A MODEL-DRIVEN GEOSPATIAL CONTENT MANAGEMENT FRAMEWORK WITH SUPPORT FOR 3D CITY MODELS

Stephan Nebiker, Stephan Schütz, Thomas Wüst

Basel University of Applied Sciences (FHBB),
CH-4132 Muttenz, Switzerland – (s.nebiker, s.schuetz, t.wuest)@fhbb.ch

First International Workshop on
Next Generation 3D City Models

KEY WORDS: Three-dimensional, Spatial, Information, Services, Internet, Modelling, Interoperability

ABSTRACT:

Today we are witnessing an increasing number of interactive 3d maps and of web-based 3d geoinformation and entertainment services covering entire regions or countries. One of the major challenges in making such interactive 3d services a lasting success is their geospatial content. In the future this content needs to be up-to-date, relevant and increasingly personalised. There is also a trend towards the integration of user's content (e.g. own hiking trails, holiday locations) within such interactive services. This paper presents concepts and mechanisms for the model-driven capturing, editing, updating and management of domain-specific 3d geospatial content in a distributed environment. It specifically addresses the support for 3d objects in general and of 3d city models in particular. One of the key elements presented is the proposed Geo eXchange Language (GXL), which enables the modelling and exchange of domain-specific geospatial content. The paper also introduces a model-driven software framework for the management of 3d geospatial content and illustrates the benefits of such a solution.

1. INTRODUCTION

1.1 Status and Motivation

3d geoinformation systems, web-based 3d geovisualisation and 3d geoinformation technologies have received considerable attention over the last few years – both in research and private industry. In the area of 3d GIS, for example, a number of research papers have addressed some of the requirements to be addressed by future systems, e.g. (Zlatanova et al., 2002). A number of authors have presented data models and system architectures capable of handling semantically and geometrically rich 3d city models, e.g. (Nebiker, 2002), (Kolbe and Gröger, 2003). From an implementation perspective, many of these papers focus on extending existing and proven (2d) geodatabase technologies (e.g. Oracle Spatial) and on established and emerging IT and geospatial standards (e.g. XML and GML 3), e.g. (Nebiker, 2002) and (Kolbe et al., 2005). Today, there are a number of operational prototype systems and even commercial 3d GIS solutions supporting many of the proposed features.

In the area of web-based 3d geovisualisation and 3d geoinformation technologies, the research focus has broadened from the streaming of large terrain and texture data sets towards the scalable and dynamic integration, querying and editing of 'intelligent' 2d and 3d geodata. Today there are a number of commercial and open source web-based 3d geoinformation solutions, which are capable of streaming large amounts of terrain and texture data as well as large numbers of 2d and 3d geospatial objects into highly interactive 3d geovisualisation environments, e.g. (GEONOVA, 2001) or (NASA, 2004). Application areas are rapidly growing from tourism and entertainment towards security, real estate, defence etc.

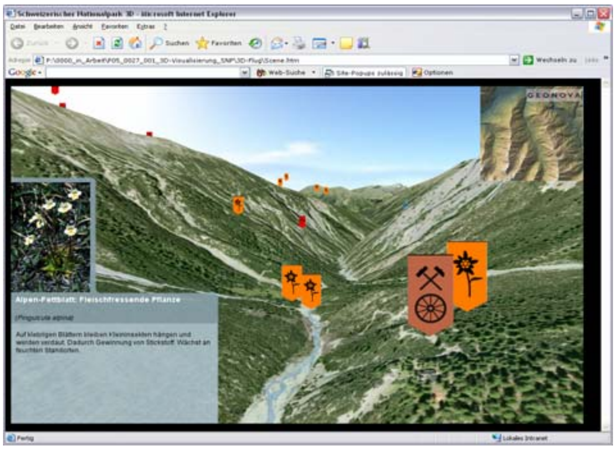
Despite the efforts and the considerable progress in the individual domains, there remains a gap between operational geodatabases – both 2d and 3d – holding an abundance of geospatial information on the one hand and web-based 3d geoinformation services on the other. This 'content gap', for example, prevents users from exploiting the potential of modern 3d city models within web-based 3d geoinformation services. One of the main reasons for this gap was the lack of mechanisms and standards for exchanging geospatial content which is both semantically rich and graphically complex. Over the last few years, this content gap has been addressed in a number of projects and geospatial infrastructure initiatives, e.g. (Nebiker et al., 2004b). The scope of these projects ranges from the development of 3d data models which can be mapped to new geospatial standards (such as GML) (Kolbe et al., 2005) and the integration of 2d GIS data into 3d services by means of web services (e.g. OGC WFS and OGC WMS) to entirely service-based 3d geoinformation solutions (e.g. OGC Web 3D Service).

While these projects share the common goal of a vastly improved exploitation of existing geospatial content within 3d geoinformation services, they typically do not address the aspects of capturing, updating and sharing such contents within the 3d environment. However, this type of content management functionality will play an important role in supporting content in 3d services which is provided by users and non-experts.

1.2 The «Geo-Roaming» Project

Geo-Roaming is an industry and government funded research project which was initiated in 2003 with the goal of improving the accessibility, usefulness and sustainability of web-based 3d geoinformation solutions. In a first project phase a service-based architecture for 3d geoinformation services was developed. The server-based 3d visualisation solution requires no software installation on the client and can be used on PCs, PDAs and Smartphones (Figure 1). The goal of the second project phase were the investigation of a model-driven mechanism and the development of a software framework for the modelling, management, exchange and updating of 3d geospatial content in a distributed 3d geoinformation infrastructure. The following paper presents findings and results of this second project phase.

Figure 1: Demonstration 3d service «ch3d.ch» on iPaq PDA with city model of Rüschlikon, Switzerland (© Osterwalder & Lehmann AG)



Auf klebrigen Blättern bleiben Kleininsekten hängen und werden verdaut. Dadurch Gewinnung von Stickstoff. Wächst an feuchten Standorten.

Figure 2: Interactive 3d map of the Swiss National Park with different types of 3d content and with an integrated quiz for park visitors (© Swiss National Park and GEONOVA AG)

1.3 Goals

The overall goal of the geospatial content management (GCM) part of the «Geo-Roaming» project was to research concepts and solutions for managing, updating and accessing distributed 3d geoinformation services. The GCM solution was to support location-based content for 3d geoinformation services, which is very rich and diverse, since it includes not only points of interest and 2d objects but also 3d objects and photo-realistic appearance, viewpoint and animation information. In addition, a 3d content management solution also has to address levels of detail and multiple-representations for all types of content – properties which are inherent to any scalable 3d geoinformation solution. The new software framework should furthermore enable the management and updating of content in distributed environments, independent of any specific application domain, storage technology or location. Among the benefits expected from this CGM framework are: a direct and comprehensive support for 3d GIS, a simple content integration and updating by non-geospatial experts and the possibility to support the integration of user's content, such as hiking trails or bike routes.

This paper introduces the GCM framework, outlines the key features of the Geospatial eXchange Language (GXL), describes the support of 3d GIS and 3d city models within the GCM framework and discusses first results.

2. GEOSPATIAL CONTENT MANAGEMENT

2.1 3d content – types and characteristics

3d maps and interactive 3d geovisualisations can consist of a large variety of map object types as indicated in a preliminary summary in (Häberling, 2003). These map objects can be considered as representing the *generic cartographic base model*, i.e. the 3d landscape model, on the one hand and the typically application- or *domain-specific model extensions* on the other hand. Among the main three content types for representing these model extensions are:

- *POI (point of interest)* – a point-oriented content object type consisting of a text label or a billboard with various spatial, thematic, graphical and behavioural properties. Examples include place names, landmarks or location indicators for persons or other tracked object.

- *2d objects* – a linear or areal vector object type. Examples include: hiking tracks, danger zones or ski slopes etc.

- *3d objects* – a volume- or surface-based object type with a potentially very complex geometry and with graphical properties such as photorealistic textures. Examples include: 3d models of buildings, traffic infrastructure or vehicles.

This short list indicates the large variety in spatial, thematic, graphical and behavioural properties of these typical content types for 3d maps. Among the common properties of such 3d map objects are multiple levels of detail (LOD) or the visible range of objects. With regards to a management solution for such cartographic content, the following requirements can be formulated:

- *Rich and extensible content* – A modern content management solution needs to support at least the three content types listed above, with the possibility to specify additional user-defined properties and to add future extensions, such as new multimedia types.

- *Application domains* – Already today, 3d geoinformation solutions are established in a broad spectrum of application domains, ranging from tourism, sports, education, gaming, aviation and simulation right through to security and defence. Each of these applications requires its own domain-specific data model, since it is inconceivable to create a universal data model satisfying all the diverse and evolving demands.

- *Timeliness and up-to-dateness* – Web-based solutions have the potential to provide up-to-date cartographic content, a potential which has been largely untapped in the past. A modern content management solution should support the automated updating of cartographic content, preferably in near real-time.

- *User content* – With the imminent integration of GPS and mobile communication technology mobile positioning is becoming a commodity. This will dramatically change the role of map users. For example, it will not only enable them to obtain precise location-based information but also to play a far more active role as users and creators of geoinformation, e.g. by recording and annotating hiking or biking trails and by sharing them with other users.

2.2 Model-based 3d content management

Due to the diverse and evolving requirements of the different application domains only a model-based content management solution was considered as future-oriented and sustainable. Such a model-based content management solution consists of the following two pillars:

- a model-based data exchange mechanism and
- a model-driven software framework

The exchange mechanism GXL (Geo eXchange Language) and the software framework of the Geo-Roaming project are outlined in the following two chapters. But first, it might be worthwhile to look at the typical processes of a content management solution for 3d geoinformation services and at the interaction between the model-driven software components and the exchange mechanism in general and the content model in particular. In interactive, web-based 3d geoinformation services there are four basic processes interacting with cartographic content (Figure 3):

- (interactive) content capturing and editing
- content management and storage
- service generation and updating
- service utilisation and content visualisation

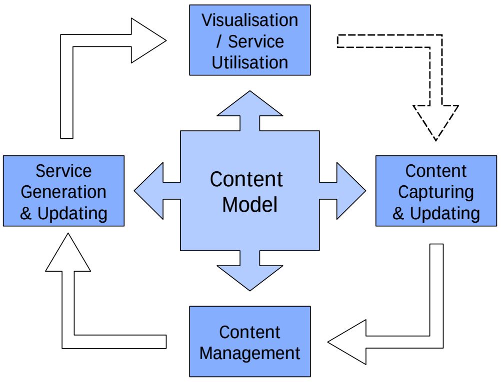


Figure 3: Model-driven content management process flow with the central GXL-based content model driving the different processes and software components.

In a model-based environment all four processes, i.e. all respective software components, are 'driven' by a common content model. The structure and encoding rules for any data objects created by and exchanged between these software components are automatically derived from the content model. In the following chapter this will be outlined in some more detail.

3. THE GEO EXCHANGE LANGUAGE GXL

One of the main challenges of the Geo-Roaming project was to find or develop a suitable mechanism for modelling and exchanging geospatial content in a 3d geoinformation environment. This mechanism should a) support geometric, semantic and graphic properties b) encompass 1d, 2d and 3d geometries c) support a model-driven system architecture and d) be suitable for a generic persistence mechanism.

None of the existing formats or standards fulfilled all of these requirements. Many graphics standards support either 2d or 3d objects but not both. Most of them lack semantic modelling features and none of them support model-driven geospatial concepts. For example, the geospatial exchange mechanism INTERLIS (Dorfschmid and Brawer, 2003) with a well proven support for model-based data exchange was lacking a support for 3d geometry and appearance (Nebiker et al., 2004a). The evolving GML standard (Lake, 2004) bears the potential to fulfil many more of the requirements. However, its 3d geometry model was (and still is) under development and it was lacking certain important features such as the support for object textures. User demands also showed that there is a need to directly support widely used 3d geometry representations (such as VRML) as part of such a content mechanism.

These results led to the development of the Geo eXchange Language (GXL). GXL is an extensible mechanism for the modelling and exchange of cartographic 3d contents and is based on GML 3.1. The main characteristics of GXL are explained below.

3.1 Basic structure of GXL

GXL is based on GML 3.1 with a small number of extensions and restrictions and is defined in XML Schema. Content objects in GXL, for example, are generally represented by GML spatial objects (features and feature collections) with their respective spatial and non-spatial properties. Based on positive experiences with INTERLIS, GXL was incorporated a predefined modelling hierarchy, which largely facilitates model-driven architectures. GXL also extends GML with a number of additional 3d data types. The resulting GXL base schema is a GML application schema and as a consequence GXL can be considered a GML application language (see Figure 4).

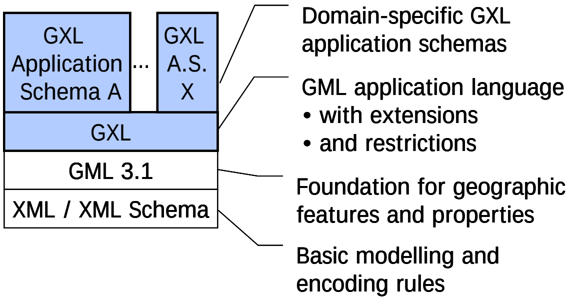


Figure 4: The GXL base schema as a GML application language and GXL application schemas for different application domains.

In contrast to domain-specific GML application schemas, GXL constitutes a general geospatial modelling language which in itself can be applied to specific application domains. The following paragraphs highlight some of the main data modelling features of GXL:

Modelling hierarchy

GXL follows a three-level modelling hierarchy of Model, Topic and Class. This constitutes a restriction of GML which itself allows modelling hierarchies of an arbitrary depth and complexity. A GXL data model starts with a Model element, which can contain any number of topics (Topic elements). A Topic can again contain any number of Class elements. At the Class level, the actual content type is defined (e.g. POI or 2d object) together with its object style (appearance) and additional application-specific properties.

Object types

The following object or content types are currently available in GXL: Point of Interest (text label), Point of Interest (symbol or image), 2d vector, 3d object (d3o) and 3d object (vrml). The first three object types use GML geometry types and extend them with additional GXL elements, for example to handle vertical and horizontal offsets between the actual point of interest and the corresponding text label or the billboard. In order to manage 3d content objects, a GXL object was defined which serves as a container for different existing 3d object types and formats. Thus, 3d objects in commonly used formats such as VRML or in the d3o representation of the DILAS 3d GIS (Nebiker, 2003) can be handled with GXL. It is conceivable that the emerging CityGML (Kolbe et al., 2005) representation for 3d city models could be added to GXL as an additional 3d type.

Object styles

GXL supports specific style properties for the different types of object types. The style or appearance of a content object can either be defined at the object level or at the class level. This allows, for example, to very quickly modify all POI of the class PostOffice but to retain the appearance of the main post office POI in bold letters.

User-defined properties

One of the key elements of a model-based mechanism is the possibility to define user-defined properties. GXL content object types can be extended with any number of user-defined properties. Content objects of the type POI holding city names could for example have an additional property of the type hyperlink to the city's official web site. Or 3d content objects representing hotel buildings could have an additional property hotelRating representing the official rating of the hotel.

3.2 GXL application schemata

Based on the GXL base schema it is now possible to create a domain specific application schema, commonly referred to as a domain-specific 'data model'. Such a GXL application schema contains the data structure (class hierarchies etc.), the content topics and classes with the corresponding geometry types and additional, user-defined properties.

Figure 5 illustrates structure and contents of a typical GXL application schema in the tourism domain. The application schema contains the above-mentioned modelling hierarchies Model, Topic and Class. From a GML perspective, each of these elements constitutes a so-called feature collection, with a number of predefined GML properties, e.g. a bounding geometry (property gml:boundedBy), a name (gml:name) and an optional description (gml:description). These GML properties are supplemented by GXL properties such as an object state (gxl:objectState) or a one-dimensional content geometry (gxl:geometry1D). The model could further contain any number of domain-specific properties such as gxl:hotelRating, gxl:averageRoomRate etc. Please note that the two content object instances (hotels Bellevue and Hilton) are not part of the data model and are shown for illustration only. The application schema in Figure 5 is complemented by the top-level element Transfer, which enables the exchange of data sets containing multiple models.

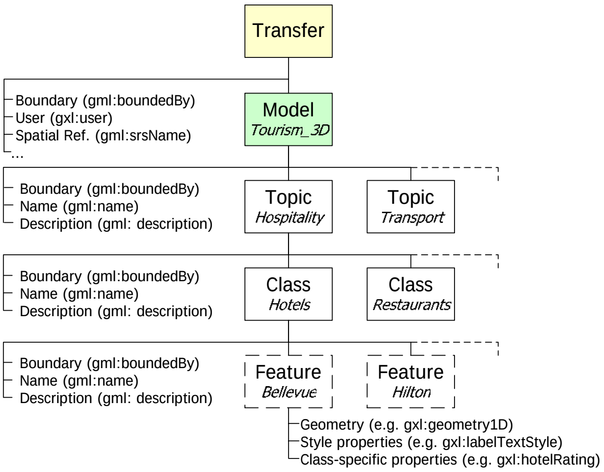


Figure 5: Illustration of a GXL application schema with the modelling hierarchies Model, Topic and Class with examples of their respective spatial and semantic properties.

3.3 GXL instance documents

The structure and encoding of GXL instance documents, i.e. of the actual 3d content data, are automatically derived from the respective GXL application schema. Such a GXL instance document could be a small XML or GXL snippet containing a single content object, e.g. a POI, or a large XML document containing the complete content of a certain topic or an entire content database. An extract of a GXL snippet containing a single POI is shown in Figure 6. In this example GML properties and GXL properties can be nicely distinguished by their different namespaces gml: and gxl:. The example also illustrates the use of the standard GML property gml:id as a unique object identifier (OID) for content objects. This OID plays a key role in identifying and handling content objects in distributed system environments.

```
<HotelObject gml:id="o467668c9-4170-4746-8045-
076f23c1eaf2">
   <gxl:lastModifiedDate>2004-11-24T2004-11-
24</gxl:lastModifiedDate>
   <gml:name>Bellevue</gml:name>
   <gml:boundedBy/>
   <gxl:objectState>NEW</gxl:objectState>
   <gxl:objectMembers>
      <gxl:LabelImage gml:id="leb744600-8e78-
4853-b784-d91e97c54af9">
         <gml:name>Hotel Bellevue</gml:name>
         <gml:boundedBy/>
         <gml:position>
            <gml:Point>
               <gml:coordinates>626274.000000
                  257420.000000 458.807098
               </gml:coordinates>
            </gml:Point>
         </gml:position>
```

```
         <gxl:heightAboveGround>100
         </gxl:heightAboveGround >
         <gxl:imageFileName>bellevue.tga
         </gxl:imageFileName>
      </gxl:LabelImage>
   </gxl:objectMembers>
   <gxl:geometry1D>
      <gml:Point>
         <gml:coordinates>626274.000000
            257420.000000 358.807098
         </gml:coordinates>
      </gml:Point>
   </gxl:geometry1D>
</HotelObject>
```

Figure 6: Extract from a GXL instance document with a single point of interest (POI).

3.4 3D city model support

The support for rich and complex 3d city models was one of the main reasons for extending the GXL mechanisms with additional 3d types. By embedding VRML objects, i.e. individual 3d objects or 3d sub-scenes encoded in VRML, into GXL, it was taken into account that VRML is currently the most popular and most widely-used 3d format. For non-geospatial experts, for example, this makes it possible to create and easily integrate their own 3d objects using any 3d modelling software. However, in order to support large 3d city models in an efficient and sustainable manner, additional aspects such as object identification, semantics, topology (inter- and intra-object relationships), texture, levels of detail etc. become important. These requirements can be fulfilled by integrating the 'd3o' type of the DILAS project (Nebiker, 2003) as an additional 3d object type into the GXL mechanism (see Figure 7). Since the d3o type is also XML-based, it ideally fits into the GXL mechanism.

```
<HotelObject gml:id="o467668c9-4170-4746-8045-
076f23c1eaf2">
   <gxl:lastModifiedDate>2004-11-24 T2004-11-24
   </gxl:lastModifiedDate>
   <gml:name>Hotel Bellevue</gml:name>
   <gml:boundedBy/>
   <gxl:objectState>NEW</gxl:objectState>
   <gxl:objectMembers>
      <gxl:LabelImage gml:id="...">
         <gml:name>Hotel Bellevue</gml:name>
         <gml:boundedBy/>
         <gml:position>
            <gml:Point>…</gml:Point>
         </gml:position>
         <gxl:heightAboveGround>100
         </gxl:heightAboveGround >
         <gxl:imageFileName>bellevue.tga
         </gxl:imageFileName>
      </gxl:LabelImage>
   </gxl:objectMembers>
   <gxl:geometry3D>
      <ObjectD3O>
         <BBoxUL>…</BBoxUL>
         <BBoxLR>…</BBoxLR>
         <PointList>…</PointList>
         <ComplexEdgeList/>
         <ElementList>…</ElementList>
         <PointCount>14</PointCount>
         <ElementCount>10</ElementCount>
         <ComplexEdgeCount>0</ComplexEdgeCount>
         <ObjectElements>…</ObjectElements>
         <AuxiliaryElements/>
      </ObjectD3O>
   </gxl:geometry3D>
</HotelObject>
```

Figure 7: Example of a GXL object containing a 3d object of the DILAS type d3o and an image label.

4. MODEL-DRIVEN GEO CONTENT MANAGEMENT FRAMEWORK

A model-driven software framework is the second pillar of a model-based content management solution. The 3d geo content management (GCM) software framework developed in the Geo-Roaming project is fully driven by GXL, i.e. by the respective GXL application schema. This GXL application schema is used for the creation of data structures within all the GCM components, for the definition and configuration of the data structures within the data repositories, for the exchange of data between software components and also for the validation of the data. Thus, the software components can handle any domain-specific 'data model' as long as it constitutes a valid GXL schema. An additional goal in the design of the software framework was to complement the modelling flexibility with a maximum of storage flexibility. This was achieved by introducing a persistence framework which supports different storage concepts. The main features of the GCM framework are presented below.

4.1 Object representations

Within the GCM framework content objects can have a number of different but largely equivalent representations (see Figure 8): a) a C++ object representation, b) a proprietary binary Format, c) a proprietary XML-based format, d) the open GXL XML format as well as e) the representation(s) in the content repository.

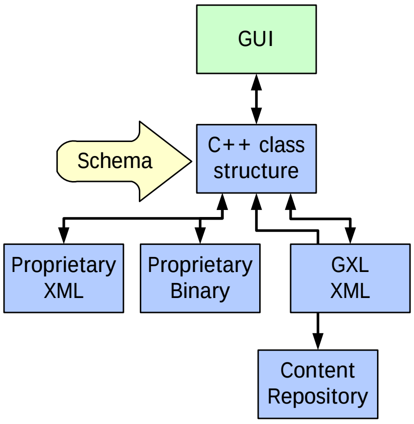


Figure 8: Different object representations within the GCM framework.

The mapping between the C++ representation and the other representations is performed by automated object serialisation and de-serialisation. The object representation in the content repository depends on the selected persistence technology. In case of an SQL DBMS the storage is based on (object-) relational tables (see below).

4.2 Query mechanism

Among the key elements of a geospatial content management framework are the querying and selection of subsets of content objects based on spatial and non-spatial predicates. Within the GCM framework an object query mechanism was developed based on the OGC Filter Encoding Specification (Vretanos, 2005). This specification defines the XML-based platform- and system-independent encoding for the querying and selection of geospatial objects. The Filter Encoding Specification was originally part of OGC WFS, the Web Feature Service

Specification but was promoted to an independent specification in order to utilise it in a number of other standards. The main benefit of the Filter Encoding Specification is the independence from any underlying repository technology, e.g. a SQL database. The mapping of an object filter statement (Figure 9) to a query for a specific repository technology (see Figure 10), e.g. to a PostgreSQL or Oracle SQL query, is handled by the respective query builder component.

```
<Filter>
   <And>
      <PropertyIsLike wildCard="*"
         singleChar="#" escapeChar="!">
         <PropertyName>name</PropertyName>
         <Literal>Bellevue</Literal>
      </PropertyIsLike>
      <Contains>
         <PropertyName>BBox</PropertyName>
         <gml:Polygon srsName=
"http://www.opengis.org/gml/srs/epsg.xml#21781">
            <gml:outerBoundaryIs>
               <gml:LinearRing>
                  <gml:pos>550000 175000
                  </gml:pos>
                  <gml:pos>650000 175000
                  </gml:pos>
                  <gml:pos>650000 275000
                  </gml:pos>
                  <gml:pos>550000 275000
                  </gml:pos>
                  <gml:pos>550000 175000
                  </gml:pos>
               </gml:LinearRing>
            </gml:outerBoundaryIs>
         </gml:Polygon>
      </Contains>
   </And>
</Filter>

SELECT "objectid" FROM "Hotels" WHERE ("name"
LIKE 'Bellevue' ESCAPE '!')
AND (Within("boundingbox",
GeometryFromText('POLYGON((550000 175000, 650000
175000,
650000 275000, 550000 275000, 550000
175000))',21781)));
```

Figure 9: Example of an OGC compliant object query statement and the equivalent, derived SQL statement

Currently, the content management framework supports three main types of object queries: an object filter, an object ID filter and a compound object filter. The first filter type represents the general case of an object filter supporting hierarchically structured spatial and thematic clauses and is compatible with the OGC specification. The second filter type permits the efficient selection of objects based on their object identifier(s) – an important and frequently occurring task. The third filter type enables the querying of objects over multiple classes – a feature which is currently not supported in the OGC standard.

XML-based object filters have the disadvantage that they are more complex to read and write than query languages such as SQL and that they are certainly less established. Thus, it was decided to implement a graphical user interface, which plugs into the GXL model hierarchy and supports users in creating valid queries.

4.3 Persistence management mechanism

Various developments in the fields of database technologies such as geospatial support in open source database systems (e.g. PostgreSQL or MySQL), the proliferation of XML in different database technologies (object-relational and native XML) and the trend towards service-based architectures (hiding the underlying database technologies) indicated the need for a platform-independent content storage solution. The persistence management mechanism developed as part of the Geo Content Management (GCM) framework is based on multiple abstraction layers hiding the underlying storage technology (see Figure 10) and on the above-mentioned platform-independent object query mechanism. The binding of a specific storage or database technology occurs at the lowest abstraction level. Components at the medium level provide functionality which is common to a certain type of database technology, e.g. the mapping to standard SQL for all (object-) relational DBMSs. Any system-specific dialects are handled on a lower abstraction level. The top level of the persistence management mechanism provides a common interface for all other components of the GCM framework and is independent from any repository technology.

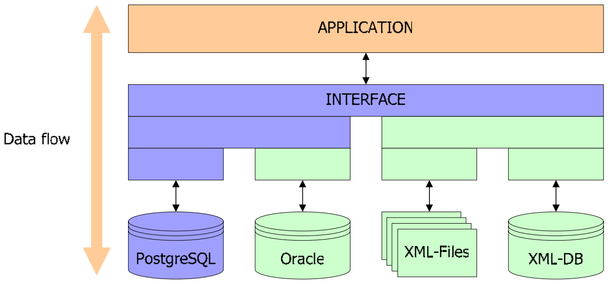
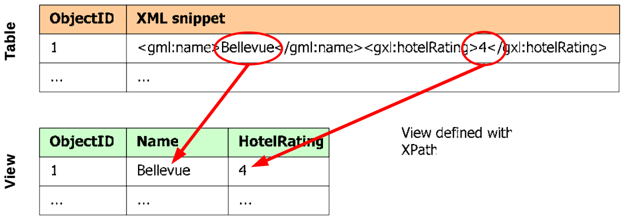


Figure 10: Persistence management components within a multi-layer architecture; dependency on specific storage technology increases towards the bottom of the diagram.

4.4 Content repositories

The two main storage technologies targeted with the GCM framework are object-relational DBMS on the one hand and XML-based storage on the other. The current implementation supports PostgreSQL, an object-relational database management system developed as an Open Source project. Currently, the Oracle-based 3d GIS DILAS is being added as an additional content repository. Since DILAS is operationally used for generating and managing large 3d city models, this 3d content will directly become accessible to web-based 3d geoinformation services.

In an object-relational environment the hierarchical structure of GXL is mapped to a table structure. During the initialisation of a new content database, the GXL application schema is used to automatically generate the domain-specific table structure. 3d content objects are then stored as XML snippets using the GXL representation (see Figure 11).



ObjectID	XML snippet
1	<gml:name>Bellevue</gml:name><gxl:hotelRating>4</gxl:hotelRating>
...	...

ObjectID	Name	HotelRating
1	Bellevue	4
...	...	...

Figure 11: Relational view on a XML snippet containing GXL data

Relational views on the XML snippets are then used for querying the contents of the GXL objects using standard SQL (Figure 11). The advantages of this solution include:

- no redundant data storage of multiple representations of the same object (e.g. XML and relational)
- no expensive, repeated parsing of XML contents for frequent queries
- no complex table structures which otherwise result from relational mappings of XML data
- no danger of users interfering with the data consistency due to the use of read-only views

In addition to this approach of relational views on XML objects, selected and important standard properties of content objects are extracted and stored as attribute values within the same table. Examples of such important properties include the bounding geometry (gxl:boundedBy) and the object identifier OID (gml:id). These key properties are extracted in order to provide a highly efficient object access. However, it should be noted that these extracted properties and the relational views serve as index structures only and that the GXL snippet remains the main object representation.

4.5 Data communication

The communication between the software components of the GCM framework is currently based on TCP/IP sockets. This relatively simple mechanism requires just the IP address and the port of a host computer with a communications server listening for incoming connection requests. With the systematic use of XML within all software components (application schema, data instances, object filter) the addition of a service-based communication, e.g. using SOAP (Simple Access Protocol) is the next logical step. The addition of Web Service functionality will also enable the utilisation and integration of geospatial web services (e.g. OGC WFS) within the 3d geo content management framework.

For the purpose of transportation or storage, geospatial content is packaged in the form of GXL snippets. If required, content is automatically converted to binary form and optionally even encrypted. This dramatically increases the system performance and also enables a secure exchange of geospatial content throughout the framework.

5. APPLICATIONS AND RESULTS

The geo content management framework was implemented as a fully operational prototype system (see Figure 12). This solution consists of the 'Geo Content Modeler' for the interactive and graphical creation of domain specific data models, the 'Geo Content Editor' for the capturing and editing of content objects in an interactive 3d viewer environment and the 'Geo Content Manager' which hosts the persistence manager and is used to configure the underlying content repositories. The GCM architecture is designed to support different types of content storage. This includes object-relational databases with XML support (currently PostgreSQL) and local file storage but it could also be extended to native XML databases.

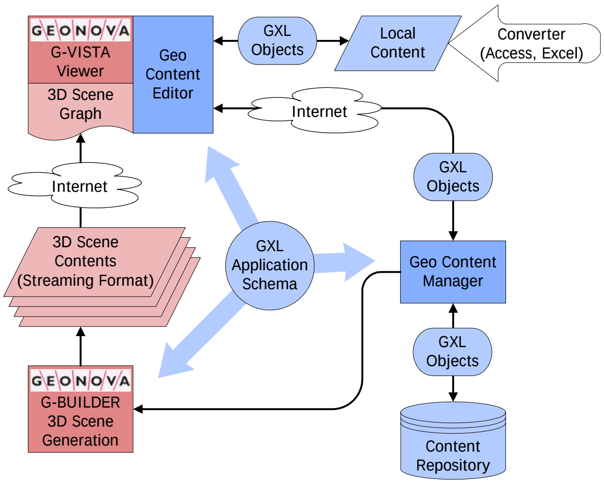


Figure 12: Geo content management – schematic view of the prototype system architecture (left side: commercial 3d viewer technology; right side: geo content components and workflow)

The geo content management framework is currently being integrated into the commercial product line of the industry partner. Figure 13 illustrates the layout and some of the key elements of the model-driven content editor component.

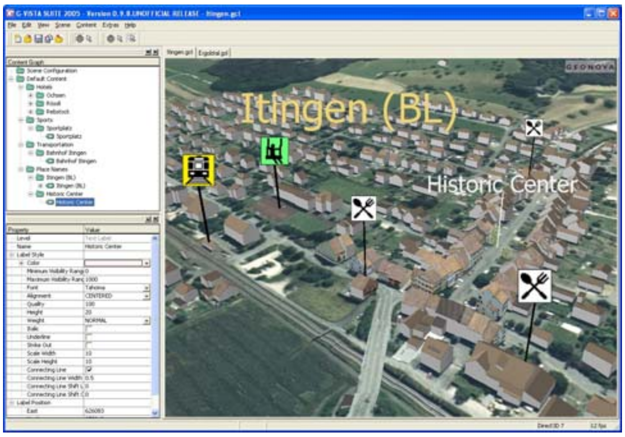


Figure 13: Geo Content Editor component integrated into G-VISTA™ SUITE with model-driven content graph (upper left), content properties dialog (lower left) and interactive 3d viewer (© GEONOVA AG)

The geo content management framework has successfully been used on a number of commercial and research projects. Figure 2, for example, illustrates an innovative application created for the Swiss National Park combining interactive 3d cartography with content-based quizzing functionality. Another interesting project using geo content management components was «bike3d». In this diploma thesis project (Schweizer and Würth, 2004) a prototype web portal for 3d mountain biking routes was developed, which allows users to upload, download and exchange 3d content based on user-generated GPS tracks.

6. CONCLUSIONS AND OUTLOOK

In this paper we presented concepts and mechanisms for the model-driven capturing, editing and management of contents for interactive 3d maps and web-based geoinformation services. The proposed Geo eXchange Language GXL combines the required modelling richness with the strictness of XML schema.

This makes it a suitable mechanism for a model-driven software framework. The presented framework is based on international standards such as XML, OGC GML and OGC Filter Encoding. GXL was primarily conceived as a powerful mechanism driving the components of the geo content software framework. However, it will be published shortly and could then provide a valuable input for the definition of a 3d geospatial content profile of GML (e.g. 3dCartoGML).

Ongoing work includes the support of additional storage technologies and the inclusion of new content types. Currently, the development of an Oracle interface for the persistence manager is under way. This development is providing the 3d geo content management framework with a direct access to the 3d GIS DILAS (Nebiker, 2003) and thus to large 3d city models. The integration of native XML databases as one of the long-term goals of the Geo-Roaming projects will be initiated once a minimal geospatial support, such as spatial indexing, will become available. Future 3d geoinformation services will include additional content such as multimedia objects and highly-interactive thematic cartographic objects – a challenge for GXL and the presented geo content management framework.

7. REFERENCES

Dorfschmid, J. and Brawer, S., 2003. Modeling of Space-Related Data - An introduction with regard to UML and INTERLIS. Coordination of geographical information and geographical information systems (KOGIS/COGIS), Seftigenstrasse 264, CH – 3084 Wabern, Switzerland, www.kogis.ch.

GEONOVA, 2001. Flug durch die Schweiz (Interactive, web-based 3D-Visualisation of Switzerland), Muttenz, Switzerland. www.geonova.ch (last accessed: April 2005).

Häberling, C., 2003. Topografische 3D-Karten - Thesen für kartografische Gestaltungsgrundsätze. ETH Nr. 15379. Dissertation Thesis, Institut für Kartographie, ETH Zürich.

Kolbe, T.H. and Gröger, G., 2003. Towards unified 3D city models, Proceedings ISPRS Commission IV Joint Workshop "Challenges in Geospatial Analysis, Integration and Visualization II", Stuttgart, Germany, 8-9 September 2003.

Kolbe, T.H., Gröger, G. and Plümer, L., 2005. CityGML – Interoperable Access to 3D City Models. In: E.M. Fendel (Editor), Proceedings of the Int. Symposium on Geo-Information for Disaster Management on 21-23 March 2005 in Delft. Springer Verlag.

Lake, R., 2004. Geography mark-up language (GML) [foundation for the geo-web]. Wiley, Chichester, 388 pp.

NASA, 2004. World Wind, Open Source Project, http://worldwind.arc.nasa.gov (last accessed: April 2005).

Nebiker, S., 2002. Design and Implementation of the High-Performance 3D Digital Landscape Server 'DILAS', ISPRS, IGU, CIG - Joint International Symposium on Geospatial Theory, Processing and Applications, Ottawa, Canada.

Nebiker, S., 2003. DILAS - The Digital Landscape Server for the Generation and Management of Large 3D City Models. In: D. Fritsch (Editor), Photogrammetric Week '03. Herbert Wichmann Verlag, Heidelberg, pp. 227-236.

Nebiker, S., Bleisch, S. and Annen, A., 2004a. INTERLIS 2 – GML 3 – Eine Vergleichsstudie, FHBB Fachhochschule beider Basel, Muttenz.

Nebiker, S., Schütz, S. and Wüst, T., 2004b. Geo-Roaming: Model-Driven Content Management for Web-based 3D Geoinformation Services, XX ISPRS Congress, Istanbul, Turkey.

Schweizer, F. and Würth, H., 2004. BIKE 3D - Inhaltsverwaltung für webbasierte Geoinformationsdienste am Beispiel von Mountain-Bike-Routen. Diploma Thesis, FHBB Basel University of Applied Sciences, Muttenz.

Vretanos, P.A. (Editor), 2005. Filter Encoding Implementation Specification, OGC 04-095, Version: 1.1.0. Open Geospatial Consortium.

Zlatanova, S., Rahman, A.A. and Pilouk, M., 2002. 3D GIS: Current Status and Perspectives, ISPRS, IGU, CIG - Joint International Symposium on Geospatial Theory, Processing and Applications, Ottawa, Canada.

8. ACKNOWLEDGMENTS

The authors would like to thank the Commission of Technology and Innovation (CTI) of the Swiss Federal Office for Professional Education and Technology for the support of this project[1]. Thanks are also due to the industry partners GEONOVA AG, Flotron AG, Osterwalder & Lehmann AG, Dütschler & Nägeli AG and Murer Geomatik AG for their support and collaboration.

---

[1] CTI Project Nr. 6338.1 FHS 'Geo-Roaming'