POLITECNICO
DI MILANO

SpatialDBgroup

GeoUML Methodology and Tools
An Overview

29 settembre 2011

CONTENTS

1. INTRODUCTION

1.1 Objectives

The ***GeoUML Methodology*** and the ***GeoUML Tools*** described in this document have been developed in order to support the management of a geographical ***Conceptual Schema*** and to perform the automatic validation of the conformance of a ***Data Product*** to a given *Conceptual Schema*. The fundamental principles of this development have been:

- to adhere to the ISO 19100 standards whenever they apply
- to be implementable on current technology
- to be independent from any specific (commercially or open) GIS product
- to keep a clear separation between the conceptual and the implementation levels.

The tool which manages the Conceptual Schema is called ***GeoUML Catalogue***; the model used for defining the schema is called ***GeoUML*** (model).
The tool used for checking whether a Data Product is conformant to a given Conceptual Schema is called ***GeoUML Validator***.
For the transfer of a Conceptual Schema between different Catalogues or from a Catalogue to the Validator a file called **Specification File** is used.
The Data Product to be validated must be implemented using one of several predefined ***Implementation Models*** **(IM)**, which transform the Conceptual Schema into a Physical Structure.
The GeoUML Tools are developed entirely in the Java programming language and can be run on every platform that support this programming language.

1.2 Applications

The GeoUML Methodology and the GeoUML Tools can be used for the design and the implementation of Spatial Databases in general because they permit to:

- specify the contents of a Database on a conceptual level while also specifying the topological properties that the data must meet
- derive from the conceptual specification not only the physical structure of the Database, but also the corresponding physical structure of the files in the Shapefile or GML format that can be used for the different tasks revolving around the Database, for example: delivery files, files to distribute the data to the users, file for exchanging the updates, etc...
- to verify the conformance of a Data Product to the specifications

A context where the GeoUML Methodology can be particularly useful is that of a strictly consistent SDI. A strictly consistent SDI is a Spatial Infrastructure where different Local Spatial Databases are independently managed by different Local Authorities, but on the whole these databases must represent *an integrated model of a global territory*.
A particular difficulty in this context is constituted by the different technical solutions adopted by different Local Authorities for physically storing their databases. For this reason the common data content and the global spatial constraints must be defined at conceptual level, but they must be checked on databases implemented with different implementation models.

1.3 Software Evolutions and Releases

The GeoUML Methodology is subject to continuous change in order to be better adapted to the requirements which emerge from real applications; this documents refers to the features which are implemented by the *consolidated release* of the software, usable through download by authorized users; other features, which have been only subject to study or prototyping and will be perhaps included in future releases of the software, are not mentioned in this document.

The numbering mechanism of the software releases is related to the evolution of the single tools and to the structure of the Specification File. The general structure of the versioning number is “x.y”, where:

- “x” refers to the evolution of the Specification File, i.e. “x” changes when the structure of this file is changed
- “y” refers to the evolution of the particular software tool (Catalogue or Validator)

Important: the specification file is produced by the Catalogue and so the version of the specification file (that is the version of its structure) is determined by the “x” value of the version of the Catalogue that produced it. A Validator, in order to read a specification file, must have a version number with the same “x”.

In order to be able to use existing specifications, even if the specification file format evolves, it will be necessary to guarantee the backward-compatibility of the Catalogue in relation to “x”, i.e. a Catalogue with version “x” should be able to read a specification with an inferior “x”.

1.4 Documentation ***(the production of the documentation in Italian and English is underway – the currently available documentation is listed on the web site)***

The documentation of the GeoUML Methodology is made of two kinds of documents:

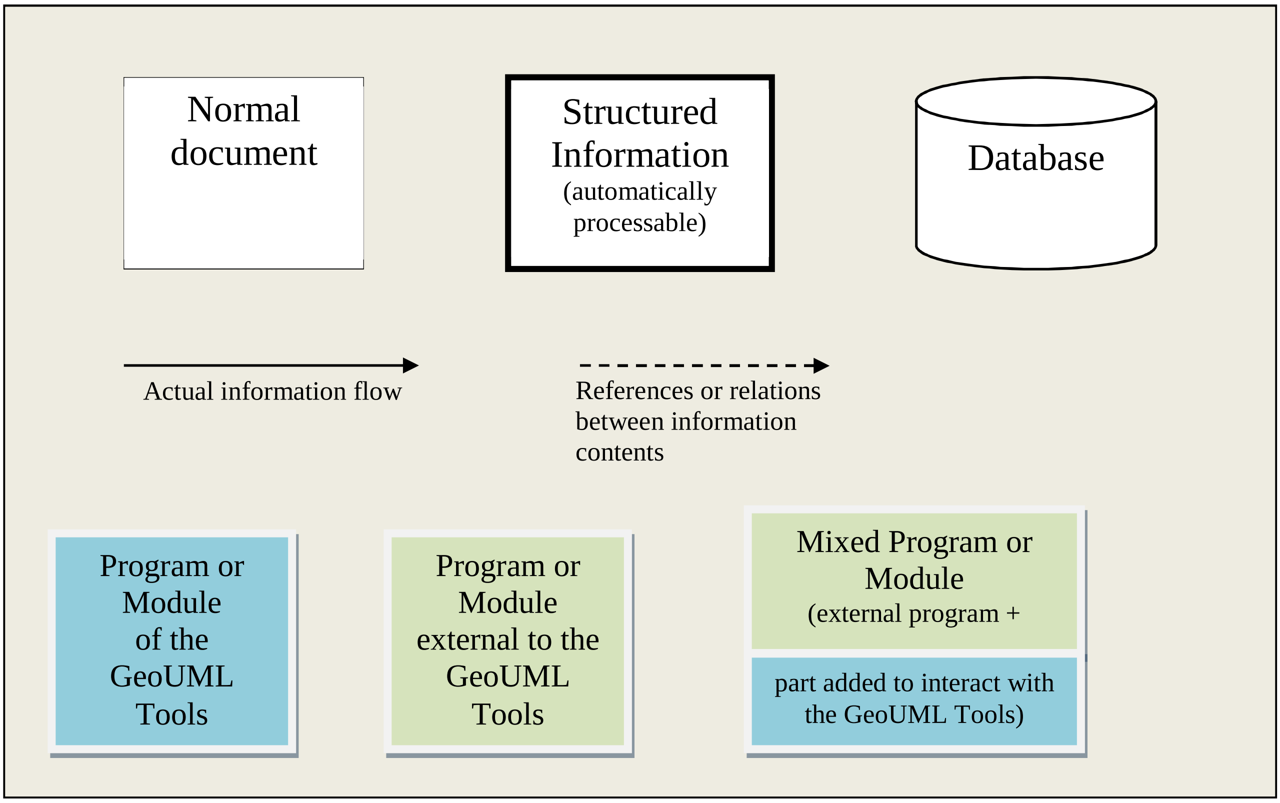
- Specifications: they are documents that focus and consider of primary importance the completeness and the precision of the definitions, they are not meant to explain the operative procedures:
  - *Il modello GeoUML (versione 2008) [The GeoUML Model (2008 version)]*
  - *Il Modello Implementativo Shape_Flat [The Shape_Flat Implementation Model]*
  - *Il Modello Implementativo Shape_Topo [The Shape_Topo Implementation Model]*
  - *Il Modello Implementativo ESF_SQL_Oracle (multigeometria) [The ESF_SQL_Oracle (multi-geometry) Implementation Model]*
  - *Il Modello Implementativo ESF_SQL_PostGIS (multigeometria) [The ESF_SQL_PostGIS (multi-geometry) Implementation Model]*
  - *Il Modello Implementativo ESF_SQL_Oracle (monogeometria) [The ESF_SQL_Oracle (single-geometry) Implementation Model]*
  - *Il Modello Implementativo ESF_SQL_PostGIS (monogeometria) [The ESF_SQL_PostGIS (single-geometry) Implementation Model (implementation to be agreed upon)]*
  - *Il Modello Implementativo ESF_GML [The ESF_GML Implementation Model]*
- User Guides: their focus is to explain how the general methodology and the tools work; the user guides are:
  - *GeoUML Methodology & Tools – An Overview (this document)*
  - *Guida all’uso del GeoUML Catalogue [User Guide of the GeoUML Catalogue]*
  - *Guida all’uso del GeoUML Validator [User Guide of the GeoUML Validator]*
  - *Guida ai Modelli Implementativi di tipo Flat [User Guide of the “Flat” Implementation Models]*
  - *Guida alla implementazione delle proprietà geometriche del GeoUML [User Guide of the Implementation of the GeoUML Geometric Properties]*

1.5 Notation Used in the Pictures

For the most part the pictures represent some flow of information between software components. In these pictures the notation illustrated in Picture 1.1 is used.

The distinction between normal documents and structured information is fundamental in this context, because one of the goals of the GeoUML Catalogue is to manage a Content Specification in terms of structured and automatically processable information, which differs from what traditionally happens in the specifications of the tenders relative to sets of geographical data.

The instruments that constitute the “GeoUML Tools” are usable in collaboration with other instruments, that's why in some pictures it is necessary to differentiate, as shown in Picture 1.1, between the GeoUML Tools own components and other software components. In some cases plug-ins for other software have been developed, thus specializing external software in order to make it able to collaborate with the GeoUML Tools, as shown in Picture 1.1 (mixed components).


Normal document
Structured Information (automatically processable)
Database
Actual information flow
References or relations between information contents
Program or Module of the GeoUML Tools
Program or Module external to the GeoUML Tools
Mixed Program or Module (external program +
part added to interact with the GeoUML Tools)

Picture 1.1

2. CONTENT SPECIFICATIONS AND THE GEOUML CATALOGUE

2.1 Content Specifications

A Content Specification (CS) describes the informative content of a Database. This is done by the use of different kinds of definitions, with different purposes:

1. definition of **informative elements**, i.e. of the elements that have to be represented in the Database (classes, attributes, geometric attributes, associations, domains) irrespective of the way in which these elements are portrayed in a particular data memorization technology.
2. definition of **integrity constraints**, i.e. the intrinsic properties that the informative elements must meet (by "intrinsic properties" we mean the properties that are verifiable on the informative elements themselves, without directly looking at the real world)
3. **descriptive elements**: in this category we can find all the pieces of information which can be used by human beings to understand how to interpret a database content in terms of represented reality and vice versa a real situation in terms of the informative elements (e.g. the geographical-data-gathering procedures)

Together the informative elements and the integrity constraints form the Conceptual Schema, i.e. the part of the specifications which is formally structured (that can be automatically processed).

Together the rules that are used to define the structured part of the specifications form the GeoUML model; the precise definition of the fore-mentioned rules can be found in the document "Il modello GeoUML (versione 2008) [The GeoUML Model (2008 version)]".

2.2 GeoUML Catalogue Features and Capabilities

To create, modify and view a Content Specification a software called "**GeoUML Catalogue**" is used.

Essentially, the GeoUML Catalogue has all the functionalities to manage a Content Specification, in particular:

- Functions to Import and Export a Specification (reserved for the Specification Administrator)
- Functions to Edit the contents (reserved for users authorized by the Administrator)
- Functions to View and Search the contents (for all users)
- Functions to prepare and generate the standard documentation (reserved for users authorized by the administrator)

Also, the GeoUML Catalogue has other functionalities which are relative to the production of Physical Schemas based on their corresponding Implementation Models. The latter functionalities are explained in chapter 4 of this document.

2.3 "Catalogue Viewer" and "Catalogue Editor" Versions

The GeoUML catalogue is distributed in two versions:

- **Catalogue Editor**: provides the full set of functionalities
- **Catalogue Viewer**: provides a reduced set of functionalities which are relative to the visualization only.

2.4 File for the Transfer of the Specifications

The GeoUML Catalogue saves the Content Specification in use in its Internal Database.

The content of the Internal Database can be exported/imported in an XML format whose schema is published (XML schema). The file which defines this XML schema is called "sc.xsd".

The file which contains a CS in this format is called Specification File and its file extension is ".scs".

The Specification File permits to transfer a CS not only to other GeoUML Catalogues, but also from the Catalogue to the Validator.

The Specification File contains, in addition to the content specification CS, also the pieces of information relative to the physical schemas (see chapter 4).

References

For a correct use of the Catalogue a good understanding of the GeoUML model is required. This model is defined in the document

"Il modello GeoUML (versione 2008)" ["The GeoUML model (2008 version)"]

The use of the Catalogue to manage a conceptual specification is described in the document

"Guida all'uso del GeoUML Catalogue" ["User guide of the GeoUML Catalogue"]

N.B. For the most part this document is about the definition aspects of the physical schemas (see chapter 4)

3 GEOUML VALIDATOR AND CONFORMITY WITH THE SPECIFICATION

3.1 Actual Conformity and Intrinsic Conformity

The conformity of a Data Product DP to a content specification CS is made of two aspects:

1. **Actual Conformity:** the actual conformity is about the correspondence between the informative content of the Dataset and the portion of the Real World (also known as the **Reference Territory**) to which the Dataset refers to; the informative elements and the related descriptive elements of the CS determine how that correspondence must be evaluated
2. **Intrinsic Conformity:** the intrinsic conformity is about the consistency of the information contained in the Data Product; the informative elements and the constraints of the CS are both used to evaluate the intrinsic conformity

The evaluation of the quality of a Data Product must consider both the Actual Conformity and the Intrinsic Conformity to the Content Specification.

The GeoUML Tools, the Validator in particular, can only support the evaluation of the intrinsic conformity (because it is the only one that can be checked automatically) and so in this document when the term “conformity” is used without any specification, we are referring to the intrinsic conformity alone.

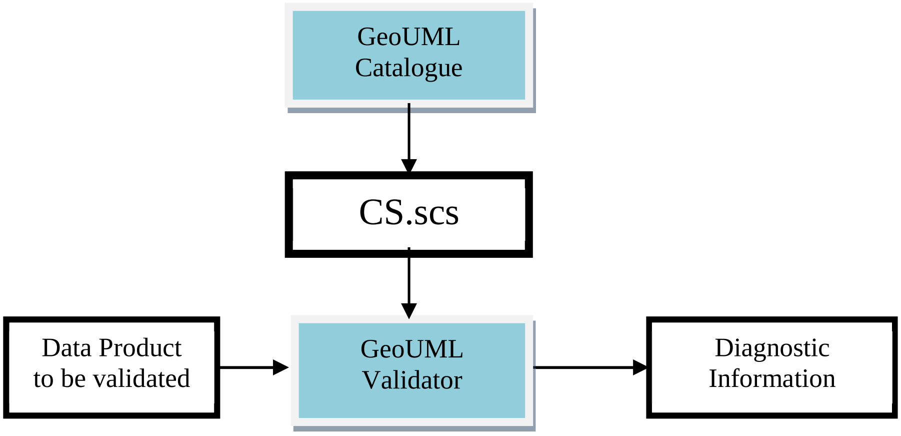
It's worth mentioning that even the traditional testing procedures, that are meant to evaluate the Actual Conformity, can take great advantage from the support provided by the Validator, because some errors of the data relative to the Intrinsic Conformity are often an indication of possible errors of Actual Conformity. For example, an indication of missing instances of a feature type can be an useful Warning for the tester; the proof that this absence corresponds to a situation in the real world or otherwise to an error is a concern of the Actual Conformity.

A Data Product DP is intrinsically compliant to a Content Specification CS if and only if:
a. all the data contained in DP correspond to informative elements of CS
b. the data contained in DP satisfy all the integrity constraints defined in CS

It should be noted that the (a) condition does not imply that for every informative element of CS some corresponding data must exist; for example, if in the Reference Territory of a Data Product DP there are no real instances of a given Feature Type FT defined in CS, in the Data Product DP there won't be any instance of that feature type and this absence will not break the Intrinsic Conformity of DP in respect to CS.

3.2 The GeoUML Validator

The **GeoUML Validator** is a tool that can be used to perform the checking of intrinsic conformity of a generic Data Product in respect to any given CS that can be managed by a GeoUML Catalogue. The Validator uses the specification exported by the Catalogue to analyse the Data Product and produce diagnostic information (see picture 3.1).



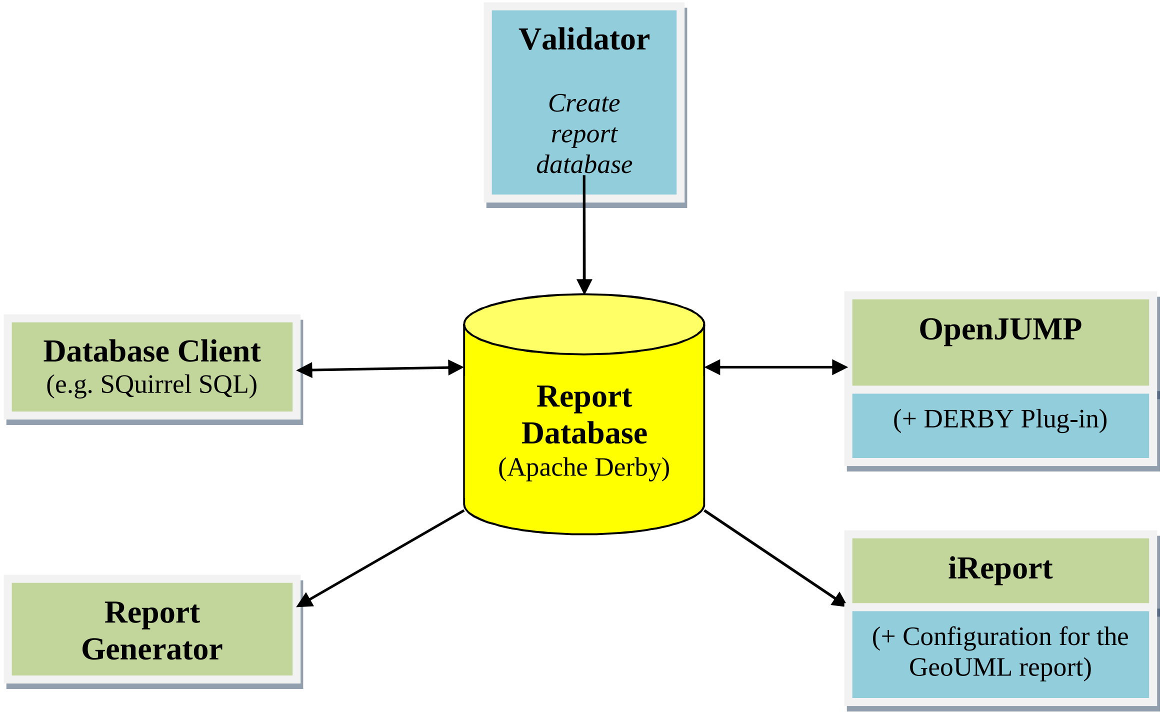
Picture 3.1

The selection and the organization of the diagnostic information is a complex problem, for various reasons:

- To be useful, the diagnostic information must refer to elements of the specification (Feature Types, Associations, Attributes of different kind and Constraints) even if the errors are found in the physical structure of the Data Product (whose reference is also important)
- The organization of the diagnostic information must be flexible because different uses require different organizations

To deal with the first problem, for every error the Validator reports both the reference to the conceptual element on which the error occurred, and the reference to the involved physical object among the many possible physical objects that represent the conceptual object.

To deal with the second problem, since it's difficult to produce a report that meets many different needs, the Validator creates a **Report Database** that can be used in different ways depending on the needs and the capabilities of the particular user, as shown in picture 3.2. The Report Database is built upon the Apache Derby Embedded technology, which is open source and freely available. The user must install Derby on his machine and then tell the Validator the destination folder for the creation of the Report Database. The pieces of diagnostic information provided by the Validator are often accompanied by a geometry, which means that they contain the geometry of the element affected by the error. The current release of DERBY does not provide Spatial Database extensions and so the geometries are memorized with the WKB standard format in a "binary object (blob)".



Picture 3.2

The 4 ways to use the database shown in the picture are:

- The use of a generic Database Client, i.e. a program which is able to execute queries on a Derby Database (there are many tools like this); this method is the most powerful and is recommended for users familiar with the use of relational databases but it has a drawback: it's impossible to look directly at the geometries (that in this case should be analysed on the data of the Data Product)
- The use of the client for geographical data OpenJUMP. A plug-in is provided that enables the connection with the Derby Database and the visualization of the geometric data.

- The use of a Report Generator to create custom-made reports; this method is recommended for users who need to reiterate particular analysis on the diagnostic, but on the first use requires someone who is able to configure the Report Generator
- The use of the Report Generator "iReport", pre-configured for the creation of the standard reports, called "**GeoUML Reports**"; this is the simplest way, suitable for the inexperienced user, but it is not flexible.

3.3 Implementation of the Geometric Properties of the GeoUML

The geometric properties of the GeoUML are defined at a very abstract level, based on logic and mathematics, more precisely on a representation in the Euclidean continuous space. In that space the coordinates of the points are represented with real numbers.

In the actual implementations the point coordinates must be represented by discrete numbers and so the space of the geometries is a discrete space. This aspect of the implementation raises many problems that must be taken into consideration for the creation of a Data Product that must conform to a GeoUML specification.

The Validator, in order to analyse the Data Product, must apply an interpretation of the existing geometries in the discrete space to verify that the geometries satisfy the properties defined at the conceptual level.

References

The procedures for the use of the Validator and the structure of the Report Database are described in the document:

"Guida all'uso del GeoUML Validator" ["User Guide of the GeoUML Validator"]

The rules of interpretation of the geometries implemented in respect of the conceptual requirements are discussed in the document

"Guida alla implementazione delle proprietà geometriche del GeoUML" ["User Guide of the Implementation of The Geometric Properties of the GeoUML"]

4. IMPLEMENTATION MODELS AND DATA PRODUCT SPECIFICATIONS (DPS)

4.1 Definitions

At the conceptual level, a Content Specification is intentionally abstract, i.e. it doesn’t define the aspects that are implementation-dependent. In order to actually create a Data Product that conforms to a CS, it is then necessary to specify a method of implementation, that in turn depends on the adopted technology.

Basic terminology:

1. **Physical Schema:** it’s the definition of the physical structure of the Data Product; for example, let’s consider the most common physical formats: an XSD for a GML file, the structure of the shape files for a dataset in the Shape format, the DDL instructions (CREATE TABLE, etc.) for a SQL Database.
2. **Conceptual-Physical Mapping (CFMapping):** it’s the definition of the existing correspondences between the elements of the physical schema and the elements of the Conceptual Specification (for example, the indication that a particular feature type is physically represented by a certain relational table or by a given shapefile)
3. **Implementation Model:** it’s a set of rules that enables to automatically generate a Physical Schema and a Physical Mapping from a Conceptual Specification

An Implementation Model is made of a set of rules that doesn’t depend on the CS to which it is applied; therefore the following statements are both valid:

- The same Implementation Model can be applied to different Content Specifications, producing different Physical Schemas that share the aspects related to the technology, but refer to different contents;
- The same Content Specification can be materialized following different Implementation Models, resulting in physical schemas that represent the same contents in a different manner.

The basic rules of an Implementation Model are fixed.
In some situations it’s advisable to introduce a certain degree of flexibility in the generation of the CFmapping: for this reason some IM are associated with particular mechanisms that permit to guide the generation process of the CFmaping – these mechanisms are of two kinds:

a. **Being Parametric**: there can be choices in the generation process of the CFmapping
b. **Adaptability**: the generated CFmapping can be modified manually

In practice, the current IM, even the most rigid ones, have a minimum level of flexibility obtained through the use of parameters.

4.2 Data Product Specifications (DPS)

A DPS is a data structure that contains the guidelines that are necessary for the system to generate a CFmapping starting from a CS. These guidelines are:

- The Implementation Model to use
- A value for every parameter that the selected IM asks for

The definition of a DPS takes place in the GeoUML Catalogue. In the Catalogue only one Conceptual Schema is present (CS), but many DPS can be defined (every DPS has its own name).
Again, in the Catalogue it is possible, following the definition of a DPS, to produce a CFmapping, that is also saved in an internal data structure. Starting from a CFmapping a Physical Schema can be generated.
The actions to be executed in the Catalogue in relation with these concepts are therefore 3 and they follow the following dependencies:

1. *Definition of a DPS:* it is independent, it can be defined at any given moment and it’s not affected by changes in the CS
2. *Generation of the CFmapping* (with possible adaptations, if the IM provides them): it depends both on the CS and the DPS used for the generation, therefore a change in the CS or in the DPS invalidates the CFmapping
3. *Generation of the Physical Schema:* it depends on the CFmapping, therefore it is invalidated by a change in the CFmapping

For a matter of flexibility, the 3 actions can be taken separately, therefore:

- Modifications made on the CS do not require changes on the DPS that were already defined, it’s only necessary to recreate the CFmapping and the Physical Schema
- The generation of the Physical Schema is supported even in the “Viewer” version of the Catalogue, because it is totally rigid and it does not change the contents of the Specification File

The Catalogue stores, in addition to the CS, all the defined DPS and the last CFmapping that has been generated, and this information can be exported in the Specification File and therefore imported in the Validator. Given the fact that for every DPS only one CFmapping can exist, the DPS name identifies the CFMapping and ultimately all the necessary information, except for CS, to interpret the structure of a Data Product.
When a DPS is modified, if the relative CFmapping is present it is deleted and must be explicitly regenerated.
Notice that the Physical Schema is not memorized in the Catalogue and so it isn't stored in the Specification File. The format of the Physical Schema can change in a radical manner according to the type of technology being used; for example, in case of an IM based on a SQL Database, the Catalogue generates the SQL definitions (CREATE TABLE, etc.) that can be used to create the Database. The user can memorize the Physical Schema in the manner that best suits his needs and his goals.

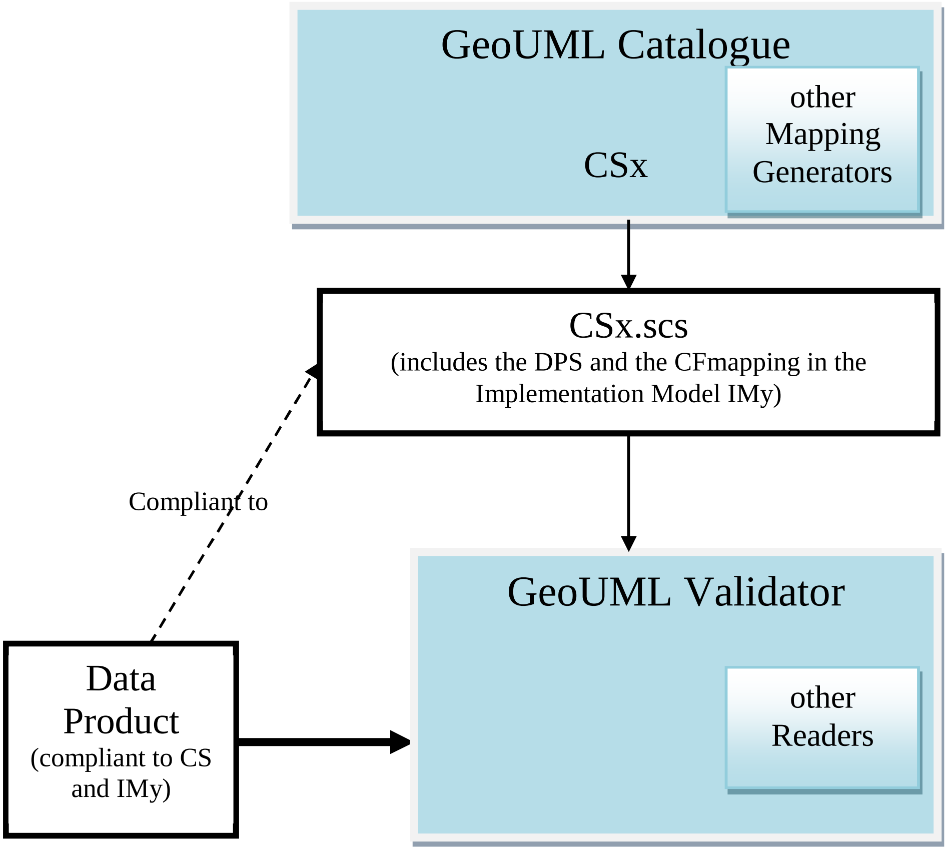
The GeoUML Catalogue can be used to generate a Database containing the CFMapping that is called the **CFMapping Database**. This database, based on the Apache Derby Embedded technology, contains all the information that defines the generated CFMapping and so it can be used as a source to create the desired documentation.
The procedure is much like the one used for the Report Database that is generated by the Validator, and the considerations relative to the access to the database and the derivation of the reports are very similar. A custom-made report has also been created to be used by the iReport software in order to generate a **GeoUML Mapping Report** that should be sufficient for most uses.

4.3 Stucture of the Tools and Implementation Models

The software architecture of the GeoUML Tools is such that it is possible to extend the capabilities of the Catalogue and the Validator to support new kinds of Implementation Models. This extension is based on the implementation of additional software components, called "Plug-ins", that can be easily added to the existing tools.
For each new Implementation Model, two new plug-ins must be developed:

- One Plug-in for the Catalogue, called **Mapping Generator**
- One Plug-in for the Validator, called **Reader**

In picture 4.1 the Validator is shown reading a Data Product that is compliant to a specification CSx and to an Implementation Model IMy; the Validator uses a Reader to interpret the Implementation Model IMy



Picture 4.1

4.4 Implementation Model Categories

The implementation models depend on the technology to which they pertain; at the moment, the technologies of reference are mainly the Shape File format, the GML, and the Georelational Databases (spatial SQL).

It's also convenient to classify the IM according to the typical function of the Data Products that use them:

- **Transfer IM** for the Data Product used for the exchange of data. For the Transfer Models a rigid ant not adaptable IM is recommended, because the receiver of the transfer must know exactly how the Data Product is structured. Typically the Transfer IM are based on the Shape File and GML technologies.
- **Work IM** for the Data Product used by an Organization to do its business. For the Work IM a flexible IM is recommended so that it is possible to adapt some aspects of the generated physical schema to the specific requirement of the organization that intends to use it. Typically, the Work IM are associated with the Georelational database technologies.

The main mapping problems that the rules of an IM should define are:

- **Structure Mapping**: how to transform a structure which consists of feature types, associations, hierarchies, enumerated and hierarchical attributes of the GeoUML, into the corresponding structures of the destination technology.
- **Mapping of the geometric model:** how to transform the geometric types of the GeoUML into the types that are available in the destination technology.
- **Mapping of the special constructs of the GeoUML**: how to implement the segmented and subregion attributes and the B3D surfaces using the constructs available in the destination technology.

It is not necessary for the IM to translate the constraints – the checking of the constraints, as we as seen, is a task reserved to a specialized tool – the GeoUML Validator.

As far as the Structure Mapping is concerned, a fundamental category of physical structures that can be used is the one that consists of **flat** structures: when using this kind of structures the GeoUML must be mapped into tables that are not nested. It does make sense to define flat IM for the GeoRelational technologies and for the Shape File technology, while for the GML, which naturally supports the articulated structure of the (Geo)UML, **nested** structures are more appropriate.

As far as the Geometric Model Mapping is concerned, the implemented geometric model which today has the most widespread adoption is for sure the one based on the Simple Feature model (SF) standard. All the implemented IM apply this standard with an extension, largely supported by actual systems, to handle the 3D representation of the points and the lines (Extended Simple Feature – ESF)

In the following picture 4.2 a structure for the classification of the implemented Implementation Models in the last release of the GeoUML Tools is shown. The highlighted rectangles represent the 7 implemented Implementation Models while the other rectangles represent the categories according to which those IM can be grouped. The classification is multiple, that is an IM can pertain to many categories; for example ESF_GML structurally belongs to the nested types category but functionally belongs to the transfer type category.

The structure of the classification is based on the following rules:

1. For the most part the IM (6 on a total of 7) are of the flat type; that's because this structure is very simple and largely supported; we can further classify this models into groups according to different principles:
   1.1. There are 4 IM that are Database-oriented; those IM can be distinguished according to these aspects:
      1.1.1. The multi-geometry SQL IM, in which a relational table can contain more than one geometric attribute; there are 2 versions based on the technology (ORACLE or PostGIS)
      1.1.2. The single-geometry SQL IM, in which a relational table can only contain one geometric attribute – this limitation has been introduced in order to support those GIS tools that are unable to handle the multi-geometry model – like seen in the previous classification, there are 2 version based on the technology (ORACLE or PostGIS)
   1.2. There are 2 transfer IM based on the Shape technology and that can be differentiated according to their representation of the geometry:
      1.2.1. SHAPE_FLAT represents the geometry object by object, like all the other IM classified as IM based on objects
      1.2.2. SHAPE_TOPO uses a topological representation of the geometries

2. A particular IM, called ESF_GML, is based on a nested structure; this model follows the ISO 19100 standard for the transfer of files in the GML format. Since the GeoUML is an extension of the UML and the standard process recommended by the ISO standard to generate a GML schema consists on the application of a set of rules coming from an Application Schema (AS) defined in UML, the fundamental step in the ESF_GML IM consists on the transformation of a GeoUML schema in a corresponding AS. This step basically consists on the conversion of the constructs that are proper of the GeoUML (segmented and subregion attributes, B3D surfaces, enumerated hierarchical domains) into the standard constructs of a particular AS. The next steps follow the rules prescribed by the standards, even though the are not fully automatic, because the rules allow some choices.

References

The operational aspects pertaining the definition of a DPS and the generation of the CFmapping and of the Physical Schema can be found in the document:

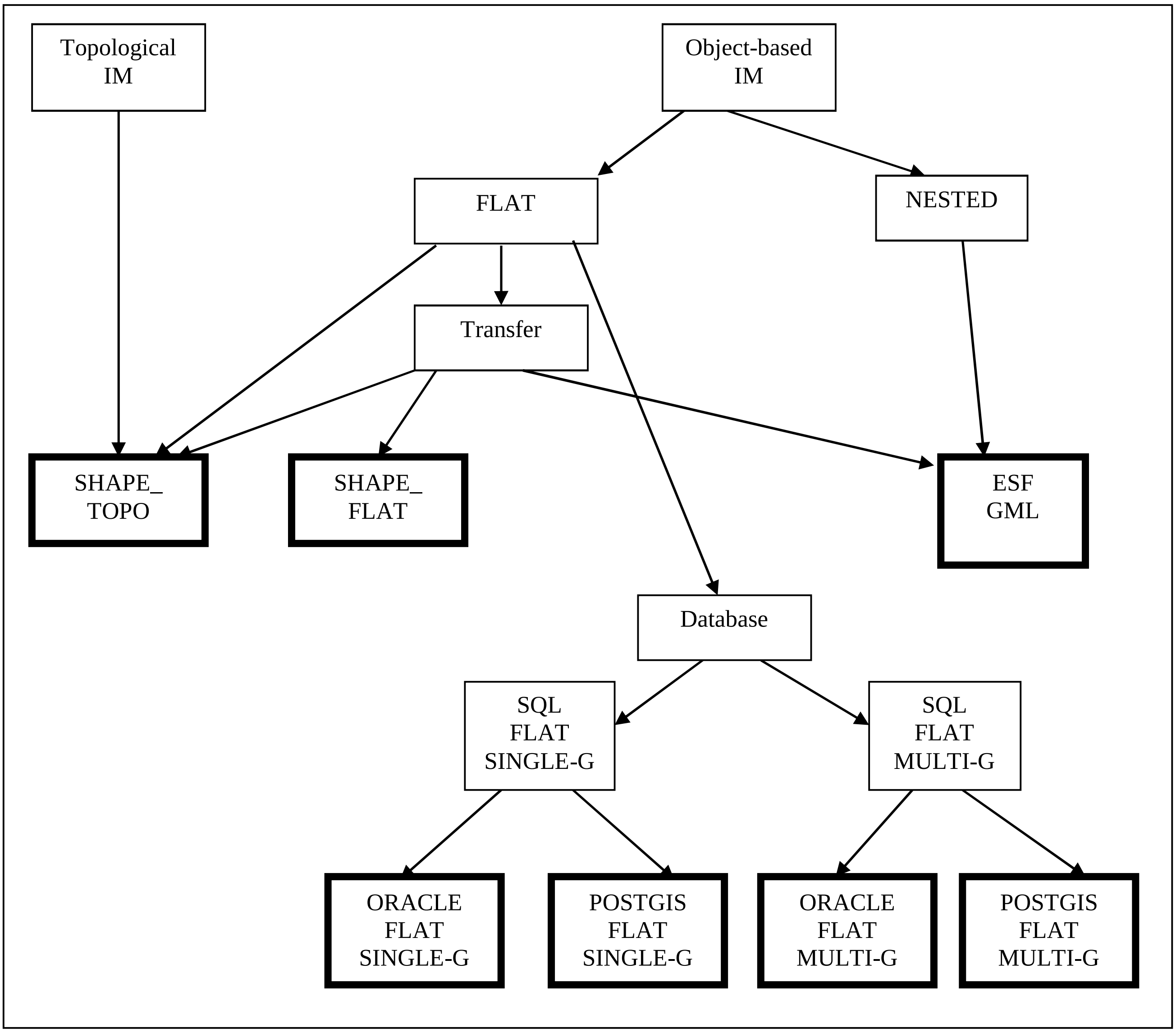
"User Guide of the GeoUML Catalogue"

The general aspects pertaining the flat structure, which 6 of the 7 IM follow, including the structure of the related CFmapping database and of the GeoUML mapping report can be found in the document:

"User Guide of the 'Flat' Implementation Models"

For every single IM xxx a specification document exists:

"The xxx Implementation Model"



Picture 4.2

APPENDIX – CORRELATION BETWEEN THE GEOUML METHODOLOGY AND THE ISO TC211 STANDARDS

General Aspects

The GeoUML methodology covers a field of problems that are already regulated by the ISO TC211 standards.

The fundamental principle of The GeoUML methodology is in being compatible with the ISO TC211 setting so that it will be possible to take advantage from all the tools compatible with the standards that will supposedly be developed; on the other hand the GeoUML methodology extends the setting of the ISO TC211 to meet three kind of needs:

1. To further increase the **decoupling between the Content Specifications and the Implementation Structures** in order to make the abstraction level of the Content Specifications more similar to the usual notion of a Conceptual Model, which is a “description of the informative contents, precise but independent from all the implementation aspects (which depend on the technology)”;
2. **To manage with a single tool** not only the structured documentation, but also the descriptive documentation intended to be read by human beings, so that those two kinds of documentation are kept aligned..

We will now analyse the reasons behind the needs for enrichment of the ISO setting mentioned before and behind the solutions brought forward in the GeoUML methodology.

Decoupling Between the Content Specifications and the Implementation Structures

In the ISO setting there are two components for the definition of the contents of a specification: the Application Schema (AS) and the Feature Catalogue (FC).

In the ISO TC211 setting the level of abstraction of the AS (and partly of the FC) is more dependent on the actual implementation than what should be appropriate for a conceptual Content Specification; in particular, the aspects that we thought needed to be changed with the introduction of the GeoUML methodology are the following:

a. The ISO geometry types define all the implementation aspects (interpolation, etc.) that are irrelevant in the analysis and comprehension of the contents (for example, at the conceptual level it is not necessary to specify if a “curve” is implemented as a LineString, a Composite Curve, etc.). In the GeoUML Methodology those geometry types are replaced by types that are more generic and are only subsequently transformed in ISO types by the Implementation Model.

b. The strict rules for the generation of the GML defined in the ISO 19136 forces the AS to contain detail choices that determine the encoding of the GML being produced, but those choices are irrelevant at the conceptual level and are incomprehensible for those who are not GML experts.

c. The specification of the constraints in the ISO TC211 is poor – the problem is sidelined referring to the use of the OCL language which is not suitable for the task for various reasons:
   - c1. OCL is not integrated with the aspects relative to geometric data, topological relations, etc.
   - c2. OCL is not syntactically appropriate for a conceptual specification because it is overly difficult for the common user
   - c3. OCL is not implementable

d. In the ISO some high-level constructs, like the Segmented Attributes and the Subregion Attributes, are missing and they must be represented through their implementation structure in terms of Composite Geometries (and thus being at the same time affected by the lack of expression of the constraints mentioned at the previous point), or in terms of linear referencing which currently does not have its own standard; in the GeoUML methodology the distinction between the different kinds of implementation for the Segmented Attributes is confined in the Implementation Model

e. The ISO TC211 lacks the high-level constructs for the management of the description of the 2D and 3D geometries that characterize the contents produced in the current state-of-the-art, which is flawed by the impossibility to realize a complete three-dimensional model.

The confinement of the aspects that do not belong to the conceptual level into the notion of Implementation Model allows the GeoUML methodology setting to ultimately obtain two benefits:

1. The definition of the Content Specifications, which consists of a complicated process of applicative analysis and of a comparison between different disciplinary point of views, can focus on the aspects that really pertain to the contents, without thinking about the implementation aspects, and allows a debate between application experts without the mediation of technology specialists

2. The Content Specifications are not affected by every evolution of the geometry technology and are therefore more durable; it is possible to develop new Implementation Models to support that evolution thus saving the investment in application analysis.

Management of the Specifications with a Single Tool

There are several available tools for managing the AS and the FC defined in the ISO approach; in particular, in the ISO TC 211 model for the specifications management the main component is the generic UML CASE TOOL which manages the AS. This choice is indeed reasonable considering the evolution history of the specifications; for example, the entire Spatial Model could not have been developed without this approach.

On the other hand, the existence of different Geometric Packages is irrelevant during the Content Specifications and, due to the scarcity of associations relatively to the presence of numerous feature types and attributes in the geographical databases, the use of UML diagrams is of little use.

In conclusion, the functionalities of a CASE UML tool which are really useful for the representation of a Content Specification are relatively few, while the functionalities needed to manage the many important descriptive aspects are missing.

To deal with this flaw, in the ISO TC211 setting the AS managed by the CASE UML is supported by a Feature Catalogue which must be managed by a different tool. The relationship between these two representations of the contents is not clear, and indeed different national organizations have proceeded in different ways (producing the AS first and then the FC or vice-versa).

The GeoUML Catalogue proposes to overcome those limitations with the addition, before the AS and the FC, of an integrated documentation which rigorously pertains to the conceptual level; from this documentation it is possible to create the Application Schema and the Feature Catalogue required by the ISO TC211. That production is realized through the definition of an Implementation Model which adds to the Content Specifications the definition of the implementation choices.

In the setting of the GeoUML Methodology the GeoUML Catalogue is more similar to a Feature Catalogue, but contains many elements of an AS, with the considerations relative to the abstraction model mentioned in the previous section, and in this way it overcomes the limitations of the CASE tools.