

A Conceptual Model Supporting a Range of 3D Parcel Representations through all Stages: Data Capture, Transfer and Storage

Rodney James THOMPSON, Australia, Peter VAN OOSTEROM, The Netherlands, Kean Huat SOON, Singapore and Russell PRIEBBENOW, Australia

Key words: 3D Spatial Unit, LandXML, LADM, Cadastral Database, 3D Cadastre

SUMMARY

The complexity of volumetric spatial units is highly variable, but in practice the vast majority are composed of simple components. Even where the spatial units are apparently complex, most boundary faces of real cadastral spatial units are either vertical or horizontal, and a significant majority of spatial units are composed of a simple polyhedral slice - a prism resulting from the vertical extrusion of a 2D polygon. A categorization of volumetric spatial units recognizing this aspect was proposed earlier (Thompson, van Oosterom et al. 2015). Further, the ISO19152 - Land Administration Domain Model (LADM) recognises that even conventional 2D spatial units are, in the real world, prismatic columns of space defined by vertical faces, with no explicitly stated upper or lower boundary faces. These are known as “Face Strings” (LA_BoundaryFaceString).

This paper explores an integrated method of defining 3D spatial units, whereby the “footprint” of the spatial unit is represented as a LA_BoundaryFaceString, associated with a (possibly empty) set of more general faces (each stored as a LA_BoundaryFace). This representation fits within the LADM (ISO-TC211 2012) and is suitable for practical encoding formats. It is shown that this provides a complete encoding scheme - from simple 2D land parcels, to the most complex volumes. It is further shown that this scheme is more parsimonious in storage than the conventional polyhedron (stored as a collection of faces) for the vast majority of real cases (using real world examples).

The approach is discussed from three perspectives representing the various stages of a volumetric parcel: as a technique for data capture and validation of mixed 2D and 3D cadastral information; as an encoding for the transport of such information (for example in LandXML or InfraGML); and as a scheme for the database storage of such information.

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016

A Conceptual Model Supporting a Range of 3D Parcel Representations through all Stages: Data Capture, Transfer and Storage

Rodney James THOMPSON, Australia, Peter VAN OOSTEROM, The Netherlands, Kean Huat SOON, Singapore and Russell PRIEBBENOW, Australia

1. INTRODUCTION

This paper presents a conceptual model based on the observation that the majority of 3D parcels are relatively simple (with vertical and horizontal faces), and mainly focusses on what has been called the “Explicit Mode” (Guo, Yu et al. 2012 Page 126) of encoding cadastral parcels - where the definition of each spatial unit contains the full geometry of that spatial unit. Likewise, a face, footprint, or any other component of a spatial unit is also self-contained, however the approach is not limited to this mode. There is a discussion of a topological structure approach in sections 5.2 and 8.1.

An initial categorization of 3D Parcels was given in (Thompson, van Oosterom 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 valued² .
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 problem of mixing 2D land parcel definitions with the range of 3D parcels in a corporate database (DCDB) 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:

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.

¹ The surface should be single valued, that it can be described with $z = f(x,y)$. The most common case is a single horizontal surface (constant z value), but other surfaces are possible; e.g. a TIN.

² 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

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 DCDB 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 respond at all, or will give incorrect answers.

For example, in Figure 1, looking for parcels adjacent to parcel F, we should get C, E and G, but not D or B (D would be returned if a weak adjacency was requested (Cohn and Varzi 1999)). Method 1 has no solution to the question, methods 2, 3 and 4 will also return parcel B, 2 and 3 will also return D. Method 4 has a correct result, but at the cost of massively increasing storage requirements. In this paper we focus on a good design for method 6 based on an alternative approach, using a combination of LA_BoundaryFace and LA_BoundaryFaceString records as defined in ISO19152, the Land Administration Domain Model (LADM) (ISO-TC211 2012) as proposed in (Lemmen, Van Oosterom et al. 2010). This paper explores that approach and will demonstrate that many of the requirements of a DCDB and exchange format encoding can be met - including the adjacency of parcels in 3D.

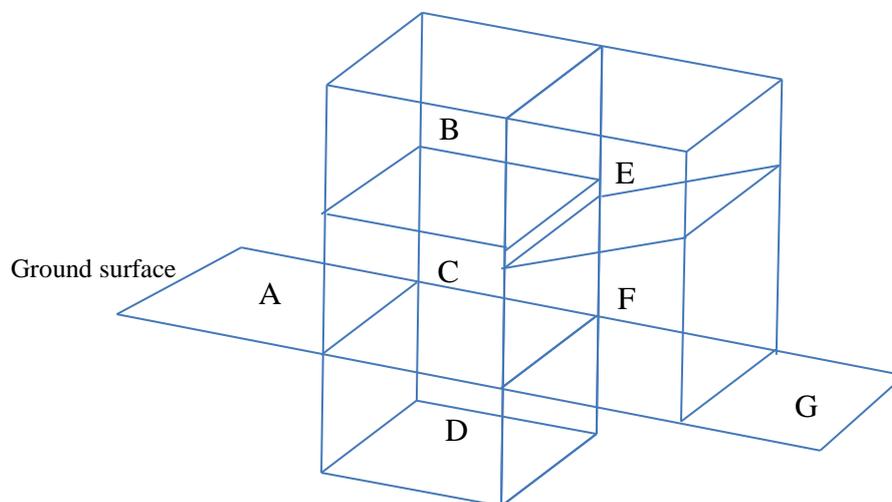


Figure 1. 2D land parcels A and G, 3D parcels B,C,D,E,F.

The remainder of the paper is structured as follows: Section 0 reprises the “Boundary Face String” concept of ISO19152 in the context of this model; Section 3 discusses the validity of the approach

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016

in terms of the range of geometric shapes to be accommodated; Section 4 considers this model as a database storage scheme; Section 5 as a transport mechanism; Section 6 as a data capture vehicle; Section 7 suggests advantages of the model in an evolving cadastral database due to positional adjustments (quality improvements); and Section 8 summarises the conclusions and future research directions.

2. THE BOUNDARYFACESTRING CONCEPT

It has been shown that in a typical cadastre, the set of 2D base layer parcels comprise most of the database; while even in the realm of 3D parcels, the majority are fairly simple in form (Thompson, van Oosterom et al. 2015). The approach being described here makes use of this fact in the encoding of all parcels, and is suggested by the LADM (ISO-TC211 2012). It is important to note that all so-called 2D parcels actually define a column of space above and below the land surface (Stoter and van Oosterom 2006), so that a 2D parcel is in truth a 3D parcel which has no explicitly defined top or bottom. This is recognised in the LADM by the concept of LA_BoundaryFaceString (Figure 2), which allows a 1D primitive (line string or ring) to fulfill the role of a 2D boundary (a surface which partly bounds a 3D region of space). Note that in Figure 2 the spatial unit is shown as bounded by five vertical faces of undefined height, but the database representation only need be a simple polygon. The defined space may be just partly bound (and partly unbounded). These spaces will never overlap. That is, for every point not on a boundary, it can be decided to which single space (spatial unit or outside space) it belongs to.

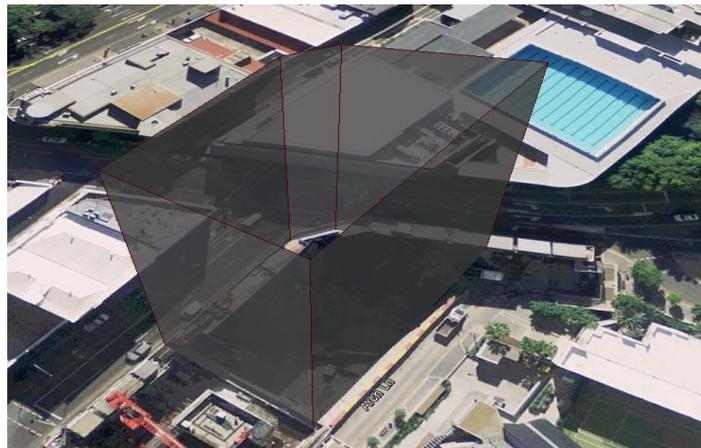


Figure 2. The concept of Face Strings from the LADM (ISO-TC211 2012), The five vertical faces should be interpreted as extending to infinity above and below the surface parcel, and together make up an LA_BoundaryFaceString

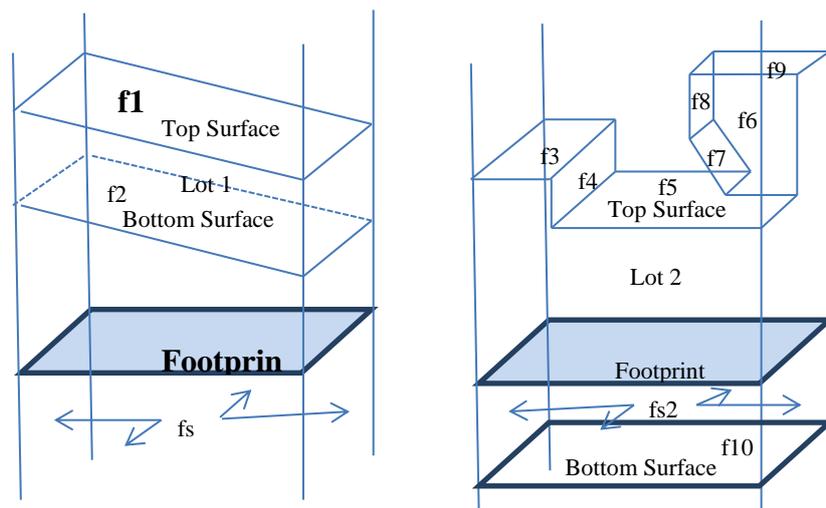


Figure 3. Definition of a volume by face string and faces. Lot 1 is defined by vertical face string fs1, and near- horizontal faces f1 and f2. Lot 2 is defined by vertical face string fs2, and faces f3 to f10. Note that the top surface of Lot 2 is quite complex (and falls according to the earlier introduced classification in to the category ‘General 3D spatial unit’

3. THE RANGE OF APPLICATION OF THE APPROACH

We now analyze in order of growing complexity the range of 3D spatial unit representations based on the LA_BoundaryFaceString (footprint) and LA_BoundaryFace (other boundaries, including both horizontal and tilted boundaries). The details depend on the complexity of the spatial unit to be defined.

3.1 2D Spatial Units

A simple 2D spatial unit is accommodated by recording a footprint only. Thus it is defined as a set of vertical faces with no defined top or bottom.

3.2 Vertical limits on 2D Spatial Units

The next more ‘complex’ class is where there is a vertical limit to the spatial unit (such as “to the depth of” limitations, for example where mining is happening below). The sub-cases here are volumes either: bounded below; bounded above; or bounded above and below. The last naturally leads to the most common form of 3D spatial unit: the polygonal slice (Thompson, van Oosterom et al. 2015). Further refinement in the classification is according to the type of bounding surface; e.g. horizontal, tilted, (see Figure 3 left) or TIN. A later variant, which occurs in definitions of tunnels and overpasses is the spatial unit with vertical walls, and the near-horizontal top and bottom surface defined by triangles (see Figure 4).

3.3 General 3D Spatial Units

In the case of a general 3D spatial unit (such as in Figure 3 right and Figure 6), it is always possible to generate a footprint (that is, a LA_BoundaryFaceString (by flattening the polyhedron), and storing this face string plus the faces (omitting those faces that lie within the facestring). In fact, as in this example, most general parcels do have at least some vertical faces over parts of the outer 2D boundary, and so even in the case of the general 3D spatial units, this may be a parsimonious storage mechanism. However, in some rare cases there can be spatial units without any vertical face. A parcel such as this can only be recorded as a LA_BoundaryFaceString plus all the original faces of the polyhedron (with no saving in storage). Even in this rare case, the method is still useful, because it means that the parcel is fully visible to 2D GIS software.

The category of 3D General 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).

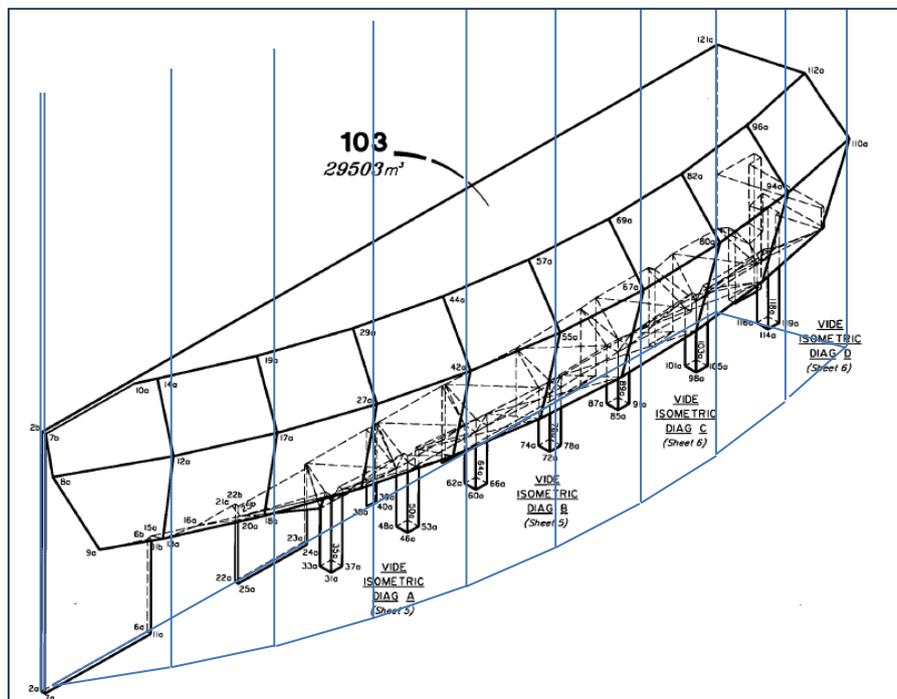


Figure 6. Complex Polyhedron, with the footprint vertical face strings shown.

3.4 Case of Interlace development

An interesting example is the case of the Interlace development in Singapore (see Figure 7). If this was one single spatial unit, it should be modelled as a multi-valued stepped slice special unit with only horizontal and vertical faces. Where the various blocks are different spatial units then these could each be modelled as a polygonal slice. The top face of a 3D spatial unit is not always identical

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016

to the bottom face of the 3D spatial unit above it. Only a part of the top/bottom boundaries are coinciding between the involved spatial units. Using a topology structure based representation this shared face could be defined once and used twice (see Sections 5.2 and 8.1). Where the block is not a single spatial unit, but contains multiple spatial units, the sharing of faces is increased further and using a topology structure based representation is able to avoid duplication.

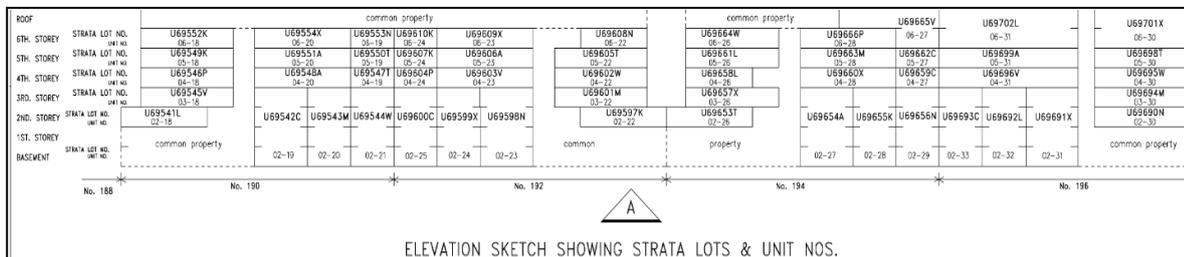


Figure 7. Interlace Development in Singapore, and its Elevation Plan (side view)

4. STORAGE SCHEME FOR THE CADASTRAL INFORMATION

After the overview of the range of 3D spatial unit types and analysis of the implications for their representation in terms of representation, Figure 8 shows a simplified database storage scheme able to represent the various types. 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.

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016

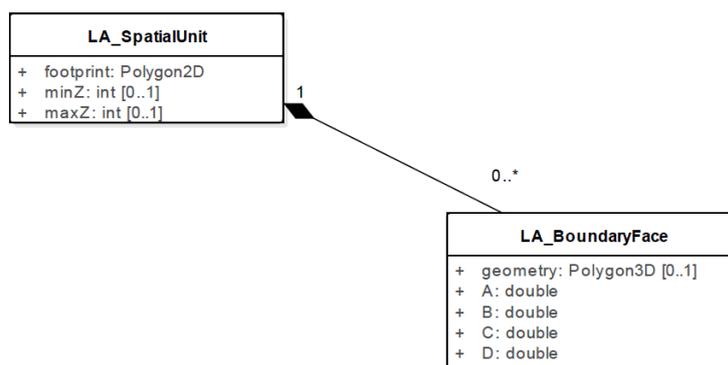


Figure 8. Simplified schema for database storage

Also note that in our simplified storage scheme, 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, and therefore is further discussed in the Sections 5.2 and 8.1.

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 see Figure 9); 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).



Figure 9. Levels of Detail in 3D City Models, from (Biljecki, Ledoux et al. 2014)

4.1 Access to the Footprint

This is the simplest, and the spatial unit table can be structured so that it can be directly accessed by a 2D-aware GIS. In fact, it is quite possible to update the spatial unit defined in this way using the 2D software - see Section 7.

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

4.2 Access to the Prism

It is sometimes useful to generate an approximation to a spatial unit - in a manner analogous to LoD1 of a 3D city model (See Figure 9). If it is desired to show the spatial units as simple prisms of space, giving an approximation to the true volume, and including all of the true volume, the footprint polygon can be extracted and extruded using the minZ and maxZ attributes.

4.3 Access as a 3D Geometry

This requires some software intervention - preferably as a database procedure to permit SQL such as: "select AsPolyhedron(footprint) from LA_SpatialUnit" (for completely bounded spaces). These routines have been developed in Java, but should equally be implementable in database procedural languages. The algorithm is fairly simple, determining the intersections of the LA_BoundaryFace(s) with the vertical faces of the footprint (interpreted as a LA_BoundaryFaceString). The result is a collection of faces which define and delimit the spatial unit. Note, that the spatial unit may not be bounded above or below, but in the case of a fully bounded volume, a conventional polyhedron is returned (and in the other cases a symbolic top or bottom is added).

4.4 Notes on the Encoding

The attributes minZ and maxZ are not strictly necessary, and could be replaced by database procedures which calculates the values from the LA_BoundaryFace table (and made available in easy manner via a database SQL view using this procedure). They are suggested here to allow a faster generation of the 'prism' form of the spatial unit. If used, they must be seen as redundant data, and must be generated automatically from the LA_BoundaryFace table, i.e. a materialized view. In our scheme the coordinate values are assumed to be in integer form (in millimeters). The attributes A, B, C and D are the definition of the plane surface on which the face is defined. They define the face as being the locus of points for which $Ax+By+Cz+D=0$ such that the points for which $Ax+By+Cz+D>0$ are on the inside (and with $A^2+B^2+C^2=1$). The encoding could work without these attributes, but at the expense of a difficult calculation which does not necessarily give consistent results. There is a slight risk of inconsistency of the Polygon3D not being exactly on the specified plane. However, proposed approach is practical. Some options to enhance consistency are: 1. specify polygon with 2D coordinates on the plane³ (and when needed derive the 3D polygon), or 2. derive the plane parameters from the 3D polygon and make available via SQL view (if 3D polygon is within the allowed tolerance for flatness otherwise this record cannot be accepted/validated on arrival).

³ Based on some conversion for the 2D coordinate reference system in plan; e.g. for non-vertical planes use projected origin and x- and y-axis. For vertical planes use z-axis instead of degenerated x- or y-axis.

5. 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 (Building Information Model) interchange specification, and as successor of LandXML for survey data. Because of the fairly early stage of development of InfraGML, this will not be discussed in detail. Other possible specifications would be expected to use a similar encoding approach.

Within the LandXML structure, the integrated footprint and LA_BoundaryFace volumetric approach can be encoded in two different schemes. One is called Simple Faces Method, which uses LandXML's <Parcel>, <CoordGeom> and <VolumeGeom> elements but does not consider the referencing between <CoordGeoms>. The other scheme is named Nested Parcels Method, which does only use <Parcel> and <CoordGeom>. It does not use the <VolumeGeom> element but considers the referencing. The following subsections describe the two schemes in more detail.

It is important to note that the transport mechanism is not strongly tied to the database storage method (as explained in Section 4), so that parcels could be transported in this mixed encoding to a database that stores them in another storage method. Likewise, spatial units transferred as full 3D polyhedra can be converted to the integrated footprint and LA_BoundaryFace form in the database.

5.1 Simple Faces Method

Note that in LandXML, the <Parcel> element is used to encode the spatial unit. Also note that a 2D (standard format) lot (parcelFormat="Standard") in this form would simply have the <VolumeGeom> element and its children omitted.

All information relating to a 3D spatial unit is contained within one <Parcel>. The footprint (LoD0, i.e. <CoordGeom desc="Footprint"> ...</CoordGeom>) is separated from the complete 3D geometry, which is encoded under the <VolumeGeom>. Programs can easily read and extract footprint and/or complete geometry of a 3D spatial unit. The trade-off of this method however is that the common faces may be encoded twice (with all possible dangers of inconsistencies). This thus will increase the number of validation checks to ensure that the common faces are indeed identical. The duplicated faces will also increase the size of LandXML especially when the 3D parcels are more complicated, e.g. a multi-storey building with 100 spatial units (strata subdivisions).

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016

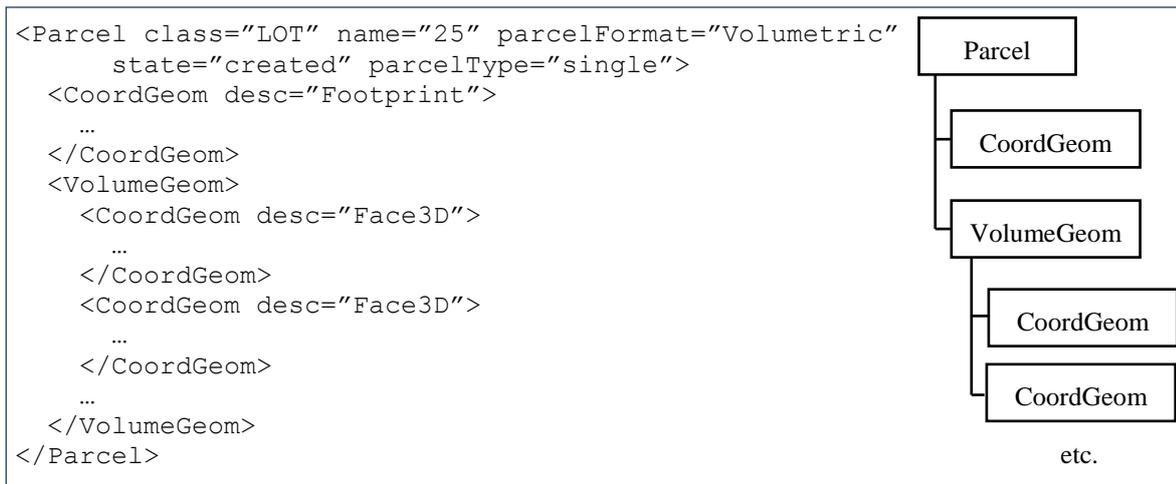


Figure 10. Simple Faces Method: "VolumeGeom" encoding of a 3D Spatial Unit

5.2 Nested Parcels Method

This method does not use the `<VolumeGeom>`, but through the referencing between `<CoordGeoms>` it forms a 3D spatial unit. This method was previously proposed in Soon, et. al. (2014) and the sample of this method is depicted in Figure 11. This allows software that does not support `<VolumeGeom>` to read 3D spatial units. Much commercial software at the moment does not support `<VolumeGeom>`. This method will facilitate the dissemination and use of 3D LandXML data in both CAD and GIS software. Further, this method will enable the representation of a topological structure (and when adding in the reference a + or a -, then the faces can be properly oriented).

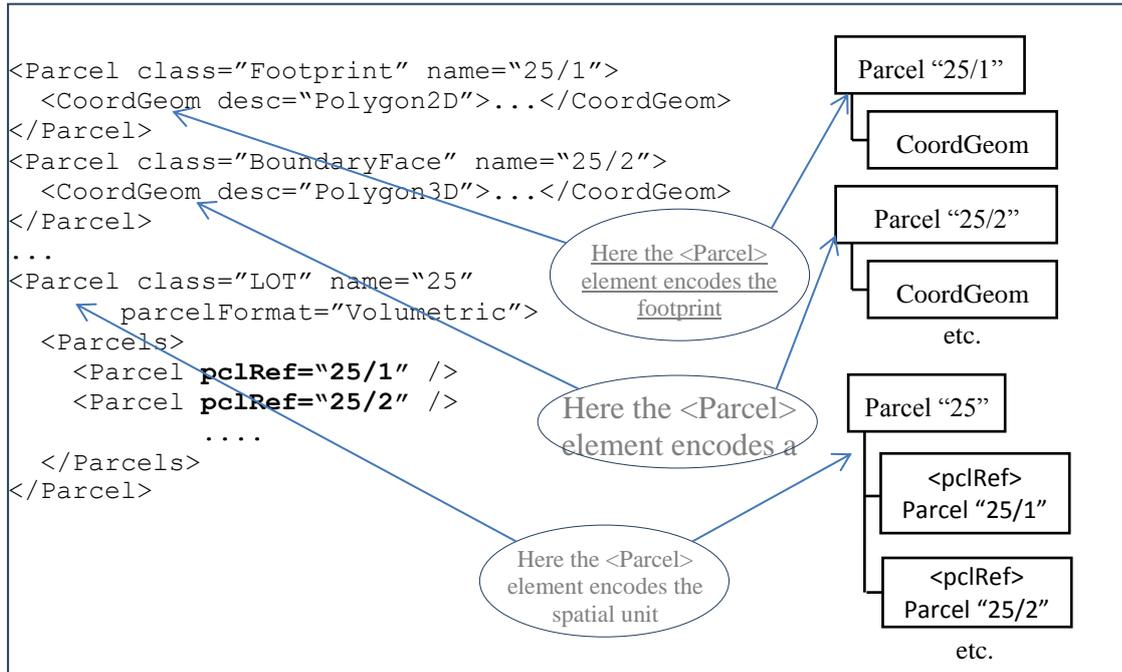


Figure 11. Nested Parcels Method: using the referencing method to form 3D Spatial Unit

As this method is structured in a federated manner through the ‘nested parcels’ referencing method, programmatically more steps will be required to read and extract the footprint of a 3D spatial unit. Thus more computing is needed especially for a 3D spatial unit that contains many faces. Another possible cause of redundant data may be the multiple storage of the same footprint shared by several 3D spatial units within the same column. Also here a referencing approach should be used to avoid this redundancy. More detail is to be found in Appendix I.

6. DATA CAPTURE AND VALIDATION

While electronic lodgement of survey plans has been introduced recently, there is in several countries a backlog of 3D plans that have been submitted on paper or in pdf form. If it is intended to back capture them, the integrated footprint and LA_BoundaryFace encoding can be of significant assistance to the process. The capture of the footprint is essentially identical to the capture of a 2D “Standard Format” parcel. In Figure 12, the bearings and distances for Lot 4 part A, Lot 2 part A, Lot 5 (with 3 disconnected components) can be located, and encoded using 2D software already in use in Queensland (where this plan is lodged).

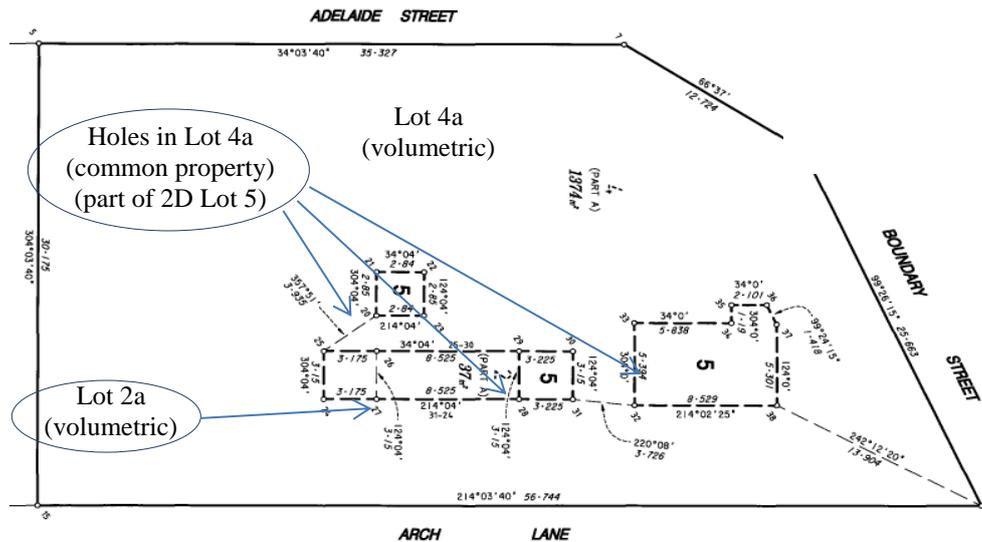


Figure 12. The Footprint information of a 3D spatial unit (here part of Lot 4 - lots 2 and 5 fill holes in lot 4). In this diagram, all points are 2D only, and so are given simple numbers: 5, 15, 10 ...

The encoding of the faces is fairly simple, as can be seen in Figure 13 which is an excerpt from the definition of lots 2a and 4a. This is used in the proof of concept to capture 3D parcels from plans of survey. For example, in the definition of Lot 4A the footprint is defined as a polygon through point numbers 5, 15, 10, 9 and 7 (which can be seen in Figure 12) while one of the bottom faces of Lot 2A is defined as a polygon through points 27b, 26b, 29a, and 28a. The alphabetic suffix to the point number indicates a 3D point, so that they are not needed in the footprint definition where only the 2D location is needed. The software in use has been written to reorganize keywords in the data capture format, such as ‘Parcel’, ‘Footprint’, ‘Top Z’, ‘Bottom’, but also to understand shortcuts such as ‘Minus’ so that a polyhedron can be excised from a spatial unit (as is shown in Figure 13 (right), where Lot 2A is excised from Lot 4A). An additional simplification in encoding is that vertical faces in a well-defined top or bottom surface can be inferred; So that in Figure 14, it is only necessary to encode the two horizontal bottom faces, inferring the marked vertical face.

Parcel Lot 2A	Parcel Lot 4A
Footprint 25 24 27 28 29 26	Footprint 5 15 10 9 7
Top Z<1.2	Top Z<1.2
Bottom1 27b 26b 29a 28a	Bottom4a Z>=-12.8
Bottom2 26a 27a 24a 25a	Minus Lot 2A
	Minus Lot 5a1
	Minus Lot 5a2
	Minus Lot 5a3

Figure 13. The encoding of the faces in Lot 2 part A and Lot 4 part A

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016

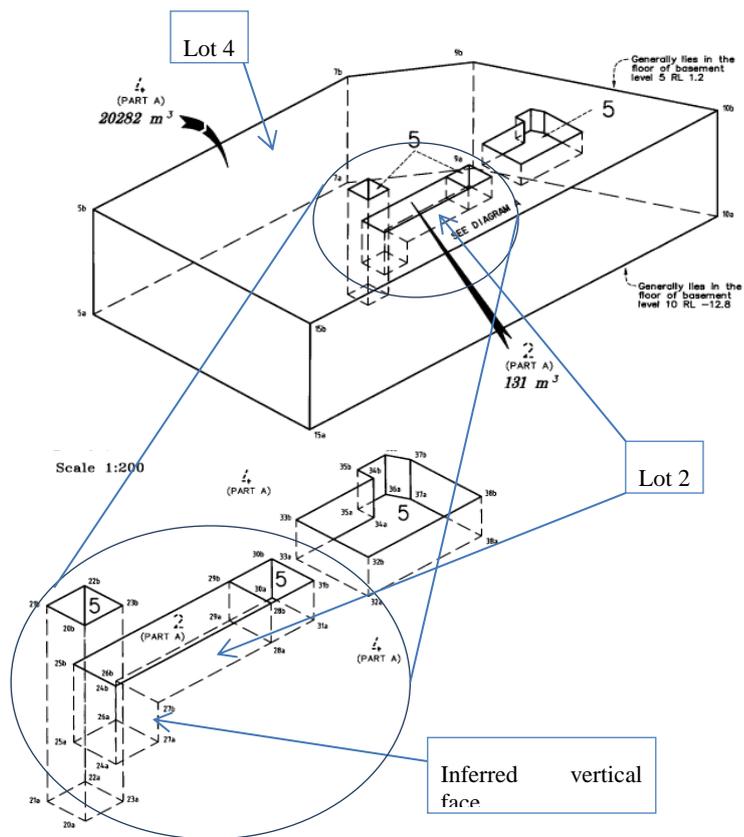


Figure 14. An isometric drawing of Lot 4 and Lot 2 of Figure 12 showing the faces to be defined (The regions marked 5 are the remainder of the common property 2D spatial unit when lots 2 and 4 are excised). Since the points in this diagram are 3D defined, they are given alphabetic suffixes - so that points 5a and 5b are 3D points with the same 2D location

7. POSITIONAL ADJUSTMENTS

In most real cadastral databases, there is need from time to time to adjust the position, and this applies equally to 3D parcels. The adjustment software should be capable of dealing with the requirement to keep some parcels rigid, even though they may need to be moved. This is not the complete story where all 3D spatial units are concerned - for example the spatial unit in Figure 15 is below a surface parcel of 4093ha (more than 40km²), and would be expected to be affected by a positional adjustment as if it were a standard format spatial unit.

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016

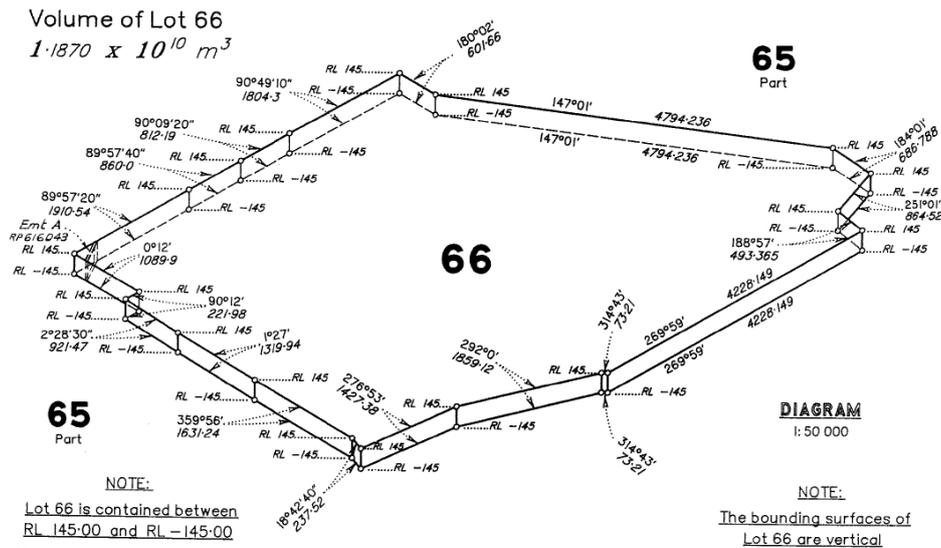


Figure 15. A volumetric lot covering a large area (4093 ha and 4 km³)

Most positional adjustment software is designed to solve the 2D adjustment problem, using techniques like least-squares fitting to develop a movement to be applied to all points in a given region. This can be used in the case of 3D spatial units using this storage approach and initially applied to the footprint (LA_BoundaryFaceString) in the adjustment software, provided that a database “trigger” is used to review the validity of the operation on the 3D parcels. The involved 3D LA_BoundaryFace’s have to be adjusted to keep their relative position and shape related to the footprint. In the majority of cases and when implemented well, the operation will result in valid 3D parcels (and only in exceptional cases the change of coordinates might cause errors, which need to be detected and corrected).

For all LA_BoundaryFaces in the definition, dissolve all matching edges.
 (This leaves only those edges which should meet the
 LA_BoundaryFaceString)
 For each of these remaining edges, snap to the nearest
 LA_BoundaryFaceString.
 Reapply the usual validation tests to ensure no failures have been
 introduced

Figure 16. Simple procedure to be applied to any 3D Spatial Unit, triggered by a positional adjustment

For all spatial units such as that of Figure 15, with a polygonal slice form, no further action is needed. For more complex cases, a simple procedure is as shown in Figure 16. This procedure may fail in one of two ways: 1. The movement may be sufficiently coarse that the snap operation fails, or 2: The validation tests may fail. A possible cause of failure in the validation tests is that a face which was planar but not horizontal or vertical may be distorted to the point of non-planarity or might introduce a topology error (with another independently defined 3D spatial unit). The non-

planarity clearly cannot happen in cases such as Figure 4, since a triangle is always planar. It could happen in the case of Figure 6 (which has quadrilateral tilted faces), but since all such faces are recognized as only “planar to a tolerance”, in most cases the spatial unit will remain valid in the presence of small distortions. It is only in a very small minority of cases that an invalid spatial unit can be generated by an adjustment or any other 2D modification to the database.

A more advanced procedure might be to include the (x,y) coordinates in the adjustment not only of the footprint, but also of the LA_BoundaryFace coordinates (and not affecting the z value). Also in this case: when done independently from nearby (higher/ lower/ intertwined) other 3D spatial units, topology errors might be introduced, so the procedure of Figure 16 should also be triggered in this case.

8. CONCLUSIONS AND FUTURE WORK

As this paper has illustrated, the conceptual model is a complete representation of the current cadastral requirements, being able to accommodate 2D (Standard Format) spatial units; above/below the depth/height of spatial units; (single and multi-valued) stepped slice spatial units; complete (“watertight”) polyhedral spatial units (with potentially tilted faces); and partially closed volumetric parcels (such as the balance of a 2D spatial unit when 3D volumes have been excised). Where the software system can only handle 2D polygonal spatial units, it can still query the data in a limited (2D) form, but with visibility of all spatial units (as overlapping footprints). The 2D spatial units take no more storage space than they would in a conventional 2D-only database.

Where a software system can only handle polygonal spatial units extruded in the z dimension, it can still query the data in a limited form, with visibility of all spatial units as a simple prism. These can be very useful, and highlight potential overlaps, but are likely to provide “pessimistic” results (indicating an encroachment that does not exist in fact, such as between lots E and F in Figure 1).

The majority of adjustments to the database can be done using conventional 2D GIS technology (supplemented by database triggers/procedures to process the involved 3D LA_BoundaryFaces) without any “drifting” of the 3D objects from connection with the 2D fabric. In only the rarest of cases will real 3D software (or interactive intervention) be needed for adjustments or updates to the cadastral database.

Some proof of concept Java code has been produced to ensure that critical algorithms are practical, and reasonably responsive. For example, the routine to convert an integrated footprint and LA_BoundaryFace encoding into a conventional polyhedron (when 3D spatial unit is supposed to be fully closed) is operational, and there is no reason to expect that it cannot be expressed as a database procedure.

8.1 Topological encoding

This paper has concentrated on explicit mode encoding - where each spatial unit contains the full geometry of that spatial unit. 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. This is not a limitation of this

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016

encoding, as has been illustrated in section 5.2, which shows that sharing of common boundaries is feasible. This paper has restricted the consideration of boundary sharing to the transport encoding (LandXML) but in future work we will further this subject. 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-TC211 2012).

8.2 Other Issues

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. In fact, many cadastres make an important distinction between fiat boundaries and boundaries defined by natural features (Smith 1994). 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). Also 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 9. 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.

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)). Recently OGC has put up a Request for Quotation (RFQ) to overcome this issue in the OGC Future City Pilot Phase 1 Project⁴.

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. Early 2016 the OGC opened the public review for the Land and Infrastructure Conceptual Model Standard, or InfraGML for short⁵. Especially sections 7.10 (with class LandDivision, p. 189), 7.11 (with class

⁴ <http://www.opengeospatial.org/standards/requests/147>

⁵ www.opengeospatial.org/standards/requests/148

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

Condominium, p. 207) and Annex D2.1 (discussing the relationship with LADM, p. 270) are relevant in this context.

There is also need for consideration of visualization tools - where large scale 2D spatial units are mixed with 3D spatial units. The visualization of partially open (above/below) spatial units in particular needs consideration.

REFERENCES

- Biljecki, F., H. Ledoux, et al. (2014). Redefining the Level of Detail for 3D Models. GIM International.
- Cohn, A. G. and A. C. Varzi (1999). Modes of Connection. Spatial Information Theory. Proceedings of the Fourth International Conference, Berlin and Heidelberg, Springer Verlag.
- 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.
- Guo, R., C. Yu, et al. (2012). Logical Design and Implementation of the Data Model for 3D Cadastre in China. 3rd International Workshop on 3D Cadastres: Developments and Practices. P. van Oosterom, R. Guo, L. Li, S. Ying and S. Angsüßer. Shenzhen, China.
- ISO-TC211 (2012). Geographic Information - Land Administration Domain Model (LADM). ISO 19152.
- Lemmen, C., P. Van Oosterom, et al. (2010). The Modelling of Spatial Units (Parcels) in the Land Administration Domain Model (LADM). FIG Congress 2010, Sydney, Australia.
- OGC (2016). OGC® Land and Infrastructure Conceptual Model Standard (LandInfra), Open Geospatial Consortium.
- Scarponcini, P. (2013). InfraGML Proposal (13-121), OGC Land and Infrastructure DWG/SWG.
- Smith, B. (1994). Fiat Objects. Parts and Wholes: Conceptual Part-Whole Relations and Formal Mereology, 11th European Conference on Artificial Intelligence, Amsterdam. N. Guarino, L. Vieu and S. Pribbenow. Amsterdam: European Coordinating Committee for Artificial Intelligence: 15-23.
- Soon, K. H., R. Thompson, et al. (2014). Semantics-based Fusion for CityGML and 3D LandXML. 4th International Workshop on 3D Cadastres. Dubai, United Arab Emirates.
- Stoter, J. and P. van Oosterom (2006). 3D Cadastre in an International Context. Boca Raton FL, Taylor & Francis.
- Thompson, R. and P. van Oosterom (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., P. van Oosterom, et al. (2015). A Taxonomy of Spatial Units in a Mixed 2D and 3D Cadastral Database. FIG Working Week 2015. Sofia, Bulgaria.

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Pribbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016

BIOGRAPHICAL NOTES

Rodney James Thompson has been working in the spatial information field since 1985. He designed and led the implementation of the Queensland Digital Cadastral Data Base, and is now advising on spatial database technology with an emphasis on 3D and temporal issues. He obtained a PhD at the Delft University of Technology in December 2007.

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

Kean Huat Soon is Senior Surveyor at the Land Survey Division of Singapore Land Authority. He is technical lead in the 3D National Mapping project and the development of the new cadastral system to support 3D Cadastres and automated cadastral processing. He earned a MSc in Geography from the Pennsylvania State University, a MSc in Geoinformatics and Bachelor of Surveying (Land) from University of Technology Malaysia. His research interests include semantic interoperability, data modeling, cadastral information system and ontology.

Russell Priebbenow is the Director of Cadastral and Geodetic Services in the Department of Natural Resources and Mines. He is a registered cadastral surveyor with over 35 years of public sector experience in surveying and mapping, including more than 25 years involvement with legislation and policy. Prior to this, he attained a PhD from the University of Queensland for research into the mapping applications of imagery from the SPOT satellite. He is a member of the Surveyors Board of Queensland and also the Queensland member of the Australian and New Zealand Intergovernmental Committee on Surveying and Mapping. He is a Fellow of the Surveying and Spatial Sciences Institute.

CONTACTS

Rod Thompson
Queensland Government, Department of Natural Resources and Mines
Landcentre, Cnr Main and Vulture Streets,
Woolloongabba, Brisbane, Queensland 4102
AUSTRALIA
Tel.: +61 7 38963286
E-mail: Rod.Thompson@qld.gov.au
Website: <http://www.dnrm.qld.gov.au/>

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016
Recovery from Disaster
Christchurch, New Zealand, May 2–6, 2016

Peter van Oosterom
Delft University of Technology
Faculty of Architecture and the Built Environment
Department OTB, GIS Technology Section
P.O. Box 5030
2600 GA Delft
THE NETHERLANDS
Tel.: +31 15 2786950
E-mail: P.J.M.vanOosterom@tudelft.nl
Website: <http://www.gdmc.nl>

Kean Huat Soon
Singapore Land Authority
55 Newton Road, #12-01, Revenue House
Singapore 307987
SINGAPORE
Tel.: +65 6478 3537
Fax: +65 6323 9937
E-mail: soon_kean_huat@sla.gov.sg
Website: <http://www.sla.gov.sg/>

Russell Priebbenow
Queensland Government, Department of Natural Resources and Mines
Landcentre, Cnr Main and Vulture Streets,
Woolloongabba, Brisbane, Queensland 4102
AUSTRALIA
E-mail: Russell.Priebbenow@dnrm.qld.gov.au
Website: <http://www.dnrm.qld.gov.au/>

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016
Recovery from Disaster
Christchurch, New Zealand, May 2–6, 2016

APPENDIX I

This is an excerpt from a sample of a LandXML format file of a 3D spatial unit (In Singapore). It is encoded in the nested parcel method presented in section 5.2 and given in summary form in Figure 11. Note that the spatial units “Basement 2” and “Basement 3” are each defined by the footprint (“Pcl1”) and top and bottom boundary faces. Note also that footprints and faces are all encoded using the <Parcel> element.

```
<Parcels>
  <Parcel class="Footprint" name="Pcl1" area="3325.1" desc="Polygon2D">
    <CoordGeom desc="Polygon2D" name="Pcl1">
      <IrregularLine>
        <Start pntRef="801"/>
        <End pntRef="801"/>
        <PntList2D>31083.666 29893.109 ... 31072.951 29884.308</PntList2D>
      </IrregularLine>
    </CoordGeom>
  </Parcel>
  <Parcel class="BoundaryFace" name="PclB2" area="3325.1" desc="Polygon3D">
    <CoordGeom name="PclB2">
      <IrregularLine>
        <Start pntRef="805_B2"/>
        <End pntRef="805_B2"/>
        <PntList3D>31083.666 29893.109 88.76 ... 31072.951 29884.308 88.76</PntList3D>
      </IrregularLine>
    </CoordGeom>
  </Parcel>
  <Parcel class="BoundaryFace" name="PclB3" area="3325.1" desc="Polygon3D">
    <CoordGeom name="PclB3">
      <IrregularLine>
        <Start pntRef="805_B3"/>
        <End pntRef="805_B3"/>
        <PntList3D>31083.666 29893.109 82.77 ... 31072.951 29884.308 82.77</PntList3D>
      </IrregularLine>
    </CoordGeom>
  </Parcel>
  <Parcel class="BoundaryFace" name="PclB4" area="3325.1" desc="Polygon3D">
    <CoordGeom name="PclB4">
      <IrregularLine>
        <Start pntRef="805_B4"/>
        <End pntRef="805_B4"/>
        <PntList3D>31083.666 29893.109 78.28 ... 31072.951 29884.308 78.28</PntList3D>
      </IrregularLine>
    </CoordGeom>
  </Parcel>
```

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016

```
</IrregularLine>
</CoordGeom>
</Parcel>
<Parcel class="LOT" name="Basement 2" parcelFormat="Volumetric">
  <Parcels>
    <Parcel pclRef="Pcl1" />  !!(reference to footprint)
    <Parcel pclRef="PclB2" />
    <Parcel pclRef="-PclB3" />  !! reverse of PclB3
  </Parcels>
</Parcel>
<Parcel class="LOT" name="Basement 3" parcelFormat="Volumetric">
  <Parcels>
    <Parcel pclRef="Pcl1" />  !!(reference to footprint)
    <Parcel pclRef="PclB3" />
    <Parcel pclRef="-PclB4" />
  </Parcels>
</Parcel>
</Parcels>
```

A Conceptual Model Supporting a Range of 3D Parcel Representations Through all Stages: Data Capture, Transfer and Storage (8173)

Rodney Thompson (Australia), Peter van Oosterom (Netherlands), Kean Soon (Singapore) and Russell Priebbenow (Australia)

FIG Working Week 2016

Recovery from Disaster

Christchurch, New Zealand, May 2–6, 2016