Formalizing Implementable Constraints in the INTERLIS Language for Modelling Legal 3D RRR Spaces and 3D Physical Objects

Kalogianni, Eftychia; Dimopoulou, Efi; Quak, Wilko; van Oosterom, P.J.M.

Publication date
2016

Document Version
Final published version

Published in
Proceedings of the 5th International FIG Workshop on 3D Cadastres

Citation (APA)
Kalogianni, E., Dimopoulou, E., Quak, W., & van Oosterom, P. J. M. (2016). Formalizing Implementable Constraints in the INTERLIS Language for Modelling Legal 3D RRR Spaces and 3D Physical Objects. In P. J. M. van Oosterom, E. Dimopoulou, & E. M. Fendel (Eds.), *Proceedings of the 5th International FIG Workshop on 3D Cadastres* (pp. 261-284). International Federation of Surveyors (FIG).

Formalizing Implementable Constraints in the INTERLIS Language for Modelling Legal 3D RRR Spaces and 3D Physical Objects

Eftychia KALOGIANNI and Efi DIMOPOULOU, Greece, Wilko QUAK and Peter van OOSTEROM, the Netherlands

Key words: Constraints, Legal and Physical Space, 3D Cadastre, THE INTERLIS Language, Spatial Data Modelling, Land Administration Domain Model (LADM)

SUMMARY

Building Information Models (e.g. BIM, IFC) and virtual 3D city models (e.g. CityGML) are revolutionising the way we manage information about cities. The main focus of such models is on the physical and functional characteristics of urban structures (Aien et al, 2015). On the other hand cadastral data models, (e.g. Land Administration Domain Model (LADM)), have been developed for legal information modelling and management purposes without providing correspondence to the object's physical counterperts. However, by definition, those two aspects need to be interrelated; i.e. a tunnel, a building, a mine, etc. always have both a legal status and boundaries as well as a physical description; while it is evident that their integration would maximise their utility and flexibility to support different applications. In order to examine the correlation between those aspects, a prototype system was developed based on model driven architecture practices, empahasizing in formalization of constraints. Constraints have been proved effective in providing solutions needed to avoid errors and enable maintenance of data quality; thus the need to specify and implement them is vital. The paper explores possibilities of linking 3D legal right, restriction, responsibilities (RRR) spaces, modelled with LADM (LADM, ISO 19152), with physical reality of 3D objects (described via CityGML, IFC, InfraGML, etc). During this implementation, INTERLIS, a standard modelling language for geodata exchange, was selected as modelling language, because of its claimed good support of constraints in vendor neutral manner and ease of obtaining actual implementation. The system's development cycle, from conceptual model to implementation of working prototype is illustrated through a proposed model for a Multipurpose Land Administation System (MLAS) for Greece. The prototype is developed in order to evaluate the strengths and limitations of the proposed conceptual model, as well as to investigate the efficiency and limitations of the existing technological (INTERLIS) tools. The process followed during the prototype development is cyclical and repetitive, providing feedback to the initial model during all the stages and improving it in terms of efficiency, reality and technical implementation. Main challenges among others during the system's development cycle include the following: 1. how to express various LADM constraints (OCL and semi-formal in standard) formally in INTERLIS, 2. how to include extensible hierarchical code lists in INTERLIS models (not natively available) and distinguish them from enumeration values, 3. how to model 3D volumetric primitives in INTERLIS (not natively available) and 4. how to link two models (legal - physical) using cross domain constraints.

Formalizing Implementable Constraints in the INTERLIS Language for Modelling Legal 3D RRR Spaces and 3D Physical Objects

Eftychia KALOGIANNI and Efi DIMOPOULOU, Greece, Wilko QUAK and Peter van OOSTEROM, the Netherlands

1. RESEARCH DESIGN AND METHODOLOGY

1.1 Introduction

In order to define conceptual schemas; i.e. the description of information systems in terms of data they should contain and the operations available to users to modify such data (Olive, 2007); the Object Management Group (OMG) has defined the UML/ OCL standards (Oriol et al, 2014). Unified Modelling Language (UML) is an OMG standard for modelling information systems, although powerful in describing and visualizing systems from a Platform Independent Model (PIM), is not capable of visually modelling every kind of constraint through its elements (e.g. classes, attributes, and associations) (Group et al, 2007). The transformation of the logical model into a physical database provides a better understanding of the model at a conceptual level while it reveals its the strengths and limitations. Additionally, the transformation from an object-oriented UML model (PIM) to an object-relational database model (in PostgreSQL/MySQL/Oracle Spatial) Platform Specific Model (PSM), requires mapping of elements in both type models.

The last decade, UML diagrams together with Object Constraint Language (OCL) notation have been found to be a suitable tool to express the designed constraints, together with the 3D geometries and associated operations defined in ISO19107 standard (ISO, 2008).

Models described with UML diagrams and constraints defined in OCL cannot be automatically converted into implementations. OCL has been defined as an extension to UML. Although it provides the possibility to model constraints in a formal language involving navigation through multiple classes and associations using a variety of condition statements, it is not capable of visually modelling every kind of constraint in an unambiguous manner.

Constraints are often initially described in natural language; however practice has shown that this results to ambiguities. Formal description of constraints of a model is a complex task; however of great importance, which needs to be operated in early stage of the conceptual design in order to reduce complexity. Consistency of (spatial) data can be checked if the underlying constraints are properly modeled and enforced. The term constraint thereby includes alpha-numerical as well as spatial definitions (Werder, S., 2009).

Various formalisation approaches applicable to 3D geo-constraints do exist today and they pave the way for extending the current state-of-art of constraint into a higher dimension. A brief review of them is presented in Section 2.3.

This paper is part of a wider research study of which some results have already been presented (Kalogianni E. et al, 2015, Kalogianni E., 2015, Kalogianni E., 2016). It explores possibilities of linking 3D legal RRR spaces, modelled with Land Administration Domain Model (LADM, ISO 19152), with their physical counterparts, described as 3D objects via CityGML, IFC, InfraGML, etc. To this purpose, standardization techniques are used, in order to explicitly define data models including their constraints, also testing the performance of advanced technological tools in terms of consistency and integrity.

1.2 Methodology

The methodological steps followed during this implementation are presented in Figure 1.

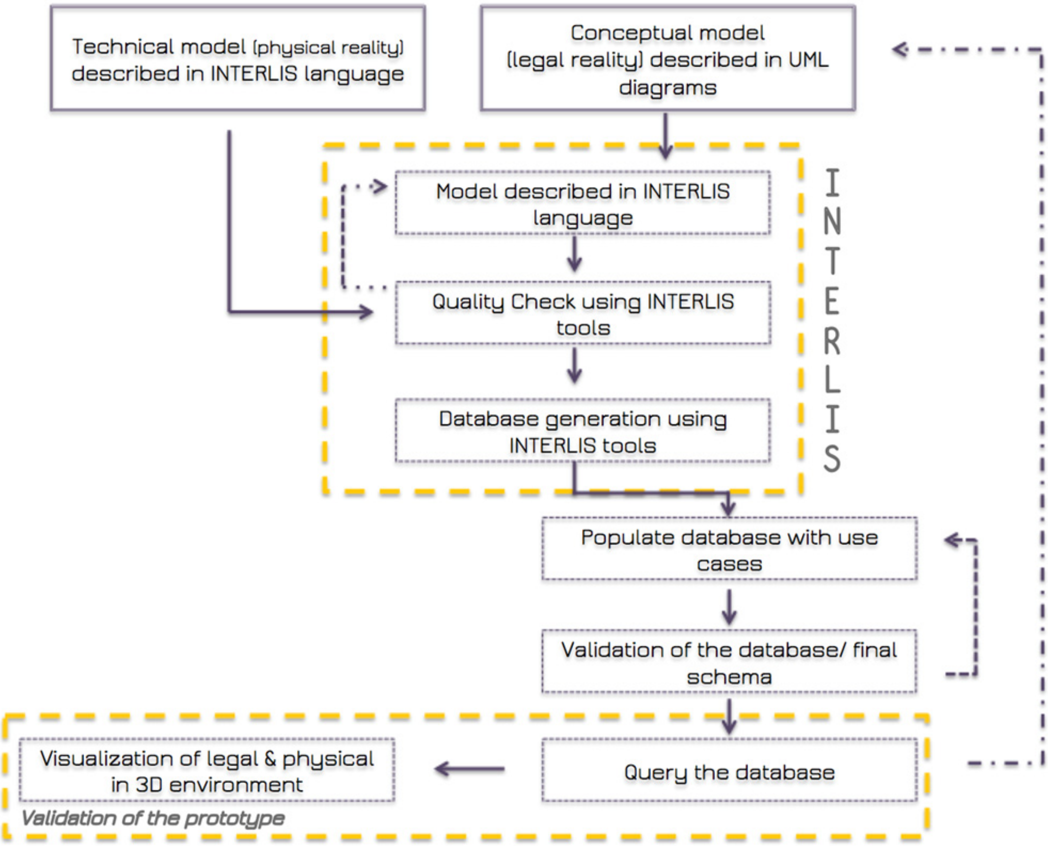
Technical model (physical reality) described in INTERLIS language
Conceptual model (legal reality) described in UML diagrams
Model described in INTERLIS language
Quality Check using INTERLIS tools
Database generation using INTERLIS tools
INTERLIS
Populate database with use cases
Validation of the database/ final schema
Visualization of legal & physical in 3D environment
Query the database
Validation of the prototype

Figure 1. Methodological steps followed during this implementation

The standardized model that offers a conceptual framework, LADM (Lemmen, 2012), is used for describing the legal reality of the objects. Based on it, the proposed model (3D Multipurpose Land Administrative System) for land administration in Greece, was described with UML diagrams using Enterprise Architect (EA) software. The proposed model, based on ISO 19152 LADM (Kalogianni, 2015) and described in UML, is first translated into into INTERLIS modelling language. Regarding the description of the physical aspect of 3D objects, a 3D City Model developed from the Swiss Land Management Group, described in in the INTERLIS language, was used.

Following, INTERLIS tools were used to automatically generate implementation components for specific environments; e.g. database schema and exchange file format. Real datasets were gathered from the Greek authorities and populated the database refering to the majority of the levels created in the spatial part of the proposed model, mostly concerning condominiums, 2D and 3D rural parcels, archaeological spaces, special real property objects and planning zones polygons. It should be noted that a small amount of sample data was created in accordance with the Greek legislative framework. The database thus populated was used for the assessment.

Although the currently available Greek datasets, do conform with the structure of the proposed model, as new elements/ components were added, sample data should be generated for terms of completeness; i.e. the class "GR_Archaeological" is a new proposed class/ level for the model for which data was created as there was no available data for such a case study. Analysis of the loading and querying of the database gave feedback and the necessary changes were made to the database schema and the initial conceptual model, in order to better represent an abstract of the reality. As a last step, the integrated physical and legal reality was visualized in 3D environment.

The paper is organized as follows: In the Theoretical Bachground Section 2, the necessary background information of the research design is briefly presented. Section 3 provides an overview of the most important (free) INTERLIS tools, analyzing the overall system architecture and INTERLIS tools during the development of the prototype. The implementation of constraints is discussed in Section 4. Finally, Section 5 provides conclusions and future work in addressing limitations and opportunites that arize from the prototype and the use of INTERLIS, as well as future trends in this domain.

2. THEORETICAL BACKGROUND

This section includes an overview of the relevant theoretical background. Examples of MDA - based applications implementing LADM profiles from other countries, related work towards the integration of 3D legal and physical objects, introduction to INTERLIS modelling language and its tools, as well as a brief description of the proposed model for the Greek LADM profile are discussed at the next paragraphs.

2.1 Other MDA based applications implementing LADM profiles

Automatic or semi-automatic methods may be tested through prototype implementation in order to avoid data loss and complicated procedures. The plethora of technological tools based on conceptual schema languages (CSL) that are currently developed and offer a computer processable result of the corresponding conceptual model resulting in direct implementable model descriptions has grown the interest in this field. Within this context, some representative MDA based applications, where LADM profiles have been implemented, are listed in this section.

In the Hespanha (2012) study, an original model was described in UML packages using Enterprise Architect (EA) software and then exported and parsed into Eclipse UML 2.0 class models and diagrams, enabling an implementation into a PostgreSQL/ PostGIS spatial database. The final result was a Java abstract layer, accessible to other applications running under the Eclipse Integrated Development Environment (IDE) and the corresponding database schema, which can be further populated with data.

Moreover, a Computer Aided Software Engineering (CASE) tool has been developed from Delft University of Technology, which performs a model transformation where the initial UML model expressed as an XMI is translated to a SQL file with a set of DDL commands (van Bennekom- Minnema, 2008). However, in order to achieve that, the LADM based model should be previously transformed into a Platform Specific Model (PSM). This implies that decisions must be taken regarding the hardware and software platform on which the model will be implemented (for instance from UML to a "PostgreSQL" environment). At another approach Hespanha et al (2015) IBM's Rational Software Architect was used for implementing an LADM country profile.

2.2 The legal and the physical reality of objects

Relations between spatial objects from legal and physical world are hard to implement, but can be supported via an Spatial Data Infrustructure (SDI) approach connecting the different registrations. The physical objects give the reference to the location and size of the legal spaces. In some cases legal spaces "contain" several physical counterparts, such as utilities, which do not constitute separate property units. Information about existence of those objects is then complementary and relevant to the context of developing the multipurpose cadastral systems (Gózdz et al, 2011).

Cadastral data models such as LADM (ISO19152, 2012) and ePlan (ePlan, 2010), manage and maintain 2D and 3D land and property ownership interests, including 3D ownership rights, restrictions and responsibilities (RRRs), only representing legal objects that do not usually integrate their physical counterparts (Aien et al, 2015). On the other side, 3D physical object models (BIM, CityGML, IFC, LandXML, etc) are focused only on the description of the geometry, while others are also semantically rich; however they do not include modelling legal objects.

As mentioned in Stadler and Kolbe (2007), since legal and physical objects are often maintained separately, integration inevitably leads to geometrical inconsistencies. Semantic information can help to reduce the ambiguities for geometric integration, provided it is coherently structured with respect to geometry. The need for integrating both legal and physical objects is underlined by Aien et al (2015), however few researchers have been undertaken to investigate this actual integration. The same authors propose a data model named 3D Cadastral Data Model (3DCDM) that supports both legal and physical information introducing Legal Property Object and Physical Object.

Additionally, researchers have proposed ADEs for CityGML in order to support also the legal aspect of the 3D objects described in the virtual city model. Dsilva et al (2009) developed an

ADE for cadastral purposes, Çagdas, (2013) developed an ADE to support the requirements of the immovable property taxation system in Turkey, while Rönsdorff et al, (2011) proposed an ADE using LADM to represent the legal space and examined how this can be mapped and encoded as a CityGML ADE; as LADM covers both the spatial (i.e. LA_SpatialUnit) and administrative (i.e. LA_Party, LA_RRR, and LA_BAUnit) aspects of land administration.

2.3 Constraints in data modelling

Constraints are rules/ conditions, which define certain properties that data must comply with. Constraints should be part of the object class definition, just as with other aspects of that definition, including attributes, methods and relationships; as they provide additional semantics to the model. Furthermore, the implementation of constraints (whether at the front-end, database level or communication level) should be driven automatically by these constraints' specifications within the model.

In order to define the constraints, the classification of typical constraints has been studied. The main types of traditional constraints include domain constraints, key and relationship structural constraints, as well as general semantic integrity constraints (Chiang et al, 1994). These classical types were extended by Cockcroft (1997) with new ones: topological, semantic and user-defined constraints, which were proven (van Oosterom, 2006) with broader context. In particular, the classification of constraints based on relationships between objects has the following components (van Oosterom, 2006): thematic constraints, temporal constraints (describing temporal relationships between objects), spatial constraints, which can be further categorized in topological constraints, direction constraints and distance constraints, as well as mixed constraints are common as the fundamental types of relationship constraints is mixed.

The approaches for modelling and enforcing the above mentioned types of constraints are plenty and diverse. Some of them are briefly described below. To start with, van Oosterom. (2006) reviews constraints in cadastral applications, focusing on the spatial side of the cadastre, in topographic applications as well as Web feautures services and presents a classification of constraints.

Constraint checking with database triggers is used by Louwsma (2006), Duboisset et al (2005) and van Oosterom (2006). Database triggers are parts of procedural code that can be used to automatically check certain conditions when data is inserted, updated or deleted. Triggers are used to avoid insertion of invalid data and therefore make use of the spatial extensions of the database management systems.

Another approach also presented in Werder (2009) is the constraint definition in an ontology presented by Mäs et al (2005). Constraints for spatial relations are defined by authors using a slightly extended version of Semantic Web Rule Language (SWRL), introducing a framework for the definition of spatial and other constraints of geospatial objects in an ontology.

Xu et al (2016) present a four-step approach to specify formal spatial constraints vai the following representations: 1. Natural Language, 2. Geometric/Topological Abstractions, 3.

UML/OCL Formulations, and 4. Model Driven Architecture (MDA). Further, they define new types of constraints: general-level constraints (applicable to all object sub-classes), parameterised constraints (containing numeric values, e.g. maximum distance), constraints allowing exceptional instances (to resolve cases that have not been defined) and constraints relating to multi-scale representations (to check the consistency between two levels of detail which model the same object). Additionally common sense rules to detect conflicting constraints are specified as well.

Object Constraint Language (OCL) is a declarative language and a standard "add on" to the UML. It is a constraint and query language at the same time designed for object-oriented modelling, which enables the automated parsing, processing and implementation of OCL constraints, referring to classes, attributes, associations, and operations (Group, 2010).
A part of the OCL constraints can be implemented by base table check constraints, but some of the more complex constraints, involving multiple classes, would require implementation through sub-selects in base table check constraints, or even with the use of Structured Query Language (SQL) assertions.

Tools implementing OCL to SQL translation do exist today; OCL2SQL from the open source OCL Dresden toolkit (Demuth, 2004, Demuth et al, 2009) running on the Eclipse Platform and MySQL4OCL code generator (Egea M., 2010) are the most well-known. OCL2SQL is part of a toolkit handling OCL constraints developed by the Dresden University of Technology, which generates a SQL check constraint, assertion or trigger for an OCL invariant based on the parsed, typechecked and normalised OCL expression (Oriol, 2014). MySQL4OCL tool is a code generator for a significant subset of OCL expressions (it does not cover the full OCL language), which is based on the use of stored procedures for mapping OCL iterators (Egea M., 2010). It is noted that those tools have an important limitation regarding implementation of spatial data types.

Moreover it should be noted that difficulties are faced to generate implementation (SQL/ DDL) for EA's UML/ OCL models. Indicative example is that there is no normative way to translate UML code lists into SQL expressions, as the corresponding enumerated types of the PostgreSQL are static and cannot be extended.

In this paper, the constraints are defined in UML models using OCL, which enables to express the constraints at a conceptual level in a formal way and platform independent/ vendor manner. Enterprise Architect (EA), which supports OCL syntax, as well as the ability to validate the OCL statement against the model itself, was used at the first steps in order to model the UML diagrams. This means that it enables the verification beyond just the syntax that the OCL statement is expressed correctly in terms of actual model elements, and that the kind of validation syntax that it uses corresponds to the actual data types defined for these elements: numbers, strings, collections, etc. Definition of constraints tested during the prototype implementation is described in Section 3.1.

2.4 The INTERLIS modelling language

As its name suggests, INTERLIS stands between (inter) land information systems (KOGIS, 2006). It is a well-known and reliable Swiss standard for geoinformation exchange, modelling and integration of geodata allowing co-operation between information systems and especially geographic information systems, which has been used and updated for many years. It is an Object Relational modelling language, very precise and standardized at the conceptual level to describe data models.

INTERLIS was choosen among other tools to be used for the transformation of the logical model into a physical database. It should be noted that it is not just another data exchange mechanism, but a data definition language based on the description of the transferable data model. INTERLIS allows the specification of data format and assures a strict separation of graphic descriptions and descriptions of underlying data structure (data definition) (KOGIS, 2006). Figure 2 depicts the multilateral character of INTERLIS.

The textual INTERLIS language offers the necessary complement to the graphic description language UML, as the INTERLIS described models are precise, unequivocal and can be interpreted without misunderstandings. INTERLIS is also based on the MDA principles, as it enables the utilization of data modelling in close connection with system neutral interface services (i.e. it comprises XML-based services), permits the implementation of a service in a computer system to be automatically derived from this model (KOGIS, 2006). By applying INTERLIS to the proposed model, directly implementable data models are provided, which speed up the technical implementation of LADM model (Germann et al, 2015).

Additionally, INTERLIS allows formal description of constraints as it includes an OCL - like constraint language and provides tools (e.g. INTERLIS checker) enabling more direct implementable models and at the same time support, check and process constraints.

Constraints in the INTERLIS language can be defined on object level (MANDATORY CONSTRAINT) or class level (SET CONSTRAINT, UNIQUE CONSTRAINT) (Germann M., et al, 2015). Constraints that refer to one single object are subdivided into requirements that must be met by each object of a class ("hard" constraints) and regulations, which in rare cases can be violated ("soft" constraints). Hard and soft constraints are further discuss in Secion 3.1.3. While constraints referring to a class demand the existence of an attribute value in an instance of a different class.

Moreover, spatial constructs are also available in the INTERLIS language and it provides a unique set of features which sets it well apart from other modelling standards (UML, XML – Shema, EXPRESS, etc) (Kalogianni.E. et al, 2015). One of the known limitation was almost no, limited support for 3D volummetric objects, which is expected to be overcome at the next INTERLIS version.

Considering all the above mentioned, the Swiss standard INTERLIS was considered as a challenging solution to get computer processable model description and transfer LADM classes via XML (Germann et al, 2015) during this prototype implementation. The INTERLIS tools are further explained in Section 3.

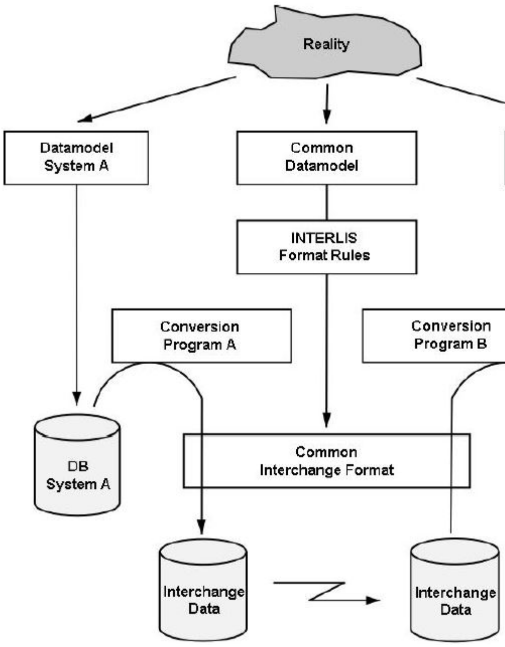

Figure 2. Data transfer between several databases via a common data model (data schema) described in a common data description language. (INTERLIS Reference manual)

2.5 The INTERLIS tools

As mentioned in Germann et al (2015) the intense use of INTERLIS in Switzerland would not be possible, if the language would not be supported by a wide range of tools. INTERLIS tools are used to automatically generate implementation components for specific environments; e.g. database schema or exchange file format. The tools (Java programs) used during this prototype are the following:

- INTERLIS compiler (ili2c): checks the syntactical correctness of an INTERLIS data model. It checks all kinds of syntactical errors: missing declaration of data types, syntax errors, wrong defined attributes or classes, etc. The compiler reads and writes INTERLIS models, permits necessary changes within and examines, whether models are in accordance with syntactic and semantic conditions of INTERLIS. Amongst others this compiler can generate automatically – in accordance with the present INTERLIS transfer service with XML – XML schema documents derived from INTERLIS models and also offers a plethora of output formats.
- INTERLIS checker (igchecker2): quality check INTERLIS XML data against INTERLIS data models (including specific constraints);
- INTERLIS UML editor: used to create INTERLIS models from UML diagrams or to visualize existing INTERLIS data models as UML diagrams;
- INTERLIS loader for PostgreSQL/ PostGIS *(*ili2pg):

Last years INTERLIS has shown "extroversion" to many GIS software and databases (Oracle), thus there are available data translator/ tools in QGIS (INTERLIS import/ export

plugin), FME Workbench (ili2 fme) as well as OGR support for INTERLIS reading and writing (Kalogianni, E., 2016).
The following figure depicts the INTERLIS tools and the sequence that were used during the implementation of the prototype.

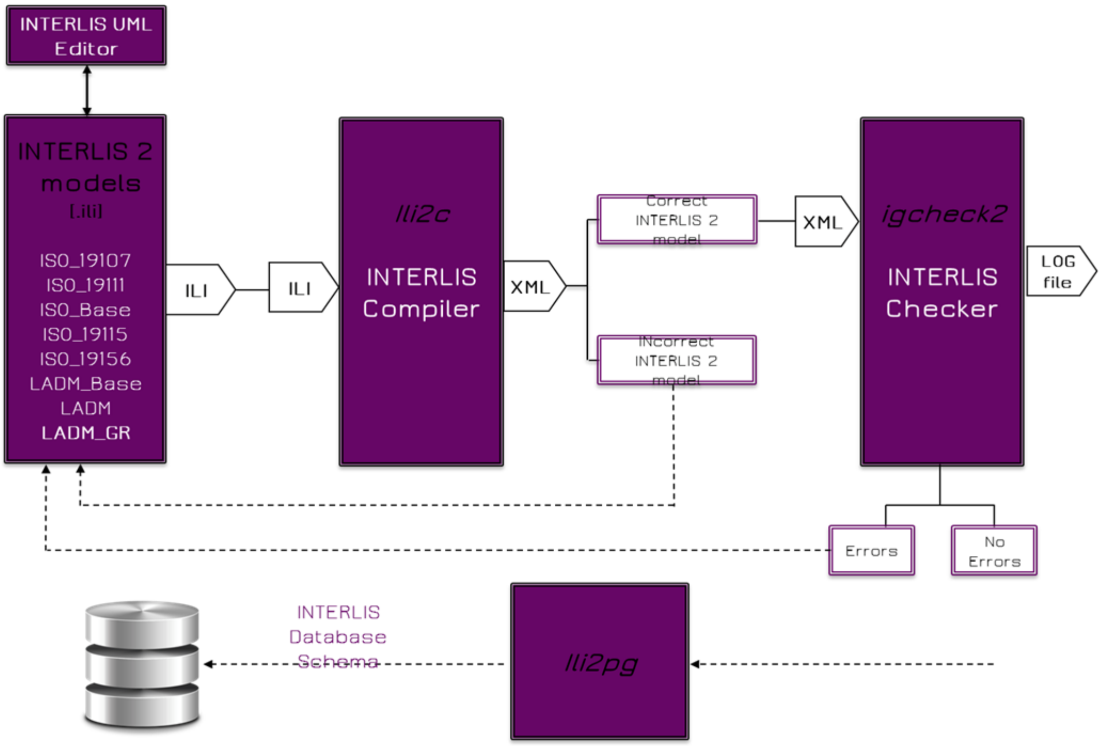

Figure 3. The INTERLIS tools used for the implementation of the prototype

2.6 Proposed 3D Multipurpose Land Administration System (MLAS) for Greece

The starting point of the development cycle of this prototype was a proposed model describing a 3D Multipurpose Land Administration System (MLAS) in Greece based on the international standard Land Administration Domain Model (LADM), ISO 19152. The proposed model both considers the current registration in 2D in Greece (which means that the code lists and attributes of this model are also included in the proposed model) as well as it is future proof covering the requirements for future registration (including 3D), serving the country's needs and inspired by the LADM standard and other country profiles.

This model is considered as an effort for overcoming current shortcomings, based on international standards, including the representation of a wide range of different types of spatial units, in 3D, aiming to establish an appropriate basis for the National Spatial Data Infrastructure (NSDI) of Greece. Within the model, an attempt is made to cover all Greek land administration related information, which are currently maintained by different organizations. This means that apart from the registrations of the Hellenic Cadastre (HC), other objects are also categorized and registered in the proposed model aiming at the creation of a multipurpose land administration system for Greece.

The different types of spatial units include areas with archeological interest, buildings and unfinished constructions, utilities (legal spaces), 2D and 3D parcels, mines, planning zones, Special Real Property Objects (SRPO) usually found in Greek islands (anogia, yposkafa) and marine parcels. What makes the development of this model unique is the support of a wide range of spatial units, each of them having different requirements, several of them having a 3D aspect. The country profile also includes the content of various code lists, which are an important aspect of standardization.

According to ISO 19152, 2012, LA_Level and therefore, the greek country profile specialization GR_Level is a collection of spatial units with a geometric or thematic coherence, an important concept for organizing the spatial units. For the proposed model, this structure allows for the flexible introduction of spatial data from different sources and accuracies, including utility networks, buildings and other 3D spatial units, such as mining claims, or construction works, etc.

The richness of different types of spatial units that can be found in Greece; the proposed MLAS which is part of authors' previous research and introduces 3D spatial types; as well as ISO 19152, LADM, which is used as basis for the country profile, emphasize the suitability of the Greek cases to be chosen as prototype basis.

3. LADM DEFINITION IN THE INTERLIS LANGUAGE

The implementation of LADM with INTERLIS follows the methodological steps as discussed in Germann et al (2015), drawing particular attention to 1. the explicit formulation of constraints, 2. the formulation of the code lists and enumeration values, as well as 3. an initial description of the third dimension, which, until now, was not included. It should be mentioned that for the description of the model in the INTERLIS language some first steps have already been made. To test the feasibility of the LADM - INTERLIS implementation, the Swiss Land Management foundation started to describe the LADM ISO 19152 standard with INTERLIS (Germann et al, 2015).

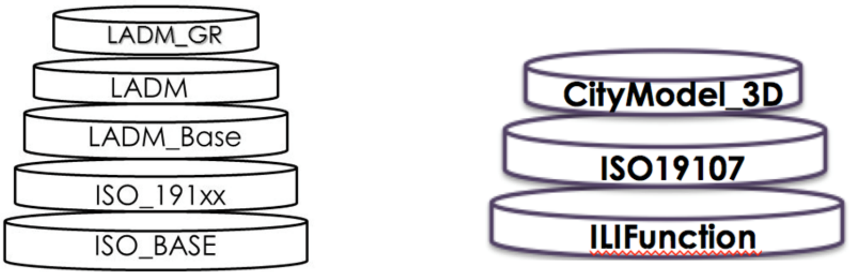

Figure 4a & b. (a) Stack of rhe INTERLIS models used to describe LADM country profiles (Kalogianni et al 2015): ISO_BASE (basic datatypes), ISO_191xx (geometric primitives and more), LADM_BASE (VersionedObject, Oid), LADM (classes from party, legal, spatial packages), and LADM_GR (Greek country profile), (b) Stack of the INTERLIS models used to describe CityModel_3D: IKIFunction , ISO_19107 and CityModel_3D (Swiss Land Management Group)

To start with, both the conceptual schema language INTERLIS and the LADM share the same MDA principles. The core work was completed covering the basic parts of the structure of the model.

Consequently, each one of the layers of the described models uses the model layer below as illustrated in Figure 4a above, starting from the base to the top. This step facilitates the progress as most of the data types needed for the description of the model were already defined. The same applies for the CityModel_3D described in INTERLIS, which also consists of layers of INTERLIS described models as illustrated in Figure 4b above.

3.1. Code lists an enumrations

Code lists are used to describe a more open and flexible enumeration. In LADM and also other ISO standards on which it is based, code lists are useful for expressing a list of potential values, aiming to allow the use of local, regional and/or national terminology (Lemmen et al, 2010). In contrast, by means of enumeration, all values admissible for this type are determined and cannot be extended without creating a new data type.

Some systems instead of maintaining the list of the values for code lists, they maintain the list of the reference of the values. This means that reference tables are created, which have as attributes an ID and a description for the code. However, those attributes have no meaning, as they are not part of the definition, and they are just being populated with a possible content. Therefore, it does not make sense to inherit such tables when created a model as they do not mean anything. However, when adding international codes to those tables as attributes (e.g. national, ISO, INSPIRE codes, etc) it prerequise that also other Organizations/ Institutions/ Authorities, would understand the content of them (Kalogianni E., 2016).
Therefore, for the GR_LADM.ili model described here, two options seem to be possible:

- In case that the list composes of fixed values, the values of the model will be defined as enumeration types.
- On the other hand, the values that can be extended (e.g. PartyRoleType); then a catalog table with referential integrity was used to express code lists.

An example is given below in order to present the extensible hierarchical enumerations that were created. At LADM.ili model the enumeration for the LA_StructureType of the LA_Level was formed as:

```
LA_StructureType = (point, line, polygon, other);
```

and the same enumeration was extended at GR_LADM.ili as:

```
GR_StructureType = (point,line, polygon,
other (text, topological, drawing, unstructured));
```

Note that LA_StructureType and GR_StructureType are two different data types. Moreover, the code list of PartyRoleType class is illustrated below as an example of the approach that was followed for the formulation of the code lists.

```
STRUCTURE GR_PartyRoleType
```

```
EXTENDS LADM.Party.LA_PartyRoleType =
   partyRoleTypeCode_ID: MANDATORY Oid;
   parentCode_ID: Oid referring to
      LADM.Party.LA_PartyRoleType.PartyRoleTypeCode_ID;
   description: CharacterString;
!! Possible code list values:
(lawyer,bank,notary,institution,tax_office,surveyor,court,metropolis
,court_of_appeal,legislative_authority,exproperiation_committee,urba
n_planning_authority,general_secretary_of_the_region,other);
END GR_PartyRoleType
```

3.2 Cardinality of the attributes

At the INTERLIS models the multiplicity of the attributes is described by using the list construct. By default in INTERLIS an attribute is optional. Examples for the definition of attributes' multiplicity are given bellow. Note that at the proposed class GR_2DParcel, the volume is "forced" to take no values.

```
CLASS GR_Party {ABSTRACT} EXTENDS LADM.Party.LA_Party =
   extLevelOfAdministrativeDivisionID:
   GR_LevelOfAdministrativeDivision;
   name (EXTENDED): CharacterString;
   role (EXTENDED): LIST {0..*} OF LA_PartyRoleType;
END GR_Party;
.....
CLASS GR_2DParcel EXTENDS GR_Parcel =
   area: LIST {1..*} OF GR_AreaValue;
   dimension: GR_DimensionType;
   volume: LIST {0} OF GR_VolumeValue;
   geometry: GM_MultiSurface;
   totalLength : Real;
END GR_2DParcel;
```

3.3 3D data types

The INTERLIS structures for geometric data types until now only describe 2D objects, while the description of data types for 3D objects is at early stages. The proposed structure for the GM_Solid is the following:

```
STRUCTURE GM_Solid EXTENDS GM_Object =
   geometry: LIST {1..*} OF Surface3DListValue;
END GM_Solid;
```

Where Surface3DListValue is the existing INTERLIS class:

```
STRUCTURE Surface3DListValue =
   value: MANDATORY Surface3D;
END Surface3DListValue;
```

and Surface3D has already been defined as:

```
Surface3D = SURFACE WITH (STRAIGHTS,ARCS) VERTEX Coord3D;
```

However, it is clear that for the description of 3D objects, only the description 3D data types is not enough. The definition of coordinate systems as well as constraints and functions that can validate the primitive and handle 3D operations are also needed.

3.4 Formulation of constraints

For the proposed model there are many types of constraints that can be found, i.e. referential integrity constraints, such as: primary key must be unique, end date of unfinished construction must be equal or smaller to start date of the (same) building (presented below), if there is a co-ownership in a parcel, the sum of the RRR share should always be 1, the boundary of 2D parcel must be closed, a 3D volume must be closed, if we refer to the marine parcel the surface relation type should only take the values below or mixed (as presented below) etc. Below some examples with constraints implemented in the INTERLIS language are presented.

```
STRUCTURE ExtArchive EXTENDS LADM_Base.External.ExtArchive =
   acceptance: DateTime;
   data: CharacterString;
   extraction: DateTime;
   recordation: DateTime;
   sID: MANDATORY Oid;
   submission: MANDATORY DateTime;
   MANDATORY CONSTRAINT
      submission>=acceptance;
END ExtArchive;
......
CLASS GR_MarineParcel EXTENDS GR_Level =
   activity: GR_MarineActivityType;
   layerType: GR_MarineLayerType;
   resourceType: GR_MarineResourceType;
   zone: GR_MarineLayerType;
   MANDATORY CONSTRAINT
      GR_SpatialUnit.surfaceRelation = "below" OR
      GR_SpatialUnit.surfaceRelation = "mixed";"
END GR_MarineParcel;
```

4. CONSTRAINTS IMPLEMENTATION

Apart from the associations and constraints that exist within a single domain (i.e. legal or physical) it is wise to define some "cross-model" associations and constraints referring to both the legal and the physical aspect. Such associations and constraints should exist both on the conceptual level of the two models and also be supported after the implementation in order to enhance efficiency and realiability of the model.

Thus, during the prototype development on cross – model constrained was defined that the buffer of the physical object should be inside the buffer of the legal object. It states that the buffer of the physical object should be inside the buffer of the legal space. However, this relation does not always have to be one per one (1 ..1), as it can be the case that multiple buildings are located into one parcel, or the other way around; two parcels can be under one tunnel; multiple interior spaces located inside a building, related to each other, form one (bigger) legal space; a party can own two or more parcels, etc. Then, the legal (or correspondingly the physical) space can be defined as a collection of 3D parcels and the association can also be many to many (*..*). Note that as the precision of the models is not always the same (and some times not always known) a buffer of 1 m both for the legal and physical reality is added. The legal aspect of a building (and 1m buffer) are presented together the corresponding physical reality (and 1 m buffer) are illustrated in Figure 5.

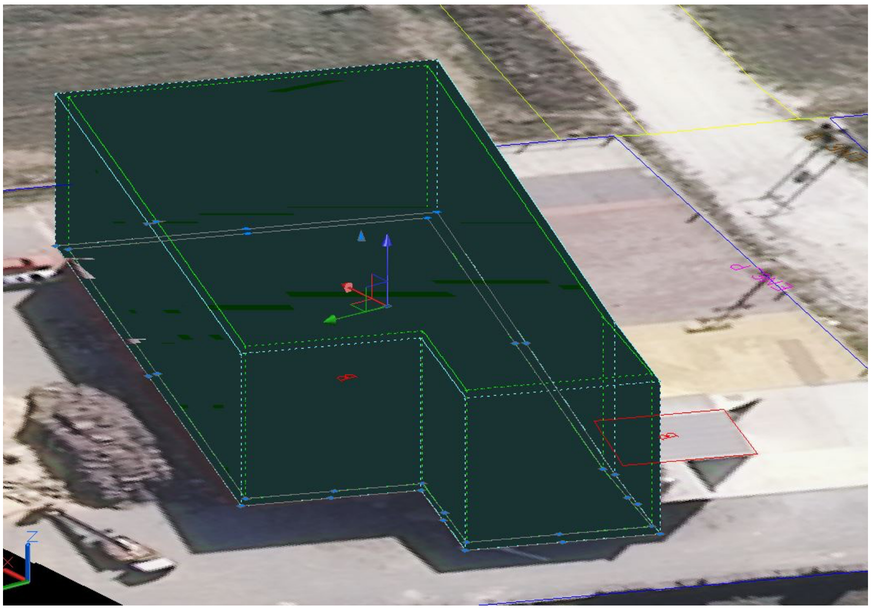

Figure 5. The cross legal – physical domain constraint in 3D (Kalogianni et al, 2016)

In general, constraints can be categorized in two core groups: hard and soft, as presented at figure 6. Hard constraints should always be true, otherwise the transaction should be canceled and an error should appear on the user. For instance, hard constraints are the following: the sum of the shares of an ownership interest should always be 1; end date after start date; etc.

On the other hand, soft constraints may not always be true as there is a number of constraints that should be true but in the real world there may be some exceptions. If they are true, then the transaction can be continued; if they are violated then it should be checked whether they belong to an exception list. This means that it is a binary decision on the constraint and its presence in the exception list. The exception list is a way to officially mark entities as allowed exception and deal with soft constraints. According to the constraints checked the exception list should be created and maintained by the corresponding authorities.

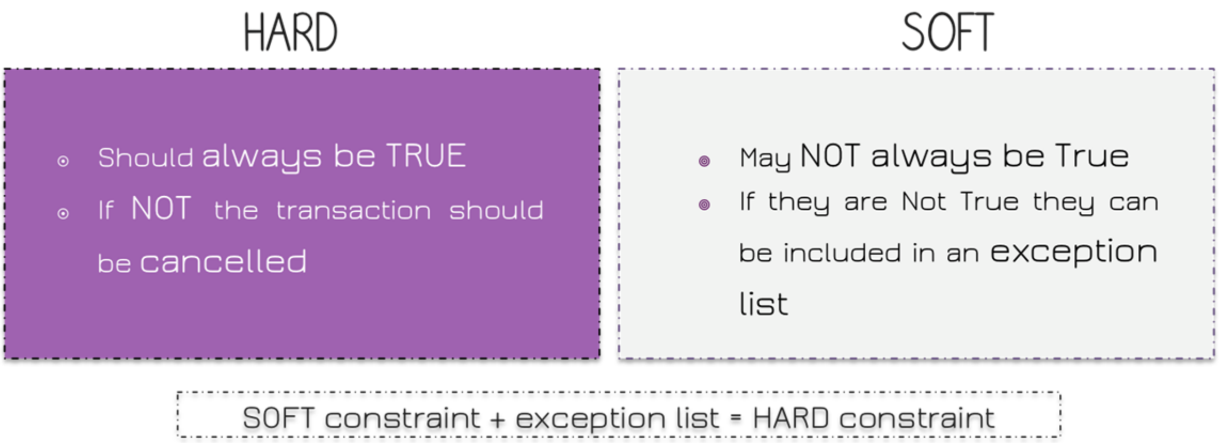
HARD
Should always be TRUE
If NOT the transaction should be cancelled
SOFT
May NOT always be True
If they are Not True they can be included in an exception list
SOFT constraint + exception list = HARD constraint

Figure 6. Hard and soft constraints during the development of the prototype

At the database level, soft constraints can be expressed with triggers for activities that should be conducted to make the different components more consistent. Finally, if a soft constraint is violated and also belongs to the exception list, then it is considered as a hard constraint.

The INTERLIS tools have been used in order to develop an implementation of the LADM-based model. As illustrated in figure 7, the 3D MLAS was graphically described with UML INTERLIS Editor only for visualization purposes.

As also mentioned in the previous section, INTERLIS also offers tools for quality check using INTERLIS compiler which checks the syntactical correctness of an INTERLIS data model. INTERLIS model (LADM_GR.ili) was imported in the compiler (ili2c) in order to check its correctness and then provide a data exchange format. When an error occurs, the tool mentions it and sometimes the compiler message is specific describing what the tool was expecting to found at the model. However, in some cases the message is not that specific, while usually the tool does not give a hint for the solution, which is time consuming in case that the user is not very familiar with the INTERLIS language. In particular, regarding the LADM_ GR model, many errors have been discovered with the use of the compiler. Thus, they were corrected at the INTERLIS model and imported again at the compiler.

After succeeding to quality check the stack of all the INTERLIS models INTERLIS checker was used to quality check INTERLIS XML data itself against the INTERLIS data models.
Having achieved to design a syntactically correct (in the INTERLIS terms) model and generate a transfer format for it, the next step is to create an empty database and import the model. Using the tool ili2pg, which translates the INTERLIS data model definitions to a PostGIS schema and loads the INTERLIS data into a PostgreSQL/ PostGIS database. Therefore, each single model of the afore mentioned was imported into ili2pg tool and the initial schema of the database was created; however a manual fine-tuning of the database was needed. Also at this tool the constraints are checked in order not to be violated.

As a database is the computerized logical representation of a conceptual model and consisting of a set of rules/ constraints and relationships the initial conceptual schema should be identical with the database schema developed. However, during this transition some limitations were noticed, as well as data loss occurred (eg a PK is generated for each table at the database without inheriting the given name at the corresponding INTERLIS class) while some constraints need to be manually added at the database (eg PK – FK constraint).

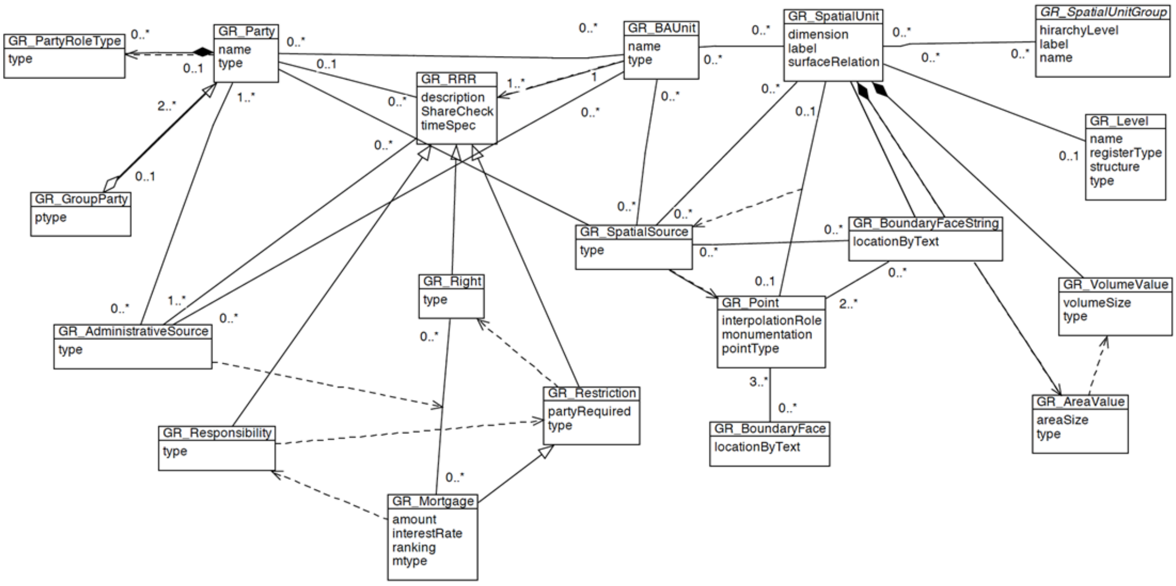

Figure 7. Proposed model graphically represented using UML INTERLIS Editor (Kalogianni E., 2016)

Below, a comparison between a class in INTERLIS and the same class as table at the database are presented.

INTERLIS:

```
CLASS GR_LegalSpaceUnfinishedConstruction (ABSTRACT) EXTENDS
GR_LegalSpaceBuildingUnit =
   unfinishedID: MANDATORY Oid;
   type (EXTENDED): GR_LegalSpacedUnfinishedConstructionType;
   startDate: MANDATORY DateTime;
   endExpected:DateTime;
   MANDATORY CONSTRAINT
    endExpected <= startDate
END GR_LegalSpaceUnfinishedConstruction;
```

Database:

```
CREATE TABLE gr_legalspaceunfinishedconstruction(
   unfinished_id integer NOT NULL,
   type integer,
   startdate character varying(20) NOT NULL,
   endexpected character varying(20),
   CONSTRAINT gr_legalspaceunfinishedconstruction_pkey PRIMARY KEY
   (t_id),CONSTRAINT start_end_dates CHECK (endexpected::text >=
   startdate::text));
INHERITS (gr_legalspacebuildingunit);
```

5. CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

The system’s development cycle, from conceptual model to implementation of working prototype has been illustrated with a proposed model for MLAS in Greece. A wide range of spatial units is supported in the model including legal spaces. Note that the process followed during the prototype development is cyclical and repetitive, providing feedback to the initial model during all the stages and improving it in terms of efficiency, reality and technical implementation. During the development, some minor omissions in the initial conceptual model were detected and corrected. Furthermore, the evaluation of the prototype with the database population brought about a series of constraints. Such needs increase when data is shared between different organizations and agencies.

Various compatibility problems arise between the INTERLIS tools and other tools used in this process. Regarding the use of the INTERLIS tools challenges faced during the implementation of the prototype include among others:

1. how to express the various LADM constraints (OCL and semi-formal in standard) formally in INTERLIS,
2. how to include extensible hierarchical code lists in the INTERLIS models (although enumeration types are included but not always appropriate; code lists are not natively available) achieving semantic meaningful content,
3. how to model 3D volumetric primitives in INTERLIS (not natively available),
4. how to link two models (legal - physical) using cross domain constraints, etc.

5.2 Future work

Regarding the INTERLIS part of the prototype, the compatibility problem between the INTERLIS tools and other systems should be faced. By introducing adequate general XML-tools for INTERLIS it is possible to render the concrete INTERLIS/XML-files available for an even wider range of application. Furthermore, it would be wise to provide more INTERLIS tools and the corresponding manuals in English language so it will be easier for non German speaking users to become familiar and use the language.

In respect of the 3D spatial data types in the INTERLIS tools further improvement of the existing 3D volume types is needed, which will lead to the explicit definition of a 3D primitive (i.e. GM_Solid). Additionally, completion of existing and definition of functions (e.g. ILIFunction.validateSolidGeometry(geometry) from CityModel_3D need to be completed), views, constraints and rules to be applied at the data types in the INTERLIS language is needed in order to check their validation. More complex constraints between different classes (tables) can be defined with a SQL function that will select all the objects that violate the constraint.

Furthermore, the development of the necessary mappings between the tools in order to recognize and check the proposed structures. As for the code lists, mappings for the proposed table with referential integrity should be created. For the top level code lists, the concept of the ”THINK” (i.e. top level ontology-parent of all the code lists) should be further examined.

At the same time, it is known how enumerations can be extended but missing how to refer them to previous enumeration types. This is also subject for further research.

What is more, more complex and extensive constraints must be defined within views (e.g. a view which connects a certain class with itself) and thus allow to compare any attribute combination with all other objects of the class. Such constraints can be expressed by creating functions or triggers at the database level. It is evident that the next version of INTERLIS will cover some of the afore mentioned proposals.

Next point of discussion could be the explicit quality checking of both 2D and 3D representations, which is closely related to the development of the mapping between the used systems. Finally, the selection of more coherent datasets which will cover a bigger area, while performing at the same time, complicated queries in order to investigate the integration of spatial and non-spatial data, and visualize them, will lead to more conclusions Morover, the temporal dimension is the need for information assurance at a national level as both current and historic versions of (legal and physical objects) should always be accessible.

Last but not least, some open questions for the future are mentioned: Is 3D topography pure reference and can it been used in such models? Given the cadastral parcel boundaries in 2D maps, the reference system, points with known coordinates and orientation, everyone can use them and recognize the described physical object. Then, is it possible that complex constructions described with survey data, BIM, point clouds, Digital Terrain Model (DTM), 3D virtual city models, etc will be used as entry data source for the description of legal space?

REFERENCES

Aien, A., Rajabifard, A., Kalantari, M. and Shojaei, D. (2015). Integrating legal and physical dimensions of urban environments. IIn: SPRS InternationalJournal of Geo-Information, 4.

Çagdas, V. (2013). An application domain extension to citygml for immovable property taxation: A turkish case study. International Journal of Applied Earth Observation and Geoinformation.

Chiang, R.H., Barron, T.M. and Storey, V.C. (1994). Reverse engineering of relational databases: Extraction of an eer model from a relational database. In: Data & Knowledge Engineering , 12 (2).

Cockcroft, S. (1997). A taxonomy of spatial data integrity constraints. In: GeoInformatica ,1.

Demuth, B. (2004). The Dresden Ocl Toolkit and its role in Information Systems development. In: Proceedings of the 13th International Conference on Information Systems Development (ISD '04), Vilnius, Lithuania.

Demuth, B. and Wilke, C. (2009). Model and object verification by using Dresden OCL. In: Proceedings of the Russian-German Workshop “Innovation Information Technologies: theory and practice”.

Dsilva, M., Speckmann, B., Westenberg, M. and van Hee, K. (2009). A feasibility study on CityGML for cadastral purposes. PhD thesis, Master’s thesis, Technische Universiteit Eindhoven, Eindhoven, the Netherlands.

Duboisset, M, Pinet, F, Kang, M-A and Schneider, M (2005). Precise modelling and verification of topological integrity constraints in spatial databases: from an expressive power study to code generation principles. In: Delcambre L, Kop C, Mayr HC, Mylopoulos J, Pastor O (eds) Conceptual Modelling – ER 2005. Springer, Berlin Heidelberg, pp 465-482.

Egea M., Dania C. and Clavel M. (2010). MySQL4OCL: A Stored Procedure-Based MySQL Code Generator for OCL. In: Proceedings of the Workshop on OCL and Textual Modelling.

ePlan. (2010). ePlan Model version 1.0 ePlan. In: Intergovernmental Committee on Surveying and Mapping (ICSM).

Germann M., Kaufmann J., Steudler D., Lemmen C., van Oosterom. P. and de Zeeuw, K. (2015). The LADM based on INTERLIS. In: FIG Working Week 2015. Sofia, Bulgaria.

Gózdz, K., Pachelski, W., van Oosterom, P. and Coors, V. (2011). The possibilities of using citygml for 3D representation of buildings in the cadastre. In: 4 International Workshop on 3D Cadastres, 9-11 November 2014, Dubai, United Arab Emirates. International Federation of Surveyors (FIG).

Group, O. M. et al (2007). Meta object facility (mof) 2 .0 /xmi mapping.

Hespanha, J., Paixao, S, Ghawana, T., Zevenbergen, J. and Carneiro, A. (2015). “Application of Land Administration Domain Model to Recognition of Indigenous Community Rights in the Philippines: Laws Examined with Spatial Dimensions (7579)”, in FIG Working Week 2015 - From the Wisdom of the Ages to the Challenges of the Modern World.

ISO (2008). ISO 19107:2003. In Geographic information – Spatial schema. ISO.

ISO (2012). ISO 19152:2012. International Standard, Geographic Information - Land Administration Domain Model (LADM).

Kalogianni, E. (2016). Master thesis: Design of a 3D multipurpose land administrative system for Greece in the context of LADM. Master’s thesis, National Technical University of Athens.

Kalogianni, E. (2015). Master thesis: Design of a 3D Multipurpose Land Administrative System for greece in the context of ladm. Master’s thesis, National Technical University of Athens.

Kalogianni, E., Dimopoulou E. and van Oosterom P. (2015). A 3D LADM prototype implementation in INTERLIS. In: 3D GeoInfo Conference 2015, Kuala Lumpur, Malaysia.

Kaufmann, J. and Steudler, D. (1998). Cadastre 2014 - A vision for a future cadastral system. F.I.G., Commission 7, Working Group 7.1. In: XXI International Congress, Brighton.

KOGIS (2006). INTERLIS 2.3 Reference Manual. Coordination, Geo-Information and Services (COGIS), a division of the Swiss Federal Office of Topography.

Lemmen, C.H.J. (2012). A Domain Model for Land Administration. PhD thesis, ITC.

Louwsma J., Zlatanova S., van Lammeren R. and van Oosterom P. (2006). Specifying and implementing constraints in GIS – with examples from a Geo-Virtual Reality System. In: GeoInformatica 10 (4).

Mäs, S., Wang, F. and Reinhardt, W. (2005). Using ontologies for integrity constraint definition. In:Proceedings of the 4th International Symposium on Spatial Data Quality'05, Beijing, China.

Olive, A. (2006). Conceptual Modelling of Information Systems. In: Springer
van Oosterom, P. (2006). Chapter 7 Constraints in Spatial Data Models, in a Dynamic Context. In "Dynamic and Mobile GIS: Investigating Changes in Space and Time" (J. Drummond, R. Billen, E. Joao, D. Forrest, eds.).

Oriol, X. Teniente, E. (2014). Incremental Checking of OCL Constraints through SQL Queries. In: Conceptual Modeling: 34th International Conference, ER 2015, Stockholm, Sweden.

Rönsdorff, C., Wilson, D. and Stoter, J. (2011). Integration of Land Administration Domain Model with CITYGML for 3D Cadastre. In: Proceedings 4th International Workshop on 3D Cadastres, 9-11 November 2014, Dubai, United Arab Emirates . International Federation of Surveyors (FIG).

van Bennekom-Minnema, J. (2008). The land administration domain model 'survey package'and model driven architecture. Unpublished MSc Thesis, Joint Programme in Geographical Information Management and Applications (GIMA).
Werder, S., (2009). Formalization of spatial constraints, in Proceedings of the 12th AGILE Conference on GIScience. Hannover, Germany: AGILE 2 – 5 June.

Xu, D., van Oosterom, P. and Zlatanova, S. (2016). A methodology for modelling of 3D spatial constraints. Presented at the Joint International Geoinformation Conference, Kuala Lumpur, 2015 and chapter in book "Advances in 3D Information", Springer (to be published).

BIOGRAPHICAL NOTES

Eftychia Kalogianni received a degree in Rural and Surveying Engineering in 2012. She has obtained an MSc in Geoinformatics in National Technical University of Athens in 2015 and an MSc in Geomtics from Deft University of Technology in 2016. Since 2015, she has been working at a consuling engineering company involved with various projects carried out by european joint ventures regarding cadastre, surveying and GIS, transport engineering and environment.

Efi Dimopoulou is Associate Professor at the School of Rural and Surveying Engineering, NTUA, in the fields of Cadastre, Spatial Information Management, Land Policy, 3D Cadastres and Cadastral Modelling. She is the Programme Director of the NTUA Inter-Departmental Postgraduate Course «Environment and Development» and President of the Hellenic Society for Geographical Information Systems (HellasGIs).

Wilko Quak has an MSc in Computer Sscience from Utrecht University, The Netherlands (UU). He worked for several years (1993-2001) as a researcher at the Dutch research center for mathematics and computer science (CWI) and University of Amsterdam (UvA) on Spatial DBMS performance. Since 2001 he is researcher at the Section GIS Technology, OTB, Delft University of Technology. At Delft University his research focus is moving towards spatial data modelling, data interoperability and standardization. Since 2007 he is working part-time for Geonovum (the National Spatial Data Infrastructure (NSDI) executive committee in the Netherlands).

Peter van Oosterom obtained an MSc in Technical Computer Science in 1985 from Delft University of Technology, the Netherlands. In 1990 he received a PhD from Leiden University. From 1985 until 1995 he worked at the TNO-FEL laboratory in The Hague. From 1995 until 2000 he was senior information manager at the Dutch Cadastre, where he was involved in the renewal of the Cadastral (Geographic) database. Since 2000, he is professor at the Delft University of Technology, and head of the 'GIS Technology' Section, Department OTB, Faculty of Architecture and the Built Environment, Delft University of Technology, the Netherlands. He is the current chair of the FIG Working Group on '3D Cadastres'.

CONTACTS

Eftychia Kalogianni
National Technical University of Athens
MSc Geoinformatics NTUA
MSc Geomatics, Delft University of Technology
10 Monis Petraki str.
115 21 Athens
GREECE
Tel.: +306944325903
E-mail: efkaloyan@gmail.com

Efi Dimopoulou
National Technical University of Athens
School of Rural & Surveying Engineering
9, Iroon Polytechneiou
15780 Zografou
GREECE
Phone: +302107722679
Mob.: +306937424666
Fax: +302107722677
E-mail: efi@survey.ntua.gr

C.W. Quak
Faculty of Architecture and the Built Environment
Department OTB Research for the Built Environment
GIS Technology Section
Delft University of Technology
Julianalaan 134
2628 BL Delft
THE NETHERLANDS
Phone: +31152783756
E-mail: c.w.quak@tudelft.nl

Peter van Oosterom
Faculty of Architecture and the Built Environment
Department OTB Research for the Built Environment
GIS Technology Section
Delft University of Technology
Julianalaan 134
2628 BL Delft
THE NETHERLANDS
Phone: +31152786950
E-mail: P.J.M.vanOosterom@tudelft.nl