Practical Approaches to 3D Cadastre Implementation: Database Schemas and Exchange Formats

Javad SHAHIDINEJAD, Mohsen KALANTARI and Abbas RAJABIFARD, Australia

Key words: 3D Cadastre, Land Administration, LADM, Physical Data Model, Database

SUMMARY

Land administration systems in Victoria currently use 2D cadastral databases to record, manage, and retrieve data, facing various challenges. These databases have been fundamental to land administration for over two decades, primarily focusing on 2D data to manage property rights, restrictions, and responsibilities (RRRs). Despite some progress, a comprehensive database schema for 3D cadastres is still lacking. Developing such a schema involves multiple steps, including creating a conceptual design, selecting a database management system, developing the logical design, and implementing the physical design.
This paper focuses on developing a database and exchange formats at the schema level based on Land Administration Domain Model (LADM) Edition II, aligned with the Victorian cadastral system. The Victorian cadastre is the result of collaborative efforts from various agencies, departments, and projects. Key components like Vicmap, Victorian Online Title System (VOTS), Surveying and Planning through Electronic Applications and Referrals (SPEAR), and ePlan are integral to the system, each corresponding to different parts of LADM II. The current 2D cadastral system in Victoria is based on an XML format, while the Intergovernmental Committee on Surveying and Mapping (ICSM) is working on developing a 3D Cadastral Survey Data Model (3D CSDM) for Australia, which is based on a JSON format. In this paper, we investigate the Victorian cadastral system and different parts of LADM Edition II and identify the similarities between them. An integrated conceptual data model for Parts 1, 2, and 5 will then be created. Since each part of LADM II is a standalone standard, there are redundancy and consistency issues that need to be addressed.
Furthermore, the challenges of transforming the conceptual data model into logical and physical data models will be discussed, and solutions to address these challenges will be analysed. The paper underscores the importance of adhering to database design principles for managing 3D spatial data within relational databases like PostgreSQL/PostGIS, highlighting the necessity of manual interventions during the transformation process. It also examines the creation of exchange formats such as XML and JSON and compares them with databases. The paper emphasises the need for a consistent approach to ensure data integrity, compatibility, and the linking of concepts to maintain consistency and interoperability.

Practical Approaches to 3D Cadastre Implementation: Database Schemas and Exchange Formats

Javad SHAHIDINEJAD, Mohsen KALANTARI and Abbas RAJABIFARD, Australia

1. INTRODUCTION

Developing a 3D cadastral system involves several key steps including data acquisition, preparation, and validation, as well as data registration and management, analysis, retrieval, visualisation, and dissemination [1-3]. Victoria, Australia, has successfully implemented both file-based and database systems for 2D cadastre management. Notable examples include ePlan [4], which utilises an XML exchange format for the digital registration of plans, and the Digital Cadastral Database (DCDB) [5], Victorian cadastral map base that provides information about land parcels and property details. However, with the development of high-rise, mixed-use, and modern multi-purpose buildings, traditional 2D systems are inadequate for accurately determining property extents and the associated interests in land. This complexity necessitates advanced data management approaches that incorporate 3D aspects of cadastre to effectively register, record, manipulate, and retrieve RRRs, and other land-related data [6]. Moreover, the importance of implementing technical models like database schemas and exchange formats was emphasised in the International Federation of Surveyors (FIG) Workshop [7].

Conceptual data modelling is a critical step for implementing both databases and exchange formats. The LADM, as an international standard, facilitates interoperability and offers a standardised terminology and formal framework for describing and managing contemporary land administration systems [8]. LADM Edition II has been developed as a multi-part series, consisting of six standards, each one is treated as an individual standard, yet all remain backward compatible with LADM Edition I [9, 10]. The Victorian cadastre shares significant similarities with LADM Edition II, particularly in Part 1 (fundamental concepts and definitions), Part 2 (land registration), and Part 5 (spatial plan information). These parts align closely with the structures and processes of the Victorian cadastral system, making compliance and integration between them feasible.

An important question is how the different parts of LADM can be merged to create an integrated conceptual model for the Victorian cadastre, based on Parts 1, 2, and 5. Once an integrated conceptual data model is designed, it can be directly transformed into exchange formats like XML using Enterprise Architecture (EA) software [11]. However, for database implementation, the model first needs to be converted into a logical data model and then into a physical data model. This process presents challenges such as managing integrity constraints, surrogate keys, operations, inheritance, data types (particularly spatial data types), domains, code lists, indexing and clustering, all of which often require manual modification to resolve [12, 13].

This paper aims to implement a 3D cadastral database for Victorian cadastre based on LADM Edition II. It discusses the alignment of LADM Edition II with the Victorian cadastral system, with a particular focus on Parts 1, 2, and 5, to achieve compliance. The paper explores technical solutions for integrating LADM's various parts and converting the conceptual data

model into a logical and physical data model. Additionally, it explores the implementation of exchange formats like XML and JSON and compares them with database implementation.
The remainder of the paper proceeds as follows. Section 2 reviews related literature and research studies. Section 3 outlines the research methodology employed in this study. Section 4 covers the implementation process, including aligning LADM with the Victorian cadastre, designing the conceptual data model, transforming it into logical and physical data models, and developing databases and exchange formats. A comparison of implementations, along with the challenges and knowledge gaps encountered during the implementation process, will be discussed in Section 5. The paper ends with conclusion and potential directions for future research in Section 6.

2. RELATED WORKS

Most of the literature has implemented and used databases without fully adhering to database design principles, often treating databases as just a component of their work rather than a primary focus of their work [6]. In this section, we review studies that have implemented a cadastral database based on LADM and database design principles.
Alattas et al. developed a technical model for indoor navigation by integrating the LADM Edition I with IndoorGML. They began by creating an integrated conceptual data model, which was then transformed into a logical data model using EA modelling tools. However, they faced challenges during this transformation, particularly with aspects like inheritance, primary key and foreign keys, multiplicity, constraints, data types, spatial data types, code list classes, and indexing. To address many of these issues, they resorted to manual adjustments. Some problems, like the failure to convert attribute multiplicity into tables, remained unresolved. The study highlighted that, despite using data modelling tools like EA, manual interventions are still required to overcome various transformation issues [12]. In another study, Alattas et al. investigated issues related to database implementation and the visualisation of query results. They designed a 3D building in Revit and imported it into the database using Open Database Connectivity (ODBC). Although they transferred semantic data directly from Revit, and linked all attributes, including the 3D geometry, to a unique geometry ID, the tables related to the LADM still required manual data entry [14]. Alattas et al. subsequently used this database to analyse user movements during evacuation exercises in indoor environments [15].
Zulkifli et al. utilised LADM to develop a cadastral registration system for Malaysia that covers both 2D and 3D spatial data. They introduced a Unique Parcel Identifier (UPI) to connect spatial data from the Department of Survey and Mapping Malaysia with non-spatial data from the Land Office. They also suggested new code lists to enhance Malaysian cadastral system. The study demonstrated the use of Oracle Spatial for storing and querying 2D and 3D cadastral data, and Bentley MicroStation for visualisation [16]. Then in another work, they developed a prototype to evaluate the Malaysian LADM country profile. They utilised EA to automatically convert the conceptual model into the technical model. However, it was necessary to make some manual adjustments to the technical model were done. As part of this process, constraints, derived attributes, multiplicity, indexing and clustering were discussed [13]. Several studies, including Zulkifli et al. [17, 18], and Nasorudin et al. [19], have also

utilised LADM and similar methodologies to transform conceptual models into physical models based on the regulations in Malaysia.
There are some studies that have focused on NoSQL databases. As an example, Višnjevac et al. developed a 3D cadastral database based on LAMD using MongoDB [20]. Their system managed 3D cadastral data to some extent but had limitations, particularly in topology rule validation, and lacked options for spatial queries, such as selecting neighbouring objects.
In summary, most studies have utilised LADM Edition I, leaving a gap in discussions regarding the integration of different parts of LADM II, as well as issues of redundancy and inconsistency. Additionally, research has concentrated on the technical model, but there is a subtle difference between technical model in EA and the concepts of logical and physical data models in database development, despite their similar end results. Another gap is the lack of emphasis on database optimisation strategies. For instance, creating separate tables for code lists increases the overall number of tables, which complicates database management and administration. This approach also leads to more frequent use of *JOIN* functions in queries, which can negatively impact performance and require more advanced SQL knowledge to manage them efficiently. Lastly, the literature fails to address the challenges of ensuring semantic consistency and effective data integration when converting LADM entities into database tables, particularly in relation to ontology and linked open data. This oversight could result in difficulties with interoperability, data integrity, and advanced data analysis.

3. RESEARCH METHODOLOGY

As this study emphasises adhering to database design fundamentals, the methodology has been adopted by the database design principles as outlined in [21]. Figure 1 illustrates the methodology and steps of implementation utilised in this research. Firstly, a literature review will be conducted to identify current practices, challenges, and research gaps. Relevant standards will then be explored, leading to the selection of LADM as the most recognised standard in the land administration domain. Victoria was chosen as the case study, and the regulations and standards of Land Victoria were analysed. The compliance of the Victorian cadastral system with LADM Edition II was assessed, and the relevant parts were selected. Based on LADM Parts 1, 2, and 5, an integrated conceptual data model was developed. For physical implementation, two approaches were adopted: a database schema and an exchange file approach. The file format could be generated directly from the conceptual data model, while the database required conversion to logical and physical data models. Finally, a comparison of the database and exchange formats will be conducted at the schema level.

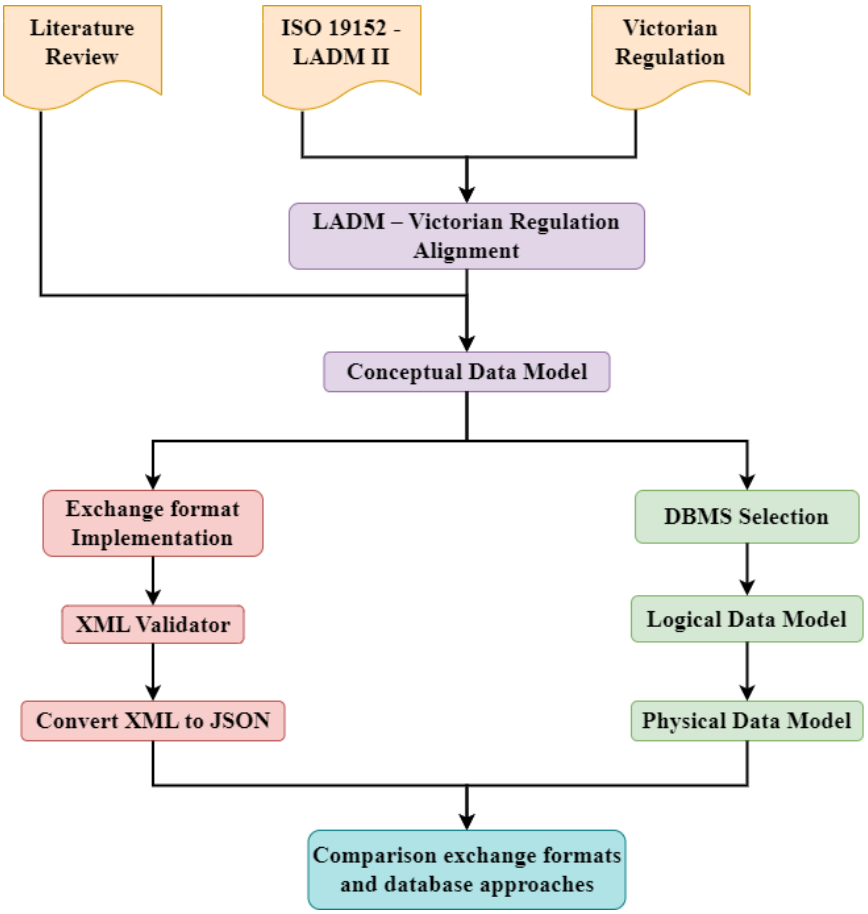

Figure 1. Overview of the research methodology

4. COMPLIANCE OF THE VICTORIAN CADASTRAL SYSTEM WITH LADM II

Prior to the data modelling and implementation, it is essential to determine which parts of LADM II need to be implemented, necessitating an alignment process between LADM II and the Victorian cadastral system. In Victoria, Land Use Victoria (LUV) is responsible for managing land administration activities such as land registration, property information, spatial data services and maps, surveying, land valuation, geographic names, government land policy and advice, and government land transaction oversight. LUV is a key state agency and is currently part of the Department of Transport and Planning (DTP) [22]. Furthermore, LUV is responsible for maintaining the Victorian cadastral system, which currently represents the state's property boundaries in 2D [23]. The Victorian cadastre is the outcome of collaborative efforts from various agencies, departments, and projects. The main components and related activities are summarised below.

Vicmap: Since 1975, Vicmap has played a vital role in Victoria primary mapping system, serving as the state's 2D cadastral map base, also known as the Digital Cadastral Database (DCDB). This database provides comprehensive information on land parcels and properties, including identifiers, standard parcel identifiers (SPI), parcel status (registered or proposed),

distinctions between freehold and Crown land, easements, and unique feature identifiers [5]. Vicmap Property can be aligned with the Spatial Unit package from both Part 1 and Part 2.
The simplified cadastral data model of Vicmap Property links parcel and property attributes to their spatial representations, making it easier to use. However, table *JOINs* are still needed to establish relationships between parcels and properties or to match properties with their addresses using SPI, lot and plan number, and council property number, with reference to the Vicmap Address data model [24]. Vicmap Address is matched with the External package (ExtAddress) of Part 1.
VOTS: land titles are maintained within the Victorian Online Title System (VOTS), a comprehensive database that serves as the authoritative record of land ownership. VOTS is a non-spatial database that not only documents ownership details but also includes information on restrictions such as mortgages, covenants, caveats, and easements that may affect the property [24]. VOTS corresponds to the Administrative and Party packages in both Part 1 and Part 2. Additionally, it can be considered as a source document.
Surveying and Planning through Electronic Applications and Referrals (SPEAR): SPEAR is an online platform that facilitates the process of managing subdivision planning permits, certification applications, and other land administration tasks. It supports the compilation, submission, management, referral, approval, and tracking of applications [25]. SPEAR is aligned with the Party and Generic Conceptual Model packages from Part 1, as well as with Part 5 – Spatial Plan Information. In addition, it is aligned with the external package of the Part 2.
ePlan: ePlan is a digital data file based on LandXML which represents cadastral and administrative information related to a plan. ePlan improves data quality, minimises duplication, and paves the way for end-to-end digital data workflows throughout the plan's lifecycle. This streamlines and automates traditional manual processes, leading to faster, more reliable, and consistent plan registration [4]. ePlan itself is part of the SPEAR process so it is corresponded to Part 1 and Part 5. In addition, it includes survey measurements and parcel dimensions which is in line with Survey and Representation package of Part 2.
Survey Marks Enquiry Service (SMES): SMES is an open-access database of survey control mark information in Victoria [26], which can be matched to the Survey and Representation of Part 2.
Abstract of field record: A document that contains detailed information recorded by a licensed surveyor. The purposes of preparing this abstract include: a) ensuring the preservation and public accessibility of records, b) providing documentary evidence of field conditions that support the chosen re-establishment method, c) offering supplementary information to verify recorded measurements, and d) documenting site conditions, such as traverse lines and instrument positions, which indicate topography and the presence of obstacles [27]. This document aligns with the Spatial Unit package and its subpackage, Surveying and Representation, from Part 2. Additionally, it serves as a valuable source of survey measurements.
Apart from these components, other databases and information are required in the cadastre, such as **Surveyors Registration Board of Victoria**, which is responsible for the registration of licensed surveyors to perform cadastral surveying in Victoria and provides a database of registered licensed surveyors [28]. This aligns with the Party package in Part 1.

Since only Part 1 of LADM II has been officially published at the time of writing this paper, the alignments with the Victorian cadastre are based on existing publications. These alignments may be further refined in the future as more detailed drafts of the other parts become available following their official release. Matching between LADM II and Victorian cadastral system components is demonstrated in Figure 2.

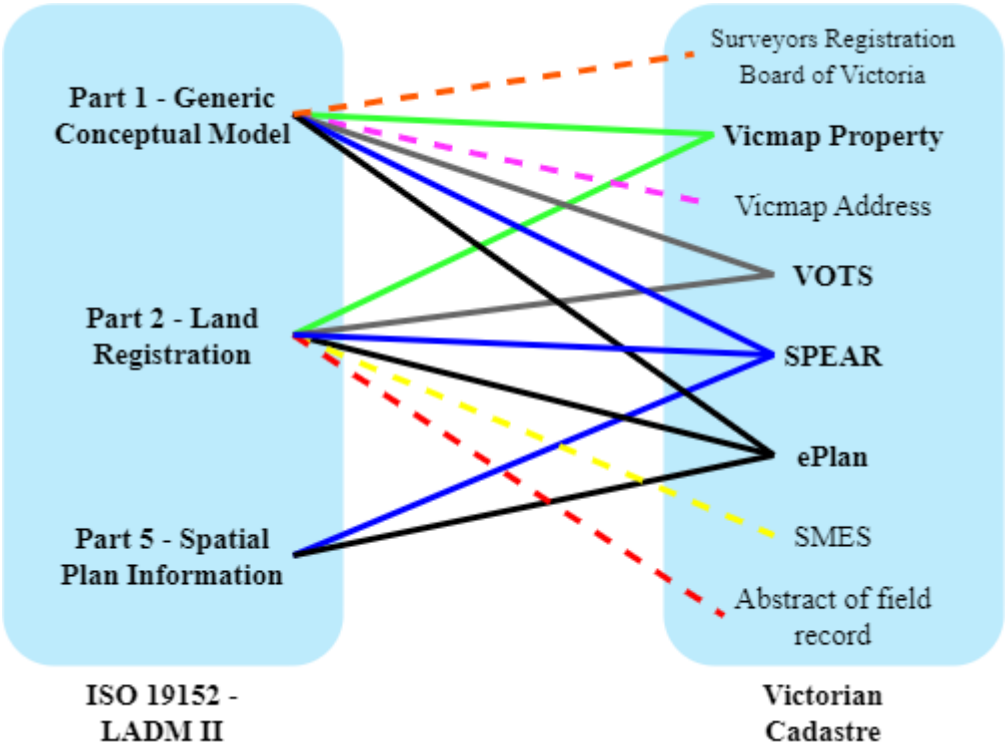

Figure 2. Compliance of LADM II with the Victorian cadastral system

5. IMPLEMENTATION

This section outlines the implementation process and explains each step. It also addresses the challenges encountered and proposes alternative solutions to enhance both execution and problem-solving strategies. The conceptual data model, which serves as the foundation for both the database and exchange formats, will be explained. Following this, the implementation of the database and file formats will be detailed.

5.1. Conceptual data modelling

Conceptual data models represent the most abstract level in data modelling, providing concepts that align with users' perceptions of the data. Based on the alignments conducted in Section 4, implementing parts 1, 2, and 5 of LADM II is beneficial for the Victorian cadastre.
In order to develop the conceptual model, the UML class diagram of ISO 19152 was used, and the model was developed utilising Enterprise Architect version 16.1. During the design of the conceptual model for arts 1, 2, and 5 of LADM, a key challenge arose from the fact that each of these parts is a standalone standard. As a result, overlapping packages and common

classes were identified across these standards, presenting challenges in effectively integrating these packages.
To avoid duplication and ensure data integrity and consistency in modelling process, we used part 1 as the core framework. The relevant packages and classes from parts 2 and 5 were manually incorporated, preserving their full structure. During this process, duplicated classes were removed, and the links between them were meticulously checked to maintain schema consistency and integrity. After integrating these components, the model was organised into 6 packages, as illustrated in Figure 3. After designing the conceptual data model, the subsequent steps vary depending on whether an exchange format or a database is needed. XML files can be directly generated from the conceptual data models; however, for database implementation, the model must be converted into a language that is readable by DBMSs.

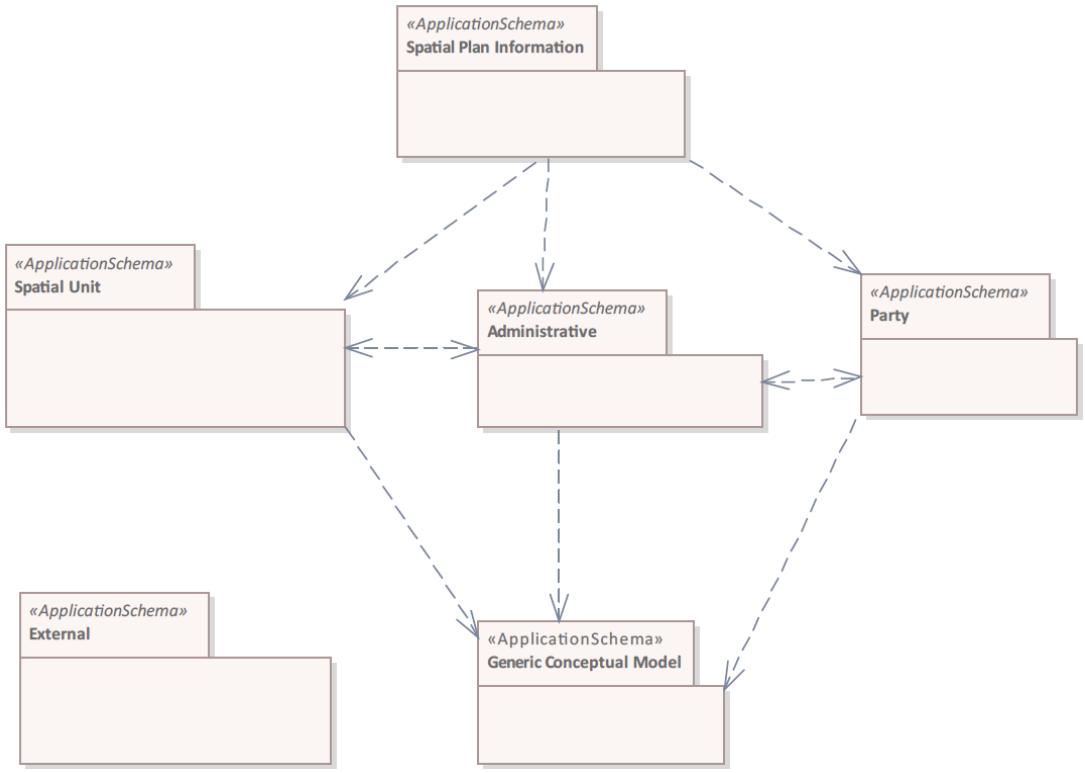

Figure 3. Integrated conceptual data model based on LADM Part 1, 2 and 3

5.2. Database implementation

After designing the conceptual model, the DBMS should be selected. Since cadastral data are mostly structured, a suitable option is to use relational data modelling and relational DBMSs. Among these, PostgreSQL/PostGIS and Oracle Spatial are the two most widely used options. We chose PostgreSQL because it supports spatial applications through its PostGIS extension and is free and open-source. Therefore, the conversion from the conceptual to the logical data model and the generation of Data Definition Language (DDL) codes in EA are based on the structure of PostgreSQL.

5.2.1. Logical design

There are several approaches for designing logical data models. As conceptual data models of ISO standards are implemented in EA, we selected this software for conceptual design. EA leverages transformation models to automatically convert UML class diagrams into DDL table diagrams. However, this process is not entirely automatic and requires some manual adjustments. Below is a step-by-step process for converting a conceptual data model into a logical data model using EA, including the challenges encountered during the transformation and suggested solutions.

- **Formalising Entities (Classes) into Relations (Tables)**: All classes are converted into tables. Association classes, such as *LA_PartyMember*, should be treated like concrete classes and converted into tables accordingly.
- **Converting Attributes to Columns/Tables**: There are three types of attributes including multivalued, composite, and derived attributes. EA does not automatically recognise these attributes, so they must be manually addressed.
    - **Multivalued Attribute:** For example, in the *LA_SpatialUnit* table, attributes like *area* (Figure 4a) may have multiple values, such as surveyed area and official area, which should be split into separate values. Thus, *area* is considered a multivalued attribute, necessitating the creation of a new table, *LA_AreaValue* (Figure 4b). In this case, the data type of FK was changed to integer (Figure 4c) and connected to the PK in *LA_AreaValue* (Figure 4d). The created FK is shown in Figure 4e.
    - **Composite Attribute**: Attributes like *extAddress* that can be divided into smaller parts such as building number, street number, street name, city, state, country, and postal code. These components should either be listed as separate attributes or treated as entities. In this case, *extAddress* should become a separate table to align with the LADM conceptual model. This approach also reflects practices in Victoria, where addresses are stored in a dedicated database like Vicmap Address, which can be considered an external class.
    - **Derived Attributes:** Attributes that can be calculated from other attributes. These attributes are usually not stored in the database because of data redundancy. For example, the *computeArea()* operation (Figure 4) can be used to calculate the area instead of storing it as a separate attribute.
- **Primary Key (PK)**: If a table has a clear primary key, such as *suID* in *LA_SpatialUnit*, it can be set as the PK. Otherwise, a surrogate key can be added. EA generates a default PK for each table, which often needs manual correction. If the table already has a PK, we remove the generated PK and assign the right attribute as the PK. If no PK exists, we retain the generated surrogate key.
- **Resolve Relationships**: Solve associations by adding foreign keys (FK) and associative entities. **Aggregation relationships** such as the relationship between *SP_PlanUnit* and *SP_PlanBlock* or *LA_Party* and *LA_GroupParty* are similar to standard associations when converting a conceptual to logical data model. **Reflexive associations**, like "update" association on *SP_PlanUnit*, are handled similarly (Figure 5).
    - **1:1 Relationships:** Adding an FK to the entity with total participation, if any, or to either side of the relation.
    - **1:M Relationships:** Adding FK to the *many* side of the relation. For example, the relationship between *LA_BAUnit* and *LA_RRR* is 1:M, so the FK is added to the *LA_RRR* table.

- **M:N Relationships:** Creating a new linking table, known as an *associative entity*, with two FKs, each referencing the PK of one of the related tables. In this case, the primary key of the associative table is typically a composite key formed by these two foreign keys, which is often referred to as a Primary Foreign Key (PFK) as illustrated in Figure 6. For instance, the relationship between *LA_BAUnit* and *LA_AdministrativeSource* requires an associative entity. However, some M:N relationships, like the *rrrSource* relationship between *LA_RRR* and *LA_AdministrativeSource*, may not be resolved automatically and need to be handled manually by creating a new table.

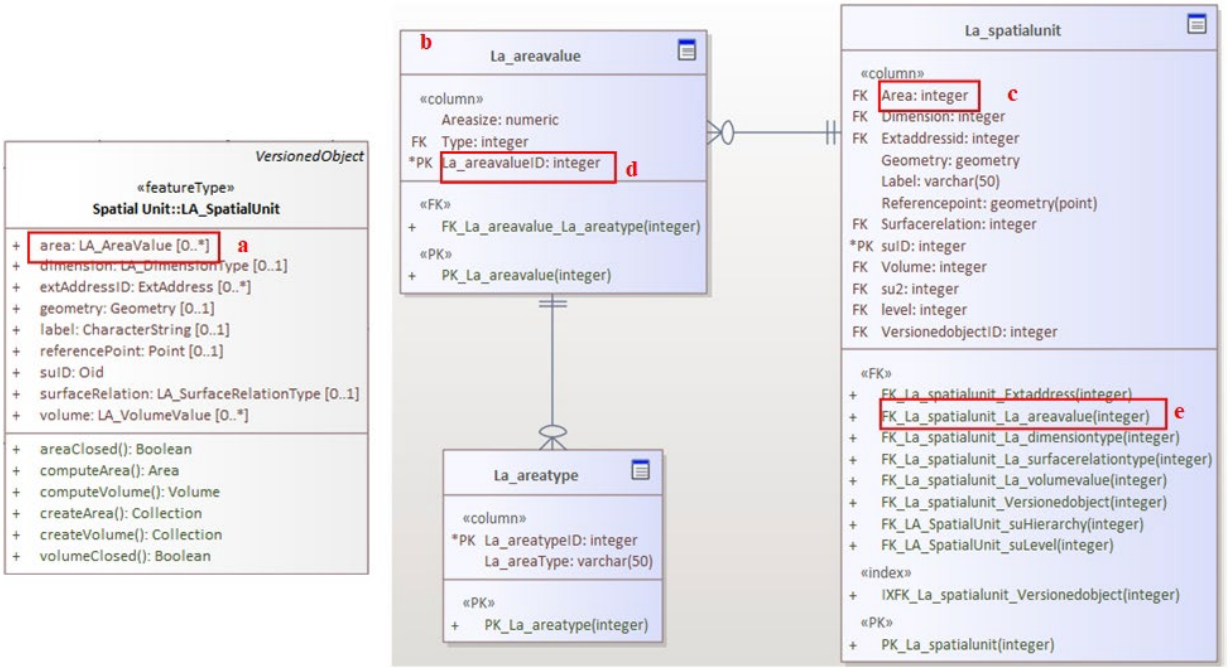

Figure 4. The process of resolving Multivalued attribute

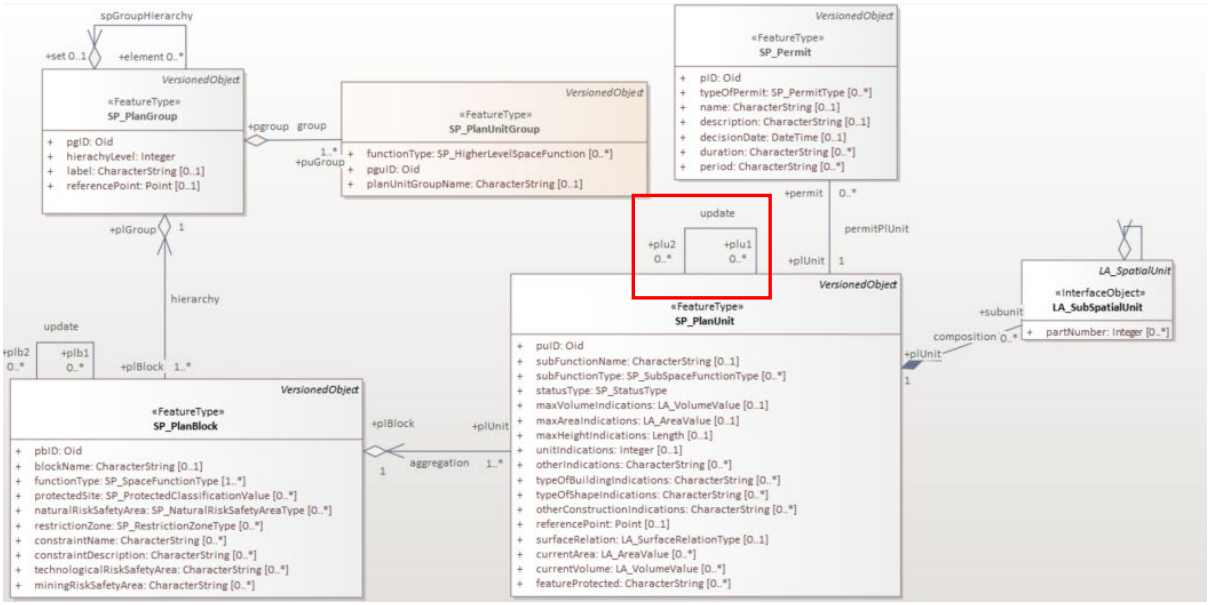

Figure 5. Example of many-to-many and Reflexive associations

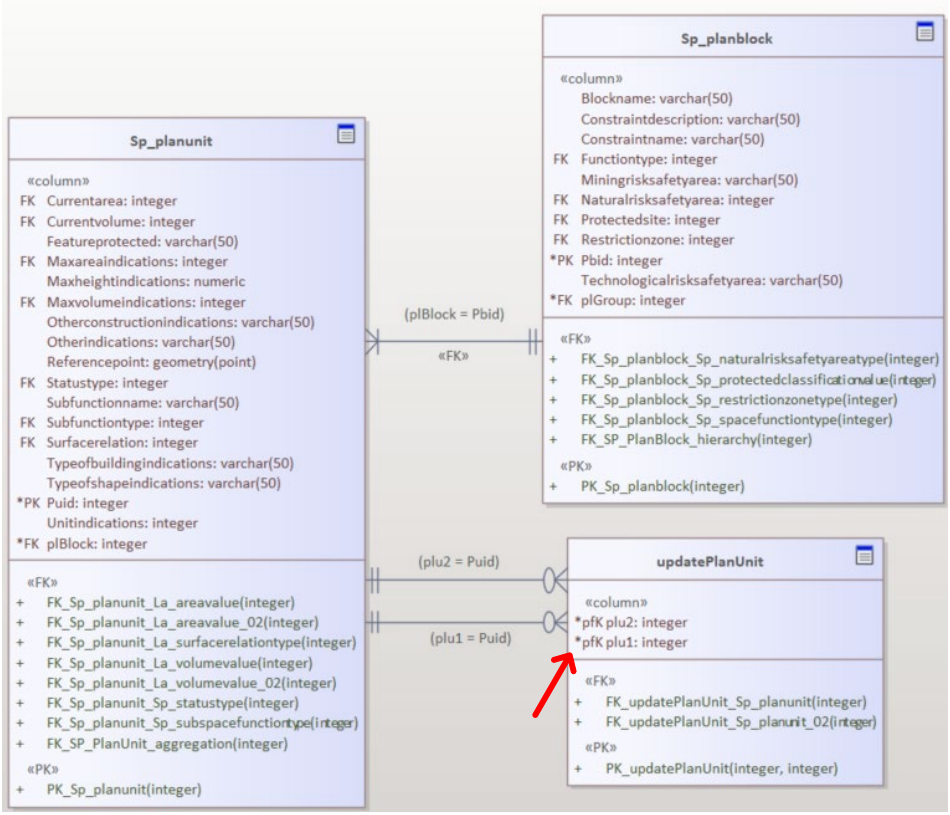

Figure 6. Resolving M:M relationship and the concept of PFK

- **Data Type**: For each attribute, it is essential to assign an appropriate data type. For spatial data, the appropriate geometry data types should be manually corrected, as EA's transformation model can not specify them. Dates and times, typically use the *timestamp* data type. **Generated data types**, such as *LA_VolumeValue*, often represent multiple values that should be stored in a separate table. However, the transformation model defines them as *varchar(50)* by default. The necessary steps are to delete the varchar(50), replace it with an integer, and designate this column as a FK that references the PK of the created table. It is important to ensure that the data type of the FK matches the PK data type of the related table.
- **Domain and Code Lists**: The range of values an attribute can take is known as its domain. There are two types of domains: enumerated and primitive types. Primitive types are similar to *varchar(50)*, which 50 is the default domain in EA. Enumerated types correspond to attributes with a limited set of values. The relevant examples in LADM are code lists. The transformation model creates tables for each code list and incorrectly converts values into columns. The solution is to create a new table with two columns: *ID* and *Type*. Then, insert the code list values into the *Type* column of this table (Figure 7). Alternatively, code lists can be managed in the user interface and limited through programming, or by using functions supported by some DBMSs for enumeration, which allows handling enumeration without creating new tables.

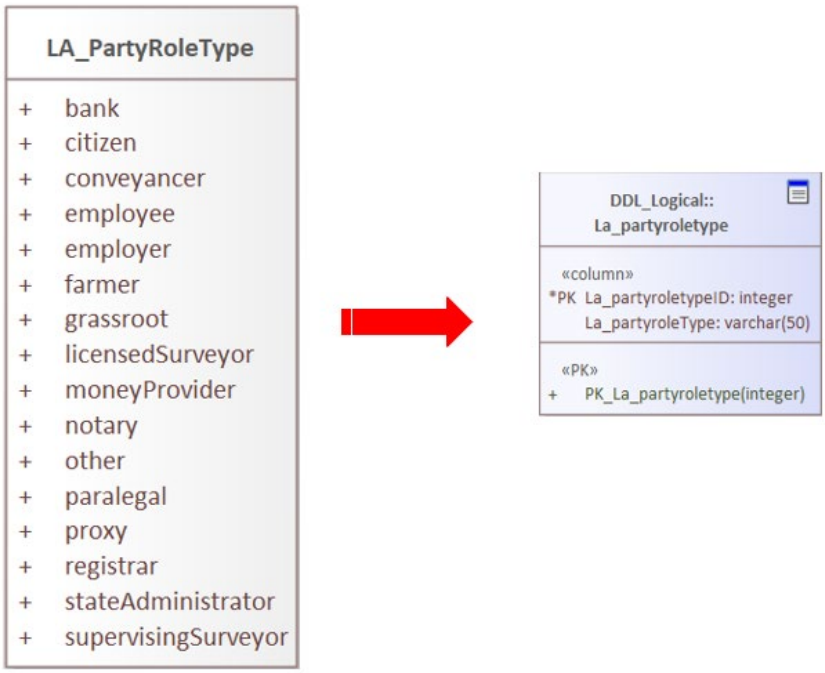

Figure 7. Solution for converting code lists

- **Inheritance/Generalisation relationship:** There are four alternative solutions:
  - **Single Table Inheritance**: All instances of the abstract class and its subclasses can be stored in a single table, with a type field added to distinguish between the different subclasses.
  - **Class Table Inheritance**: A separate table is created for each subclass, including both the fields inherited from the abstract class and those specific to the subclass.
  - **Concrete Table Inheritance**: Create a table for the abstract class to hold common fields, and separate tables for each subclass that include only their specific fields, with foreign keys linking to the abstract class table.
  - **Object-Oriented Inheritance in RDBMSs:** In RDBMSs like PostgreSQL that support object-oriented features, inheritance can be used to create a hierarchical structure of tables.
- **Multiplicity / Cardinality / Participation:** There are limitations in logical and physical data models when implementing these constraints on relationships. The implementation of these constraints varies depending on the DBMS. For instance, MySQL supports participation constraints, whether partial or total. For attribute multiplicity, it is possible to limit them through domain definitions, enumeration, and not-null constraints. For example, multiplicity values like 1 and 1..* can be managed by applying a not-null constraint to attributes that must have more than one value. However, there is not always a direct method to enforce these constraints using standard constraints alone. An alternative is to use SQL coding and triggers to mandate the required values. Another solution for handling attribute multiplicity is to create a new table for attributes which can have multiple values like 0..*. For instance, the multiplicity of the *volume* attribute in *LA_SpatialUnit* is [0..*]. We treat such attributes as multivalued and create a separate table to accommodate them.

- **Operations (Methods / Constraints):** Transformation model does not implement operations such as *areaClosed()* in *LA_SpatialUnit* or "count(part)+count(element)>0" in *LA_SpatialUnitGroup*. These operations must be defined by developing functions and triggers in the DBMS.
- **Normalisation:** Normalisation is a technique used to iteratively improve relations to remove undesired redundancy by decomposing relations and eliminating anomalies such as deletion, insertion and update. In First Normal Form (1NF), multivalued attributes and repeating groups will be resolved. In Second Normal Form (2NF), all the partial dependencies are resolved. The next stage is Third Normal Form (3NF) where all the transitive dependencies will be resolved.
- **Indexing:** Indexes are automatically created when defining PKs and FKs. However, it is necessary to manually implement spatial indexes in a DBMS that supports spatial data by using SQL codes.

5.2.2. Physical design

After resolving all issues in the logical data model, physical model can be implemented into the targeted DBMS. There are two approaches to designing the physical data model. Table 1 illustrates the advantages and disadvantages of each approach.

Table 1. Comparison of approaches for implementing physical data models

Generating SQL Codes	Using Database Builder in EA
Advantages: • **Flexibility and full control over SQL code.** • **Allows precise customisation and correction.** • **Independence from EA.** • **Allowing management in any DBMS.**	**Advantages:** • Automates direct changes in the DBMS. • Seamless integration with DBMS for real-time updates. • Provides additional functionalities like views, functions, and queries, and user-friendly interface. • Requires no advanced database knowledge. • Integrates conceptual, logical, and physical models in a single environment.
Disadvantages • **Requires manual modifications.** • **Time-consuming** • **Higher risk of human error during manual editing.** • **Less efficient for large projects.** • **Required more advanced SQL knowledge**	**Disadvantages:** • Requires manual modifications. • Less control over the SQL code. • Dependency on EA. • Complex initial setup and connection process. • Requires a corporate or higher license for EA, which increases costs.

Generating SQL Codes: The first approach we implemented involves extracting SQL code from the DDL codes generated by EA's transformation model, manually correcting any issues in the SQL code as discussed in section 5.2.1, and then executing it in the PostgreSQL DBMS Query Tool.

Using Database Builder in EA: The second approach utilises the Database Builder in EA, which can be directly connected to the DBMS. Any required changes to tables or data are automatically executed in the DBMS. To connect EA to the database, there are several options including a direct native connection or an ODBC-based connection. We used native

connection to link EA to PostgreSQL. Once connected, a database package is created, which includes conceptual, logical, and database packages. It is important to note that the results of converting the conceptual to the logical data model using the transformation models (generated DDL tables) must be moved to database package to allow for manipulation. In addition, there are some functionalities such as views, functions, sequences and queries that facilitate the database management. Figure 8 shows a view of the tables created under the Database Builder in EA.

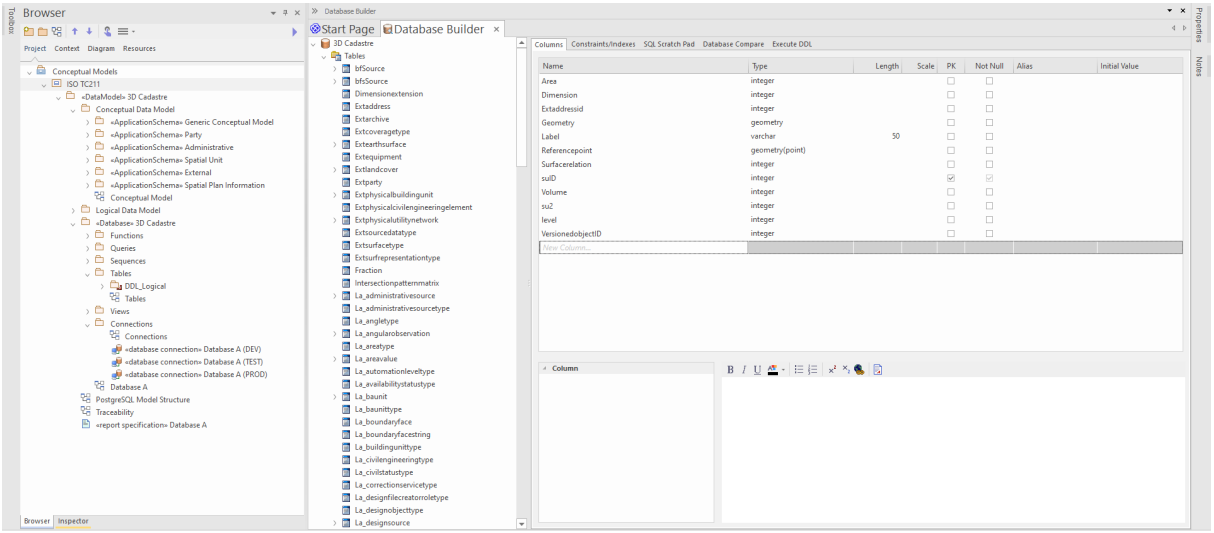

Figure 8. Connecting Enterprise Architect to Database Management System

5.3. Exchange formats – physical data modelling

Implementing exchange formats is crucial for ensuring that cadastral data can be efficiently shared and understood across different systems. For implementing the exchange formats and their physical data models, the first step is having a conceptual data model, which we developed in section 5.1. Then, EA allows us to export this conceptual data model's structure to formats like XSD or XML. These formats provide the foundation for the physical data model and the implementation of exchange formats across different systems. The reason we implemented XML is that ePlan is currently stored in XML format. Moreover, we are interested in implementing JSON because ICSM is developing a 3D conceptual data model for Australia that uses JSON as its format [29]. However, once the project is finalised, this model will need to be customised for use in Victoria.

After exporting the XML, we validated it using an XML validator to ensure its accuracy and consistency. Following validation, there are several options for converting XML to JSON. Since XML is a well-known format supported by most software, it can easily be converted to other formats, such as JSON, and utilised in various projects. The importance of implementing exchange formats like XML and JSON lies in their ability to standardise data, enabling seamless communication and interoperability between various software and platforms. Figure 9 shows a part of the XML and JSON files that were implemented.

```xml
<?xml  version='1.0' encoding='utf-8' ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="bfSource" type="bfSource"/>
    <xs:complexType name="bfSource">
        <xs:sequence>
            <xs:element name="bf" type="xs:integer" minOccurs="1" maxOccurs="1"/>
            <xs:element name="source" type="xs:integer" minOccurs="1" maxOccurs="1"/>
            <xs:element name="PK_La_boundaryface" type="La_boundaryface" minOccurs="1" maxOccurs="1"/>
            <xs:element name="PK_La_spatialsource" type="La_spatialsource" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="bfsSource" type="bfsSource"/>
    <xs:complexType name="bfsSource">
        <xs:sequence>
            <xs:element name="bfs" type="xs:integer" minOccurs="1" maxOccurs="1"/>
            <xs:element name="source" type="xs:integer" minOccurs="1" maxOccurs="1"/>
            <xs:element name="PK_La_boundaryfacestring" type="La_boundaryfacestring" minOccurs="1" maxOccurs="1"/>
            <xs:element name="PK_La_spatialsource" type="La_spatialsource" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="Dimensionextension" type="Dimensionextension"/>
    <xs:complexType name="Dimensionextension">
        <xs:annotation>
            <xs:documentation>The dimension extension type of topology, see ISO 19107:2019</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="DimensionextensionID" type="xs:integer" minOccurs="1" maxOccurs="1"/>
            <xs:element name="DimensionextensionType" type="xs:string" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
```


```
{
    "schema": {
        "element": [
            {
                "_name": "bfSource",
                "_type": "bfSource",
                "__prefix": "xs"
            },
            {
                "_name": "bfsSource",
                "_type": "bfsSource",
                "__prefix": "xs"
            },
            {
                "_name": "Dimensionextension",
                "_type": "Dimensionextension",
                "__prefix": "xs"
            },
            {
                "_name": "Extaddress",
                "_type": "Extaddress",
                "__prefix": "xs"
            },
            {
                "_name": "Extarchive",
                "_type": "Extarchive",
                "__prefix": "xs"
            },
            {
                "_name": "Extcoveragetype",
                "_type": "Extcoveragetype",
                "__prefix": "xs"
            },
            {
                "_name": "Extearthsurface",
                "_type": "Extearthsurface",
                "__prefix": "xs"
            },
```


Figure 9. Implementation of XML and JSON files

6. DISCUSSION

In this section, an assessment of the database implementation and exchange formats (XML and JSON) at the schema level is conducted. Table 2 provides a summary comparison of database schema and exchange format schema (XML, JSON) across various dimensions. Then, the challenges encountered during the implementation will be examined.

- **Conceptual data model integration**

The first step in the process is designing a conceptual data model. Given that LADM II is divided into different parts, each functioning as a separate standard with some overlapping elements, a key challenge is integrating these packages while maintaining consistency and integrity. We addressed this challenge by manually copying the full structure of packages to the core package (in our case, core package is Part 1), ensuring that all connections and links were preserved. However, it is crucial to perform a complete structure copy; otherwise, the integrity and links between the packages could be compromised. Avoiding duplication, maintaining consistency, and ensuring interoperability at the schema level are essential when integrating various standards and data models. However, the challenge of effectively integrating conceptual data models across different systems still remains.

- **Challenges in converting conceptual data model into logical data model.**

Another issue is the challenges in converting a conceptual data model to a logical data model. Currently, there is no fully automated method for this conversion, and many steps still need to be done manually. There are numerous challenges, such as when using software like Enterprise Architect, which automatically creates primary keys and some relationships but struggles with handling certain many-to-many relationships. It can generate indexes but not spatial indexes, and it does not automatically recognise spatial and user-generated data types, constraints, or multivalued and composite attributes.

Table 2. Overview of database schema and exchange format differences

Criteria	Database Schema	Exchange Format Schema (XML, JSON)
Conceptual Model	Both methods share the same conceptual model, ensuring consistency at the conceptual level.	Both methods share the same conceptual model, ensuring consistency at the conceptual level.
User Interface and Ease of Use	Comes with a user-friendly interface that allows easy access and editing of tabular data.	Lacks a native user interface; requires external tools for editing and manipulation
Data Entry	Can be performed manually or automatically using SQL commands or a database interface. Manual entry of geometrical data is particularly challenging.	More challenging without specialised tools; External tools or XML editors are often needed to manage data entry efficiently.
Semantic and Linked Data	Limited. Traditional relational databases are not inherently designed for semantic data or linked data, but they can be extended to support semantic structure. Some DBMSs, such as Oracle, support semantic technologies. Additionally, there are triple stores that support RDF; however, it is necessary to investigate whether they also support 3D spatial data.	XML is naturally compatible with RDF and other semantic web standards. It is easier to convert XML to RDF or import XML to platforms that support semantic formats.
Integrity	Strong data integrity with enforced integrity constraints.	Weaker data integrity.
Flexibility	Changes to schema can be complex, because of constraints	Schemas can be modified more easily

- **Linked data and semantic consistency**

When conceptual model based on LADM is converted into a database, the direct connection with the original model is severed, resulting in a set of tables with independent names that no longer automatically reflect changes made to LADM, particularly when database tables are not linked to the evolving LADM terminology. This can lead to potential inconsistencies or need ongoing updates and alignments. The literature lacks discussion on the semantic implications of converting LADM entities into database tables, which can result in a loss of meaning and alignment with the original standard. To address these issues, the adoption of linked data, ontology, and semantic web principles could enhance semantic consistency, interoperability, and data integration, ensuring that the database and exchange formats remain relevant and connected to evolving standards.

- **Distributed Cadastral Systems in Victoria**

Another issue is that, in Victoria, as in other states or countries, the cadastre is highly distributed across various projects, departments, and agencies. The challenge lies in connecting these disparate systems, especially since there is no centralised database encompassing all of them. An alternative solution is to use the unique identifier in each system. For example, VOTS uses volume and folio method for indexing, which can be linked

to spatial cadastral databases through lot and plan numbers and cross referencing with volume and folio numbers [24]. Similarly, the Standard Parcel Identifier (SPI) can be used in the DCDB, and the SPEAR reference number aligns with the SPEAR identifier. Although unique identifiers are accessible in each of these systems that allow us to link and read their data, the problem is that these databases do not connect seamlessly. Integration remains difficult unless a web service from them or access to their online platforms or similar resources is available.

- **Optimisation at schema level**

Another gap is the lack of emphasis on database optimisation strategies. For instance, creating separate tables for code lists increases the overall number of tables, which complicates database management and administration. This approach also leads to more frequent use of *JOIN* functions in queries, which can negatively impact performance and require more advanced SQL knowledge to manage them efficiently. It is essential to consider alternative approaches to improve the quality of database by optimisation strategies.

7. CONCLUSION AND FUTURE WORKS

In this paper, we investigated the Victorian cadastral system and aligned it with LADM II. Then, an integrated data model of LADM Parts 1, 2 and 5 was developed and the challenges in converting the conceptual model to logical and physical models using Enterprise Architect, proposing solutions. We also discussed two approaches for database implementation: using the Database Builder and generating SQL codes. Additionally, we implemented exchange formats based on XML, which is the format used in the current version of ePlan, and JSON, which is a potential future exchange format for Australia. In conclusion, while there are automated tools available for converting conceptual models into logical and implementation models, some issues still require manual intervention. LADM II, particularly Parts 1, 2, and 5, shows some alignment with the Victorian cadastral systems. However, the new version of LADM introduces additional challenges, such as integrating its different parts to avoid data redundancy and maintain consistency and integrity.

There are several gaps in the literature that can be addressed in future works. While our current implementation covers key aspects at the schema level, it is crucial to fully populate all relevant tables to evaluate the database's performance and efficiency at the instance level. Additionally, there is a gap in automatically converting conceptual data models into logical and physical data models. To address this, developing a program that can automate this conversion with less issues is essential. Furthermore, using linked data and semantic web technologies could help connect different conceptual data models more effectively. Storing file formats in both relational and document databases, followed by a comparison of results, is another area that requires exploration.

REFERENCES

1. Aien, A., *3D cadastral data modelling*. 2013, Department of Infrastructure Engineering, The University of Melbourne: Australia.

2. Olfat, H., B. Atazadeh, F. Badiee, Y. Chen, D. Shojaei, and A. Rajabifard, *A Proposal for Streamlining 3D Digital Cadastral Data Lifecycle.* Land, 2021. **10**(6): p. 642.
3. Kalogianni, E., E. Dimopoulou, and P. van Oosterom. *3D Cadastre and LADM-Needs and Expectations towards LADM Revision*. In *Proceedings of the 7th Land Administration Domain Model Workshop*. 2018. International Federation of Surveyors (FIG).
4. SPEAR. *ePlan Overview*. 2024; Available online: https://www.spear.land.vic.gov.au/spear/pages/eplan/about/what-is-eplan.shtml (Accessed on 19 March 2024).
5. Vicmap. *Vicmap Property*. 2024; Available online: https://www.land.vic.gov.au/maps-and-spatial/spatial-data/vicmap-catalogue (Accessed on 24 Jan 2024).
6. Shahidinejad, J., M. Kalantari, and A. Rajabifard, *3D Cadastral Database Systems—A Systematic Literature Review.* ISPRS International Journal of Geo-Information, 2024. **13**(1): p. 30.
7. Lemmen, C., P.J. van Oosterom, A. Kara, E. Kalogianni, A. Shnaidman, A. Indrajit, and A. Alattas. *The scope of LADM revision is shaping-up*. In *8th Land Administration Domain Model Workshop*. 2019.
8. ISO, *ISO 19152:2012, Geographic Information–Land Administration Domain Model (LADM)*. 2012: International Organization for Standardization (ISO), Geneva, Switzerland.
9. Kara, A., C. Lemmen, P. van Oosterom, E. Kalogianni, A. Alattas, and A. Indrajit, *Design of the new structure and capabilities of LADM edition II including 3D aspects.* Land Use Policy, 2024. **137**: p. 107003.
10. ISO, *ISO 19152-1:2024, Geographic information - Land Administration Domain Model (LADM) - Part 1: Generic conceptual model*. 2024: International Organization for Standardization (ISO), Geneva, Switzerland.
11. Sparx Systems. *Enterprise Architect (version 16.1)*. 2024; Available online: https://sparxsystems.com/ (Accessed on 03 August 2024).
12. Alattas, A., P. Van Oosterom, and S. Zlatanova. *Deriving the technical model for the indoor navigation prototype based on the integration of IndoorGML and LADM conceptual model*. In *Proceedings of the 7th International FIG Workshop on the Land Administration Domain Model, Zagreb, Croatia*. 2018.
13. Zulkifli, N.A., A.A. Rahman, H. Jamil, T.C. Hua, T.L. Choon, L.K. Seng, C.K. Lim, and P. van Oosterom. *Development of a Prototype for the Assessment of the Malaysian LADM Country Profile*. In *FIG Congress 2014, Engaging the Challenges – Enhancing the Relevance. Kuala Lumpur, Malaysia, 16-21 June.* 2014.
14. Alattas, A., P. van Oosterom, S. Zlatanova, A.A. Diakité, and J. Yan. *Developing a database for the LADM-IndoorGML model*. In *6th International FIG Workshop on 3D Cadastres*. 2018. International Federation of Surveyors (FIG).
15. Alattas, A., P. van Oosterom, S. Zlatanova, D. Hoeneveld, and E. Verbree, *LADM-IndoorGML for exploring user movements in evacuation exercise.* Land Use Policy, 2020. **98**: p. 104219.
16. Zulkifli, N.A., A.A. Rahman, and P. Van Oosterom. *Developing 2D and 3D cadastral registration system based on LADM: illustrated with Malaysian cases*. In *Proceedings of the 5 th FIG Land Administration Domain Model Workshop*. 2013.

17. Zulkifli, N.A., A. Abdoel Rahman, and C.B. Siew. *Database Design and Development of 3D Cadastral Registration based on LADM*. In *8th International FIG Workshop on the Land Administration Domain Model. Kuala Lumpur, Malaysia, 1-3 October*. 2019.
18. Zulkifli, N.A., A.A. Rahman, and S.C. Bernad, *Design and implementation of 3D strata objects registration based on LADM–A case study in Malaysia.* Land Use Policy, 2021. **108**: p. 105497.
19. Nasorudin, N.N., M.I. Hassan, N.A. Zulkifli, and A.A. Rahman, *Geospatial Database for Strata Objects Based on Land Administration Domain Model (LADM).* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016. **42**: p. 329.
20. Višnjevac, N., R. Mihajlović, M. Šoškić, Ž. Cvijetinović, and B. Bajat, *Prototype of the 3D cadastral system based on a NoSQL database and a Javascript visualization application.* ISPRS International Journal of Geo-Information, 2019. **8**(5): p. 227.
21. Coronel, C. and S. Morris, *Database systems: design, implementation, & management*. 13 ed. 2017: Cengage Learning.
22. Land Victoria. *An introduction to Land Registry Services*. 2024; Available online: https://www.land.vic.gov.au/land-registration/first-time-here/what-we-do (Accessed on 03 August 2024).
23. Olfat, H., B. Atazadeh, D. Shojaei, and A. Rajabifard, *The feasibility of a BIM-driven approach to support building subdivision workflows—case study of Victoria, Australia.* ISPRS International Journal of Geo-Information, 2019. **8**(11): p. 499.
24. Kalantari, M., *Cadastral Data Modelling-A Tool for e-Land Administration*. 2008, The University of Melbourne: Australia.
25. SPEAR. *Surveying and Planning through Electronic Applications and Referrals*. 2024; Available online: https://www.spear.land.vic.gov.au/spear/index.shtml (Accessed on 03 August 2024).
26. Land Victoria. *SMES, Survey Marks Enquiry Service*. 2024; Available online: https://www.land.vic.gov.au/surveying/services/survey-marks-enquiry-service (Accessed on 03 August 2024).
27. DELWP, *Victorian Cadastral Surveys Practice Directives*. 2021, Department of Environment, Land, Water and Planning.
28. Surveyors Registration Board of Victoria. 2024; Available online: https://www.surveyorsboard.vic.gov.au/ (Accessed on 03 August 2024).
29. ICSM. *ICSM Conceptual Model for 3D Cadastral Survey Dataset Submissions*. 2022; Available online: https://icsm-au.github.io/3d-csdm-design/2022/spec.html (Accessed on 03 August 2024).

BIOGRAPHICAL NOTES

Javad Shahidinejad is a PhD candidate in urban land administration at the Centre for Spatial Data Infrastructures & Land Administration (CSDILA) at the University of Melbourne. His research is focused on designing and developing a 3D cadastral database. He has a strong background in national and international industry projects, holding positions like Geospatial Data Scientist, Big Data Analyst, Artificial Intelligence Specialist, and Database

Administrator. His academic research covers areas such as 3D Cadastre, Semantic Web, NSDI, Open Data, and Recommender Systems.

Mohsen Kalantari is an Honorary Associate Prof of Geospatial Engineering at the Department of Infrastructure Engineering at the University of Melbourne and is A/Prof at the School of Civil and Environmental Engineering of the University of New South Wales. His research covers the geospatial information value chain, including sourcing and capturing fit-for-purpose data, engineering data models for optimised storage and visualisation, organising and disseminating geospatial data and leveraging geospatial data by integrating it with other data. He brings a wealth of experience in developing successful research grant applications, engaging with the industry and profession, building strong research teams, conducting research in Australia and overseas, and leading a research centre.

Abbas Rajabifard is a Professor and Associate Dean Sustainability, and Discipline Leader of Geomatics, at the Faculty of Engineering and IT, the University of Melbourne. He is also Director of the Centre for Spatial Data Infrastructures and Land Administration. He has a strong track record in research and teaching, and academic leadership, and is internationally recognised scholar and engineer. His academic background is in Surveying and Mapping, Land Administration and Urban Systems modernisation, and has continued to maintain a high level of performance across the areas of research, teaching, supervision, and service to the surveying and spatial sciences.

CONTACTS

Javad Shahidinejad
Department of Infrastructure Engineering, The University of Melbourne,
VIC 3010, Australia
E-mail: javad.shahidinejad@student.unimelb.edu.au
Website: https://eng.unimelb.edu.au/csdila/people/graduate-researchers/current/javad-shahidinejad

Mohsen Kalantari
School of Civil and Environmental Engineering, University of New South Wales (UNSW)
NSW 2052, Australia
E-mail: mohsen.kalantari@unsw.edu.au
Website: https://www.unsw.edu.au/staff/mohsen-kalantari

Abbas Rajabifard
Department of Infrastructure Engineering, The University of Melbourne, Parkville, Victoria, Australia
VIC 3010, Australia
E-mail: abbas.r@unimelb.edu.au
Website: https://findanexpert.unimelb.edu.au/profile/6142-abbas-rajabifard