simpleType name="stroomrichting">
    <restriction base="string">
      <enumeration value="Eenrichting"/>
      <enumeration value="Twee richtingen (getijde invloed)"/>
      <enumeration value="Stilstaand"/>
```

```
    </restriction>
</simpleType>

<simpleType name="toegankelijkheid">
    <restriction base="string">
      <enumeration value="Openbaar"/>
      <enumeration value="Niet-openbaar"/>
      <enumeration value="Onbekend"/>
    </restriction>
</simpleType>

<simpleType name="typeAdminGebied">
    <restriction base="string">
      <enumeration value="Land"/>
      <enumeration value="Provincie"/>
      <enumeration value="Gemeente"/>
      <enumeration value="Bundesland"/>
      <enumeration value="Regierungsbezirk"/>
      <enumeration value="Kreis"/>
    </restriction>
</simpleType>

<simpleType name="typeGebouw">
    <restriction base="string">
      <enumeration value="Gebouw"/>
      <enumeration value="Huizenblok"/>
      <enumeration value="Toren"/>
      <enumeration value="Installatie"/>
    </restriction>
</simpleType>

<simpleType name="typeBeheersGebied">
    <restriction base="string">
      <enumeration value="Natuurgebied, natuurreservaat"/>
      <enumeration value="Boswachterij"/>
      <enumeration value="Nationaal park"/>
    </restriction>
</simpleType>

<simpleType name="typeFuncGebied">
    <restriction base="string">
      <enumeration value="Industriegebied"/>
      <enumeration value="Militair oefengebied, schietterrein"/>
      <enumeration value="Begraafplaats"/>
      <enumeration value="Sportterrein"/>
      <enumeration value="Vliegveld, luchthaven"/>
      <enumeration value="Bungalowpark"/>
      <enumeration value="Camping"/>
      <enumeration value="Recreatiegebied"/>
      <enumeration value="Zwembad"/>
      <enumeration value="Golfterrein"/>
      <enumeration value="Sluizencomplex"/>
      <enumeration value="Gebouwencomplex"/>
      <enumeration value="Jachthaven"/>
      <enumeration value="Haven"/>
      <enumeration value="Dierentuin, safaripark"/>
```

```
    </restriction>
</simpleType>

<simpleType name="typeGeografGebied">
    <restriction base="string">
      <enumeration value="Streek, veld"/>
      <enumeration value="Heuvel, duin, berg"/>
      <enumeration value="Eiland"/>
      <enumeration value="Polder"/>
      <enumeration value="Bosgebied"/>
      <enumeration value="Heidegebied"/>
      <enumeration value="Kaap, hoek"/>
      <enumeration value="Zee, zeegat, zeearm"/>
      <enumeration value="Meer, plas, vijver"/>
      <enumeration value="Geul, vaargeul"/>
      <enumeration value="Wad"/>
      <enumeration value="Bank, ondiepte, plaat"/>
      <enumeration value="Plaats, bewoond oord"/>
      <enumeration value="Wijk, buurt"/>
      <enumeration value="Overig"/>
    </restriction>
</simpleType>

<simpleType name="typeInfra">
    <restriction base="string">
      <enumeration value="Verbinding"/>
      <enumeration value="Kruising"/>
      <enumeration value="Vlak"/>
    </restriction>
</simpleType>

<simpleType name="typeInrichtingsElement">
    <restriction base="string">
      <enumeration value="Hoogspanningsleiding"/>
      <enumeration value="Kabelbaan"/>
      <enumeration value="Leiding"/>
      <enumeration value="Paalwerk"/>
      <enumeration value="Heg, haag"/>
      <enumeration value="Bomenrij"/>
      <enumeration value="Hoogspanningmast"/>
      <enumeration value="Paal"/>
      <enumeration value="Grenspaal, grenssteen"/>
      <enumeration value="Vlampijp"/>
      <enumeration value="Baak"/>
      <enumeration value="Zendmast"/>
      <enumeration value="Seinmast"/>
      <enumeration value="Peilschaal"/>
      <enumeration value="Oliepompinstallatie"/>
      <enumeration value="Muur"/>
      <enumeration value="Hekwerk"/>
      <enumeration value="Gedenkteken, monument"/>
      <enumeration value="Dukdalf"/>
      <enumeration value="Wegwijzer"/>
      <enumeration value="Boom "/>
      <enumeration value="Hunebed"/>
      <enumeration value="Grafheuvel"/>
```

```
      <enumeration value="Stuw"/>
      <enumeration value="Sluis"/>
      <enumeration value="Windmolentje"/>
      <enumeration value="Windmolen"/>
      <enumeration value="Watermolen"/>
      <enumeration value="Kilometerpaal"/>
      <enumeration value="Boom"/>
      <enumeration value="Hoogspanningsmast"/>
      <enumeration value="Heg"/>
      <enumeration value="Wegafsluiting"/>
      <enumeration value="GPS Kernpunt"/>
      <enumeration value="Kruis"/>
      <enumeration value="Toren"/>
      <enumeration value="Grenspaal"/>
      <enumeration value="Kilometerraaibord"/>
      <enumeration value="Lichtopstand"/>
      <enumeration value="Windturbine"/>
      <enumeration value="Gemaal"/>
      <enumeration value="Opslagtank"/>
      <enumeration value="Aanlegsteiger"/>
      <enumeration value="Wegafsluiter"/>
      <enumeration value="Vuurtoren"/>
      <enumeration value="Veer"/>
      <enumeration value="Sluisdeur"/>
   </restriction>
</simpleType>

<simpleType name="verhardingsbreedteklasse">
    <restriction base="string">
      <enumeration value="&gt;7m"/>
      <enumeration value="4-7m"/>
      <enumeration value="2-4m"/>
      <enumeration value="&lt;2m"/>
    </restriction>
</simpleType>

<simpleType name="verhardingsmateriaal">
    <restriction base="string">
      <enumeration value="Asfalt"/>
      <enumeration value="Zeer open asfalt beton"/>
      <enumeration value="Klinkers"/>
      <enumeration value="Grint"/>
      <enumeration value="Zand"/>
      <enumeration value="Overig"/>
      <enumeration value="Onbekend"/>
    </restriction>
</simpleType>

<simpleType name="verhardingstype">
    <restriction base="string">
      <enumeration value="Verhard"/>
      <enumeration value="Half verhard"/>
      <enumeration value="Onverhard"/>
      <enumeration value="Onbekend"/>
    </restriction>
</simpleType>
```

```
  <simpleType name="verkeersgebruik">
      <restriction base="string">
        <enumeration value="Trein"/>
        <enumeration value="Tram"/>
        <enumeration value="Metro"/>
        <enumeration value="Overig"/>
      </restriction>
  </simpleType>

  <simpleType name="voorkomen">
      <restriction base="string">
        <enumeration value="Met riet"/>
        <enumeration value="Dras, moerassig"/>
        <enumeration value="Dras, moerassig met riet"/>
        <enumeration value="Riet"/>
        <enumeration value="Overig"/>
     </restriction>
  </simpleType>

  <simpleType name="watertype">
      <restriction base="string">
        <enumeration value="Waterloop"/>
        <enumeration value="Meer, plas, ven, vijver"/>
        <enumeration value="Sloot, greppel"/>
        <enumeration value="Zee"/>
        <enumeration value="Droogvallend"/>
        <enumeration value="Greppel"/>
        <enumeration value="Greppel, droge sloot"/>
      </restriction>
  </simpleType>

  <simpleType name="wegtype">
      <restriction base="string">
        <enumeration value="Autosnelweg"/>
        <enumeration value="Hoofdweg"/>
        <enumeration value="Regionale weg"/>
        <enumeration value="Lokale weg"/>
        <enumeration value="Straat"/>
        <enumeration value="Overige weg"/>
      </restriction>
  </simpleType>

  <simpleType name="zoutgehalte">
      <restriction base="string">
        <enumeration value="Zoet"/>
        <enumeration value="Zout"/>
        <enumeration value="Brak"/>
      </restriction>
  </simpleType>

</schema>
```

A.2 metadata2.0.xsd

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- File: metadata.xsd  -->
<schema targetNamespace="http://www.gdmc.nl/tdn"
        xmlns:tdn="http://www.gdmc.nl/tdn"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:gml="http://www.opengis.net/gml"
        xmlns="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified"
        version="1.0">

  <annotation>
    <appinfo>metadata.xsd</appinfo>
    <documentation xml:lang="en">
    </documentation>
  </annotation>


  <!-- ================================================================
       Declarations for Top10Themas (= root element)
  ================================================================= -->
  <element name="MetaData" type="tdn:MetaDataType" />
  <complexType name="MetaDataType">
          <sequence minOccurs="0" maxOccurs="unbounded">
            <element ref="tdn:_MetaDataFeature"/>
          </sequence>
  </complexType>

  <element name="_MetaDataFeature" abstract="true" />

  <element name="Bron" type="tdn:BronType"  substitutionGroup="tdn:_MetaDataFeature"/>
  <complexType name="BronType">
        <sequence>
           <element name="bron_id" type="integer"/>
           <element name="brontype" type="tdn:brontype"/>
           <element name="bronbeschrijving" type="string"/>
           <element name="bronactualiteit" type="string"/>
           <element name="bronnauwkeurigheid" type="string"/>
        </sequence>
        <!--attribute name="id" type="ID" use="optional"/-->
        <attribute name="id" type="ID"/>
  </complexType>



  <!-- ================================================================
       Attribute type definitions (permitted values)
  ================================================================= -->
  <simpleType name="brontype">
      <restriction base="string">
        <enumeration value="Luchtfoto"/>
        <enumeration value="Kaart"/>
        <enumeration value="RD"/>
        <enumeration value="GBKN"/>
```

```
        <enumeration value="Overig"/>
        <enumeration value="Onbekend"/>
        <enumeration value="Kadaster"/>
        <enumeration value="Gemeente"/>
        <enumeration value="Genterpreteerde luchtfoto"/>
        <enumeration value="Terreinverkenning"/>
      </restriction>
  </simpleType>

</schema>
```

A.3 Extract from tielEstafette.gml

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<!-- File: tiel_20020401.gml  -->
<tdn:Top10Themas
  xmlns:tdn="http://www.gdmc.nl/tdn"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.gdmc.nl/tdn tdn_strict2.1.xsd">

  <gml:description>Situatie op 20020401</gml:description>

  <gml:boundedBy>
    <gml:Box srsName="EPSG:28992">
      <gml:coordinates>
        158000,432000 160000,434000
      </gml:coordinates>
    </gml:Box>
  </gml:boundedBy>

  <tdn:top10ThemasMember>
    <tdn:RuimtelijkeObjecten>

      <gml:boundedBy>
        <gml:Box srsName="EPSG:28992">
          <gml:coordinates>
            158000,432000 160000,434000
          </gml:coordinates>
        </gml:Box>
      </gml:boundedBy>

      <tdn:ruimtelijkeObjectenMember>
        <tdn:SpoorbaanDeel fid="TOP10.100672">
          <tdn:top10_id>4100001</tdn:top10_id>
          <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000016"/>
          <tdn:object_begindatum>2001-12-11T11:38:15+02:00</tdn:object_begindatum>
          <tdn:versienummer>1</tdn:versienummer>
          <tdn:versie_begindatum>2001-12-11T11:38:15+02:00</tdn:versie_begindatum>
          <tdn:dimensie>2D</tdn:dimensie>
          <tdn:tdncode>4003</tdn:tdncode>
          <tdn:type>Verbinding</tdn:type>
          <tdn:toegankelijkheid>Openbaar</tdn:toegankelijkheid>
          <tdn:status>In gebruik</tdn:status>
          <gml:polygonProperty>
            <gml:Polygon srsName="EPSG:28992">
              <gml:outerBoundaryIs>
                <gml:LinearRing>
                  <gml:coordinates>
                    158000,433909.096 158010.625,433918.383 158025.551,433930.232
                    ...
                    158000,433909.096
                  </gml:coordinates>
```

```
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </gml:Polygon>
    </gml:polygonProperty>
    <gml:geometryProperty>
      <gml:LineString srsName="EPSG:28992">
        <gml:coordinates>
          158094.951,434000 158085.081,433990.561 158075.216,433982.329
          158022.477,433933.828 158007.598,433922.017 158000,433915.376
        </gml:coordinates>
      </gml:LineString>
    </gml:geometryProperty>
    <tdn:verkeersgebruik>Trein</tdn:verkeersgebruik>
    <tdn:fysiek_voorkomen>Op verhoogd vlak</tdn:fysiek_voorkomen>
    <tdn:spoorbreedte>Normaalspoor</tdn:spoorbreedte>
    <tdn:aantal_sporen>1</tdn:aantal_sporen>
    <tdn:functie>Gemengd gebruik</tdn:functie>
    <tdn:elektrificatie>Gelektrificeerd</tdn:elektrificatie>
    <tdn:hoogteniveau>0</tdn:hoogteniveau>
    <tdn:naam>Onbekend</tdn:naam>
  </tdn:SpoorbaanDeel>
</tdn:ruimtelijkeObjectenMember>

<tdn:ruimtelijkeObjectenMember>
  <tdn:WegDeel fid="TOP10.102514">
    <tdn:top10_id>2100001</tdn:top10_id>
    <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000017"/>
    <tdn:object_begindatum>2001-12-11T11:38:15+02:00</tdn:object_begindatum>
    <tdn:versienummer>1</tdn:versienummer>
    <tdn:versie_begindatum>2001-12-11T11:38:15+02:00</tdn:versie_begindatum>
    <tdn:dimensie>2D</tdn:dimensie>
    <tdn:tdncode>3603</tdn:tdncode>
    <tdn:type>Verbinding</tdn:type>
    <tdn:toegankelijkheid>Openbaar</tdn:toegankelijkheid>
    <tdn:status>In gebruik</tdn:status>
    <gml:polygonProperty>
      <gml:Polygon srsName="EPSG:28992">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates>
              158337.725,432529.178 158350.006,432546.275 158362.61,432564.881
              ...
              158337.725,432529.178
            </gml:coordinates>
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </gml:Polygon>
    </gml:polygonProperty>
    <gml:geometryProperty>
      <gml:LineString srsName="EPSG:28992">
        <gml:coordinates>
          158000,432272.351 158035.303,432291.129 158070.085,432306.506
          ...
          158466.974,432835.804 158467.235,432847.795
        </gml:coordinates>
      </gml:LineString>
    </gml:geometryProperty>
    <tdn:wegtype>Overige weg</tdn:wegtype>
    <tdn:hoofdverkeersgebruik>Fietsers/bromfietsers</tdn:hoofdverkeersgebruik>
    <tdn:fysiek_voorkomen>Overig</tdn:fysiek_voorkomen>
    <tdn:kruisingstype>Overig</tdn:kruisingstype>
    <tdn:verhardingsbreedteklasse>Onbekend</tdn:verhardingsbreedteklasse>
    <tdn:verhardingsbreedte>Onbekend</tdn:verhardingsbreedte>
    <tdn:verhardingstype>Onverhard</tdn:verhardingstype>
    <tdn:verhardingsmateriaal>Onbekend</tdn:verhardingsmateriaal>
    <tdn:aantal_rijstroken>Onbekend</tdn:aantal_rijstroken>
    <tdn:rijrichting>Tweerichting</tdn:rijrichting>
    <tdn:hoogteniveau>0</tdn:hoogteniveau>
    <tdn:straatnaam>Ophemertsedijk</tdn:straatnaam>
```

```
    <tdn:wegnummer>Onbekend</tdn:wegnummer>
  </tdn:WegDeel>
</tdn:ruimtelijkeObjectenMember>

<tdn:ruimtelijkeObjectenMember>
  <tdn:WaterDeel fid="TOP10.101224">
    <tdn:top10_id>6100003</tdn:top10_id>
    <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000017"/>
    <tdn:object_begindatum>2001-12-11T11:38:15+02:00</tdn:object_begindatum>
    <tdn:versienummer>1</tdn:versienummer>
    <tdn:versie_begindatum>2001-12-11T11:38:15+02:00</tdn:versie_begindatum>
    <tdn:dimensie>2D</tdn:dimensie>
    <tdn:tdncode>6113</tdn:tdncode>
    <tdn:type>Verbinding</tdn:type>
    <tdn:toegankelijkheid>Openbaar</tdn:toegankelijkheid>
    <tdn:status>In gebruik</tdn:status>
    <gml:polygonProperty>
      <gml:Polygon srsName="EPSG:28992">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates>
              160000,433963.863 159985.942,434000 159816.025,434000
              ...
              160000,433459.682 160000,433963.863
            </gml:coordinates>
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </gml:Polygon>
    </gml:polygonProperty>
    <gml:geometryProperty>
      <gml:LineString srsName="EPSG:28992">
        <gml:coordinates>
          160000,433706.733 159911.78,433972.412 159906.865,434000
        </gml:coordinates>
      </gml:LineString>
    </gml:geometryProperty>
    <tdn:watertype>Waterloop</tdn:watertype>
    <tdn:breedteklasse>Onbekend</tdn:breedteklasse>
    <tdn:breedte>Onbekend</tdn:breedte>
    <tdn:hoofdafwatering>Ja</tdn:hoofdafwatering>
    <tdn:zoutgehalte>Zoet</tdn:zoutgehalte>
    <tdn:fysiek_voorkomen>Overig</tdn:fysiek_voorkomen>
    <tdn:gebruik>Overig</tdn:gebruik>
    <tdn:stroomrichting>Stilstaand</tdn:stroomrichting>
    <tdn:hoogteniveau>0</tdn:hoogteniveau>
    <tdn:naam>Amsterdam Rijnkanaal</tdn:naam>
  </tdn:WaterDeel>
</tdn:ruimtelijkeObjectenMember>

<tdn:ruimtelijkeObjectenMember>
  <tdn:Terrein fid="TOP10.100785">
    <tdn:top10_id>5100002</tdn:top10_id>
    <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000017"/>
    <tdn:object_begindatum>2001-12-11T11:38:15+02:00</tdn:object_begindatum>
    <tdn:versienummer>1</tdn:versienummer>
    <tdn:versie_begindatum>2001-12-11T11:38:15+02:00</tdn:versie_begindatum>
    <tdn:dimensie>2D</tdn:dimensie>
    <tdn:tdncode>5213</tdn:tdncode>
    <tdn:landgebruik>Grasland</tdn:landgebruik>
    <tdn:fysiek_voorkomen>Overig</tdn:fysiek_voorkomen>
    <tdn:toegankelijkheid>Openbaar</tdn:toegankelijkheid>
    <tdn:voorkomen>Overig</tdn:voorkomen>
    <gml:polygonProperty>
      <gml:Polygon srsName="EPSG:28992">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates>
              159671.118,433063.953 159578.017,433022.543 159444.877,432959.128
              ...
```

```
                159858.176,433134.422 159716.524,433081.567 159671.118,433063.953
              </gml:coordinates>
            </gml:LinearRing>
          </gml:outerBoundaryIs>
          <gml:innerBoundaryIs>
            <gml:LinearRing>
              <gml:coordinates>
                159991.163,433156.694 159994.564,433150.83 159964.283,433136.21
                159961.259,433142.256 159991.163,433156.694
              </gml:coordinates>
            </gml:LinearRing>
          </gml:innerBoundaryIs>
        </gml:Polygon>
      </gml:polygonProperty>
      <tdn:hoogteniveau>0</tdn:hoogteniveau>
      <tdn:naam>Onbekend</tdn:naam>
    </tdn:Terrein>
  </tdn:ruimtelijkeObjectenMember>

 </tdn:RuimtelijkeObjecten>
</tdn:top10ThemasMember>

<tdn:top10ThemasMember>
 <tdn:Gebouwen>

  <gml:boundedBy>
    <gml:Box srsName="EPSG:28992">
      <gml:coordinates>
        158000,432000 160000,434000
      </gml:coordinates>
    </gml:Box>
  </gml:boundedBy>

  <tdn:gebouwenMember>
    <tdn:Gebouw fid="TOP10.100151">
      <tdn:top10_id>1100001</tdn:top10_id>
      <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000014"/>
      <tdn:object_begindatum>2001-12-11T11:38:15+02:00</tdn:object_begindatum>
      <tdn:versienummer>1</tdn:versienummer>
      <tdn:versie_begindatum>2001-12-11T11:38:15+02:00</tdn:versie_begindatum>
      <tdn:dimensie>2D</tdn:dimensie>
      <tdn:tdncode>1000</tdn:tdncode>
      <tdn:type>Gebouw</tdn:type>
      <tdn:functie>Overig</tdn:functie>
      <tdn:hoogteklasse>Laagbouw</tdn:hoogteklasse>
      <tdn:hoogte>Onbekend</tdn:hoogte>
      <tdn:status>In gebruik</tdn:status>
      <gml:geometryProperty>
        <gml:Polygon srsName="EPSG:28992">
          <gml:outerBoundaryIs>
            <gml:LinearRing>
              <gml:coordinates>
                158005.969,433786.089 158014.962,433792.586 158000,433817.916
                158000,433796.957 158005.969,433786.089
              </gml:coordinates>
            </gml:LinearRing>
          </gml:outerBoundaryIs>
        </gml:Polygon>
      </gml:geometryProperty>
      <tdn:hoogteniveau>0</tdn:hoogteniveau>
      <tdn:naam>Onbekend</tdn:naam>
    </tdn:Gebouw>
  </tdn:gebouwenMember>

 </tdn:Gebouwen>
</tdn:top10ThemasMember>

<tdn:top10ThemasMember>
 <tdn:InrichtingsElementen>
```

```
    <gml:boundedBy>
      <gml:Box srsName="EPSG:28992">
        <gml:coordinates>
          158000,432017.751 160000,433993.218
        </gml:coordinates>
      </gml:Box>
    </gml:boundedBy>

    <tdn:inrichtingsElementenMember>
      <tdn:InrichtingsElement fid="TOP10.100618">
        <tdn:top10_id>3100001</tdn:top10_id>
        <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000015"/>
        <tdn:object_begindatum>2001-12-11T11:38:15+02:00</tdn:object_begindatum>
        <tdn:versienummer>1</tdn:versienummer>
        <tdn:versie_begindatum>2001-12-11T11:38:15+02:00</tdn:versie_begindatum>
        <tdn:dimensie>2D</tdn:dimensie>
        <tdn:tdncode>6654</tdn:tdncode>
        <tdn:type>Lichtopstand</tdn:type>
        <tdn:functie>Overig</tdn:functie>
        <tdn:hoogte>Onbekend</tdn:hoogte>
        <tdn:status>In gebruik</tdn:status>
        <gml:geometryProperty>
          <gml:Point srsName="EPSG:28992">
            <gml:coordinates>
              158750.022,433041.006
            </gml:coordinates>
          </gml:Point>
        </gml:geometryProperty>
        <tdn:hoogteniveau>0</tdn:hoogteniveau>
        <tdn:naam>Onbekend</tdn:naam>
        <tdn:nummer>Onbekend</tdn:nummer>
      </tdn:InrichtingsElement>
    </tdn:inrichtingsElementenMember>

  </tdn:InrichtingsElementen>
</tdn:top10ThemasMember>

<tdn:top10ThemasMember>
  <tdn:AdministratieveGebieden>

    <gml:boundedBy>
      <gml:Box srsName="EPSG:28992">
        <gml:coordinates>
          158000,432000 160000,434000
        </gml:coordinates>
      </gml:Box>
    </gml:boundedBy>

    <tdn:administratieveGebiedenMember>
      <tdn:AdministratiefGebied fid="TOP10.100001">
        <tdn:top10_id>8100001</tdn:top10_id>
        <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000002"/>
        <tdn:object_begindatum>2001-12-11T11:38:15+02:00</tdn:object_begindatum>
        <tdn:versienummer>1</tdn:versienummer>
        <tdn:versie_begindatum>2001-12-11T11:38:15+02:00</tdn:versie_begindatum>
        <tdn:dimensie>2D</tdn:dimensie>
        <tdn:tdncode>8013</tdn:tdncode>
        <tdn:type>Gemeente</tdn:type>
        <gml:geometryProperty>
          <gml:Polygon srsName="EPSG:28992">
            <gml:outerBoundaryIs>
              <gml:LinearRing>
                <gml:coordinates>
                  159269.43,433055.23 159350.78,433102.26 159488.82,433177.26
                  ...
                  159269.43,433055.23
                </gml:coordinates>
              </gml:LinearRing>
```

```
            </gml:outerBoundaryIs>
          </gml:Polygon>
        </gml:geometryProperty>
        <tdn:naam>Tiel</tdn:naam>
      </tdn:AdministratiefGebied>
    </tdn:administratieveGebiedenMember>

  </tdn:AdministratieveGebieden>
</tdn:top10ThemasMember>

<tdn:top10ThemasMember>
  <tdn:GeografischeGebieden>

    <gml:boundedBy>
      <gml:Box srsName="EPSG:28992">
        <gml:coordinates>
          158000,432000 160000,434000
        </gml:coordinates>
      </gml:Box>
    </gml:boundedBy>

    <tdn:geografischeGebiedenMember>
      <tdn:GeografischGebied fid="TOP10.100605">
        <tdn:top10_id>7150001</tdn:top10_id>
        <tdn:bronRef xlink:type="simple" xlink:href="metadata.xml#TOP10.9000005"/>
        <tdn:object_begindatum>2001-12-11T11:38:15+02:00</tdn:object_begindatum>
        <tdn:versienummer>1</tdn:versienummer>
        <tdn:versie_begindatum>2001-12-11T11:38:15+02:00</tdn:versie_begindatum>
        <tdn:dimensie>2D</tdn:dimensie>
        <tdn:tdncode>8533</tdn:tdncode>
        <tdn:type>Plaats, bewoond oord</tdn:type>
        <gml:geometryProperty>
          <gml:Polygon srsName="EPSG:28992">
            <gml:outerBoundaryIs>
              <gml:LinearRing>
                <gml:coordinates>
                  158120.536,432185.663 158340.067,432451.396 158328.387,432471.162
                  ...
                  158120.536,432185.663
                </gml:coordinates>
              </gml:LinearRing>
            </gml:outerBoundaryIs>
          </gml:Polygon>
        </gml:geometryProperty>
        <tdn:naam>Tiel</tdn:naam>
      </tdn:GeografischGebied>
    </tdn:geografischeGebiedenMember>

  </tdn:GeografischeGebieden>
</tdn:top10ThemasMember>

</tdn:Top10Themas>
```

Bibliography

[1] The Pros and Cons of XML. ZapThink Research Report. Technical report, http://www.zapthink.com, September 2001.

[2] T. Badard and D. Richard. Using XML for the exchange of updating information between geographical information systems. *Computers, Environment and Urban Systems*, 25(1-5):17–31, 2001.

[3] Nico Bakker and Bert Kolk. A new generation TOP10vector data in the Netherlands. In *Proceedings of the 20th International Cartographic Conference ICA, Beijing*, August 2001.

[4] David Carlson. *Modeling XML Applications with UML.* Addison Wesley, 2001.

[5] Eddie Curtis. XML Schema: Reconciling Diversity with Standardisation. In *GML Developers Conference, Vancourver, Canada*, July 2002.

[6] Paul Daisey. Implementing GML Schema for US Census TIGER/LINE files. In *EOGEO 2001/Digital Earth Congres, Fredericton, NB, Canada*, June 2001.

[7] C. J. Date. *An introduction to database systems: vol. 1 (5th ed.).* Addison-Wesley Longman Publishing Co., Inc., 1990.

[8] Drs. M.E. de Vries, drs. T.P.M. Tijssen, drs. J.E. Stoter, drs. C.W. Quak, and prof. dr.ir. P.J.M. van Oosterom. The GML prototype of the new TOP10vector object model. Technical report, TU Delft, Faculty CiTG, Department of Geodesy, December 2001. GISt No. 9.

[9] Marian de Vries, Theo Tijssen, Jantien Stoter, Wilko Quak, and Peter van Oosterom. Topographic data, object orientation and GML. In *EOGEO 2001/Digital Earth Congres, Fredericton, NB, Canada*, June 2001.

[10] C.J. de Zeeuw, J.D. Bulens, A.K. Bregt, R. Knapen, P.J. Lentjes, and R. van der Schans. Gebruikersspecificaties TOP10-21ste eeuw. Technical Report Alterra-rapport 158, ISSN 1566-7197 CGN-rapport 5, Centrum voor Geo-Informatie (CGI), 2000. (in Dutch).

[11] Roy Gronmo, Ida Solheim, and David Skogan. Experiences of UML-to-GML encoding. In *AGILE 2002, proceedings of the 5th AGILE Conference on Geographic Information Science*, http://www.agile-online.org, April 2002.

[12] Richard Knippers and Menno-Jan Kraak. Objectgerichte beschrijving TOP10vector – concept ontwerp gegevensmodel, versie 1.0. Technical report, ITC, Mei 2001. (in Dutch).

[13] Bert Kolk. Op weg naar objectgerichtheid in een 'state of the art'-omgeving. Technical report, Topografische Dienst Nederland, Augustus 1999. (version 3, in Dutch).

[14] Open GIS Consortium, Inc. OpenGIS Simple Features Specification for SQL. Technical Report Revision 1.1, OGC, May 1999.

[15] Open GIS Consortium, Inc. Open GIS Specification - Geography Markup Language (GML). Technical Report Version 2.0 (01-029), OGC, March 2001.

[16] Open GIS Consortium, Inc. Open GIS Geography Markup Language (GML) Implementation Specification. Technical Report Version 2.1.2 (02-069), OGC, September 2002.

[17] Open GIS Consortium, Inc. Open GIS Geography Markup Language (GML) Implementation Specification. Technical Report Draft version 3.0 (02-023r4), OGC, December 2002.

[18] Ordnance Survey. DNF data in GML – DNF release 1 product data: a description of how DNF data is represented in the Geography Markup Language. Technical report, OS, May 2001. version 1.0.

[19] Paul van Asperen. Objectgerichte beschrijving TOP10vector – concept ontwerp gegevensmodel, versie 1.0. Technical report, Topografische Dienst Nederland, Augustus 2000. (version 4.0, in Dutch).

[20] Hans van der Linde, Gerard Vrijkotte, and Aart Jan Klijnjan. Ict2002 in ka@@rt – investeringsplan tdn (periode 1999-2002). Technical report, Topografische Dienst Nederland, Februari 1999. (final version, in Dutch).

[21] Peter van Oosterom. OpenGIS technologie als basis voor de nieuwe TDN datastructuur. In *Nederlandse Vereninging voor Kartografie (NVK) studiemiddag, Alterra, Wageningen*, April 2001. (in Dutch).

[22] L.A.E. Vullings, J.D. Bulens, A.K. Bregt, R. Knapen, P.J. Lentjes, A.J.W. de Wit, and C.J. de Zeeuw. Evaluatie prototype TOP10-21ste eeuw. Technical Report Alterra-rapport 373, CGN-rapport 15, Centrum voor Geo-Informatie (CGI), 2001. (in Dutch).

[23] *XML Linking Language (XLink) version 1.0 W3C Recommendation.* http://www.w3.org/TR/xlink, June 2001.

[24] *XML Pointer Language (XPointer) version 1.0 W3C Working Draft.* http://www.w3.org/TR/xptr, August 2002.

Reports published before in this series:

- GISt Report No. 1, Oosterom, P.J. van, Research issues in integrated querying of geometric and thematic cadastral information (1), Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 29 p.p.
- GISt Report No. 2, Stoter, J.E., Considerations for a 3D Cadastre, Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 30.p.
- GISt Report No. 3, Fendel, E.M. en A.B. Smits (eds.), Java GIS Seminar, Opening GDMC, Delft, 15 November 2000, Delft University of Technology, GISt. No. 3, 25 p.p.
- GISt Report No. 4, Oosterom, P.J.M. van, Research issues in integrated querying of geometric and thematic cadastral information (2), Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 29 p.p.
- GISt Report No. 5, Oosterom, P.J.M. van, C.W. Quak, J.E. Stoter, T.P.M. Tijssen en M.E. de Vries, Objectgerichtheid TOP10vector: Achtergrond en commentaar op de gebruikersspecificaties en het conceptuele gegevensmodel, Rapport aan Topografische Dienst Nederland, E.M. Fendel (eds.), Delft University of Technology, Delft 2000, 18 p.p.
- GISt Report No. 6, Quak, C.W., An implementation of a classification algorithm for houses, Rapport aan Concernstaf Kadaster, Delft 2001, 13.p.
- GISt Report No. 7, Tijssen, T.P.M., C.W. Quak and P.J.M. van Oosterom, Spatial DBMS testing with data from the Cadastre and TNO NITG, Delft 2001, 119 p.
- GISt Report No. 8, Vries, M.E. de en E. Verbree, Internet GIS met ArcIMS, Delft 2001, 38 p.
- GISt Report No. 9, Vries, M.E. de, T.P.M. Tijssen, J.E. Stoter, C.W. Quak and P.J.M. van Oosterom, The GML prototype of the new TOP10vector object model, Report for the Topographic Service, Delft 2001, 132 p.
- GISt Report No. 10, Stoter, J.E., Nauwkeurig bepalen van grondverzet op basis van CAD ontgravingsprofielen en GIS, een haalbaarheidsstudie, Rapport aan de Bouwdienst van Rijkswaterstaat, Delft, 2001, 23 p.
- GISt Report No. 11, Geo DBMS, De basis van GIS-toepassingen, KvAG/AGGN Themamiddag, 14 november 2001, J. Flim (eds.), Delft 2001, 37 p.