# Implementation of UML Schema to RDBM

M. Nagni, S. Ventouras, and G. Parton
Centre for Environmental Data Archival, STFC, Harwell Oxford, Oxfordshire, UK OX11 0QX
maurizio.nagni@stfc.ac.uk

## 1. Why map an UML schema to a RDBM?

Numerous disciplines require information concerning phenomena implicitly or explicitly associated with a location relative to the Earth. Disciplines using Geographic Information (GI) in particular are those within the earth and physical sciences, and increasingly those within social science and medical fields. Therefore geographic datasets are increasingly being shared, exchanged and frequently re-purposed for uses beynd their original intended use.

The ISO Technical Committee 211 (ISO/TC 211) together with Open Geospatial Consortium (OGC) provide a series of standards and guidelines for developing application schemas which should:

a) capture relevant conceptual aspects of the data involved;

b) be sufficient to satisfy previously defined use-cases of a specific or cross-domain concerns.

In addition, the HollowWorld technology offers an accessible and industry-standardised methodology for creating and editing Application Schema UML models which conform to international standards for interoperable GI [2]. We present a technology which seamlessly transforms an Application Schema UML model to a relational database model (RDBM). This technology, using the same UML information model, complements the XML transformation of an information model produced by the FullMoon tool [2].

RDBMs exist to enable searching within a data collection, a process that has, over the decades, been heavily optimized. Moreover, modern non-relational DB [3] flavours (MongoDB, Cassandra, NewSQL) are still in their infancy with associated disadvantages for ease of adoption, software reliability, etc. A UML schema, or better its XMI description, has, in contrast, an almost natural mapping/transformation to an XSD schema and ISO19136 with well known applications, e.g. Fullmoon or Shape Change, supporting this approach. However, describing geographic information within a widely accepted XML-encoded vendor-neutral format such as GML may not be the best option for persisting or searching operations.

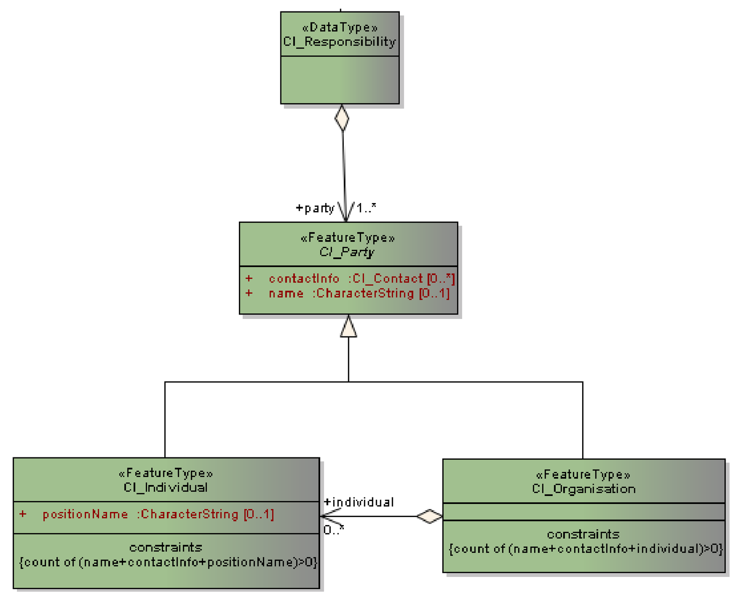
Within a full model-driven approach the UML should remain at the centre of any implementation claiming to represent the model itself. In this context, a UML -> XSD -> RDBM transformation is not possible because the both the XSD and RDBM, and even an OWL implementation, have the same goal: to represent the same model. In a typical scenario an ingested XML document is separated into core and ancillary data: the core data map to a set of relational tables, the ancillary to a single XML-type field. This approach works well when the core data are a fraction of the whole document, and even better if the main aim of the RDBM is not to simply return other XML objects.

## 3. Inheritance vs Composition

In both UML and OO a child class can inherit attributes and/or methods either from one or many parent classes. However, only a few RDBMs can partially handle inheritance rules. To address this issue the **Composition over Inheritance** [4] technique is utilised which transforms an inheritance relation to a composite class. Composition may appear less "natural" than inheritance but it provides a more stable environment overall, having the advantage over inheritance by being a more thorough isolation of interests which may be described by a hierarchy of descendant classes.

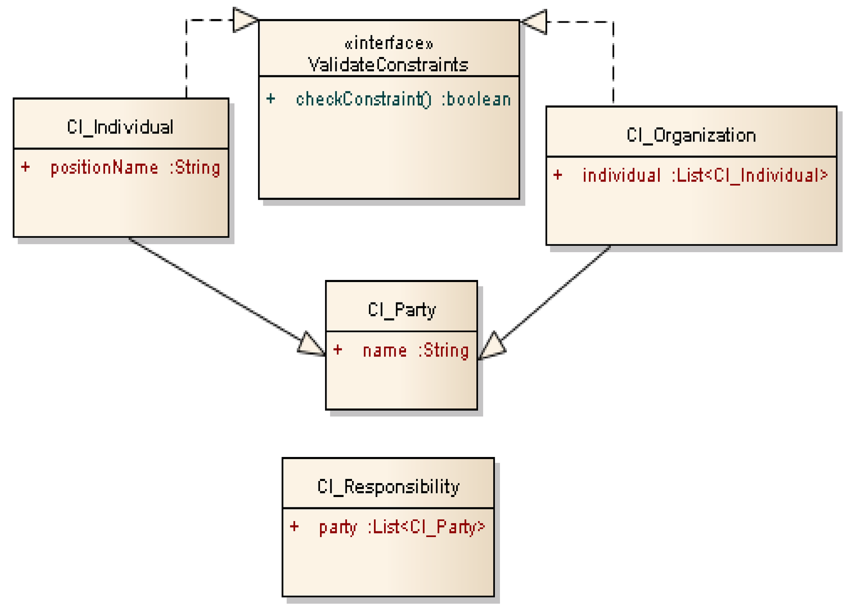
A RDBM may then exploit such isolation through use of multiple foreign key relationships between tables.

### UML Model



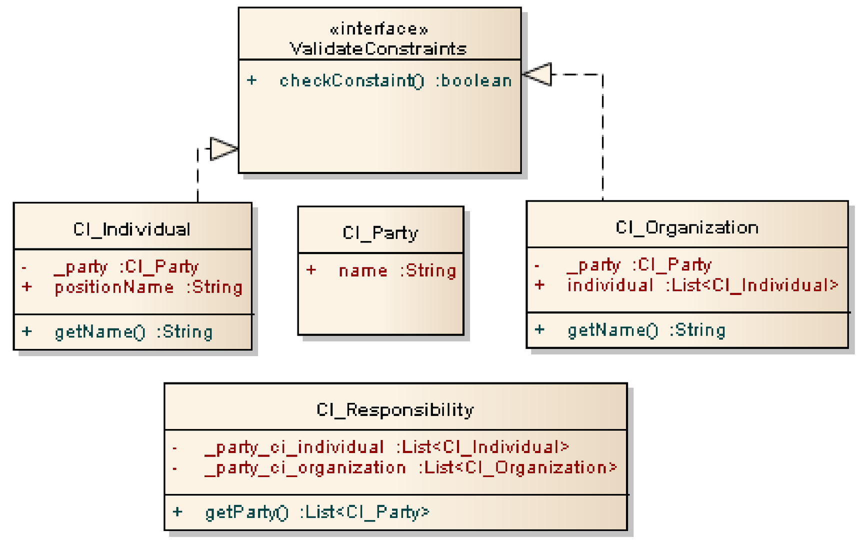
- Classes defined through inheritance and relations
- All attributes are public
- Constraints are expressed in a human understandable format. However, an Object Constraint Language (OCL) expression would fit better a model driven approach

### OO Model (with Inheritance)



- The classes are defined through inheritance
- All attributes are public
- Constraints are realised implementing a general interface method (through the interface we define an abstract class containing only methods or class constants)
- A "List" class realises the attribute multiplicity of the class

### OO Model (with Composition)

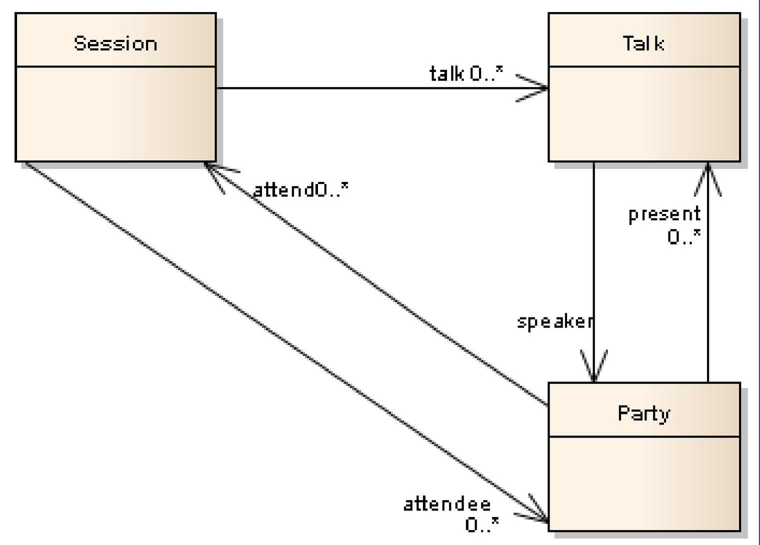


- The classes contains their parent as a private attribute
- Attributes coming from a parent are accessed through a "getter" method

## 5. Reference

1. ISO 19101. Geographic information — Reference model. International Organization for Standardization, 2002.
2. P. Colodoniuc and S. Cox, Application schema modelling for interoperable geospatial information using the ISO 19100 series of standards
3. http://en.wikipedia.org/wiki/NoSQL
4. http://en.wikipedia.org/wiki/Composition_over_inheritance

## 2. From hierarchical model to RDBM

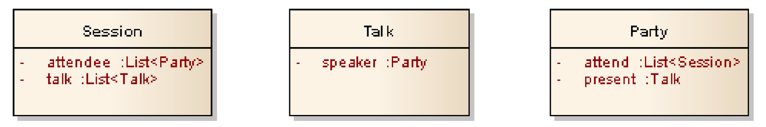


The essence of a UML model consists of three components: *object, relation, constraint*.

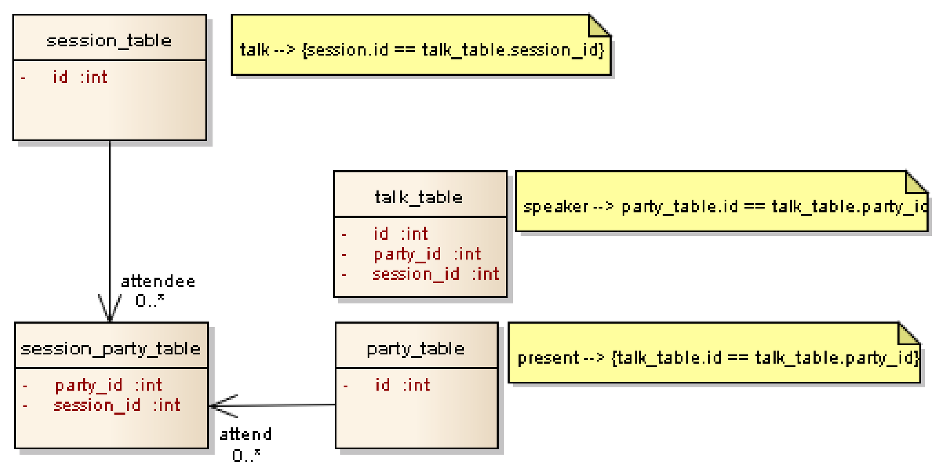
- An **object** is typically a common, language independent, Object Oriented (OO) class with its attributes and methods
- A **relation** connects two objects, whose types are harmonised either when defined at class level, i.e. inheritance, or at the attribute level, i.e. via multiplicity
- A **constraint** is a logical expression limiting the object's features

**An RDBM is established through:**

1. creating a collection of OO classes and further enriching these classes with appropriate methods to satisfy the model constraints.



2. enforcing the UML relationships in the RDBM with an appropriate collection of foreign keys and subsequently create additional tables (*many-to-many* or *circular relations*)

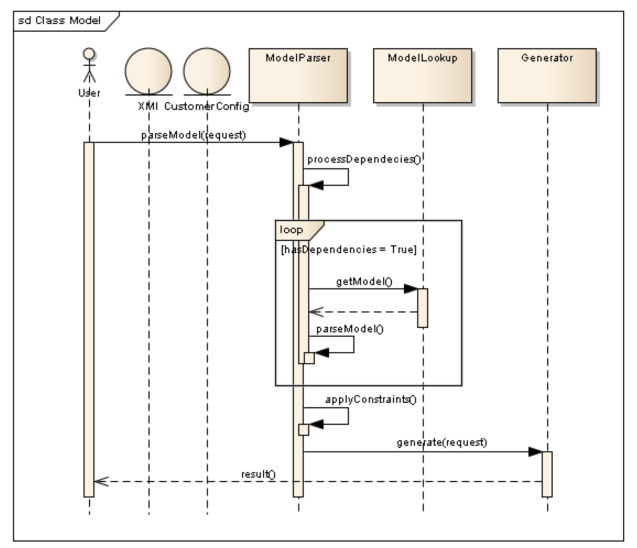


However, it should be noted that:

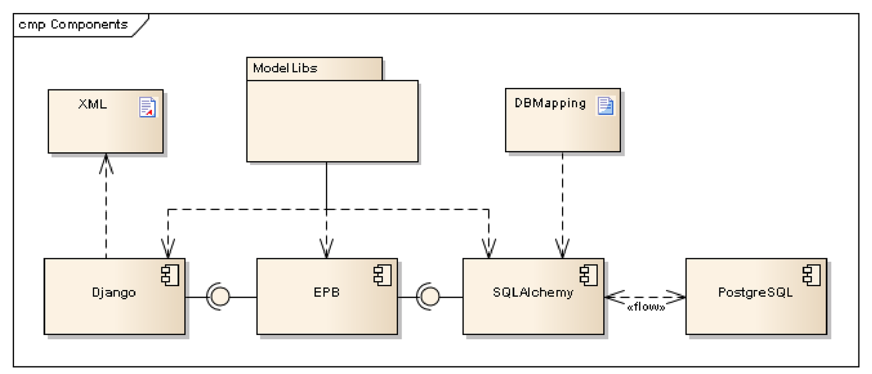
- as the RDBM consist of collections of interconnected class instances the tables, and the classes that are their counterpart, gain one or more "ID" attributes in order to reconstruct specific model/submodel instances
- those "ID" attributes are assumed to be *private,* being part of the persistence mechanism and not of the UML model itself
- an additional table is needed where a *many-to-many* relation exists, e.g. a **Session** contains many **Parties** but also a **Party** may contains many sessions and is resolved creating an ad-hoc table, **session_party_table**, in order to fully decouple the **session_table** and the **party_table**
- the additional table for the *many-to-many* relationship (the **session_party_table** above) also resolves also problem arising from *circular relations, e.g.* **Session** → **Talk** → **Party** → **Session**

## 4. Implementation details

Once a XMI format of the UML model has been exported a Java application parses the XMI retrieving the necessary models to import "on-the-fly" through an inner name:httpLink lookup table, creating an internal description of the model. The application then uses Apache Velocity to generate a python collection of OO classes and an SQLAlchemy mapping script. Within a complex UML model the class relationships are hard to handle directly; consequently the access to the RDBM is done using Object Relational Mapping (ORM) software, in this case SQLAlchemy. Moreover, due to the internal model description that has been created, the application can potentially generate either the collection of classes in any specific OO language and XSD or OWL schemas.



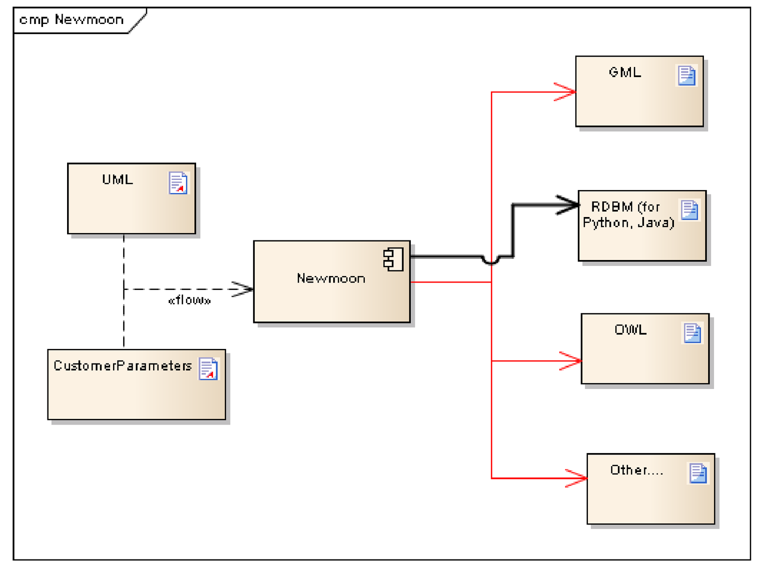
Clearly the application result object(s) depend on the specific transformation (UML to PythonRDBM, to JavaRDBM, to GML....) that is required.



A typical python application based on xmi2rdbm uses the UML generated library (ModelLibs) and class-to-rdbm mapping (DBMapping) to configure a rdbm-side (SQLAlchemy) and client-side (Django) application.

The EPB (a user written class/interface) is a single point of access from the client-side to the rdbm-side. It exposes to the client-side methods to create/update/delete/seach Model objects and relays on the rdbm-side for the db persistence mapping.

The XML document generated from the Django application complains about the GML generated from the same UML model.



Future work aims to integrate the UML2RDBM core parser into Newmoon (http://bond.badc.rl.ac.uk/newmoon) which offers an online version of Fullmoon. The new core parser could easily be extend to offer these UML transformation services.

British Atmospheric Data Centre
NATIONAL CENTRE FOR ATMOSPHERIC SCIENCE
NATURAL ENVIRONMENT RESEARCH COUNCIL

Centre for Environmental Data Archival
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESERACH COUNCIL

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

Science & Technology Facilities Council