3D Data Management - Overview Report

Karel JANEČKA, Czech Republic and Sudarshan KARKI, Australia

Key words: 3D Data Management, 3D Cadastre, 3D Database, 3D Representation, 3D Analysis

SUMMARY

The paper has explored 3D data management from multiple perspective. The focus of the data management issue in this paper has not been restricted to 3D Cadastre, but rather to a broader 3D GIS to ensure that all capabilities and issues that exist in different related fields will assist and affect in the data management of 3D cadastral data. In functional requirements for 3D cadastral data management, the categorisation of 3D parcels at an increasing level of complexity is discussed. This lead to a discussion on options for storing 3D cadastral data in an existing 2D cadastral database that traditionally exists in current jurisdictions. The issues related to adding the time dimension in a 4D cadastre from a database point of view was discussed. A discussion of 3D geometric models based on current research on standards, solid geometry and LADM schema, which in turn led to 3D topological models. The LADM provides a data model that recognises and describes the relationships of a 3D spatial unit to other levels of encodings. BIMS are a good source of 3D cadastral data and has already been used by many jurisdictions. The link between the various geometrical and semantic aspects of BIM vs other data sources can cause differences and issues when data are to be integrated. The current standards such as ISO LADM, GML, CityGML etc. and their inter-relationship were then discussed. In current DBMS 3D capabilities, current software and methods of storing 3D data were discussed which led to a discussion on recent developments of spatial databases and the physical capacity of existing hardware to cope with the large volume of 3D data. The analysis of a gap between what is available and what is needed was based on 3D geometry and topology, validation, standards and ontology, data and hardware, 3D data use and transfer and implementation of a 3D LADM prototype. 3D data management capability and technology exist, however these have not been transferrable to 3D cadastre. The problem is, established cadastre are traditionally 2D and the nature of the cadastral data does not easily extend itself to 3D modelling. While 3D GIS data may be easy to extrude to create a 3D visualisation, because 3D cadastre deals with absolute ownership of 3D spaces it becomes much more complex to convert a 2D database to a 3D operational data structure. The extrusion of 2D to 3D might still be a feasible solution for a cadastre if the purpose is just visualisation, however, if the purpose is to define ownership of defined space, information about the adjoining 3D spaces, checks to determine encroachment or slivers among the spaces, then a simple extrude does not fulfil the requirements.

3D Data Management - Overview Report

Karel JANEČKA, Czech Republic and Sudarshan KARKI, Australia

1. INTRODUCTION

It is obvious, that the amount and use of three-dimensional data has rapidly increased in the last few year. Boss and Streilein (2014) observed four major technology and business drivers for 3D:

1. There are massive new sensor hardware capabilities, such as automated data capture and model creation on the sensor side, LIDAR with masses of point clouds and automated photogrammetric workflows and processes.
2. 3D visualisation has now come into mainstream, but 3D analysis not. But there is as yet no mass market with consumer-focused systems.
3. Managing 3D data in enterprise workflows with improved performance and scalability of existing workflows and bridging the gap between point cloud surveys, GIS, CAD, BIM. Traditional file handling moves to database management.
4. There is a necessity for 3D data, where 2D data is not sufficient to describe our world and the consumer expectation demands three dimensions, as we all live and act in a three dimensional environment.

For cadastral organizations, who traditionally describe their cadastral data in two dimensions and hold their information in 2D (often graphical) files, concepts for entering the third dimensions are not yet available, mainly due to the facts that (Boss and Streilein, 2014):

- 3D modelling is much more heterogeneous and complex compared to 2D modelling,
- converting 2D data to 3D data on an operational level, with not just adding a Z-Coordinate onto each planimetric pair of coordinates, is quite cumbersome and there is no 'best' solution obvious, as the existing datasets are usually quite specific,
- one has to migrate from simple data structures to complex data structures,
- newly on has to deal with the economic and sustainability issues of handling and storing high data volumes compared to (relatively) low data volumes in the current years,
- and last but not least, user-friendly tools for 3D analysis are still missing.

Stoter et al (2016) mention that as techniques for 3D mapping are maturing and at the same the need for 3D data is increasing, that this has pushed national (and regional) mapping agencies (NMAs) to consider extending their traditional task of providing topographic data into the third dimension. They show that some NMAs are still in the initial (experimental) phase of 3D mapping, while others have already built solid databases to maintain 2.5D and 3D topographic data covering their whole country.

Several investigations have proved that only some additional information is needed to build up a 3D-spatial data set out of the existing 2D-spatial cadastral data and to keep the information up to date. What is needed are the number of floors, ridge direction, and the building height. Most of this information already exists in the planning process; additional data is collected during the cadastral survey. With this approach and the integration the aforementioned information a future 3D-cadaster could be implemented sustainable (Seifert et al, 2016). Gue et al (2014) describe the practical application of 3D data management and development of 3D Cadastre in Shenzhen. Their solution is seamlessly integrated with 2D cadastral system and in close combination with the cadastral business framework. They conclude, that although there are some advances, the 3D cadastral administrative still faces many difficulties and challenges, such as the supported laws and regulations. They define an improvement of the practicability and convenience as the strategic research direction of 3D Cadastre.

For the establishment of a 3D cadastre there are several challenges and key issues to deal with. There are still open questions to overcome such as (Boss and Streilein, 2014):

- Is existing GIS software capable to handle the requirements of managing 3D data?
- What should be the main important developments of software manufactures in the near future?
- Where is still need of (scientific) research?
- There exists until today no mass market for 3D data management and 3D data analysis. Potential users don't know that they could solve their problems using 3D GIS and a 3D cadastre. How could we reach these future clients to stimulate the demand and thus indirectly accelerate the development of user friendly software?
- Identification of country specific similarities and differences (in the regions or in the world).

Several questions have to be posed and answered on a generic or operational level (Boss and Streilein, 2014):

- What about data acquisition?
- Is crowd sourcing usable for 3D cadastre?
- What about automatic processes?
- What about software?
- What about data standards?
- What about system architectures?
- What are the types of 3D cadastral objects that need to be registered?
- What about the segmentation of objects? What about 3D data analysis?
- What about data presentation/visualisation?
- What about robust data management?
- What about temporal aspects?

2. FUNCTIONAL NEEDS (REQUIREMENTS) FOR 3D CADASTRE DATA MANAGEMENT (INCLUDING 4D TIME)

In our contemporary social context, the development of land use has subdivided land parcels into three-dimensional (3D) spaces according to certain property rights, especially in metropolitan areas with dense population. This results in 3D parcels (ISO, 2012) above or below the land surface. In such circumstances, the local government needs to construct and manage 3D cadastral objects to be able to manage the development of real urban 3D spaces appropriately (Ying et al, 2015).

Constructing of 3D data models and their topological relationship are two important parts of 3D cadastre (Ying et al, 2011). As stated in Ding et al (2016), one can construct a mass of complex 3D buildings models by LIDAR techniques or oblique photography technique. However, these approaches do not take consideration of current 2D parcels and provide much meaningless information such as texture for 3D cadastre. So 3D model construction approaches based on the 2D parcels are imperative for 3D cadastre.

2.1 Types of 3D parcels

An initial categorization of 3D Parcels was given in Thompson et al (2015) and forms the starting point for the further investigations into suitable corresponding database representations exchange format, and data capture encodings. The following categories were introduced, now listed in the order of growing complexity:

1. 2D spatial unit (actually prism of 3D space): defined by a 2 dimensional shape.
2. Building format spatial unit: defined by the extents of an existing or planned structure.
3. Semi-open spatial unit: defined by 2D shape with upper or lower surface1.
4. Polygonal slice spatial unit: defined by 2D shape with upper and lower surface.
5. Single-valued stepped spatial unit: defined by only horizontal and vertical boundaries (among others the facestring from 2D space) and single valued[1].
6. Multi-valued stepped spatial unit: as above but now multi valued.
7. General 3D spatial unit: defined also by other boundaries than horizontal and vertical.

The category of General 3D spatial units can be further refined: 2-manifold required or not, partly open/completely closed volume, planar/curved boundaries, multi-valued single/multi-volume, etc. (Thompson and van Oosterom 2012).

The problem of mixing 2D land parcel definitions with the range of 3D parcels in a corporate database and exchange format encodings is one of the most basic issues to be solved in creating a modern approach to Cadastral modelling. Various approaches have been suggested in Thompson et al (2015):

1. Keep the 3D parcels in a separate database from the rest of the 2D database.
2. Simply store footprints only, with no (reference to) 3D definitions at all.
3. Keep a representation all parcels in the main database in 2D form only (with the 3D parcels represented by "footprints"). The full 3D definition of the 3D spatial units are kept in another form (in CAD or pdf format) and to be obtained from a document archive.
4. Store all parcels in the same database, with 3D parcels being approximated by a "slice" (a polygon with a horizontal top and bottom surfaces) which contains the parcel (but may be a loose fit).
5. Convert all parcels to 3D form and store in a single database.
6. Integrate 2D parcels and 3D parcels in the same database, make sure they fit well together.

Beyond simple mapping applications, a basic requirement to be satisfied by a corporate database is to answer the query "*given a spatial unit, what are its adjoiners?*" Of the above methods only methods 5 and 6 can satisfy this query directly. The others either cannot

[1] The volume is called single valued if there is no pair of points within the spatial unit with the same (x,y) coordinates which have a point from outside the spatial unit between them.

respond at all, or will give incorrect answers (Thompson et al, 2016). Thompson (2015) published the finding that levels of encoding can co-exist within the same cadastral database and that 2D and 3D parcels can be mixed.

2.2 4D time

The principle of an efficient management of object life cycle was proposed e.g. in Seifert et al (2016), where the used data model requires for each object a unique identifier together with a designated time stamp for creation and deletion of an object. However, once an object has to be deleted during an updating process the object will not be physically removed from the data base. Only the life cycle of the thematic relevance has ended, but not the existence of the object as an instance. A "deleted" object is then considered the as a historical information which can be easily distinguished from the actual information. Sometimes there are changes of an object which do not require the deletion of the object (e.g. only a name of the person changes). In that case also the different versions of an object can be stored. Since every object carries life cycle information the storage of historical objects and versions of objects is not limited to any specific object type.

2.3 3D geometric models

Practically most of the work on geometry model has been completed by the Open Geospatial Consortium Inc. (OGC, formerly the Open GIS Consortium) (Lee and Zlatanova, 2008). ISO has also independently from OGC developed ISO/TC 211 19107:2003, Geographic information – Spatial Schema (Hering, 2001).

The OGC Implementation Standard for Geographic information – Simple feature access – Part 1: Common architecture (OGC, 2011) describes the common architecture for simple feature geometry. The simple feature geometry object model is Distributed Computing Platform neutral and uses UML notation. The base Geometry class has subclasses for Point, Curve, Surface and GeometryCollection. Each geometric object is associated with a Spatial Reference System, which describes the coordinate space in which the geometric object is defined. This part of OGC Simple feature access implements a profile of the spatial schema described in ISO 19107:2003, Geographic information – Spatial schema..

The OGC Implementation Standard for Geographic information – Simple feature access – Part 2: SQL option (OGC, 2010) defines a standard Structured Query Language (SQL) scheme that supports storage, retrieval, query and update of feature collections via the SQL Call-Level Interface (SQL/CLI). A feature has both spatial and non-spatial attributes. Spatial attributes are geometry valued, and simple features are based on two-or-fewer dimensional geometric (point, curve and surface) entities in 2 or 3 spatial dimensions with linear or planar interpolation between vertices.

Kazar et al (2008) and Verbree and Si (2008) observe that the ISO 19107 solids are not sufficient for 3D cadastral applications: the ISO19107 solid is a simple solid whose shell is not allowed to touch (they have to be 2-manifold).

Proper 3D geometries are required for 3D cadastres. Surveying data can be investigated by the surveyors or the engineers, thus the creation and submission of 3D volumetric objects are the key phases in a 3D cadastre system. However, what are acceptable (valid) 3D cadastral object representations and how to create their 3D geometries (even the non-2-manifold geometries) are still challenges (Van Oosterom 2013). The non-manifold 3D representations (self-touching in edge or node; see 1) are not well supported by current GIS, CAD, and DBMS software or by generic ISO standards such as ISO 19107 (van Oosterom 2013).

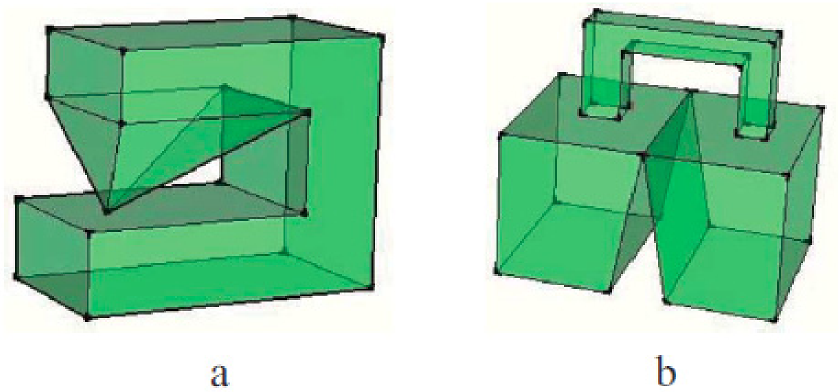

Figure 1. Solids with non-manifold conditions: (a) point non-manifold condition; and (b) edge non-manifold condition (Ying et al, 2015)

Kazar et al (2008) and Thompson and Van Oosterom (2012) give the definition of a 3D parcel for 3D cadastre purposes. The main rule is that the volumetric object is internally connected, which means that a shell can self-touch, as long as the interior of the solid stays connected. Ying et al (2015) follow this definition and state that a valid volumetric object is a 3D primitive that can be represented by one close polyhedron, refined by a set of connected faces. The volumetric object satisfies the following characteristics: closeness, interior connection, face-construction and proper orientation. Evidently, the volumetric object here can have through-hole/ring or cavity that allows its boundary faces to touch each other, which is not a 3-manifold in some cases.

Figure 2 shows a simplified database storage scheme proposed by Thompson et al, (2016) able to represent the various types of spatial units. Compared to ISO 19152, the classes LA_SpatialUnit and LA_BoundaryFaceString have been combined into a single class (LA_SpatialUnit) as there is in this context a 1-to-1 relationship between the two classes. This is conformant with ISO 19152. There are two reasons why a polyhedron attribute of type GM_Solid for 3D spatial units is not appropriate: 1. in most cases there is overlap between the vertical faces of polyhedron and the LA_BoundaryFaceString defined by the footprint (redundant and possible cause of inconsistency), and 2. the GM_Solid can only represent fully bound spaces. Therefore, this is not a suitable solution and the association with LA_BoundaryFace is used instead.

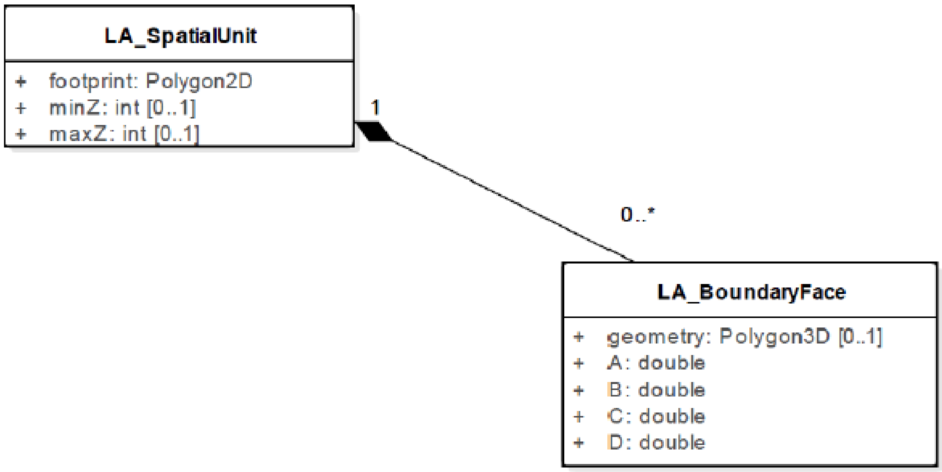

Figure 2. Simplified schema for database storage (Thompson et al, 2016)

There is no sharing of LA_BoundaryFace's among different LA_SpatialUnit's and the association between LA_SpatialUnit and LA_BoundaryFace is also not signed (indication + or – orientation of face when used in a 3D LA_SpatialUnit). This is possible in ISO 19152 and also fits quite well in the proposed style of LandXML encoding. In a DBMS that allows in-row storage of simple geometries, this form is highly efficient. For example in PostgreSQL/PostGIS or Oracle Spatial, simple 2D spatial units (such as four sided city blocks) will be stored in-row, permitting very fast retrieval. In addition, access can be in one of three forms: 1: as a 2D footprint (this could be compared to LoD0 in City Models); 2: as a "Prism" (footprint with top and/or bottom, this could be compared to LoD1 in City Models); 3: as a complete 3D geometry (the higher LoD's in City Models, including indoor, as one building may contain multiple spatial units) (Thompson et al, 2016).

Thompson et al (2016) further elaborate that the down-side of this mode is that there is duplication of the definition of boundaries that separate spatial units (one copy for each spatial unit involved), leading to the potential for incompatible definitions of the same boundary. The broad approach in terms of a storage scheme is that a more-or-less conventional 2D complete, non-overlapping topological coverage of the region of interest would be generated (sharing 2D boundaries), while 3D surfaces would be shared by and would separate spatial units that are adjacent in 3D, but overlapping in 2D. A secondary advantage of this approach is that it effectively supports liminal parcels as defined in the LADM (ISO, 2012).
Another issue is that if a footprint is stored as a polygon, most DBMSs do not permit any attributes to be recorded on the individual lines - such as the nature of the line. This is an area needing consideration and in principle the LADM supports management attributes on the boundary level: both for lines (LA_BoundaryFaceString) and faces (LA_BoundaryFace) (Thompson et al, 2016).

2.4 3D topologic models

Topology is defined as the identification of spatial relationships between adjacent or neighbouring objects (Ellul, 2007*)*. To model 3D topology, a number of 3D topological frameworks have been introduced. As Zulkifli et al (2015a) mention, these can be distinguished into two types of frameworks:

1. classification of topological relationships between two objects (e.g. Egenhofer, 1995; Billen et al, 2002) and
2. topological structures representing the structural relationship between many primitives and objects.

In the context of the second type of framework, several 3D topological models and approaches have been developed to construct a topologically correct datasets, e.g. (Penninga and van Oosterom, 2008; Ledoux and Meijers, 2009; Bormann and Rank, 2009; Ghawana and Zlatanova, 2010; Boguslawski et al, 2011; Brugman et al, 2011).

2.4.1 Considering LADM standard

However, these previously mentioned topological models have not discussed on LADM standard (Zulkifli et al, 2015a). A comprehensive land administration model is essential to build the cadastral management system. The LADM (Land Administration Domain Model) provides a conceptual description for a land administration system, including a 3D topology spatial profile (Thompson and Van Oosterom 2011).

The LADM provides conceptual descriptions for land administration, including 3D topology. The LADM also allows for organizing land related data in a standardized and interoperable way to support different types of spatial data. According to the requirements of LADM, topological information alone is not sufficient to describe a 3D spatial unit. Geometrical information must also be associated with each topological primitive; either direct geometries, or indirect (via related topological primitives with geometries). For 3D topology model in LADM as described in Spatial profiles of Annex E7 (ISO, 2012), there are no overlapping volumes (3D_SpatialUnit). However, volumes may be open at the bottom or at the top, corresponding to non-bounded 3D_SpatialUnits (in this case, the size of the volume cannot be computed). Note that in 3D_Level, the attribute structure is fixed to '3D', and there still is an optional referencePoint, which should be provided via 3D GM_Point. There is a set of constraints defining a valid topological structure for a 3D volume partition. In case of the 3D topology representation, a 3D boundary has plus/minus orientation information included in the association to a 3D spatial unit (see figure 3). All topological boundary faces are used once in plus and also exactly once in minus direction. Unless the boundary face is on the edge of the domain, then either the plus or the minus direction is used once (and the other zero times). The boundary faces do not self-intersect and do meet other boundary faces at their boundaries. All 3D_BoundaryFaces have outward orientation (normal vector points to the outside). All the 3D_BoundaryFaces together form at least one outer shell and zero or more inner shells. In principle, the shells are closed, with the exception that they may open (unbound) to the top (sky) and bottom (earth) direction (Zulkifli et al, 2015a).

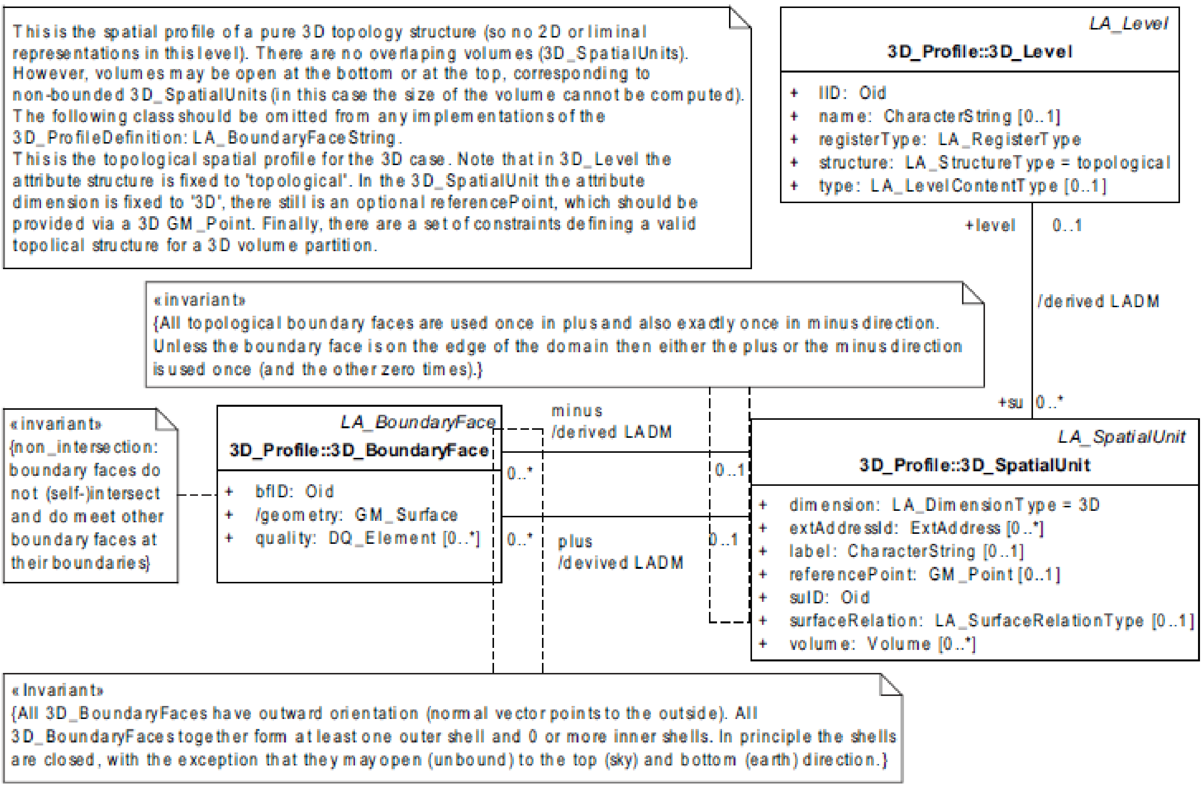

Figure 3. 3D topology based on LADM (ISO, 2012)

Zulkifli et al (2015a) review 3D topology within LADM. They review characteristics of the different 3D topological models in order to choose the most suitable model for certain applications. The characteristics of the different 3D topological models are based on several main aspects (e.g. space or plane partition, used primitives, constructive rules, orientation and explicit or implicit relationships). The most suitable 3D topological model depends on the type of application it is used for. They conclude, that there is no single 3D topology model best suitable for all types of applications. Therefore, it is very important do define the requirements of the 3D topology model. They further conclude, that based on the reviews of the 3D topological models, a very suitable 3D topology model is the approach based on a Tetrahedral Network (TEN), proposed by Penninga and Van Oosterom (2008).

Ying et al (2015) present an effective straightforward approach to identifying and constructing the valid volumetric cadastral object from the given faces, and build the topological relationships among 3D cadastral objects on-the-fly, based on input consisting of loose boundary 3D faces made by surveyors. These 3D faces as the cadastral boundaries with official identifications are stored in a database. The method does not change the faces themselves and faces in a given input are independently specified. Various volumetric objects, including non-manifold 3D cadastral objects (legal spaces), can be constructed correctly. They also aimed to develop a more direct method of the solid validation process, describing the steps below:

1. To build valid solids at the beginning of object generation to satisfy the validation requirements.

2. If a valid solid is built and the sets of solids directly there is no need to validate its existence afterwards.

They propose a data model oriented towards the application and storage of a 3D cadastral system. Especially, they extend the geometric-topological model in LADM, which is based on ISO 19107, and redesign the model to support non-manifold 3D objects to represent realistic 3D cadastral objects. They propose a method for creation of both 3D volumetric objects – 3D solids and non-manifold solids (shapes with self-touching or hole) along with topological relationships that are already valid. This is important to model some realistic cadastral objects. Also the 3D volumetric objects in relation to the outer complementary space (named by Maximal Minimal Solid) can be generated. The presented approach ensures volumetric objects (polyhedral shapes) that satisfy the valid solid characteristics: face-based construction, closeness and uniqueness. Against the mainstream methods, that require one to assume that the shapes (solids) already exist in the 3D object and then test to see if this existence assumption holds, in the proposed method this assumption step is no longer required as a necessary research process. The input faces themselves are stable and they are independently specified. This direct 3D volume construction conforms to normal sequential data flow and business logic to provide valid 3D volumetric objects for 3D cadastral systems without the need for a post production validity check. The algorithm is capable of supporting various 3D shapes and non-manifold volumetric objects with holes or caves, and causes no problems with regard to the topological consistency. Real 3D volumetric objects are constructed first with the input faces, storing the references in the 3D topological model (see fig. 4). A valid volume is made up of and closed by at least four faces with their normal directions. Class Plane is designed to emphasize the face's normal direction, which means that every face used in the body is only a half-plane face. A 3D volume is a 3D primitive to describe the volumetric object and is basically incident to faces, the lower dimensional 2D geometric primitive. The volumetric model is defined as a seamless 3D space with interior orientation, and commonly its shells which, closed and made up of the faces, together completely separate the interior and exterior of the volume; volumes cannot intersect and penetrate mutually. An important condition of Face is that its normal direction points outward or inward to the volume, which is essential for volume construction. The face's normal direction determines the interior orientation of the 3D volume, and Class Face is an oriented facet or patch with one outer loop, and zero or more inner loops. In general, the term face denotes a simple flat face that is used to define a part of the boundary.

Ding et al (2016) propose a modelling approach for the 3D cadastral object based on extrusion. The approach does not allow overlapping among footprints which are used to construct one or more 3D objects. Based on this approach, one can extract 2D topological features from 2D footprints. Then 2D topological features and height values are used to present topological features. Using 2D feature to present 3D feature can save storage space. They used this approach in a case study of Pozi Street in Taizhou and conclude, that there is still need a lot of practice to verify its availability for 3D cadastre.

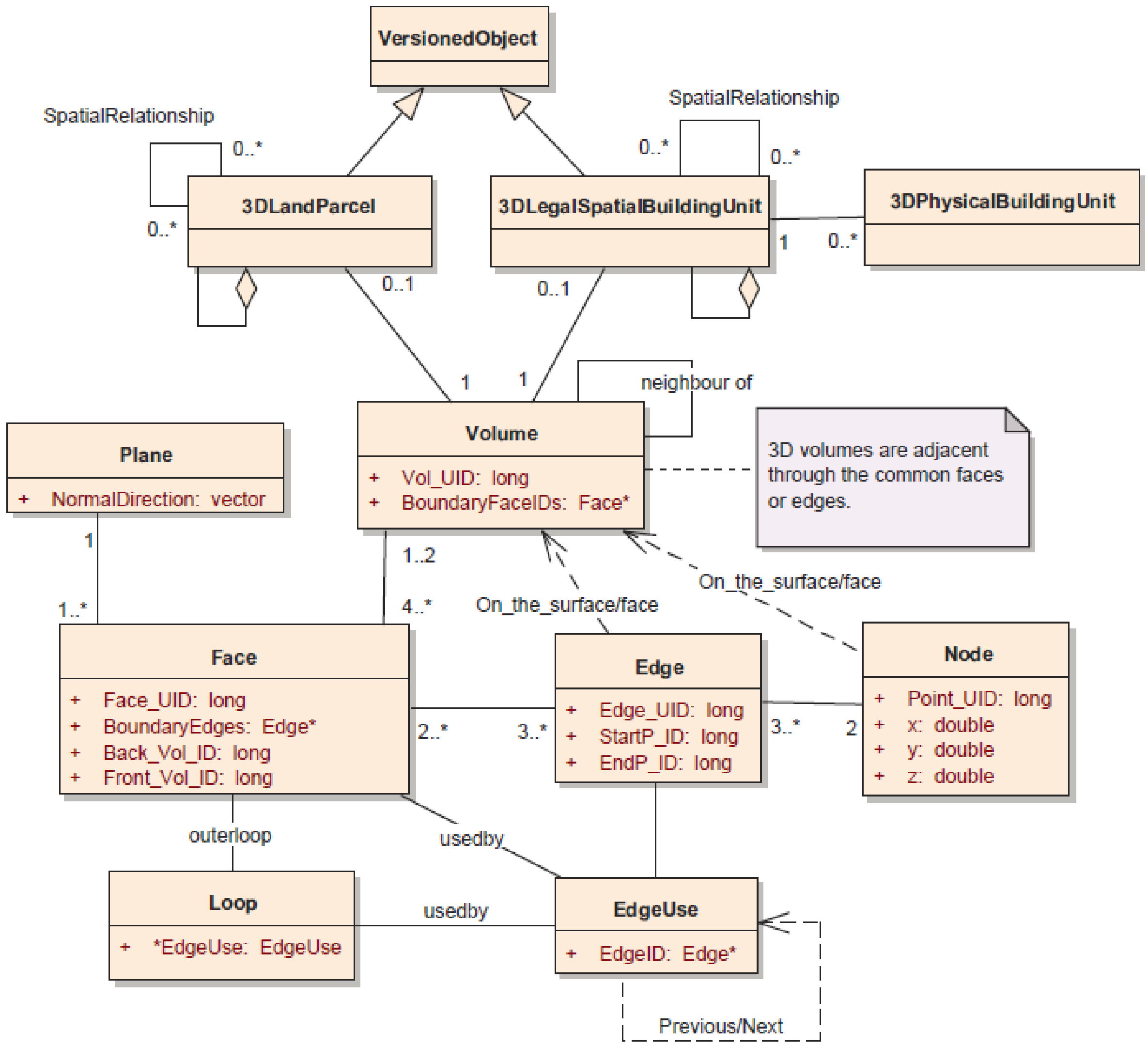

Figure 4. Data model in the prototype system (Ying et al, 2015)

2.5 (SDI-) link to 3D topography/BIM

There is a need for 3D topography data (in various level of detail), currently the cities are producing the city models with buildings in several LODs (according to the CityGML). Such data could be then potentially reused for 3D cadastre purposes.

For example, Building Information Models (BIM) are used to update the cadastre in Costa Rica (Van Oosterom et al, 2014). Behnam et al (2016) present usage of BIM as a feasible approach for managing land and property information in high-rise administration. They propose an extension to the BIM standard to show the potential capability of using BIM for modeling 3D ownership rights.

For any developments that require spatial data, often the fusion of diverse spatial datasets is unavoidable. For instance in developing a 3D cadastral database serving various purposes, data may need to be sourced from different spatial datasets such as: building design models in

BIM format, topographic and built environment information in CityGML, and cadastral legal boundaries in LandXML (Soon et al, 2014).

In the context of cadastral requirements, the CityGML does not contain any features describing the legal information about spatial objects (Góźdź et al, 2014).

As also stated in Góźdź et al (2014), the Land Administration Domain Model also constitutes a generic expandable domain model, designed to be connected in SDI-setting to data from other domain models and other standards (e.g. CityGML, INSPIRE Data Specifications).

Exploring the link with ExtPhysicalBuildingUnit (as represented according to CityGML or IndoorGML or BIM/IFC) is an important topic to explore further; e.g. which LOD level is being referred to (see figure 5). Obviously, when a single building contains multiple spatial units, then indoor is needed (LoD4 in CityGML or perhaps better use IndoorGML or BIM/IFC representations). Note that the link between the LA_SpatialUnit and ExtPhysicalBuildingUnit (or ExtPhysicalUtilityNetwork) does not have direct legal implication. However, if corresponding 3D spaces are very different, then someone should take action. Actual reusing of (3D) topographic objects as boundaries of legal spaces could be a dangerous step (if physical object moves / changes, then also legal spaces might be affected unintendedly), so care is needed (Thompson et al, 2016).

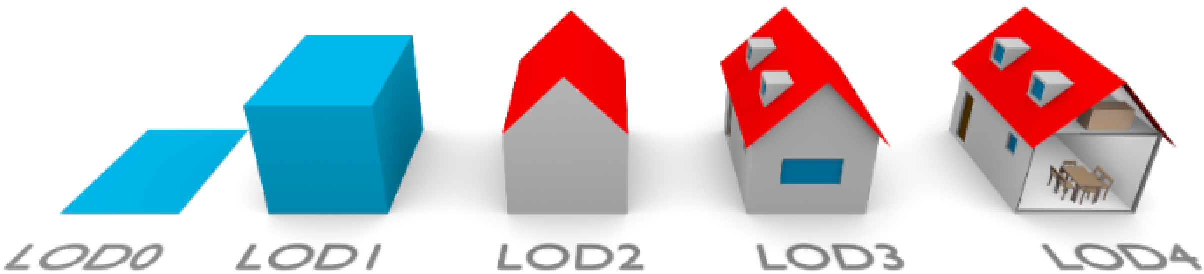

Figure 5. The five LODs of CityGML 2.0. The geometric detail and semantic complexity increase, ending with LOD4 containing indoor features (Biljecki et al, 2016)

Not only the geometrical aspect, the semantic aspect of data sources should also be considered. Building data in BIM/IFC, CityGML and LandXML are produced based on different domain knowledge (design, physical and legal). This causes conceptual and terminological differences between data sources if these data sources are to be integrated (Soon et. al. (2014)).

Rönsdorf et al (2014) demonstrated how the OGC CityGML standard can be used to provide an encoding for 3D land administration information. The basic principles of the integration by mapping key feature classes in both standards are shown. Further they conclude, that the same approach will be applicable for country or region specific profiles of ISO 19152 and encourage practical experimentation with this.

The possibilities of applying CityGML for cadastral purposes are elaborated in Góźdź et al (2014) with particular attention to the 3D representation of buildings. A proposal of the

CityGML-LADM ADE is presented. Drawing particular attention to the buildings, both addressing their physical aspects, and their legal counterparts. Technical realization of the issue has been executed at the conceptual level by integration the CityGML OGC Standard and the International Standard ISO 19152. Practical implementation of the CityGML-LADM ADE model has demonstrated the benefits of providing relations between spatial objects from legal and physical world. The insight into the third dimension of physical objects helps to understand the location and size of the legal spaces as well as it is relevant in the context of developing the multipurpose cadastral systems.

Ying et al (2014) provide a framework and workflow of the conversion from CityGML data to 3D Cadastral unit with the test of city data of CityGML LOD3.

Roschlaub and Batscheider (2016) used 3D City Database (3DCityDB[2]) to store the 3D buildings (at LOD2 level) created as a combination of 2D digital building ground plans derived from the official digital cadastral map and LIDAR (*Light Detection And Ranging*) data. 3D City Database is a free 3D geo database to store, represent, and manage virtual 3D city models on top of a standard spatial relational database. The database model contains semantically rich, hierarchically structured, multi-scale urban objects facilitating complex GIS modeling and analysis tasks. With a database scheme the user has the possibility to create a CityGML conformant data model in the database. Seifert et al (2016) add, that this data participates in the existing national and international spatial data infrastructure (SDI), for example through simple export to the defined INSPIRE topics (e.g. Buildings).

2.6 Operations on the 3D data types

2.6.1 Creation and validation

With the utilization and development of dense urban space, true 3D geometric volume primitives are needed to represent 3D parcels with the adjacency and incidence relationship. A volumetric primitive is a complete representation of a polyhedron able to support the various calculations and analysis related to the 3D cadastral objects. The volumetric primitives in 3D space need to be mutually exclusive and they need to exhaustively partition the extent of the domain (i.e. no gaps are allowed) (Ying et al, 2015).

SQL Geometry Types

The SQL Geometry Types (OGC, 2010) extend the set of available predefined data types to include Geometry Types. A conforming implementation shall support a subset of the following set of Geometry Types: {Geometry, Point, Curve, LineString, Surface, Polygon, PolyhedralSurface, GeomCollection, MultiCurve, MultiLineString, MultiSurface, MultiPolygon, and MultiPoint}.

OGC (2010) presents a new SQL geometry type – PolyhedralSurface, which shall be subtyped from Surface, and implements the required constructors routines and interfaces of Surface and MultiSurface. A PolyhedralSurface is a contiguous collection of polygons, which share common boundary segments and which as a unit have the topological attributes of a surface. For each pair of polygons that “touch”, the common boundary shall be expressible as

[2] http://www.3dcitydb.org/3dcitydb/3dcitydbhomepage *(accessed on 21 August 2016).*

a finite collection of LineStrings. Each such LineString shall be part of the boundary of at most 2 Polygon patches. The PolyhedralSurface could be a simple, closed polyhedron (OGC, 2011).

While there exists definition for solids (given by the international standards for geographic information), Ledoux (2014) states that these definitions for solids are ignored by most researchers and software vendors. He states, that several different definitions are indeed used, and none is compliant with the standards: e.g. solids are often defined as 2-manifold objects only, while in fact they can be non-manifold objects. Exchanging and converting datasets from one format/platform to another is thus highly problematic. Ledoux (2014) presents a methodology to validate solids according to the international standards. He implemented the methodology in a prototype called *val3dity*[3].

The validator for solids in Oracle Spatial permits us to validate solids (although, as explained it is neither according to the ISO rules nor complete) but returns only one error when the solid is not valid: the first one encountered (even if a given solid contains hundreds of errors). The error comes with a code explaining its nature and, when suitable, its location (for example if a shell is not closed the centre of the hole is given). This means that a user has to fix the solid for the error mentioned, and to run again the validation function. This step has to be followed for all the errors present, which can be a rather long and painful process for the user. Ideally, all the errors in a solid should be reported so that a user can fix them in one operation. However, cascading effects when validating should be avoid—one example is if a surface is not a valid polygon in 2D, then the validation of the shell whose boundary contains that surface should not be attempted as it will most likely not be valid. In the prototype *val3dity*, a "hierarchical validation" is used and efforts are made to avoid cascading errors (Ledoux, 2014).

2.6.2 Spatial indexing

The important aspect of 3D data management is spatial indexing. Spatial indexes are used in DBMS for fast search especially when spatial functions are applied. Without indexing, any searches for a feature would require a sequential scan of every record in the database. Indexing speeds up searching by organizing the data into a search tree that could be quickly traversed to find a particular record.

The review of spatial indexing give Breunig and Zlatanova (2011). Within the current SDBMSs, e.g. PostGIS and Oracle Spatial, there are several types if indexes (Khuan et al, 2008): they are B-Tree indexes, R-Tree indexes (Guttman, 1984), and GiST indexes.

- B-Trees are used for data, which can be sorted along one axis; for example, numbers, letters, dates. GIS data cannot be rationally sorted along one axis (which is greater, (0,0) or (0,1) or (1,0)) so B-Tree indexing is of no use for GIS user.
- R-Trees break up data into rectangles, and sub-rectangles, and sub-sub rectangles, etc. R-Trees are used by some spatial databases to index GIS data, e.g. Oracle Spatial implemented the 3D R-Trees.

[3] https://github.com/tudelft3d/val3dity *(accessed on 20 August 2016)*

- GiST (Generalized Search Trees) indexes break up data into ‘things to one side’, ‘things which overlap’, ‘things which are inside’ and can be used on a wide range of data-types, including GIS data. PostGIS uses GiST to index GIS data.

2.6.3 Analysis in DBMS

In the implementation specification, OGC (2011) provides the geometry functions that are not limited to any dimension.

Some of the standard functions given by OGC (Simple feature access – Part 1: Common Architecture (OGC, 2011)):

- Envelope ( ): Geometry – The minimum bounding box for the *Geometry*, returned as a Geometry. Minimums for Z and M may be added.
- IsSimple ( ): Integer – Returns 1 (TRUE) if *this* geometric object has no anomalous geometric points, such self intersection or self tangency. The description of each instantiable class will include the specific conditions that cause as instance of that class to be classified as not simple.
- Is3D ( ): Integer – Returns 1 (TRUE) if *this* geometric object has z coordinate values.
- etc.

Furthermore, OGC (2011) define methods for testing spatial relations between geometric objects:

- Equals (anotherGeometry: Geometry): Integer – Returns 1 (TRUE) if *this* geometric object is “spatially equal” to anotherGeometry.
- Intersects (anotherGeometry: Geometry): Integer – Returns 1 (TRUE) if *this* geometric object “spatially intersects” anotherGeometry.
- Touches (anotherGeometry: Geometry): Integer – Returns 1 (TRUE) if *this* geometric object “spatially touches” anotherGeometry.
- etc.

Only DBMS itself decides the implementation of the standard functions (specified by OGC) that considers the third dimension or not (Khuan, 2008).

2.7 Standardization

2.7.1 ISO

ISO is an independent, non-governmental international organization with a membership of 163 national standards bodies[4]. Through its members, it brings together experts to share knowledge and develop voluntary, consensus-based, market relevant International Standards that support innovation and provide solutions to global challenges.

The ISO 19100 is a series of standards for defining, describing, and managing geographic information. This standard defines the architectural framework of the ISO 19100 series of standards and sets forth the principles by which this standardization takes place. Standardization of geographic information can best be served by a set of standards that integrates a detailed description of the concepts of geographic information with the concepts of information technology. A goal of this standardization effort is to facilitate interoperability of geographic information systems, including interoperability in distributed computing environments. The ISO 19100 series of geographic information standards establishes a

[4] http://www.iso.org/iso/home/about.htm *(accessed on 19 August 2016)*

structured set of standards for information concerning objects or phenomena that are directly or indirectly associated with a location relative to the Earth. This standard specifies methods, tools and services for management of geographic information, including the definition, acquisition, analysis, access, presentation, and transfer of such data in digital/electronic form between different users, systems and locations.

The overall objectives of ISO/TC 211 are (ISO/TC 211, 2009):

- increase the understanding and usage of geographic information;
- increase the availability, access, integration, and sharing of geographic information;
- promote the efficient, effective, and economic use of digital geographic information and associated hardware and software systems;
- contribute to a unified approach to ecological and humanitarian problems.

2.7.2 OGC

The Open Geospatial Consortium (OGC) is a non-profit organization that deals with the development of standards for modelling real-world objects. These standards deal with conceptual schemes for describing and manipulating the spatial characteristics of geographic features. The specification defines three important areas, namely (Khuan et al, 2008):

- Data types: the need to have data types that represent real world object is obvious. Different kinds of data types and different kinds of objects could be modelled within DBMS.
- Functions/operations: there must be functions and operators to support the management of multi-dimensional objects that work for spatial analysis in DBMS.
- Spatial index: the main purpose is to deal with spatial searching (query), and sometimes it implements in different operators to speed up the query process.

2.7.3 Cooperation between ISO and OGC

By 1995, ISO/TC 211 developing international standards for spatial data and the OGC developing computer interface specifications became highly visible and prominent players on the international geographic agenda. Afterwards, ISO/TC 211 and the OGC formed a joint coordination group to leverage mutual development and minimize technical overlap. The OGC is submitting their specifications for ISO standardization via ISO/TC 211. Achieving more interoperability requires a proactive coordination of spatial standards at both the abstract and implementation levels. Proactive cooperation among spatial standards activities of ISO/TC 211 and the OGC should also help to use available resources more efficiently by minimizing technical overlap, wherever this occurs. Such coordination and cooperation should lead to more market-relevant spatial standards, and could serve as a useful roadmap for all interested parties (ISO/TC 211, 2009).

2.7.4 ISO 19152 LADM

LADM is of one of the first spatial domain standards within ISO TC 211. There is a need for domain specific standardisation to capture the semantics of the land administration domain on top of the agreed foundation of basic standards for geometry, temporal aspects, metadata, and also observations and measurements from the field. This is required for communication between professionals, for system design, system development and system implementation purposes and for purposes of data exchange and data quality management. Such a standard will enable Geographical Information Systems (GIS) and database providers and/or open source communities to develop products and applications. And in turn this will enable land registry and cadastral organisations to use these components to develop, implement and maintain systems in an even more efficient way. LADM provides a shared ontology, defining a terminology for land administration. It provides a flexible conceptual schema with three basic packages: parties, rights (and restrictions/responsibilities) and spatial units. LADM supports the development of application software for land administration, and facilitates data exchange with and from distributed land administration systems (Van Oosterom and Lemmen, 2015).

In LADM, 2D and 3D representations of spatial units use boundary face strings and boundary faces as key concepts (see figure 6 and 7).

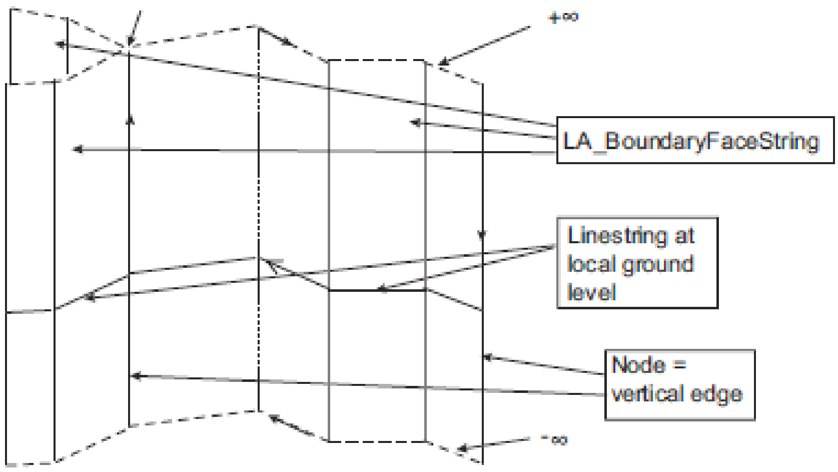

Figure 6. Boundary face string concepts (ISO, 2012)

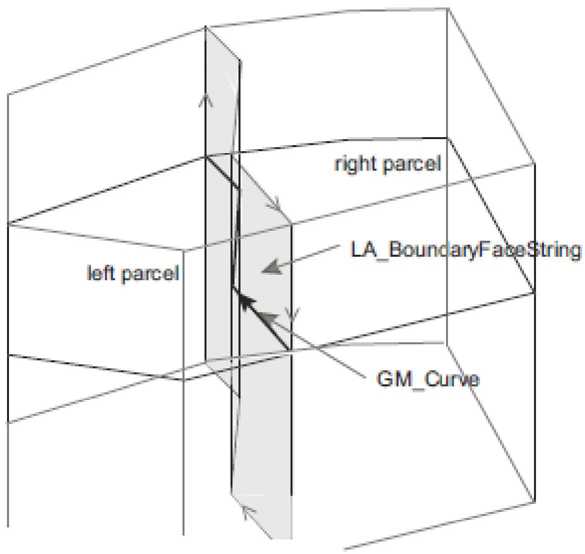

Figure 7. Spatial units defined by boundary face strings (ISO, 2012)

2.7.5 GML

GML is an XML grammar defined by OGC to express geographical features (ISO, 2007). GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet. As with most XML based grammars, there are two parts to the grammar – the schema that describes the document and the instance document that contains the actual data. A GML document is described using a GML Schema. This allows users and developers to describe generic geographic data sets that contain points, lines and polygons. However, the developers of GML envision communities working to define community-specific application schemas that are specialized extensions of GML. Using application schemas, users can refer to roads, highways, and bridges instead of points, lines and polygons.

Aien et al (2014) convert the logical data model of the 3D Cadastral Data Model (3DCDM) to a physical data model. The physical data model of the 3DCDM has been developed as an application scheme of the GML (in version 3.2.1). For this purpose, eleven XML schemes were developed.

2.7.6 CityGML

There are many formats for the storage and visualization of the spatial data, however they are usually focused only on a description of geometry. In contrast, the CityGML which provides a geographic information model for urban landscapes, not only represents the shape and graphical appearance of the 3D city objects, but also addresses the representation of the semantic and thematic properties, taxonomies and aggregations (Góźdź et al, 2014).

Open Geospatial Consortium has defined CityGML (City Geography Markup Language) for modeling 3D city models. The current version of CityGML is 2.0 and contains modules like Relief, Building, City Furniture, Water Body, Bridge, Tunnel, Vegetation, Land Use, and Transportation. CityGML defines classes, attributes and relations for topographic features with aspects of geometrical, topological, semantic and appearance. Different level of details can be captured from LOD 0 to LOD 4. LOD 0 represents the earth surface (i.e. the terrain) be it as Digital Terrain Model (DTM) or Digital Surface Model (DSM). LOD 1 represents topographic and constructed features as simple 3D blocks (i.e. no texturing or appearance). LOD 2 shows topographic features with texturing and refined top structure. As the case of building for example, instead of a flat roof surface in LOD 1, LOD 2 models the actual shape of a rooftop. LOD 3 models more detailed topographic features and includes other external installations for example windows and doors. LOD 4 includes internal installation modeling (van den Brink et. al., 2012).

In the Building module of CityGML, Abstract Building is an important class, which has two subclasses called Building and Building Part. The attributes for Abstract Building class include Class, Function, Usage, RoofType, MeasuredHeight, etc. Abstract Building class also has geometries, which support for the level of details from LOD 0 to LOD 4. As Abstract Building class' specializations, Building and Building Part inherit all attributes and relations of Abstract Building (Soon et al, 2014).

The CityGML schema can be extended to have additional modules such as Cadastre using the Application Domain Extension (ADE) (Stoter et al, (2011); van den Brink et al, (2012); Góźdź et al, 2014).

2.7.7 W3C

The World Wide Web Consortium (W3C) is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards. W3C publishes documents that define Web technologies. These documents follow a process designed to promote consensus, fairness, public accountability, and quality. At the end of this process, W3C publishes Recommendations, which are considered Web standards[5].

2.7.8 LADM OWL ontology

The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs, e.g., to verify the consistency of that knowledge or to make implicit knowledge explicit. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies. The current version of OWL, also referred to as "OWL 2" is an extension and revision of the 2004 version of OWL[6].

With the current ISO 19152 - Land Administration Domain Model (LADM) standard (ISO, 2012) that is modelled in Unified Modeling Language (UML) and additional explanatory natural text and tables, it will facilitate the software development and database design for the proper implementation of land administration systems. The use of UML supports generating a database schema or exchange format (Soon et al, 2014). To support reasoning and inference, Soon (2013) has formalized LADM in OWL. LADM OWL ontology also support automated integration for land administration information (Boskovic, et al, 2010; Sladić, et al, 2013).

With the intention to use the LADM OWL ontology for automated integration of land administration information, Soon et al (2014) proposed to augment the LADM OWL ontology with a concept Physical Space Building Unit (see fig. 8). In addition, as a physical building sometimes can have more than one legal boundary, for example through strata subdivision, a relation is defined as *hasLegalSpace* between Physical Space Building Unit and Legal Space Building Unit. *hasLegalSpace* is an ObjectProperty in the LADM OWL ontology. The same also applies to utility network where a new concept Physical Space Utility Network is added. The relation *hasLegalSpace* also links Physical Space Utility Network with Legal Space Utility Network (Soon et al, 2014).

[5] https://www.w3.org *(accessed on 19 August 2016)*
[6] https://www.w3.org/OWL *(accessed on 19 August 2016)*

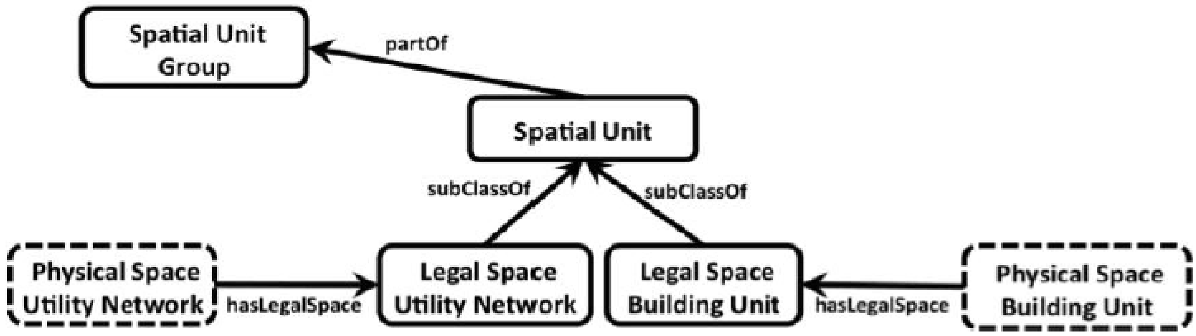

Figure 8 Extension to the LADM OWL ontology with the concept Physical Space Utility Network and Physical Space Building Unit (highlighted in dash-lined boxes) and with a new relation *hasLegalSpace* (Soon et al, 2014)

With the addition of new concepts (Physical Space Building Unit, Physical Space Utility Network) in the LADM OWL ontology, it helps to integrate information about building from CityGML and LandXML as discussed in detail by Soon et al (2014).

2.7.9 BIM/IFC

ISO 16739:2013 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries, specifies a conceptual data schema and an exchange file format for Building Information Model (BIM) data (ISO, 2013).

Under development is ISO/AWI 19166 Geographic information -- BIM to GIS conceptual mapping (B2GM)[7]. This international standard defines the conceptual framework and mechanisms for mapping of information elements from BIM to GIS to access the needed information based on specific user requirements. The conceptual framework for mapping BIM information to GIS are defined with the following three mapping mechanisms:

- BIM to GIS Element Mapping (B2G EM);
- BIM to GIS LOD (Level of Detail) Mapping (B2G LM);
- BIM to GIS Perspective Definition (B2G PD).

The conceptual mapping mechanism defined in this international standard uses existing international standards such as Geography Markup Language (GML), CityGML (OGC standard) and Industry Foundation Classes (IFC).

2.7.10 Transport encoding of cadastral information

There are currently two transport specifications in discussion for the interchange of survey plan data: 1: LandXML which is currently in use in New Zealand and being implemented in Australia and Singapore, and 2: InfraGML which is being developed by the OGC as a BIM interchange specification, and as successor of LandXML for survey data (Thompson et al, 2016). LandXML can also be used for capturing other types of engineering data, such as pipe networks and roadways (Soon et al, 2014).

Soon et al (2014) extend LandXML to model 3D parcels and introduce the Nested Parcels Approach, which makes use of the element of PntList3D of LandXML, to store 3D coordinates.

[7] http://www.iso.org/iso/catalogue_detail.htm?csnumber=32584 (*accessed on 19 August 2016*)

In addition to LandXML, the expression in InfraGML (currently in development by the Open Geospatial Consortium) (Scarponcini 2013; OGC 2016) should be considered for the integrated footprint (LA_BoundaryFaceString) and LA_BoundaryFace integrated footprint (LA_BoundaryFaceString) and LA_BoundaryFace volumetric encoding of spatial units (Thompson et al, 2016).

3. CURRENT DBMS 3D CAPABILITIES

3.1 General 3D/4D geometry/topology capabilities

Due to the complexity of real-world spatial objects, various types of representations (e.g. vector, raster, constructive solid geometry, etc.) and spatial data models (topology, and geometry) have been investigated and developed. Promising developments were observed in the SDBMS domain where more spatial data types, functions and indexing mechanism were supported. In this respect, SDBMSs are expected to become a critical component developing of an operational 3D GIS. However, the native 3D support at SDBMS level has to be achieved (Khuan et al, 2008).

Mostly all the main spatial database management systems support the Simple Feature Access international standard. Some database systems (e.g. MySQL) support only version 1.0 but most of them (e.g. PostGIS, Microsoft SQL Server, Oracle Spatial) provide user with version 1.2 allowing modelling 3D geometries (Janečka and Kára, 2012).

3.2 Oracle Spatial

The spatial features in Oracle Spatial consist of a set of object data types, type methods, and operators, functions, and procedures that use these types. A geometry is stored as an object, in a single row, in a column of type SDO_GEOMETRY. Spatial index creation and maintenance is done using basic DDL (CREATE, ALTER, DROP) and DML (INSERT, UPDATE, DELETE) statements[8].

3.2.1 <u>Geometry types</u>

A geometry (in Oracle Spatial) is an ordered sequence of vertices that are connected by straight line segments or circular arcs. The semantics of the geometry are determined by its type. Spatial supports several primitive types, and geometries composed of collections of these types, including two-dimensional: points and point clusters, line string, n-point polygons, arc line strings (all arcs are generated as circular arcs), arc polygons, compound polygons, compound line string, circles, optimized rectangles.

Spatial also supports the storage, indexing (R-tree) and retrieval of three-dimensional and four-dimensional geometric types, where three of four coordinates are used to define each vertex of the object being defined.

The three-dimensional spatial data can include: points, point clouds (collection of points), lines, polygons, surfaces, and solids.

---

[8] https://docs.oracle.com/database/121/SPATL/toc.htm *(accessed on 21 August 2016)*

Table 1. SDO_GEOMETRY attributes for three-dimensional geometries (*here only Solid and Multisolid are mentioned*)

Type of 3D Data	SDO_GTYPE	Element Type, Interpretation on SDO_ELEM_INFO
Solid	3008	Simple solid formed by a single closed surface: one element type (*SDO_ETYPE, see table 2*) 1007, followed by one element type 1006 (the external surface) and optionally one or more element type 2006 (internal surfaces) Composite solid formed by multiple adjacent simple solids: one element type 1008 (holding the count of simple solids), followed by any number of element type 1007 (each describing one simple solid)
Multisolid	3009	Element definitions for one or more simple solids (element type 1007) or composite solids (element type 1008)

Table 2. Values and semantics in SDO_ELEM_INFO

SDO_ETYPE	SDO_INTERPRETATION	Meaning
1006 or 2006	$n > 1$	Surface consisting of one or more polygons, with each edge shared by no more than two polygons. A surface contains an area but not a volume. The value n in the Interpretation column specifies the number of polygons that make up the surface. The next n triplets in the SDO_ELEM_INFO array describe each of these polygon subelements. A surface must be three-dimensional.
1007	$n = 1$ or 3	Solid consisting of multiple surfaces that are completely enclosed in a three-dimensional space, so that the solid has an interior volume. A solid element can have one exterior surface defined by the 1006 elements and zero or more interior boundaries defined by the 2006 elements. The value n in the Interpretation column must be 1 or 3. Subsequent triplets in the SDO_ELEM_INFO array describe the exterior 1006 and optional interior 2006 surfaces that make up the solid element. If n is 3, the solid is an optimized box, such that only two three-dimensional points are required to define it: one with minimum values for the box in the X, Y, and Z dimensions and another with maximum values for the box in the X, Y, and Z dimensions.

3.2.2 Extending spatial indexing capabilities

Oracle Spatial enables one to create and use spatial indexes on objects other than a geometry column. The SDO_GEOMETRY object can be embed in a user-defined object type, and then the geometry attribute of that type can be indexed. Further, one can create and use a function-based index where the function returns the SDO_GEOMETRY object.

3.3 PostGIS

PostGIS is a spatial database extender for PostgreSQL object-relational database. It adds support for geographic objects allowing location queries to be run in SQL. In addition to basic location awareness, PostGIS offers many features rarely found in other competing spatial databases such as Oracle Locator/Spatial and SQL Server. PostGIS adds extra types (geometry, geography, raster and others) to the PostgreSQL database.

It also adds functions, operators, and index (Generalized Search Tree (GiST)) enhancements that apply to these spatial types. These additional functions, operators, index bindings and types augment the power of the core PostgreSQL DBMS, making it a fast, feature-plenty, and robust spatial database management system[9].

The GIS objects supported by PostGIS are a superset of the "Simple Features" defined by the OGC. PostGIS supports all the objects and functions specified in the OGC "Simple Features for SQL" specification. PostGIS extends the standard with support for 3DZ, 3DM and 4D coordinates.

Some PostGIS functions related to solids:

- ST_IsSolid — Tests if the geometry is a solid. No validity check is performed.
- ST_MakeSolid — Casts the geometry into a solid. No check is performed. To obtain a valid solid, the input geometry must be a closed Polyhedral Surface or a closed TIN.
- ST_Volume — Computes the volume of a 3D solid. If applied to surface (even closed) geometries will return 0.

3.4 3D Topology

In the widely used SDBMSs such as Oracle Spatial, PostGIS, ESRI Geodatabase, 2D topology is well supported and documented. However, in most of current SDBMSs, 3D topology is not natively supported. So one must construct and store topology by his own approach.

3.5 Point clouds and TINs

ESRI Geodatabase allows storing TIN as a planar graph where nodes are connected by edges to form triangles. Edges connect nodes that are close to one another. PostGIS has constructors for creating 3D geometry. *pgpointcloud*[10] is a PostgreSQL extension and loader for storing point cloud data in PostgreSQL. Also includes extension for casting between point cloud data type and PostGIS geometry. TIN in PostGIS is modelled as a special case of polyhedral surface which is collection of adjacent triangles. Very similar situation is for Microsoft SQL Server. From the data structures point of view, Oracle Spatial is an example of SDBMS providing suitable data structures and mechanisms directly for TINs and point clouds. When the available specialized object types are used, then a point cloud can be stored in a single row, in a single column in a user-defined table in Spatial. These object types related to point clouds and TINs are elaborated e.g. in (Janečka and Kára, 2012).

Martinez et al (2014) used MonetDB and PostgreSQL with the point cloud data to understand the impact of the point cloud data on the different layers of a DBMS. It touches from key issues from (adaptive) data loading to optimization of queries over point clouds. The results obtained through a micro benchmark illustrate both the capabilities to handle point cloud queries efficiently, but also the relative merits of traditional index structures and compression techniques on the performance characteristics. They conclude, that MonetDB can be considered more modern than PostgreSQL, because it is designed from an in-memory perspective and relies on the operating system to move data between the storage hierarchies in

[9] http://postgis.net/docs/manual-2.2
[10] https://github.com/pgpointcloud/pointcloud *(accessed on 21 August 2016)*

an efficient manner. All queries are also highly parallel, using the cores available wherever possible. On the contrary, PostgreSQL represents the traditional buffer-based and iterator query engine approach. Tuning the buffer size to use all available memory by itself does not help because the logic of chasing data in buffers remains. Further they mention, that PostgreSQL does not by default support multi-core query processing.

Van Oosterom et al (2015) design a point cloud benchmark based on requirements from different groups of users within government, industry and academia. They analyse various data management systems: PostgreSQL, MonetDB, Oracle, and LAStools. They further state that the Oracle Exadata[11] with flat table model proved to be a very effective environment, both with respect to data loading and querying. Due to the massive parallel hardware engineered towards DBMS support, it was possible to load 23 billion points in less than 4:39 hours and storing the 12 Tb data from LAS files into a 2.2 Tb database (using 'query high' compression). In case of queries returning a very large number of points (from 10 million to over 1 billion), the system outperformed the other platforms.

3.6 Voxelization algorithms for geospatial applications

Nourian et al (2016) present algorithms that generate voxels (volumetric pixels) out of point cloud, curve, or surface objects. The aim of their research is to provide easy access to methods for making large-scale voxel models of built environment for environmental modelling studies while ensuring they are spatially correct, meaning they correctly represent topological and semantic relations among objects. The algorithms for Voxelization of surfaces and curves are a customization of the topological voxelization approach (Laine, 2013).

3.7 Tetrahedral networks for modelling 3D topographic objects

For storing and modelling three-dimensional topographic objects (e.g. buildings, roads and terrain), tetrahedralisation have been proposed as an alternative to boundary representations.
Penninga (2005) presented a modelling approach for 3D topography modelling based on tetrahedral network (TEN).
The approach is based on two fundamental observations:

- The ISO 19101 Geographic information - Reference model defines a feature as an 'abstraction of real world phenomena'. These real world phenomena have by definition a volumetric shape. In modelling often a less-dimensional representation is used in order to simplify the real world. Fundamentally there are no such things as point, line or area features; there are only features with a point, line or area representation (at a certain level of abstraction/generalization).
- The real world can be considered to be a volume partition. A volume partition can be defined (analogously to a planar partition) as a set of non-overlapping volumes that form a closed modelled space. As a consequence objects like 'air' or 'earth' are explicitly part of the real world and thus have to be modelled.

Four types of topographic features can be determined: 0D (point features), 1D (line features), 2D (area features) and 3D (volume features). For each type of feature simplexes of corresponding dimension are available to represent the features with, i.e. nodes, edges, triangles and tetrahedrons. A great advantage of using these simplexes is the well-defined character of the mutual relationships: a kD simplex is bounded by k+1 (k-1)D-simplexes

[11] http://www.oracle.com/technetwork/database/exadata/overview/index.html *(accessed on 21 August 2016)*

(Pilouk, 1996). The important advantage of simplexes is the flatness of the faces, which enables one to describe a face using only three points. The next advantage is that every simplex, regardless its dimension, is convex, thus making convexity testing unnecessary (Penninga, 2005).

The topographic model is stored as a full TEN. The process of modelling topographic features consist of four discernible steps:

1. Start with four initial tetrahedrons, two 'air' and two 'earth' tetrahedrons;
2. Refine the earth's surface by inserting height information from a DEM;
3. Refine 'air' and 'earth' tetrahedrons in case of ill-shaped tetrahedrons by insertion of Steiner points;
4. Add real topographic features.

Triangulating or tetrahedronizing the features one-by-one before insertion in the topographic model reduces computational complexity and thus saves computer time. The results need to be inserted into the full topographic model. This requires the use of an incremental algorithm to avoid recomputing the whole model. As the complete topographic model (the TEN) will be stored in a spatial database, it is necessary to implement the incremental algorithm within the database. As a result a full DBMS approach is required, instead of using the database just to store results of the computations (Penninga, 2005).

Penninga (2008) proposed a DBMS data structure for storage of a constrained TEN. His simplicial complex-based method requires only explicit storage of tetrahedrons, while simplexes of lower dimensions (triangles, edges, and nodes), constraints and topological relationships can be derived in views. In this implementation, simplexes are encoded by their vertices. He demonstrates, that storage requirements for 3D objects in tetrahedronised form (excluding the space in between these objects) and 3D objects stored as polyhedrons are in the same order of magnitude.

A TEN has favourable characteristics from a computational point of view. All elements of the tetrahedral network consist of flat faces (important for clear inside/outside decisions), all elements are convex and they are well defined, thus allowing relatively easy implementation of operations, such as validation of 3D objects (Penninga, 2008).

A full volumetric approach contributes not only to improved analytical and validation capabilities, but also enables future integration of topography and other 3D data within the same volume partition (Penninga, 2008).

Since the edit operations act as locally as possible, the resulting tetrahedronization is not necessarily of the best quality. To overcome this drawback, periodical quality improvements need to be made. Three types are distinguished: operators that add vertices, operators that remove vertices and operators that modify the TEN configuration through flips. Every now and then a complete TEN rebuild might be feasible to optimise TEN quality (Penninga, 2008).

Ledoux and Meijers (2013) proposed an alternative data structure for storing tetrahedralisation in a DBMS (see figure 10). It is based on the idea of storing only the vertices and *stars* of edges; triangles and tetrahedra are represented implicitly. The structure permits one to store attributes for any primitives, and has the added benefit of being topological, which permits one to query it efficiently.

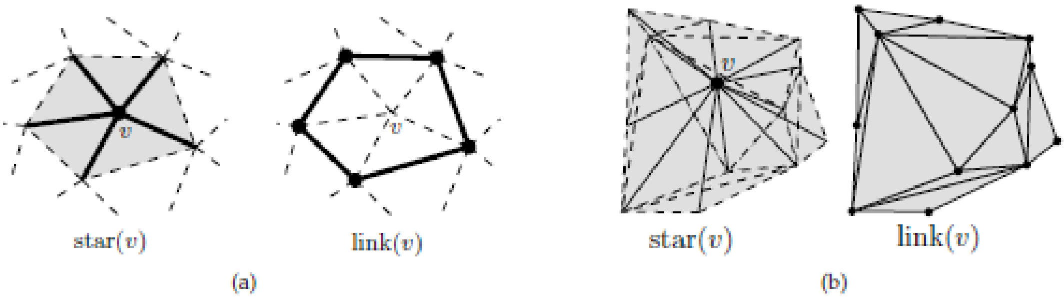

Figure 9. The star and the link of a vertex v in (a) 2D and (b) 3D (Ledoux and Meijers, 2013).

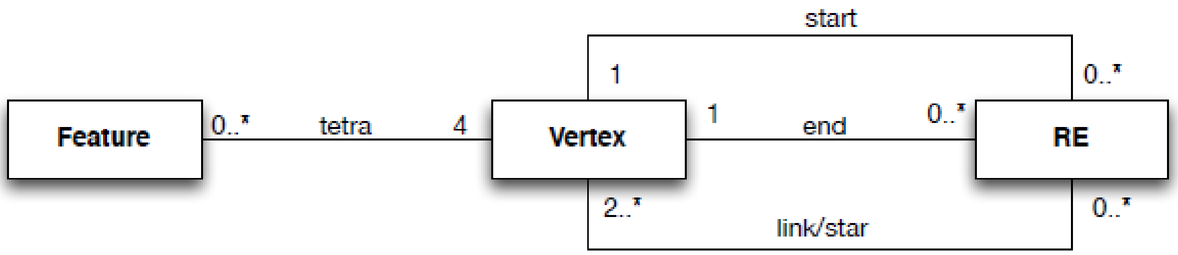

Figure 10. The UML diagram of the data model for star-based data structure (Ledoux and Meijers, 2013)

The strong point of the star-based structure is that it can be easily implemented in any DBMS supporting variable length arrays with two simple tables, and that no complex spatial index is needed (Ledoux and Meijers, 2013).

3.8 Recent developments of spatial databases

3.8.1 n-D arrays

In terms of Big Data, Baumann (2014) finds in particular three main contributors today: grids (both regular and irregular), point clouds, and general meshes.

The term array is seen here in a programming language sense and synonymously to raster data, regularly gridded data, and Multi-Dimensional Discrete Data (MDD) (Furtado and Baumann, 1999). MDD is array of arbitrary size, dimension and base type.

Since computer memory is inherently linear - a one-dimensional structure, mapping multi-dimensional data on it can be done in several ways. By far the two most common memory layouts for multi-dimensional array data are row-major and column-major. When working with 2D arrays (matrices), row-major vs. column-major are easy to describe. The row-major layout of a matrix puts the first row in contiguous memory, then the second row right after it, then the third, and so on. Column-major layout puts the first column in contiguous memory, then the second, etc. (Bendersky, 2015).

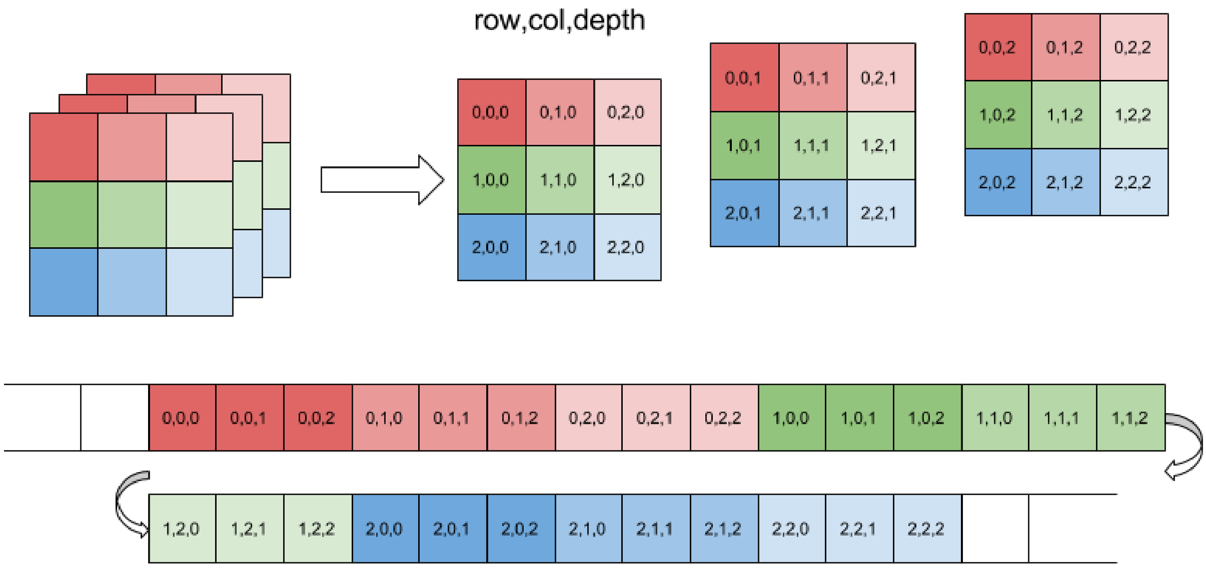

Figure 11. Mapping 3D array with $N_1 = N_2 = N_3$ in row-major (Bendersky, 2015)

The offset for a given element is:

$$offset = n_3 + N_3 * (n_2 + N_2 * n_1)$$

For example, the offset of the element with indices 2,1,1 is 22 (Bendersky, 2015).
While the database collection types set, list, and record have received in-depth attention, the fourth type, array, is still far from being integrated into database modeling. Due to this lack of attention there is only insufficient array support by today's database technology. This is surprising given that large, multi-dimensional arrays have manifold practical applications in earth sciences (such as remote sensing and climate modeling), life sciences (such as microarray data and human brain imagery), and many more areas. (Bauman and Holsten, 2010).

To overcome this, large, multi-dimensional arrays as first-class database citizens have been studied by various groups worldwide. Several formalisms and languages tailored for use in array databases have been proposed and more or less completely implemented, sometimes even in operational use (Bauman and Holsten, 2010). Array Databases close a gap in the database ecosystem by adding modeling, storage, and processing support on multi-dimensional arrays (Baumann and Merticariu, 2015).

In the attempt towards a consolidation of the field Bauman and Holsten (2010) compare four important array database models: AQL, AML, ARRAY ALGEBRA, and RAM. As it turns out, ARRAY ALGEBRA is capable of expressing all other models, and additionally offers functionality not present in the other models. They show this by mapping all approaches to ARRAY ALGEBRA. This establishes a common representation suitable for comparison and allows us discussing the commonalities and differences found. Finally, a feasibility of conceptual array models for describing optimization and architecture was showed.
ARRAY ALGEBRA adopts an algebraic approach to array modeling. The targeted application domains of ARRAY ALGEBRA encompass sensor, image, and statistics data

services. However, at stated in Bauman and Holsten (2010), current emphasis is on large-scale Earth Science (Gutierrez and Baumann, 2007) data.

The *RasDaMan* array DBMS with its query language, rasql [12], implements ARRAY ALGEBRA. This system is in operational use since many years, among others as the geo raster server of the French National Geographic Institut where an airborne image map of dozen TB size is maintained. The *RasDaMan* implementation employs a middleware architecture where multidimensional arrays are partitioned into multi-dimensional sub-arrays called tiles. These tiles, which represent the units of disk access, are stored in BLOBs (binary large objects) inside some relational or object-oriented database, such as PostgreSQL or O2[13]. A spatial index helps to quickly determine the tiles affected by a query. Query processing relies on tile streaming: Physical query operators follow the open-next-close (ONC) protocol for reading their inputs tile by tile, and likewise they deliver their results in units of tiles. Based on this processing paradigm, the *RasDaMan* architecture follows a conventional multi-user DBMS approach, however, with all components crafted individually to accommodate the special needs of array processing. Array definition and query languages, rasdl and rasql, are available to the application via command line tools, visual tools, and C++ and Java APIs. The client/server communication protocol connects clients to the DBMS server. A dispatcher distributes incoming queries among the rasdaman server processes running. Each server process (see Figure 12) receives queries and parses, optimizes, and executes them. Auxiliary modules include catalogue manager, index manager, as well as cache and transaction manager. For example, the catalogue contains the array and collection type definitions against which semantic checks (like boundary checks for array dimensions not containing open limits) are performed during query analysis. The base DBMS interface layer abstracts from the particularities of the underlying DBMS. Adaptors exist for PostgreSQL, MySQL, Oracle, DB2, Informix, and the file system. Thereby, both array data, rasdaman-internal array metadata, and non-array application data all end up in the same underlying database. As practice shows, this information integration considerable eases database administration (Baumann and Holsten, 2010).

In industrial world, e.g. Oracle offers the GeoRaster cartridge for 2-D geo raster imagery stored in a database. Instead of a rigorous embedding into SQL there are procedural constructs in PL/SQL which accomplish raster access as well as invocation of a set of predefined functions (Baumann and Holsten, 2010). PostGIS Raster is an extension to PostGIS which supports 2-D raster imagery through so-called map algebra functions; unlike in rasdaman, these are implemented as user-defined data types and, hence, not as tightly integrated and optimizable. PostGIS Raster generally is considered suitable for small and medium size rasters[14]. In the application domain, ARRAY ALGEBRA concepts have had much impact on the design of the Open GeoSpatial Consortium (OGC) Web Coverage Processing Service (WCPS) geo service standard (OGC, 2008) and several related OGC standards. Baumann and Holsten (2010) work on extending the framework beyond arrays towards general meshes so as to allow retrieval on further spatiotemporal scientific data, such as Voronoi-type structures

[12] http://rasdaman.org/browser/manuals_and_examples/manuals/doc-guides/ql-guide.pdf?order=name *(accessed on 16 August 2016)*

[13] http://www.sai.msu.su/sal/H/2/O2.html *(accessed on 16 August 2016)*

[14] http://lists.osgeo.org/pipermail/postgis-users/2014-April/039024.html *(accessed on 21 August 2016)*

(adaptive grids can be handled already). They also investigate the seamless integration of arrays as first-class abstractions with standard SQL.

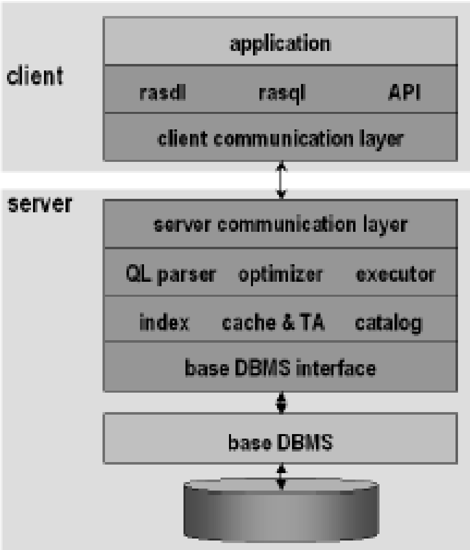

Figure 12. *RasDaMan* system architecture (dark grey) situated between application and base DBMS layers (light grey) (Baumann and Holsten, 2010)

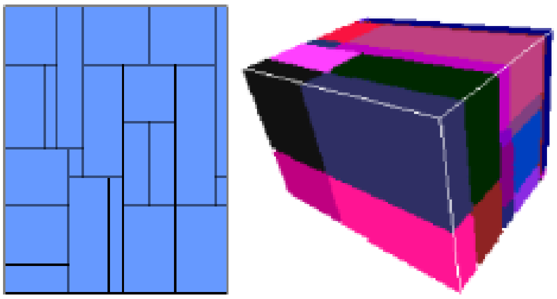

Figure 13. Sample 2-D and 3-D array tilings (Baumann and Holsten, 2010)

In scaling out on point clouds, that are characterized by large numbers of points (going into the billion, and growing), relational databases possibly have a say again. For example, MonetDB[15] shows promising handling of point clouds in its column-store architecture (Martinez et al, 2014).
The rasdaman system utilizes PostgreSQL as a backend to support point clouds through its WCS interface, thereby unifying grid and point cloud access (Baumann and Holsten, 2010).
ISO initiative is under way (Misev et al, 2014) to extend the simple, incomplete 1-D array support into fully-fledge n-D arrays with powerful image / signal processing and statistics operations (Baumann and Holsten, 2010).

15 https://www.monetdb.org/Home *(accessed on 18 August 2016)*

3.8.2 GPU use and massive parallel architectures for processing large-scale geospatial data

Modern Graphics Processing Units (GPUs) are now capable of general computing (Hennessy and Patterson, 2011). GPUs that are capable of general computing are facilitated with Software Development Toolkits (SDKs) provided by hardware vendors (Zhang et al, 2015c) see figure 14.

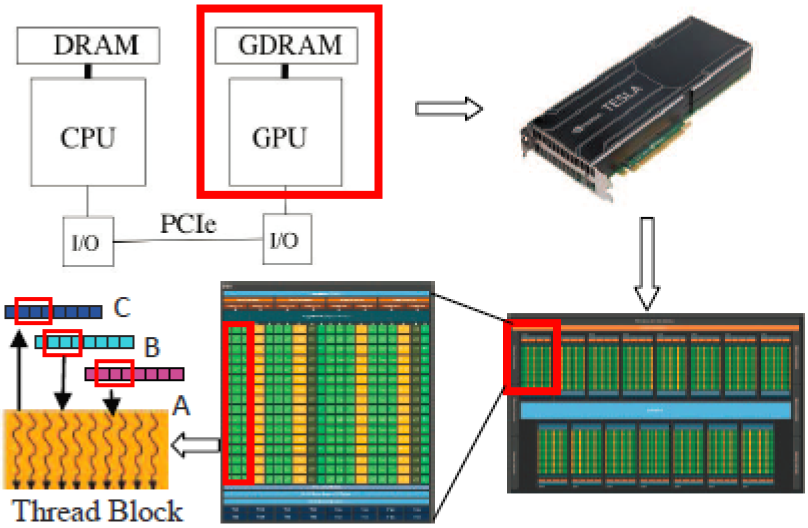

Figure 14. Illustration of GPU hardware Architecture (according to (Zhang et al, 2015c)

While geospatial data management techniques have been provided by both Spatial Databases and Geographical Information Systems (GIS), existing software is incapable of processing large-scale geospatial data for practical applications (Zhang et al, 2014). Quickly evolving processor, storage and networking technologies require new Big Data research to understand how new hardware impacts the performance of large-scale data processing.

In the past few years, the simplicity of the MapReduce computing model and its support in the open source Hadoop system have made it attractive to develop distributed geospatial computing techniques on top of MapReduce/Hadoop (Cary et al, 2009). The success of SpatialHadoop (Eldawy and Mokbel, 2013) and HadoopGIS (Aji et al, 2013) has demonstrated the effectiveness of MapReduce-based techniques for large-scale geospatial data management where parallelisms are typically identified at the spatial partition level which allows adapting traditional serial algorithms and implementations within a partition (Zhang et al, 2015c).

GPU-equipped computing nodes have much higher ratios between floating point computing power (in the order of floating point operations per second (flops), nowadays teraflops (Tflops) and fast growing) and network bandwidth (in the order of Gbps and remains stable) than regular computing nodes at which Hadoop-based systems are targeting. The gap makes efficient and scalable processing of large-scale data challenging, especially for geospatial data, whose processing is both data intensive and computing intensive (Zhang et al, 2015b).
Several techniques for processing large-scale geospatial data have been develop on both single computing nodes and clusters equipped with GPUs (You et al, 2015a; You et al, 2015b; Zhang et al, 2015a; Zhang et al, 2014).

Zhang et al (2015c) report their work on data parallel designs for several geospatial data processing techniques. By further integrating these GPU-based techniques with distributed computing tools, including Message Passing Interface[16] (MPI) library in the traditional High-Performance Computing (HPC) clusters and newer generation of Big Data systems (such as Impala[17] and Spark[18]) for Cloud computing, it is possible to scale the data parallel geospatial processing techniques to cluster computers with good scalability.

While being aware of the complexities in developing a spatial database on GPUs, Zhang et al (2015c) demonstrated the feasibility and efficiency of GPU-based geospatial processing, especially for large-scale data, developed modules for major geospatial data types and operations that can be directly applied to practical applications and developed a framework to integrate multiple GPU-based geospatial processing modules into an open system that can be shared by the community.

4. ANALYZE

4.1 Gap between what is available and what is needed 3D parcels

What are acceptable (valid) 3D cadastral object representations and how to create their 3D geometries (even the non-2-manifold geometries) are still challenges. The non-manifold 3D representations (self-touching in edge or node) are not well supported by current GIS, CAD, and DBMS software or by generic ISO standards such as ISO 19107 (Van Oosterom, 2013).

How to create and maintain valid 3D parcels is still a challenge in practice Ying et al (2015). At least three aspects should be clearly developed in order to manage the 3D parcels correctly (Ying et al, 2015): (1) precise geometric models that describe the shapes and geographic locations of various 3D parcels, based on the flat faces; (2) volumetric or solid models that indicate all its boundary faces with orientation to present the corresponding 3D parcel objects; and (3) the topological relationships that encode all the information about the adjacencies among 3D parcels shared common faces/edges to keep the consistency of the objects' geometries and support spatial query and management.

4.1.1 3D topology

As previously elaborated, a suitable 3D topology model for 3D cadastre seems to be an approach based on a Tetrahedral Network (TEN), proposed by Penninga and van Oosterom (2008): the "topological structure to organize tetrahedrons". However, the TEN model need to be synchronized, described in a new spatial profile, with LADM specifications. As mentioned in Zulkifli et al (2015a), the future work is to develop a conceptual model of the TEN based on LADM standard. Then, the proposed conceptual models (i.e. 2D and 3D topology) should be translated into physical model to develop a prototype cadastral registration.

[16] http://en.wikipedia.org/wiki/Message Passing Interface *(accessed on 22 August 2016)*
[17] http://impala.io *(accessed on 22 August 2016)*
[18] http://spark.apache.org *(accessed on 22 August 2016)*

A full topological model for the 3D cadastre, land planning and management is needed for the following reasons: (1) to utilize the surveying boundaries to generate the 3D cadastral objects (the term "volumetric model" is used geometrically and topologically); (2) to represent the 3D volumetric objects with high quality, and consistent topology without intersection; and (3) for the rapid topological queries necessary for real-time user interaction and management (Ying et al, 2015).

Another important aspect is the development of (spatial) indexes for topological models. Last but not least, operations on topological models, including conversion to geometric models, are important (Breunig and Zlatanova, 2011).

4.1.2 Validation of 3D solids

Ledoux (2014) mentions several possible extensions of validation of 3D solids. For the modelling of 3D buildings, the semantics information can be used. For example, if for instance one surface is labelled as the roof of the building, then an extra validation rule (over the geometry) would be to ensure that the roof is located "above" the surface labelled as the ground floor. Furthermore, the automatic repair of invalid solids could be considered.

4.1.3 Interrelation CityGML – LADM ADE

An important trend which can be observed is the use of building information models/ construction plans to update the cadastral database, as done in Costa Rica (Van Oosterom et al, 2014).

Further research will aim at investigating other possible alternatives of combining the LADM and CityGML standards (Góźdź et al, 2014) that is:

- embedding the selected CityGML classes into (broader) LADM framework,
- introducing a link between both domain models (in SDI setting) using references between object instances.

Unfortunately, it is not possible to indicate classes corresponding to LA_Party, LA_RRR and LA_BAUnit in CityGML. Due to that fact there are many problems during transformation of the model from conceptual to technical level. The results of this investigation entitle to make a statement that introducing the semantic representation for land administration within CityGML will be advisable. That issue is included in the list of work packages that define the scope of next version of CityGML (Góźdź et al, 2014).

4.1.4 Ontology

For any developments that require spatial data, often the fusion of diverse spatial datasets is required. This becomes non trivial when semantic heterogeneity occurs between schemas like CityGML and LandXML. Soon et al (2014) introduced a semantics-based fusion framework to integrate CityGML and LandXML using the LADM OWL ontology previously developed. The LADM OWL ontology is augmented with concepts of Physical Space Building Unit and Physical Space Utility Network, which are related to Legal Space Building Unit and Legal Space Utility Network respectively through a new relation *hasLegalSpace*. Soon et al (2014) looked into how the extended LADM OWL ontology is linked with CityGML schema and ePlan model through the equivalent Class relation. Syntactically, the equivalent Class relation can be realized using the ExternalReference and DocFileRef elements of CityGML and LandXML respectively. The framework ultimately attempts to integrate not only the semantic models inherent in the schemas but also the geometries from CityGML and LandXML. Through this semantics based fusion, it is expected that a computer system will be able to do reasoning and inference in the OWL ontology. The computer system will also be able to retrieve the geometries of building's legal space or physical space, or both, through the ExternalReference and DocFileRef elements. The intention of the framework is to utilize the best of all worlds (i.e. CityGML, LandXML and OWL) without affecting the existing schemas, which have been comprehensively developed for different applications (Soon et al, 2014).

4.1.5 Point clouds and TINs

Van Oosterom et al (2015) state that at least two closely related level of standardization must be considered: (a) Database Structure Query Language (SQL) extension for point clouds, and (b) Web Point Cloud Services (WPCS) for progressive transfer based on multi-scale or vario-scale LoD.

Janečka and Kára (2012) suggest to extend the point cloud and TIN related data structures available in production spatial databases to enable storage of additional non-spatial attributes (semantic) related e.g. to the particular point (or set of points). Such information can be then used, for example, during the update of the stored 3D geometries directly inside the spatial database.

4.1.6 Usage of GPU clusters for processing geospatial data

Balancing latency and throughput has profound implications in Big Data research. While traditional parallel and distributed databases mostly targeted at reducing data processing latency for moderately sized datasets, Big Data systems need to take ownership costs and energy consumption into consideration. Using large quantities of small processors to achieve similar throughputs while reducing energy footprint is becoming an increasingly important topic in Big Data research (Zhang et al, 2015b). Motivated by the increasing gap between the computing power of GPU-equipped clusters and network bandwidth and disk I/O throughput, Zhan et al, (2015b) proposed a low-cost prototype research cluster made of NVidia TK1 SoC[19] boards that can be interconnected with standard 1Gbps network to facilitate Big Data research. They evaluate the performance of the tiny GPU cluster for spatial join query processing on large-scale geospatial data. Experiments on point-in-polygon test based spatial join using two real world applications with tens to hundreds of millions of points and tens of thousands of polygons have demonstrated the efficiency of the solution when compared with SpatialSpark. The future work should incorporate not only including processors, but also memory, disk and network components. Furthermore, the performance of GPU cluster should be evaluated using more real world geospatial datasets and applications, e.g., distance and nearest neighbour based spatial joins (Zhan et al, 2015b).
In the age of Big Data it is not sufficient any longer that each research domain pursues its own ways of finding solutions, often reinventing the wheel or, conversely, inventing inadequate wheels. Specifically, the geoinformatics domain and core computer science domains like databases, Web services, programming languages, and supercomputing, share challenges seen from different angles. It is not too infrequent that similar ideas appear in different fields. For

[19] http://www.nvidia.com/object/jetson-tk1-embedded-dev-kit.html *(accessed on 21 August 2016)*

example, array databases offer declarative query languages on large n-D arrays which internally are partitioned for efficient access to subsets. SciHadoop is an approach independent from databases where an array-tuned query language is put on top of Hadoop. Data formats like TIFF and NetCDF also support the concept of array partitioning. It is worthwhile, therefore, to extend this small, focused survey into a larger one incorporating more domains and also implementation aspects. Fostering exchange, therefore, seems promising (Baumann, 2014).

4.1.7 Open data and smart cities initiatives

One of the new areas is the creation of the 2D and 3D registries in the context of open data and smart cities initiatives that are aimed at providing a platform for city data. The inclusion of geospatial and building data in this context is paramount and was highlighted by the British Standard Institutions City Data Survey Report[20].

4.1.8 Compression and transfer of spatial data

3D models generally result in large data sets, which require special techniques for rapid visualisation and navigation (Breunig and Zlatanova, 2011). As the speed of geodata collection is still increasing, Janečka and Váša (2016) suggest that also the need for the effective geodata compression will be essential, for example to deliver the data to the final user/application via internet. They proposed a compression approach for geographical objects at various level of detail. For complex geographical objects, after the compression the amount of data is even lower than 4% of the original file size.

4.2 3D LADM example implementation/ prototypes

Góźdź and Pachelski (2014) introduced the 3D LADM based country profile, see fig. 15. They mention the fact, that Polish cadastral system meets serious complications with providing information about the legal status of properties in case of 3D complex situations, when different property units are located above each other or constructed in more complex structures, i.e. interlocking one another. For that reason, the presented Spatial Package is extended to new classes, e.g. PL_3DParcel.

In the last few years several prototypes of 3D LADM based country profiles have been developed, for example: Trinidad and Tobago (Griffith-Charles and Edwards, 2014), Malaysia (Zulkifli et al, 2014; Zulkifli et al, 2015b), Israel (Felus et al, 2014) or Turkey (Alkan and Polat, 2016).

---

[20] http://www.bsigroup.com/Documents/BSI_City Data Report_Singles FINAL.pdf *(accessed on 23 August 2016)*

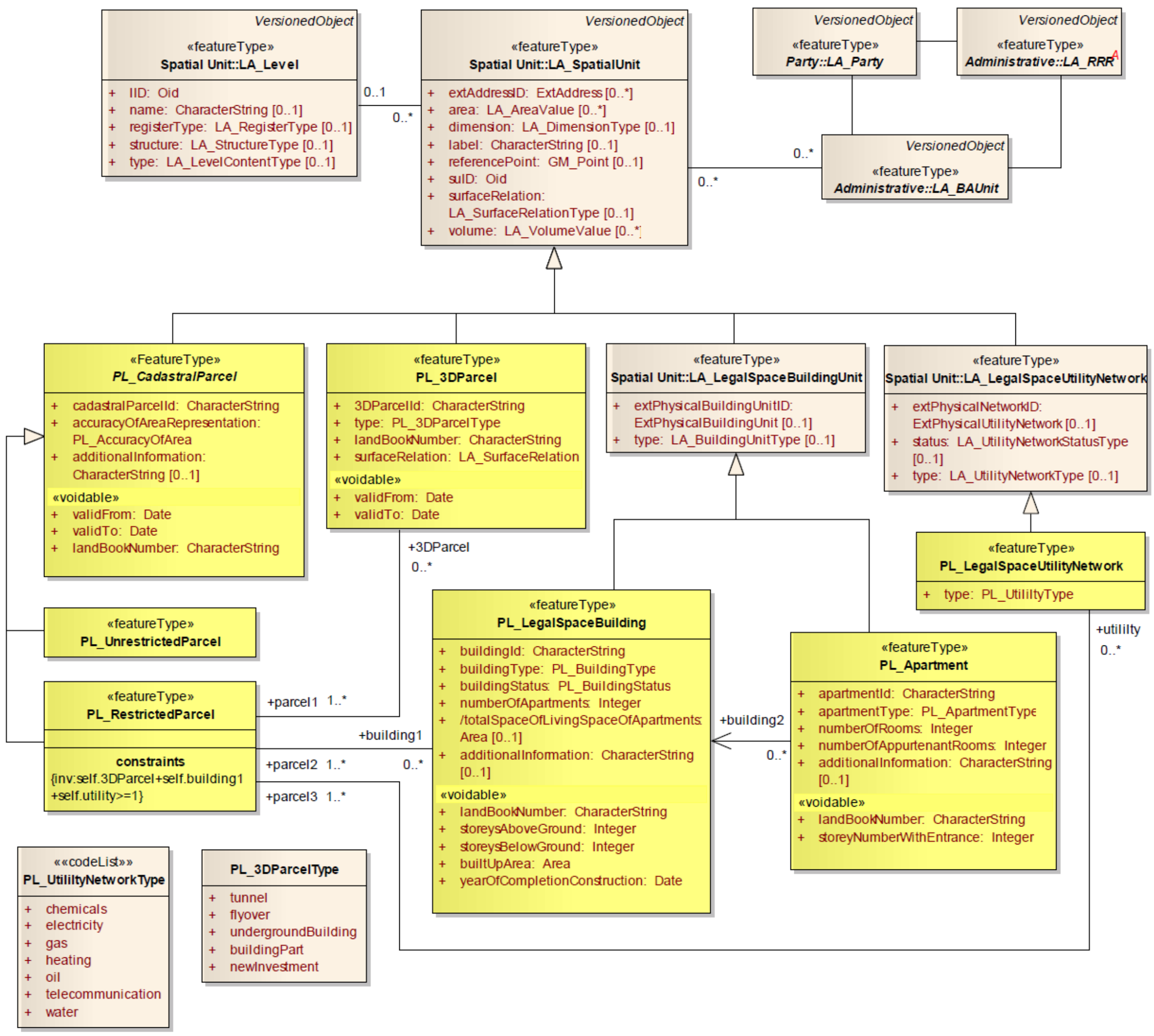

Figure 15. Spatial package of the Polish 3D LADM country profile (Góźdź and Pachelski, 2014)

5. CONCLUSION

The paper has explored 3D data management from multiple perspective. The focus of the data management issue in this paper has not been restricted to 3D Cadastre, but rather to a broader 3D GIS to ensure that all capabilities and issues that exist in different related fields will assist and affect in the data management of 3D Cadastral data.

In functional requirements for 3D cadastral data management, the categorisation of 3D parcels at an increasing level of complexity is discussed. This lead to a discussion on options for storing 3D cadastral data in an existing 2D cadastral database that traditionally exists in current jurisdictions. The issues related to adding the time dimension in a 4D cadastre from a database point of view was discussed. A discussion of 3D geometric models based on current research on standards, solid geometry and LADM schema, which in turn led to 3D topological models. The LADM provides a data model that recognises and describes the relationships of a

3D spatial unit to other levels of encodings. BIMS are a good source of 3D cadastral data and has already been used by many jurisdictions. The link between the various geometrical and semantic aspects of BIM vs other data sources can cause differences and issues when data are to be integrated. The current standards such as ISO LADM, GML, CityGML etc. and their inter-relationship were then discussed. In current DBMS 3D capabilities, current software and methods of storing 3D data were discussed which led to a discussion on recent developments of spatial databases and the physical capacity of existing hardware to cope with the large volume of 3D data. The analysis of a gap between what is available and what is needed was based on 3D geometry and topology, validation, standards and ontology, data and hardware, 3D data use and transfer and implementation of a 3D LADM prototype.

3D data management capability and technology exist, however these have not been transferrable to 3D cadastre. The problem is, established cadastre are traditionally 2D and the nature of the cadastral data does not easily extend itself to 3D modelling. While 3D GIS data may be easy to extrude to create a 3D visualisation, because 3D cadastre deals with absolute ownership of 3D spaces it becomes much more complex to convert a 2D database to a 3D operational data structure. The extrusion of 2D to 3D might still be a feasible solution for a cadastre if the purpose is just visualisation, however, if the purpose is to define ownership of defined space, information about the adjoining 3D spaces, checks to determine encroachment or slivers among the spaces, then a simple extrude does not fulfil the requirements.

Future direction

Important directions for 3D SDBMS research include the integration of 2D and 3D data models and development of dimension-independent topological and geometric models. These data models should be uniformly usable for both 2D and 3D worlds. The challenges for future research include spatial data integration, new user interfaces for SDBMSs (augmented reality), development of 3D/4D geo-information systems, and geo-sensor databases. A database should be able to efficiently manage un-processed raw data such a large point clouds. Persistent storage of raw data will support the 3D modelling process, allowing re-use of source data while constructing a 3D model (Breunig and Zlatanova, 2011).

REFERENCES

Aien, A., Rajabifard, A., Kalantari, M., Williamson, I. and Shojaei, D. (2014). Development of XML Schemas for Implementation of a 3D Cadastral Data Model. In: Proceedings of the 4th International Workshop on 3D Cadastres. 9-11 November 2014, Dubai, United Arab Emirates. ISBN 978-87-92853-28-8.

Aji, A., Wang, F., Vo, H., Lee, R., Liu, Q., Zhang, X. and Saltz, J. (2013). Hadoop-gis: A high performance spatial data warehousing system over mapreduce. In VLDB, 6(11), pp. 1009–1020.

Alkan, M. and Polat, Z.A. (2016). Design and development of LADM-based infrastructure for Turkey, Survey Review, doi: 10.1080/00396265.2016.1180777.

Baumann, P. (2014). Are Databases of Any Use in Modern Geo Services? In: FOSS4G-Europe, Bremen, Germany.

Baumann, P. and Holsten, S. (2010). A Comparative Analysis of Array Models for Databases. In: Database Theory and Application, Bio-Science and Bio-Technology. Volume 258 of the series Communications in Computer and Information Science, Springer, pp. 80-89. doi: 10.1007/978-3-642-27157-1_9.

Baumann, P. and Merticariu, V. (2015). On the Efficient Evaluation of Array Joins. IEEE International Conference on Big Data. Santa Clara, CA. doi: 10.1109/BigData.2015.7363986.

Behnam, A., Kalantari, M., Rajabifard, A., Ho, S. and Ngo, T. (2016). Building Information Modelling for High-rise Land Administration. Transactions in GIS. doi: 10.1111/tgis.12199

Bendersky, E. (2016). Memory layout of-multi-dimensional arrays. Available online: http://eli.thegreenplace.net/2015/memory-layout-of-multi-dimensional-arrays/ (accessed on 16 Aug 2016).

Biljecki, F., Ledoux, H. and Stoter, J. (2016). An improved LOD specification for 3D building models. Computers, Environments and Urban Systems, 59:25-37. doi:10.1016/j.compenvurbsys.2016.04.005.

Billen R., Zlatanova, S., Mathonet, P. and Boniver, F. (2002). The Dimensional Model: a framework to distinguish spatial relationships, in: Advances in Spatial Data handling, D. Richardson, P. van Oosterom (Eds.), Springer, pp. 285-298.

Borrmann, A. and Rank, E. (2009). Topological analysis of 3D building models using a spatial query language. Advanced Engineering Informatics 23(4), pp. 370–385.

Boskovic, D., Ristić, A., Govedarica, M., Pržulj, D. (2010). Ontology Development for Land Administration. Proceedings of 8th International Symposium on Intelligent Systems and Informatics (SISY), 437-442. IEEE.

Boss, H., Å. and Streilein, A. (2014). 3D Data Management – Relevance for a 3D Cadastre Position Paper 3. In: Proceedings of the 4th International Workshop on 3D Cadastres. 9-11 November 2014, Dubai, United Arab Emirates. ISBN 978-87-92853-28-8.

Breunig, M. and Zlatanova, S. (2011). 3D geo-database research: Retrospective and future directions. Computers & Geosciences 37, pp. 791-803. doi:10.1016/j.cageo.2010.04.016

Brugman, B., Tijssen, T. and van Oosterom, P. (2011). Validating a 3D topological structure of a 3D space partition. In: Advancing Geoinformation Science for a ChangingWorld, Springer, pp. 359–378.

Cary, A., Sun, Z., Hristidis, V., Rishe, N. (2009). Experiences on processing spatial data with mapreduce. In SSDBM, pp. 302–319.

Ding, Y., C. Wu et al (2016). Construction of geometric model and topology for 3D cadastre – Case study in Taizhou, Jiangsu. FIG working week 2016. Christchurch, New Zealand.

Ellul, C. (2007). Functionality and Performance – Two Important Considerations when implementing Topology in 3D. Ph.D. Thesis. University of London.

Egenhofer, M.J. (1995). Topological relations in 3-D. Technical report, National Center for Geographic Information and Analysis and Department of Spatial Information Science and Engineering Department of Computer Science university of Maine.

Eldawy, A. and Mokbel, M. (2013). A demonstration of spatialhadoop: an efficient mapreduce framework for spatial data. In VLDB, 6(2), pp. 1230–1233.

Felus, Y, Barzani, S., Caine, A., Blumkine, N. and van Oosterom, P. (2014). Steps towards 3D Cadastre and ISO 19152 (LADM) in Israel. In: Proceedings of the 4th International Workshop on 3D Cadastres. 9-11 November 2014, Dubai, United Arab Emirates. ISBN 978-87-92853-28-8.

Furtado, P. and Baumann, P. (1999). Storage of multidimensional arrays based on arbitrary tiling. In Proceedings of the 15th International Conference on Data Engineering, pages 328–336. IEEE Computer Society.

Ghawana, T. and Zlatanova, S. (2010). Data consistency checks for building a 3D model: a case study of Technical University, Delft Campus, The Netherlands. Geospatial World (4). ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume II-2/W1, ISPRS 8th 3DGeoInfo Conference & WG II/2 Workshop, 27 – 29 November 2013, Istanbul, Turkey.

Griffith-Charles, Ch. and Edwards, E. (2014). Proposal for Taking the Current Cadastre to a 3D, LADM Based Cadastre in Trinidad and Tobago. In: Proceedings of the 4th International Workshop on 3D Cadastres. 9-11 November 2014, Dubai, United Arab Emirates. ISBN 978-87-92853-28-8.

Góźdź, K. and Pachelski, W. (2014). The LADM as a core for developing three-dimensional cadastral data model for Poland. The 14th International Multidisciplinary Scientific GeoConference SGEM 2014. Albena, Bulgaria.

Góźdź, K., Pachelski, W., van Oosterom, P. and Coors, V. (2014). The Possibilities of Using CityGML for 3D Representation of Buildings in the Cadastre. In: Proceedings of the 4th International Workshop on 3D Cadastres. 9-11 November 2014, Dubai, United Arab Emirates, pp. 339-362. ISBN 978-87-92853-28-8.

Guo, R., Luo, F., Zhao, Z., He, B., Li, L., Luo, P. and Ying, S. (2014). The applications and Practices of 3D Cadastre in Shenzhen. In: Proceedings of the 4th International Workshop on 3D Cadastres. 9-11 November 2014, Dubai, United Arab Emirates, pp. 339-362. ISBN 978-87-92853-28-8.

Gutierrez, A., G. and Baumann, P. (2007). Modeling fundamental geo-raster operations with array algebra. In Workshops Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA, pages 607–612. IEEE Computer Society.

Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. In: Proceedings of ACM SIGMOD, International Conference on Management of Data, Boston, MA, pp. 47-57.

Hennessy, J.L. and Patterson, D. A. (2011). Computer Architecture: A Quantitative Approach, 5th ed. Morgan Kaufmann.

Herring, J. (2001). Topic 1 Feature Geometry (Same as ISO 19107 Spatial Schema), available at http://www.iso.org.

ISO (2013). ISO 16739, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.

ISO (2012). ISO 19152, Geographic information – Land Administration Domain Model (LADM), ed. 1. ISO, Geneva, Switzerland.

ISO (2007). ISO 19136, Geographic information – Geography Markup Language (GML).

ISO/TC 211 (2009). Standards Guide. Available online/accessed on 19 August 2016 http://www.isotc211.org/Outreach/ISO_TC_211_Standards_Guide.pdf.

Janečka, K. and Kára, M. (2012). Advanced Data Structures for Surface Storage. In: Proceedings of GIS Ostrava 2012 – Surface models for geosciences. VŠB-TUO, Ostrava. ISBN 978-80-248-2667-7.

Janečka, K. and Váša, L. (2016). Compression of 3D geographical objects at various level of detail. In: The Rise of Big Spatial Data. Lecture Notes in Geoinformation and Cartography. Springer. 978-3-319-45122-0. (*in print*).

Kazar, B., M, Kothuri, R., van Oosterom, P. and Ravada, S. (2008). On valid and invalid three-dimensional geometries. In Van Oosterom P, Zlatanova S, Penninga F, and Fendel E (eds) Advances in 3D GeoInformation Systems. Berlin, Springer: 19–46.

Khuan, Ch., T., Rahman, A., A. and Zlatanova, S. (2008). 3D Solids and Their Management in DBMS. In: Advances in 3D Geoinformation Systems. Lecture Notes in Geoinformation and Cartography, pp. 279-311. 10.1007/978-3-540-72135-2_16.

Laine, S. (2013). A topological approach to voxelization. Comput. Graph. Forum 32 (4), pp. 77–86.

Ledoux, H. (2014). On the validation of solids represented with the international standards for geographic information. Computer-Aided Civil and Infrastructure Engineering, 28(9):693–706. doi: http://dx.doi.org/10.1111/mice.12043.

Ledoux, H. and Meijers, M. (2013). A star-based data structure to store efficiently 3D topography in a database. Geo-spatial Information Science, 16(4):256–266.

Ledoux, H. and Meijers, M. (2009). Extruding building footprints to create topologically consistent 3d city models. Urban and Regional Data Management, UDMS Annuals pp. 39–48.

Lee, J. and Zlatanova, S. (2008). A 3D data model and topological analyses for emergency response in urban areas. In: Geospatial information technology for emergency response, Publisher: Taylor and Francis, pp.143-168.

Martinez-Rubi, O., Kersten, M.L., Goncalves, R. and Ivanova, M. (2014). A Column-Store Meets the Point Clouds. FOSS4G-Europe Academic Track.

Misev, D., Baumann, P. (2014). Extending the SQL Array Concept to Support Scientific Analytics. In: Proceedings of the 26th International Conference on Scientific and Statistical Database Management. ISBN: 978-1-4503-2722-0 doi:10.1145/2618243.2618255

Nourian, P., Goncalves, R., Zlatanova, S., Ohori, A.K. and Vo, V.A. (2016). Voxelization Algorithms for Geospatial Applications: Computational methods for voxelating spatial datasets of 3D city models containing 3D surface, curve and point data models. MethodsX 3 pp. 69-86. http://dx.doi.org/10.1016/j.mex.2016.01.001.

OGC (2016). OGC® Land and Infrastructure Conceptual Model Standard (LandInfra), Open Geospatial Consortium.

OGC (2011). OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture.
OGC (2010). OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option.

OGC (2008). Web Coverage Processing Service (WCPS) Implementation Specification. Number 08-068. Open Geospatial Consortium, 1.0.0 edition. Editor P. Baumann.

Penninga, F. (2008). 3D Topography A Simplicial Complex-based Solution in a Spatial DBMS. Ph.D. thesis, TU Delft, the Netherlands.

Penninga, F. (2005). 3D topographic data modelling: why rigidity is preferable to pragmatism. In: Spatial Information Theory, Cosit'05, Vol. 3693 of Lecture Notes on Computer Science, Springer. pp 409-425.

Penninga, F. and van Oosterom, P.J.M. (2008). A Simplicial Complex-Based DBMS Approach to 3D Topographic Data Modelling. International Journal of Geographic Information Science, 22, pp. 751-779.

Pilouk, M. (1996). Integrated Modelling for 3D GIS, PhD thesis, ITC Enschede, the Netherlands.

Reed, C. (2006). Data integration and interoperability: OGC standards for geo-information, in: Zlatanova, S & Prosperi, D. (eds). 3D large-scale data integrations: challenges and opportunities, Boca Raton, Taylor & Francis (CRCpress) pp. 163-174.

Roschlaub, R. and Batscheider, J. (2016). An INSPIRE-conform 3D model building model of Bavaria using cadastre information, Lidar and image matching. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B4, 2016, pp.747-754.

Rönsdorf, C., Wilson, D. and Stoter, J. (2014). Integration of Land Administration Domain Model with CityGML for 3D Cadastre. In: Proceedings of the 4th International Workshop on 3D Cadastres. 9-11 November 2014, Dubai, United Arab Emirates. ISBN 978-87-92853-28-8.

Scarponcini, P. (2013). InfraGML Proposal (13-121), OGC Land and Infrastructure DWG/SWG.

Seifert, M., Gruber, U. and Riecken, J. (2016). Multidimensional Cadastral System in Germany. In: Proceedings of the FIG Working Week 2016. Christchurch, New Zealand. ISBN 978-87-92853-52-3.

Sladić, D., Govedarica, M., Pržulj, D., Radulović, A. and Jovanović, D. (2013). Ontology for Real Estate Cadastre. In Survey Review. 45 (332): 357-371. Maney Publishing.

Soon, K. H. (2013). Representing Roles in Formalizing Domain Ontology for Land Administration. Proceedings of 5th Land Administration Domain Model Workshop. Kuala Lumpur, Malaysia. 24-25 September 2013. FIG.
Soon, K. H., Thompson, R. and Khoo, V. (2014). Semantics-based Fusion for CityGML and 3D LandXML. In: Proceedings of the 4th International Workshop on 3D Cadastres. 9-11 November 2014, Dubai, United Arab Emirates. ISBN 978-87-92853-28-8.

Stoter, J., van den Brink, L., Vosselman, G., Goos, J., Zlatanova, S., Verbree, E., Klooster, R., van Berlo, L., Vestjens, G., Reuvers, M. and Thorn, S. (2011). A Generic Approach for 3D SDI in the Netherlands. Proceedings of the Joint ISPRS Workshop on 3D City Modelling & Applications and the 6th 3D GeoInfo Conference. Wuhan, China.

Stoter, J., Vallet, B., Lithen, T., Pla, M., Wozniak, P., Kellenberger, T., Streilein, A., Ilves, R. and Ledoux, H. (2016). State-of-the-art of 3D national mapping in 2016. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B4. doi:10.5194/isprsarchives-XLI-B4-653-2016.

Streilein, A. (2011). Position Paper 3; 2nd international Workshop on 3D cadastre, 2011, Delft. Available online: http://3dcadastres2011.nl/documents/036_III_PP.pdf (accessed on 10 August 2016) .

Thompson, R. (2015). A model for the creation and progressive improvement of a digital cadastral data base. Land Use Policy 49, pp. 565-576. http://dx.doi.org/10.1016/j.landusepol.2014.12.016

Thompson, R. and van Oosterom, P. (2012). Modelling and validation of 3D cadastral objects. Urban and Regional Data Management. S. Zlatanova, H. Ledoux, E. Fendel and M. Rumor. Leiden, Taylor & Francis. UDMS Annual 2011.

Thompson, R., van Oosterom, P., Karki, S. NS Cowie, B. (2015). A Taxonomy of Spatial Units in a Mixed 2D and 3D Cadastral Database. FIG Working Week 2015 - From the Wisdom of the Ages to the Challenges of the Modern World. Sofia, Bulgaria.

Thompson, R., Van Oosterom, P., Soon, K.H. and, Priebbenow, R. (2016). A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage. FIG Working Week 2016. Christchurch, New Zealand.

van den Brink, L., Stoter, J. and Zlatanova, S. (2012). Establishing A National Standard for 3D Topographic Data Compliant to CityGML. International Journal of Geographical Information Science. 27(1): 92-113. Taylor & Francis.

van Oosterom, P. (2013). Research and development in 3D cadastres. Computers, Environment and Urban Systems 40: pp. 1–6.

van Oosterom, P., Lemmen, Ch. (2015). The Land Administration Domain Model (LADM): Motivation, standardisation, application and further development. Land Use Policy 49, pp. 527-534. http://dx.doi.org/10.1016/j.landusepol.2015.09.032

van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S., Tijssen, T., Kodde, M. and Gonçalves, R. (2015). Massive point cloud data management: Design, implementation and execution of a point cloud benchmark. Computers & Graphics. Volume 49, pp. 92-125. http://dx.doi.org/10.1016/j.cag.2015.01.007

van Oosterom, P., Stoter, J., Ploeger, H., Lemmen, Ch., Thompson, R. and Karki, S. (2014). Initial Analysis of the Second FIG 3D Cadastres Questionnaire: Status in 2014 and Expectations for 2018. In: Proceedings of the 4th International Workshop on 3D Cadastres. 9-11 November 2014, Dubai, United Arab Emirates. ISBN 978-87-92853-28-8.

Verbree, E. and Si, H. (2008). Validation and storage of polyhedra through constrained Delaunay tetrahedralization. In Cova T J, Miller H J, Beard K, Frank A U, and Goodchild M F (eds) GIScience 2008: Proceedings of the Fifth International Conference. Berlin, Springer-Verlag: 354–69

Ying, S., Guo, R., Li, L., van Oosterom, P., Ledoux, H. and Stoter, J. (2011). Design and Development of a 3D Cadastral System Prototype based on the LADM and 3D Topology. In 2nd International Workshop on 3D Cadastres. Delft, the Netherlands.

Ying, S., Guo, R., Li, L., van Oosterom, P. and Stoter, J. (2015). Construction of 3D Volumetric Objects for a 3D Cadastral System. Transactions in GIS. Vol. 19 Issue 5, pp. 758-779. 10.1111/tgis.12129.

Ying, S., Jin, F., Guo, R., Li, L., Yang, J. and Zhou, Y. (2014). The Conversion from CityGML to 3D Property Units. In: Proceedings of the 4th International Workshop on 3D Cadastres. 9-11 November 2014, Dubai, United Arab Emirates. ISBN 978-87-92853-28-8.

You, S., Zhang, J., Gruenwald, L. (2015a). Scalable and Efficient Spatial Data Management on Multi-Core CPU and GPU Clusters: A Preliminary Implementation based on Impala. In: Proceedings of International Workshop on Big Data Management on Emerging Hardware (HardBD'15), Seoul, Korea.

You, S., Zhang, J. and Gruenwald, L. (2015b). Large-Scale Spatial Join Query Processing in Cloud. In: Proceedings of International Workshop on Cloud Data Management (CloudDM'15), Seoul, Korea.

Zhang, J., You, S. and, Gruenwald, L. (2015a). A Lightweight Distributed Execution Engine for Large-Scale Spatial Join Query Processing. Technical report. Available online: http://wwwcs.engr.ccny.cuny.edu/~jzhang/papers/lde_spatial_tr.pdf (accessed on 15 August 2016).

Zhang, J., You, S. and Gruenwald, L. (2015b). Tiny GPU Cluster for Big Spatial Data: A Preliminary Performance Evaluation. In: 2015 IEEE 35th International Conference on Distributed Computing Systems Workshops.

Zhang, J., You, S. and Gruenwald, L. (2015c). Large-Scale Spatial Data Processing on GPUs and GPU-Accelerated Clusters. SIGSPATIAL Special. Vol. 6 Issue 3, pp. 27-34. doi: 10.1145/2766196.2766201.

Zhang, J., You, S. and Gruenwald, L. (2014). Parallel Online Spatial and Temporal Aggregations on Multi-core CPUs and Many-Core GPUs. Information Systems, vol. 44, p. 134–154.

Zulkifli, N., A., Abdul Rahman, A., Jamil, H., Teng, C.H., Tan, L., C., Looi, K.S., Chan, K.L. and van Oosterom, P. (2014). Development of a prototype for the assessment of the Malaysian LADM country profile. In: Proceedings of FIG Congress 2014, Malaysia.

Zulkifli, N. A., Abdul Rahman, A. and van Oosterom, P. (2015a). An overview of 3D topology for LADM-based objects. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-2/W4, 2015. doi: 10.5194/isprsarchives-XL-2-W4-71-2015.

Zulkifli, N.A., Abdul Rahman, A., van Oosterom, P., Choon, T.L., Jamil, H., Hua, T.Ch., Seng, L.K. and Lim, Ch.K. (2015b). The importance of Malaysian Land Administration Domain Model country profile in land policy. Land Use Policy 49, pp. 649-659. http://dx.doi.org/10.1016/j.landusepol.2015.07.015.

ACKNOWLEDGEMENTS

The first author of the publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports.

BIOGRAPHICAL NOTES

Karel Janečka Ph.D. (2009) Geomatics, University of West Bohemia in Pilsen. He had been working as a database programmer at the Czech Office for Surveying, Mapping and Cadastre in Section of cadastral central database between 2006 and 2008. Since 2009 he is a researcher at University of West Bohemia, Department of Geomatics. His research activities are spatial data infrastructures (SDI), geographical information systems (GIS), spatial databases, spatial data mining, and 3D cadastre. He has experience with coordination of several EU projects and is also reviewer of several international scientific journals. Since 2012 he is the president of the Czech Association for Geoinformation and member of National Mirror Committee 122 Geographic information/Geomatics.

Sudarshan Karki is the manager of the Digital Cadastral Database (DCDB) in the Department of Natural Resource and Mines, Queensland Government, Australia. He is a surveyor and has completed a Master of Spatial Science by Research at the University of Southern Queensland (USQ) in 2013 and a professional Master's Degree in Geo-informatics from ITC, The Netherlands in 2003. He has continued his research interest in 3D cadastre and is currently undertaking his PhD research at USQ.

CONTACTS

Karel Janečka
University of West Bohemia
Technická 8
Pilsen
CZECH REPUBLIC
Phone: + 420 607982581
E-mail: kjanecka@kgm.zcu.cz
Website: http://gis.zcu.cz

Sudarshan Karki
Queensland Government, Department of Natural Resources and Mines
Landcentre, Cnr Main and Vulture Streets,
Woolloongabba, Brisbane, Queensland 4102
AUSTRALIA
Tel.: +61 7 3896 3110
E-mail: Sudarshan.Karki@dnrm.qld.gov.au
Website: http://www.dnrm.qld.gov.au/