An Object-Oriented Graphic Library for Independent Cadastral Mapping Software Development

Huynh, T. P.
Hanoi University of Mining and Geology, 18 Vien street, Dong Ngac Ward, Hanoi 100000, Vietnam
E-mail: phamthehuynh@humg.edu.vn, ORCID: 0009-0009-8794-148X
DOI: https://doi.org/10.52939/ijg.v22i10.5240

Abstract
Cadastral mapping applications in Vietnam have traditionally relied on commercial graphic platforms such as AutoCAD and MicroStation for core graphic processing and application development. While these platforms provide mature graphic and extension capabilities, there remains a need to investigate independent software architectures tailored to cadastral-specific graphic and topology requirements. This study aims to design and functionally validate an object-oriented graphic library as a core software component for specialized cadastral mapping application development. A design-and-development approach was adopted, comprising cadastral requirement analysis, review of selected CAD graphic platforms, design of a layered object-oriented architecture, integration of established geometric and topological processing techniques, and prototype implementation. The proposed library integrates fundamental graphic entities, including points, lines, polylines, circles, arcs, text, and blocks, with cadastral-specific classes such as parcels, parcel labels, and map frames. Functional validation of the prototype covered graphic-object management, DXF data exchange, topology processing, parcel-region generation, cadastral map-frame generation, and cadastral annotation. The results show that these components could be integrated and executed within the prototype and produced the intended functional outputs. These findings provide proof-of-concept evidence for the functional feasibility of organizing fundamental graphic processing, cadastral-specific objects, and topology-processing functions within an independent object-oriented core library. The contribution of the study is therefore primarily architectural and implementation-oriented rather than the development of new geometric algorithms or a new cadastral information model. Further evaluation using large-scale cadastral datasets, quantitative performance metrics, professional user assessment, and interoperability with contemporary land-administration and geospatial standards is required to establish the scalability and practical applicability of the proposed architecture.

Keywords: Cadastral Information Systems, Cadastral Mapping, Digital Mapping Software, Graphic Object Library, Topology, Object-Oriented Design

1. Introduction

The rapid advancement of information technology and software engineering has supported the development of powerful computer-aided design (CAD) and geospatial systems, among which AutoCAD and MicroStation have been widely used in surveying, mapping, civil engineering, construction, and infrastructure management [1][2][3][4] and [5]. Their capabilities in graphic editing, geometric processing, spatial data handling, and functional extension have contributed substantially to the transition from conventional paper-based mapping to digital surveying and mapping workflows [3][4] and [5].

In Vietnam, cadastral mapping and related surveying applications have frequently been developed on AutoCAD- or MicroStation-based environments because of their established graphic-processing capabilities and support for application extensions [6] and [7]. However, reliance on proprietary CAD/GIS environments may constrain software customization, system integration, and adaptation to specific domestic cadastral workflows [8] and [9]. In sectors requiring greater control over software and data environments, dependence on foreign platforms may also raise concerns regarding technological autonomy and digital sovereignty [10]. These considerations create continuing demand for software architectures that can support cadastral mapping functions while reducing dependence on a particular proprietary graphic-processing environment.

Cadastral mapping has requirements that extend beyond general graphic editing. A cadastral system must manage not only fundamental geometric

entities such as points, lines, polygons, text, and symbols, but also domain-specific objects and relationships, including land parcels, parcel identifiers, cadastral annotations, map frames, parcel adjacency, containment, and topology [11] [12] [13] [14] and [15]. Earlier cadastral and GIS research has therefore emphasized spatial data models, cadastral databases, topology management, and the integration of spatial and attribute information [11] [14] and [15]. However, contemporary geospatial platforms have also evolved considerably. ArcGIS Pro Parcel Fabric, for example, provides parcel-specific data structures, geodatabase topology, parcel editing workflows, and programmatic access through the ArcGIS Pro SDK [16]. Open-source environments such as QGIS similarly support extensible geospatial processing and customized application development through PyQGIS and plugin mechanisms [17]. Consequently, modern cadastral software development can no longer be characterized simply as the extension of generic CAD graphic primitives; it increasingly involves specialized parcel-management functions, extensible GIS environments, and domain-aware spatial data structures.

In parallel, international land-administration research has increasingly emphasized standardized conceptual models and interoperable cadastral information structures. The second edition of the Land Administration Domain Model (LADM) provides an updated international framework for this purpose. ISO 19152-1:2024 defines the generic conceptual model and common structure of the land-administration domain, while ISO 19152-2:2025 addresses land registration and incorporates concepts related to spatial units, surveying, spatial representation, geometry, and topology [18] and [19]. The development and broader structure of LADM Edition II are further summarized by [20]. These developments extend earlier research on the adoption of LADM in cadastral information systems [15] and demonstrate the continuing movement toward standardized and interoperable approaches to land administration.

Recent studies also demonstrate that cadastral application development is increasingly moving toward open-source, standards-aware, and integrated digital environments. [21] developed a mobile cadastral application based on open-source technologies, illustrating the potential of non-proprietary tools for land-information management. [22] examined pathways for integrating BIM/IFC, crowdsourced information, and LADM in three-dimensional cadastral mapping, reflecting the broader evolution toward interoperable and multi-source cadastral systems. These studies complement recent work on three-dimensional cadastral databases [11] and [14] and indicate that contemporary cadastral research increasingly addresses data models, interoperability, three-dimensional representation, mobile and Web-based environments, and integration with broader land-administration infrastructures.

Interoperability has consequently become an important consideration in geospatial software architecture. OGC API - Features provides a standardized mechanism for accessing and querying feature-based geospatial resources through Web-oriented interfaces [23], while the OGC GeoPackage Encoding Standard provides an open and platform-independent container for storing and exchanging geospatial features, associated attributes, and other spatial information [24]. These standards illustrate a broader shift toward separating application functionality from proprietary data structures and improving portability across heterogeneous software environments. At the software-component level, general-purpose geometry kernels such as Open CASCADE similarly demonstrate the value of reusable object-oriented components for separating geometric processing functions from application-specific implementations [25].

Object-oriented software engineering provides an appropriate foundation for such modular development because data structures and their associated processing functions can be encapsulated within reusable and inheritable object classes [26]. In cadastral applications, this principle can be extended beyond generic geometric entities by defining specialized classes for parcels, parcel labels, map frames, and related topological operations. Such an approach allows fundamental graphic-processing and cadastral-specific functions to be organized as reusable software components rather than being embedded entirely within a particular application or proprietary CAD/GIS platform. Despite these advances, an important distinction remains between comprehensive cadastral information systems and reusable core graphic libraries for cadastral software development. Contemporary GIS environments provide increasingly sophisticated parcel-management, topology, extension, and spatial-data capabilities [16] and [17], while LADM provides standardized conceptual models for land administration [18] [19] and [20], and recent studies have advanced open-source and three-dimensional cadastral solutions [21] and [22]. However, comparatively limited attention has been given to the design of a cadastral-oriented graphic-library architecture in which fundamental geometric entities, cadastral-specific objects, and topology-processing functions are integrated as an independent core

software layer specifically for specialized cadastral mapping application development. This software-architecture gap constitutes the primary motivation for the present study.

Accordingly, this study aims to design and functionally validate an object-oriented graphic library that integrates fundamental graphic entities, cadastral-specific objects, and geometric and topological processing functions within an independent core software architecture for specialized cadastral mapping application development. The proposed library adopts the DXF format as its primary data-exchange mechanism, maintaining compatibility with established CAD-based workflows while implementing graphic processing and cadastral-specific functions within an independent object-oriented software architecture.

The study makes three main contributions. First, it proposes an object-oriented graphic library architecture specifically designed for cadastral mapping, in which fundamental graphic entities and cadastral-specific objects are organized within a unified class hierarchy. Second, the proposed library integrates cadastral-specific graphic objects with geometric and topological processing functions, including parcel construction, spatial relationship management, and cadastral map generation, within a unified software framework. Third, the study demonstrates the functional feasibility of implementing the proposed library as an independent core software layer for cadastral mapping applications without requiring a proprietary CAD platform as the primary graphic-processing environment. Accordingly, the novelty of this study lies primarily in the software architecture and integration of cadastral-specific graphic and topology components, rather than in the development of new geometric algorithms, a new cadastral information model, or a new GIS platform.

2. Research Methodology

This study followed a design and development research (DDR) methodology, consisting of four main stages: (1) reviewing existing graphic systems, (2) analyzing cadastral mapping requirements, (3) designing an object-oriented graphic object library and integrating established geometric and topological processing techniques, and (4) implementing and validating a prototype system.

Reviewing and analyzing graphic systems that are widely used in digital mapping applications, particularly AutoCAD and MicroStation;

Analyzing the functional and operational requirements of the Vietnamese cadastral mapping system to identify the necessary object classes and system functionalities; Designing a graphic object library based on object-oriented programming principles and integrating established geometric and topological processing techniques for cadastral spatial-data management;

Developing an experimental application using the Visual Basic programming language to evaluate the functional integration and operational feasibility of the designed components. Figure 1 illustrates the research workflow adopted in this study.

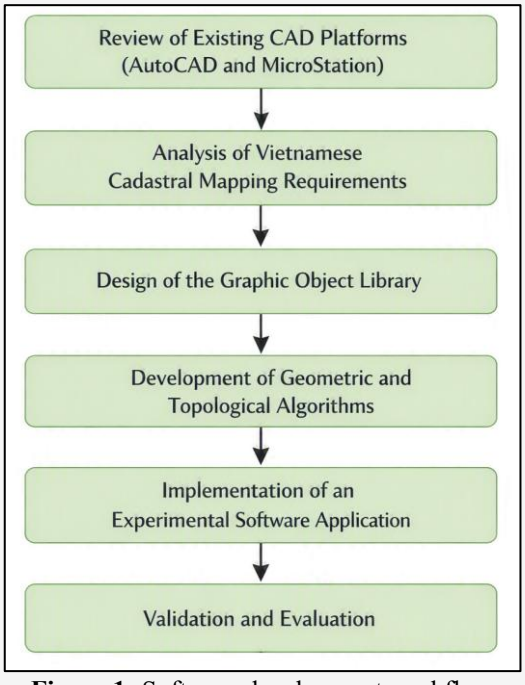

Figure 1: Software development workflow

2.1 System Requirements Analysis

2.1.1 Characteristics of cadastral maps

Cadastral maps are the most fundamental documents in cadastral records, representing the location, shape, area, parcel number, and land-use type of each land parcel for land administration purposes. They are used in land registration, issuance of land-use right certificates, land statistics, land-use planning, and the resolution of land disputes [27] and [28]. Unlike many other types of thematic maps, cadastral maps require a high level of geometric accuracy and must simultaneously manage both spatial data and attribute data for each land parcel [29]. Each parcel is represented not only by its geometric shape but also by associated information such as parcel number, area, land-use type, and other administrative attributes [30]. In addition, land parcels on a cadastral map must ensure continuity, closure, and non-overlap; therefore, topological relationships among objects play a particularly important role [14]. Due to the parcel-based nature of land administration, software systems for cadastral map production must

support the accurate management of spatial objects, cadastral attributes, and topological relationships among objects [28] and [31]. This requires the development of specialized graphic objects such as land parcels, parcel labels, and cadastral map frames, rather than relying only on general graphic objects such as points, lines, or polygons [11] and [12]. This provides the basis for developing a specialized graphic object library to support cadastral mapping software development.

2.1.2 Requirements for the graphic system

According to the Software Requirements Specification for Cadastral Mapping Solution [32], digital cadastral mapping requires appropriate support for graphic feature representation, standardized symbology and layer organization, coordinate-based spatial data handling, map editing, and standardized data output. Based on these requirements and the functional needs of the proposed cadastral mapping environment, the present study further considers object selection, search, snapping, printing, and data exchange as operational requirements of the graphic library. To meet these requirements, the graphic system should be built upon an object library capable of providing unified management of graphic objects, attributes, and spatial relationships. Such a library should not only support basic graphic operations, including object creation, visualization, and editing, but also accommodate cadastral-specific requirements such as the management of land parcels, parcel labels, cadastral map frames, and topological relationships. Therefore, the design of the graphic object library architecture constitutes a fundamental component in the development of cadastral mapping software. Table 1 summarizes the principal functional requirements of the proposed cadastral graphic system. These requirements form the design criteria for the proposed graphic object library architecture, which is presented in the following section to support the unified management of graphic objects, cadastral entities, and topological relationships.

2.2 Design of the Graphic Object Library Architecture

2.2.1 Review of existing graphic systems

To identify suitable design principles for the proposed graphic object library, two CAD-based graphic platforms widely used in digital mapping workflows in Vietnam, AutoCAD and MicroStation, were examined. The analysis focused specifically on their core graphic-layer characteristics, including data organization, rendering mechanisms, object management, editing functions, and support for software extension. These two platforms were selected because they have historically provided the graphic-processing environments on which many surveying and cadastral mapping applications in Vietnam have been developed [6] and [7]. The comparison is therefore intended to inform the design of the proposed core graphic library rather than to provide a comprehensive comparison with contemporary GIS or land-administration platforms such as QGIS, ArcGIS Pro Parcel Fabric, or LADM-based systems.

According to Table 3, both AutoCAD and MicroStation provide mature mechanisms for managing and editing graphic objects and support application extension through programming interfaces. Their approaches to graphic-object organization, coordinate handling, rendering, and object manipulation therefore provide useful architectural references for the development of an independent graphic library. In the present study, these characteristics were examined primarily from a software-architecture perspective rather than through quantitative performance comparison. As shown in Table 2, all three environments support fundamental graphic entities, whereas the proposed library implements cadastral-specific entities and topology functions directly within its core object architecture.

Table 1: Functional requirements of the cadastral graphic system

Requirement group	Description
Object Management	Point, Line, Polyline, Text, Object selection and search
Attribute Management	Layer, color, line style
Graphic Editing	Copy, Move, Rotate, Trim, Coordinate input and snapping
Topology Management	Polygon creation, spatial relationships
Cadastral Functions	Land parcels, parcel labels, cadastral map frames
Data Export	DXF export, map printing
Display	Zoom, Pan, Window Zoom

Table 2: Comparison of selected core graphic-layer features of the proposed library, AutoCAD, and MicroStation

Feature	Proposed Library	AutoCAD	MicroStation
Fundamental point object	✓	✓	✓
Fundamental line object	✓	✓	✓
Cadastral parcel object implemented as a core library class	✓	Not native to the core graphic layer	Not native to the core graphic layer
Parcel-label object implemented as a core library class	✓	Not native to the core graphic layer	Not native to the core graphic layer
Cadastral topology functions integrated into the proposed core library	✓	Requires application-specific extension/integration	Requires application-specific extension/integration
Cadastral map-frame object implemented as a core library class	✓	Not native to the core graphic layer	Not native to the core graphic layer

Table 3: Design Characteristics of the CAD graphic platforms examined

Design characteristic	AutoCAD	MicroStation	Relevance to proposed library
Graphic-object management	Supported	Supported	Basis for object-oriented entity management
Digital map editing	Supported	Supported	Reference for core editing functions
Programming/API extension	Supported	Supported	Reference for modular software organization
Separation of graphic data and application functions	Supported through platform architecture	Supported through platform architecture	Informs layered architectural design
External data exchange	Supported	Supported	Informs compatibility-oriented data exchange

This distinction reflects the design objective of the proposed library rather than a claim of functional superiority over AutoCAD or MicroStation, both of which can support cadastral applications through customization and extension mechanisms. The analysis of these platforms provided architectural references concerning graphic-object organization, object manipulation, and software extensibility. Building on these principles, the proposed library organizes fundamental graphic entities, cadastral-specific objects, and geometric and topological processing functions within a unified object-oriented framework specifically intended for cadastral mapping application development.

The comparative review identified several architectural characteristics relevant to the proposed library, particularly object-based graphic management, modular software extension, separation of graphic data from application functions, and support for external data exchange. These characteristics, together with the cadastral-specific requirements identified in Section 2.1, informed the design of the proposed object-oriented graphic library. The resulting architecture integrates fundamental graphic entities, cadastral-specific objects, and geometric and topological processing functions within a unified software framework.

2.2.2 Proposed graphic object library architecture

Based on the analysis of the selected CAD graphic platforms and the cadastral mapping requirements identified in Section 2.1, the proposed graphic object library was designed using a layered architecture. The architecture separates the application layer,

graphic object library, geometric and topological processing functions, and data-management components. This separation provides a modular structure intended to support functional extension, software maintenance, and the development of specialized cadastral mapping applications. Figure 2 illustrates the overall layered architecture and the relationships among these components.

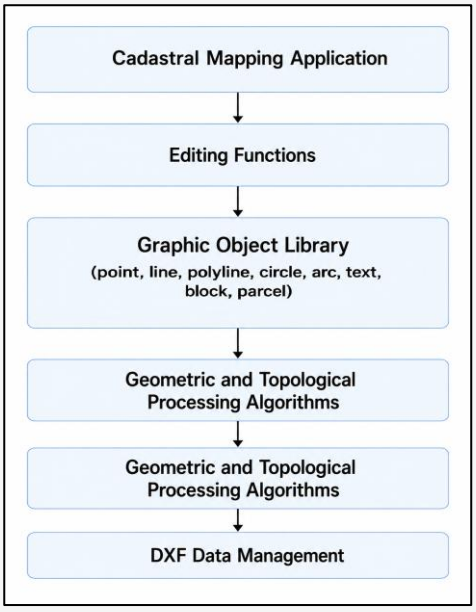

Figure 2: Layered architecture of the proposed graphic object library

The proposed architecture is organized into a set of functional layers that separate data management, geometric and topological processing, graphic-object management, and application-level functions. The DXF Data Management layer supports the storage and exchange of digital map data. The Geometric and Topological Processing layer performs geometric computations, intersection detection, closed-region construction, and management of spatial relationships. The Graphic Object Library constitutes the core of the architecture and contains both fundamental graphic entities such as points, lines, polylines, circles, text, and blocks, and cadastral-specific objects required for cadastral mapping. At the application level, editing and cadastral mapping functions provide tools for creating, modifying, and managing cadastral map data. The separation of these components is intended to limit dependencies between application-specific functions and core graphic-processing components and to facilitate the extension of individual classes or processing functions without requiring changes to the overall architectural structure. This layered structure therefore provides the architectural basis for the prototype implementation evaluated in this study.

2.3 Design of the Graphic Object Classes

The graphic object library was designed using an object-oriented programming (OOP) approach in which each object class encapsulates the attributes and methods associated with a specific graphic or cadastral entity. A common base class, Entity, defines the shared properties and operations required by the graphic objects, while derived classes represent fundamental geometric entities, annotation objects, and cadastral-specific objects. This hierarchical organization provides a consistent structure for managing different object types within the library [26]. In the prototype implementation, the graphic object classes were compiled into a Dynamic Link Library (DLL), enabling the core graphic components to be separated from application-level cadastral functions. Based on the functional requirements identified in Section 2.1, the library includes both fundamental graphic classes and cadastral-specific classes. Table 4 summarizes the principal classes and their primary responsibilities, while Figure 3 presents the class hierarchy and selected class attributes using UML class-diagram notation.

Table 4: Principal graphic object classes and their responsibilities

Object Class	Primary Responsibility
Point	Represents and displays point objects
Line	Represents and manages line segments
Polyline	Represents and manages polylines
Circle	Represents and manages circles
Arc	Represents and manages circular arcs
Text	Displays and manages text annotations
Block	Manages map symbols (blocks)
Entity	Provides a generic base class for graphic objects
Parcel	Represents and manages cadastral parcels
Parcel Label	Represents and manages parcel annotation information
Map Frame	Represents and manages cadastral map frames

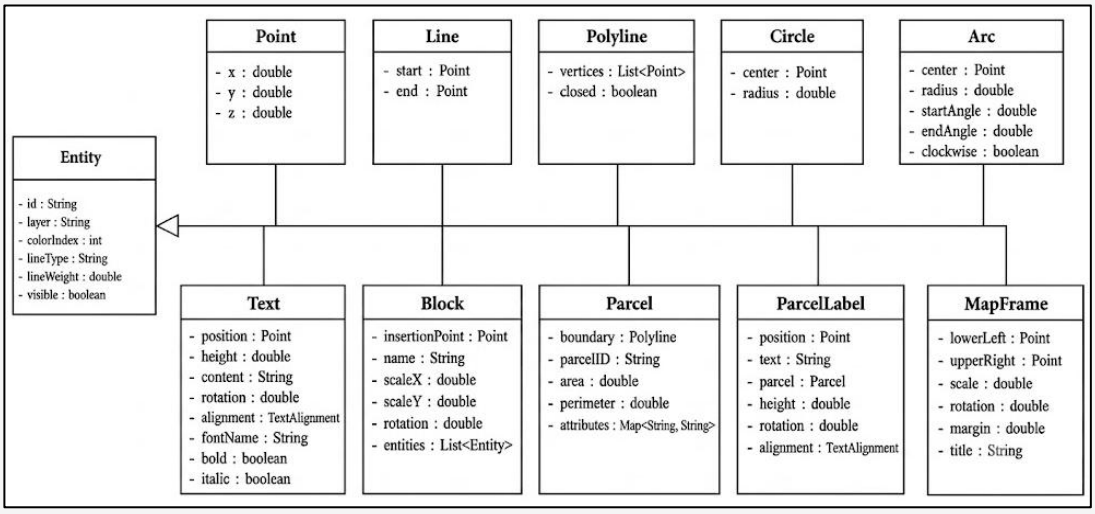

Figure 3: UML class diagram of the proposed graphic object library

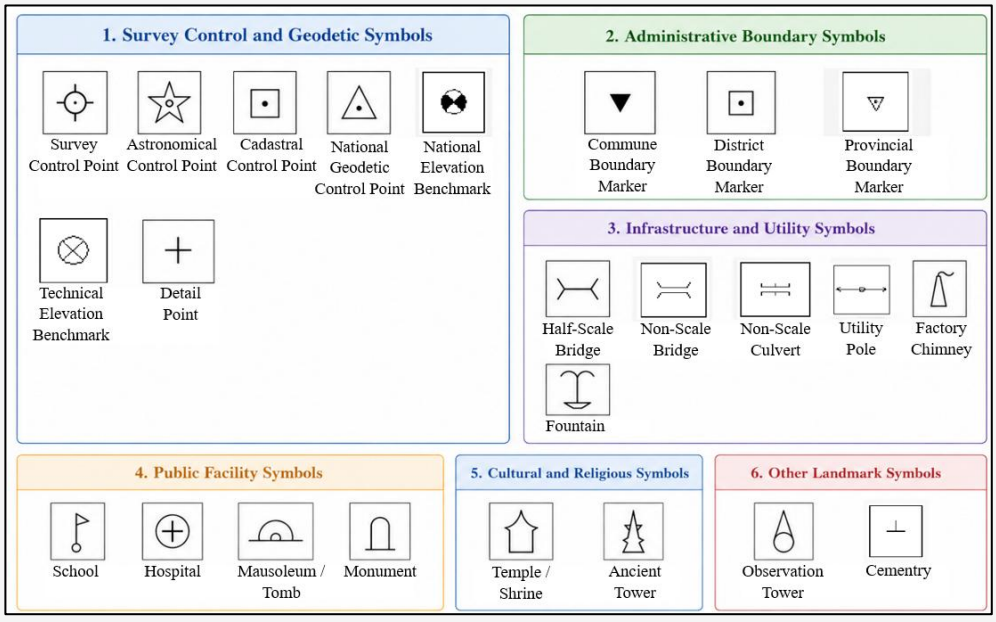

Figure 4: Representative cadastral and cartographic symbols implemented in the proposed graphic object library

As illustrated in Figure 3, the Entity class serves as the common base class of the graphic-object hierarchy and contains attributes shared by the derived object classes. The diagram also shows selected class-specific attributes for Point, Line, Polyline, Circle, Arc, Text, Block, Parcel, ParcelLabel, and MapFrame. Fundamental geometric classes include Point, Line, Polyline, Circle, and Arc; Text and Block provide annotation and symbol-related functions; and Parcel, ParcelLabel, and MapFrame represent cadastral-specific and map-related entities. In the prototype implementation, these classes are organized within the same object-oriented library to support object identification, editing, rendering, and subsequent geometric or topological processing. Figure 3 therefore clarifies both the common Entity-based class structure and the cadastral-specific specialization of the proposed library. Figure 4 presents representative graphic symbols implemented in the proposed library, including survey control points, administrative boundary markers, infrastructure facilities, public facilities, and cultural landmarks. These symbols illustrate how reusable cadastral and cartographic representations can be incorporated into the object library in addition to fundamental geometric entities. As summarized in Table 5, the proposed library distinguishes fundamental geometric objects from annotation, cadastral-specific, and map-related objects.

Table 5: Classification of graphic object classes in the proposed library

Category	Object Classes
Fundamental geometric objects	Point, Line, Polyline, Circle, Arc
Annotation and symbol objects	Text, Block
Cadastral-specific objects	Parcel, Parcel Label
Map-related objects	Map Frame

This classification reflects the domain-oriented design of the library by incorporating Parcel, ParcelLabel, and MapFrame directly into the object architecture rather than treating them solely as application-level constructs. These classes provide the object-level foundation for subsequent cadastral operations. In particular, cadastral parcel management requires geometric and topological operations for identifying intersections, containment, adjacency, and closed regions. The geometric and topological processing functions used to establish these relationships are described in Section 2.4.

2.4 Design of the Topology-Construction Workflow

A key component of the proposed graphic object library is the integration of geometric, search, and topological processing algorithms into a unified workflow for constructing and managing cadastral parcel topology. In cadastral maps, graphic objects are spatially related through intersections, containment, adjacency, and closed-region relationships. Explicit representation of these relationships is required to construct parcel regions from graphic elements and to support subsequent cadastral data management and spatial operations. The topology-construction workflow incorporates six principal algorithmic functions:

- QuickSort;
- Binary search;
- Line intersection detection;
- Point-in-Polygon;
- Closed-region construction;
- Adjacent-region identification.

These algorithms are established computational and geometric techniques rather than newly developed algorithms in this study. Their role in the proposed library lies in their integration within a cadastral-specific processing workflow. QuickSort and Binary Search support the organization and retrieval of graphic objects, while Line Intersection Detection identifies intersections required for topology-node generation. Point-in-Polygon supports containment testing, and the Closed-Region Construction and Adjacent-Region Identification functions are used to construct parcel regions and establish relationships among neighboring regions. Figure 5 illustrates how these functions are combined in the topology-construction process.

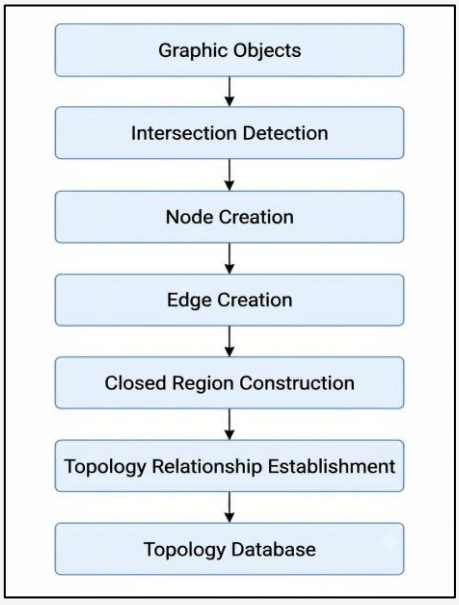

Figure 5: Topology construction process

As illustrated in Figure 5, the topology construction process begins with identifying the intersections among line objects and generating topology nodes. The resulting nodes and line segments are then used to construct topology edges and identify closed regions representing candidate parcel areas. Once closed regions have been established, spatial relationships such as containment and adjacency are identified and associated with the corresponding cadastral objects. The resulting topology information is subsequently used by the library to support parcel construction and related cadastral operations. Table 6 summarizes the functional roles of the principal algorithms and processing functions within this workflow. The table is intended to clarify how the individual techniques contribute to the integrated topology-construction process rather than to present them as novel algorithms. The integrated workflow provides the geometric and topological processing required by the Parcel and related cadastral object classes described in Section 2.3. In the present study, validation focuses on whether these functions can be successfully integrated and executed within the proposed graphic object library; quantitative comparison of computational performance, processing time, or scalability against alternative

topology-processing implementations is outside the scope of the prototype validation.

2.5 Prototype Validation

The proposed graphic object library was validated through its integration into a prototype cadastral mapping application developed on the Microsoft Windows platform. The purpose of the prototype was to assess the functional feasibility and internal integration of the proposed architecture rather than to conduct a comparative performance benchmark against existing CAD/GIS platforms. Validation therefore focused on whether the principal components designed in Sections 2.2-2.4 could be successfully implemented and executed within an independent cadastral mapping application. The validation covered four principal functional areas: (i) graphic-object creation, management, and editing; (ii) DXF-based data import and exchange; (iii) geometric and topological processing, including intersection detection, closed-region construction, and identification of spatial relationships; and (iv) cadastral-specific operations, including parcel representation, parcel labeling, and cadastral map-frame generation. These functional areas were selected because they correspond directly to the principal architectural components and cadastral-specific object classes defined in the proposed library.

Each functional area was assessed by executing the corresponding operations in the prototype and verifying whether the expected functional output was produced. Successful validation was defined as the ability of the prototype to execute the implemented function without functional failure and to produce an output consistent with the intended cadastral operation. For example, graphic-object validation examined whether supported entities could be created, displayed, selected, and edited; DXF validation examined whether supported graphic data could be imported and represented by the corresponding library objects; topology validation examined whether line intersections and closed regions could be identified and used to establish parcel relationships; and cadastral-function validation examined whether parcel-related objects and cadastral map frames could be generated and managed through the prototype.

The validation was therefore functional and proof-of-concept in scope. It was intended to demonstrate that the proposed object-oriented architecture, cadastral-specific object classes, and integrated topology-processing workflow could operate together within a working prototype. Quantitative benchmarking of execution time, memory consumption, computational scalability, and comparative performance against AutoCAD, MicroStation, QGIS, ArcGIS Pro, or other cadastral/GIS platforms was not undertaken and is outside the scope of the present validation. The corresponding implementation and functional validation results are reported in Section 3. The current prototype and its source code are not publicly available, as the implementation remains a research proof-of-concept rather than a released open-source software package.

3. Results and Evaluation

3.1 Prototype Implementation and Functional Results

The proposed graphic object library was implemented and integrated into a prototype cadastral mapping application to verify the architectural and functional components described in Section 2. Table 7 summarizes the principal software modules implemented in the prototype. The modules include fundamental geometric objects, cadastral-specific objects, geometric and topological processing functions, display and navigation functions, and graphic attribute management. Together, these modules constitute the implemented software components used for the functional validation described in Section 2.5.

Figure 6 presents the graphical user interface (GUI) of the prototype application. The interface integrates the implemented drawing, editing, visualization, topology-processing, and cadastral mapping functions within a single application environment. The main menu provides access to five functional groups: File, Format, Draw, View, and Topology, through which the implemented library functions can be executed and visually inspected.

Table 6: Roles of algorithms and processing functions in the topology-construction workflow

Algorithm/processing function	Role in the proposed workflow
QuickSort	Organizes graphic objects for subsequent processing
Binary Search	Retrieves graphic objects from organized object collections
Line Intersection Detection	Identifies intersections and supports topology-node generation
Point-in-Polygon	Determines containment relationships between points and polygonal regions
Closed Region Construction	Constructs closed regions from connected line objects
Adjacent-Region Identification	Identifies adjacency relationships between constructed regions

Table 7: Principal software modules implemented in the prototype

Module	Main Objects	Primary Functions	Software Role
Geometry	Point, Line, Circle, Arc, Text	Drawing	Basic graphic processing
Cadastral	Parcel, Parcel Label, Map Frame	Representation	Cadastral modeling
Topology	Region, Boundary	Region generation	Spatial relationship management
Display	Display Window	Display and Navigation	Visualization
Attribute	Layer, Color, Line Type	Graphic properties	Object management

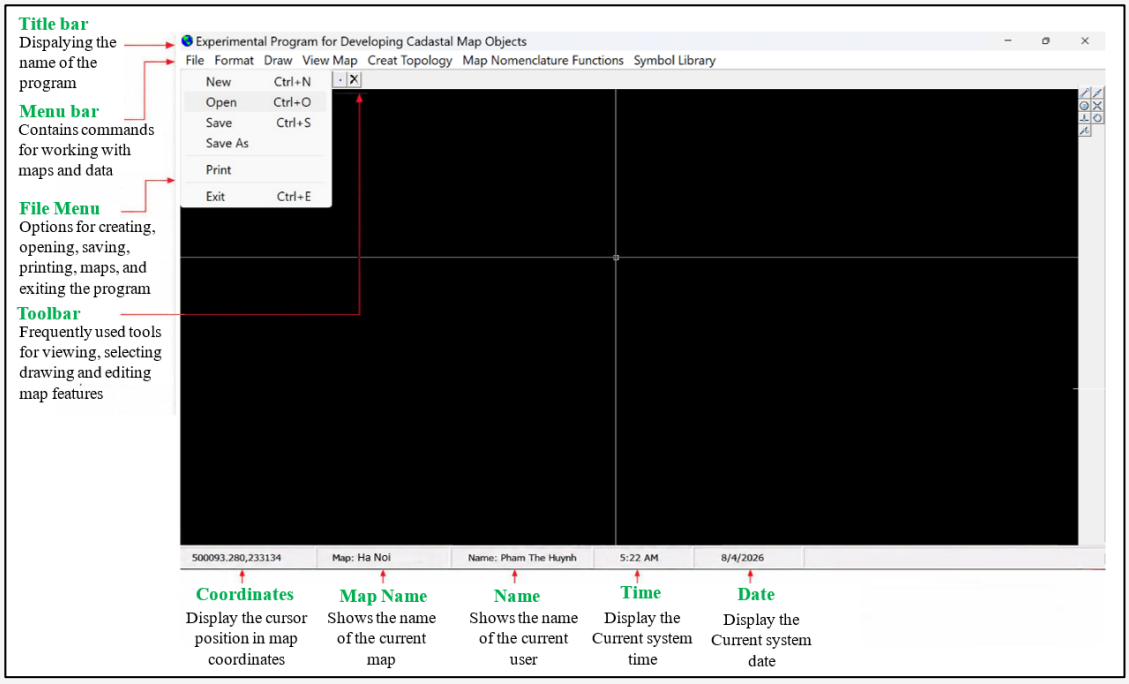

Figure 6: Prototype system interface and main functional components

Functional testing was conducted using representative cadastral mapping operations corresponding to the validation areas defined in Section 2.5. The tested operations included DXF import, graphic-object creation and editing, topology construction, parcel-region generation, cadastral map-frame generation, and cadastral annotation.

As shown in Figure 7, connected cadastral boundary lines were processed by the topology workflow to identify and construct closed regions representing candidate cadastral parcels. The generated regions could subsequently be used for parcel-related operations and the representation of spatial relationships among neighboring parcels. This result verifies the functional execution of the topology-construction workflow described in Section 2.4. Figure 8 shows the cadastral map frame generated by the prototype from the implemented map-frame functions. The output includes the principal components required by the prototype implementation, such as the map title, scale information, map-sheet designation, and frame-related annotations. Figure 9 presents the annotation table generated for small parcels for which textual information cannot be accommodated directly within the parcel boundary. The generated table provides fields for parcel identification and associated cadastral information and demonstrates the implementation of the corresponding cadastral annotation function.

3.2 Functional Validation of the Proposed Library

Functional validation was conducted according to the scope and criteria defined in Section 2.5. The purpose was to verify whether the principal library components could be integrated and executed within the prototype and whether each tested operation produced the intended functional output. Table 8 summarizes the validation operations and observed results. The validation results show that the tested components were integrated and executed within the prototype and produced the intended functional

outputs. Supported DXF data could be imported and represented within the library environment; fundamental graphic objects could be created and manipulated; the topology-processing workflow generated closed parcel regions from connected cadastral boundaries; and the cadastral-specific functions generated parcel-related objects, map-frame components, and annotation outputs. These observations provide functional evidence that the principal architectural components described in Sections 2.2-2.4 can operate together within the prototype. The results should, however, be interpreted as functional proof-of-concept validation rather than quantitative performance evaluation. The tests demonstrate implementation and integration of the proposed components but do not establish comparative superiority, computational efficiency, scalability, or production-level robustness relative to existing CAD/GIS platforms.

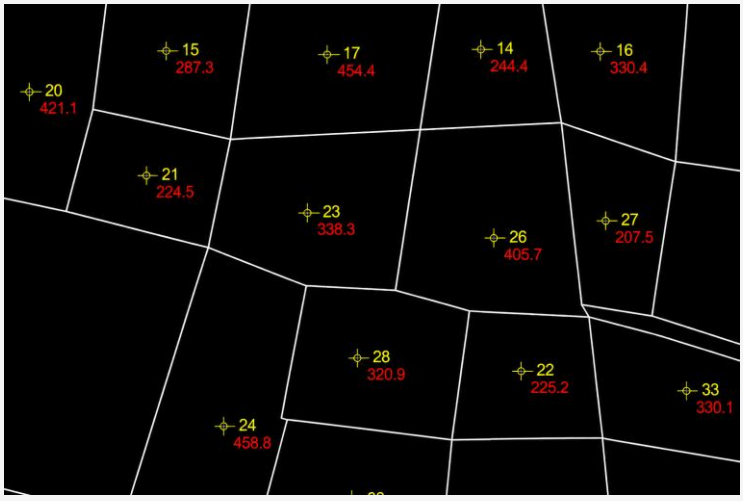

Figure 7: Topology-based generation of closed cadastral regions

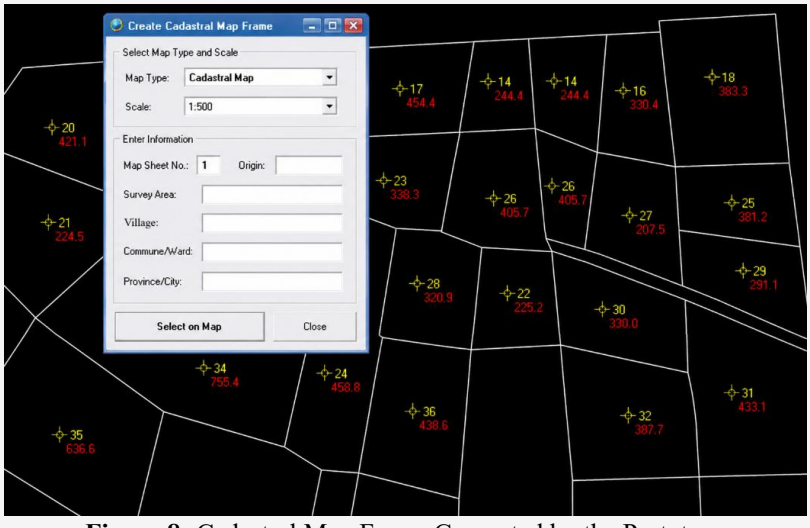

Figure 8: Cadastral Map Frame Generated by the Prototype

Annotation Table for Small Land Parcels

Parcel No.	79	84	103	121	136	146	165	
Area	93.5	102.3	131.6	99.3	113.4	108.5	124.1	
Building Type/ Land Use Type	LUC	HNK	HNK	LUC	LUC	HNK	ONT	

Figure 9: Automatically generated annotation table for small land parcels

Table 8: Functional validation results of the prototype

Functional area	Validation operation	Expected output	Observed result
Graphic-object management	Create, display, select, and edit supported graphic entities	Supported entities can be represented and manipulated in the prototype	Supported graphic entities created, displayed, selected, and edited
DXF data exchange	Import supported DXF graphic data	Imported entities are represented by corresponding library objects	Supported DXF data imported and displayed
Topology processing	Detect intersections and construct closed regions from connected boundaries	Topology nodes/edges and closed regions are generated	Closed regions generated
Parcel generation	Generate parcel objects from constructed regions	Closed cadastral parcel regions are represented by parcel objects	Parcel regions generated
Map-frame generation	Generate cadastral map frame	Implemented map-frame components are produced	Map frame generated
Cadastral annotation	Generate annotation for parcels requiring external labels	Parcel annotation information is generated	Annotation output generated

Table 9: Architectural characteristics of the proposed library in relation to the cad-based development approach examined in this study

Architectural aspect	CAD-based development examined in this study	Proposed library
Fundamental graphic entities	Provided by CAD core environment	Implemented within library
Cadastral-specific object classes	Can be implemented through application-specific customization/extensions	Implemented within core library architecture
Topology-processing functions	Added through application-specific implementation/integration	Integrated within proposed library workflow
Parcel-region construction	Depends on application-specific implementation	Implemented in prototype
Cadastral map-frame function	Depends on application-specific implementation	Implemented in prototype
Parcel annotation function	Depends on application-specific implementation	Implemented in prototype

3.3 Architectural Evaluation

In addition to functional validation, the implemented prototype was examined in relation to the architectural objective of the study: integrating fundamental graphic entities, cadastral-specific objects, and topology-processing functions within an independent core library. Table 9 summarizes the principal architectural distinctions between the proposed library and the CAD-based development approach considered in this study.

The comparison indicates that the principal distinction of the proposed approach lies in the location and organization of cadastral-specific functionality within the software architecture. In the CAD-based development approach examined in Section 2.2, cadastral-specific functions can be implemented through application-level customization or extensions to the underlying graphic environment. In the proposed architecture, Parcel, ParcelLabel, MapFrame, and the associated topology-processing functions are incorporated directly into the graphic object library and can therefore be accessed by applications built on the library. The prototype results demonstrate the feasibility of this architectural arrangement but do not establish that the proposed library is more computationally efficient, scalable, reusable, maintainable, or extensible than existing CAD/GIS platforms. These characteristics remain architectural design objectives that require quantitative software metrics, large-dataset testing, and comparative evaluation in future work.

4. Discussion

4.1 Significance of the Proposed Graphic Object Library

The principal contribution of this study lies in the software architecture used to integrate fundamental graphic entities, cadastral-specific objects, and geometric and topological processing functions within a unified core library. Previous studies have primarily addressed cadastral databases, GIS-based cadastral systems, three-dimensional cadastral models, and related spatial-data management issues [11][12][13][14] and [15]. More recent developments have further expanded the cadastral software landscape through standardized land-administration models, extensible GIS environments, and open-source cadastral applications [18][19][20] and [21]. Within this broader context, the present

study addresses a narrower software-engineering problem: the organization of fundamental graphic processing and cadastral-specific functionality as integrated components of an independent core library intended to support reuse in specialized cadastral applications.

The functional validation results demonstrate that the principal components of the proposed architecture can be integrated and executed within a working prototype. As reported in Section 3, the prototype supported graphic-object management, DXF data exchange, topology processing, parcel-region generation, cadastral map-frame generation, and cadastral annotation. These results provide proof-of-concept evidence for the feasibility of combining fundamental graphic entities, cadastral-specific object classes, and topology-processing functions within the same object-oriented architecture. They should not, however, be interpreted as evidence that the proposed library provides superior computational performance, scalability, maintainability, or reusability compared with existing CAD/GIS platforms.

The architectural contribution is particularly evident in the representation of Parcel, ParcelLabel, and MapFrame as cadastral-specific classes and in the integration of topology-processing functions within the library workflow. Contemporary platforms such as QGIS and ArcGIS Pro provide extensive mechanisms for geospatial processing, customization, and parcel management [16] and [17]. The proposed library is not intended to replace or outperform these platforms. Rather, it explores a different architectural level by implementing a cadastral-oriented graphic core that can serve as a software component for specialized application development. In this respect, the contribution is architectural and implementation-oriented rather than a new cadastral information model, GIS platform, or geometric algorithm.

This distinction is also relevant to international developments in land administration. LADM Edition II provides standardized conceptual structures for land-administration information, including land-registration and spatial components [18][19] and [20], whereas the present library operates primarily at the graphic-object and geometric/topological processing level. The two approaches therefore address different but potentially complementary layers of cadastral software architecture. Future mapping between the library's cadastral-specific object classes and LADM concepts could provide a pathway toward greater semantic interoperability without changing the core architectural contribution demonstrated in this study. From a software-engineering perspective, the layered organization and DLL implementation are intended to separate core graphic processing from application-specific cadastral functions and to support subsequent functional extension. However, because maintainability, reusability, and extensibility were not quantitatively evaluated, these characteristics should be regarded as design objectives and potential architectural benefits rather than empirically demonstrated performance advantages.

4.2 Application Potential and Interoperability

The proposed library has potential to serve as a common software component for specialized cadastral mapping applications that require direct control over graphic objects and cadastral-specific processing functions. Because fundamental geometric entities, cadastral objects, and topology-processing functions are organized within the same library, future applications could access these components without implementing the entire graphic-processing layer independently. This represents a potential software-development benefit of the architecture; however, the extent of code reuse, development-time reduction, and maintenance improvement remains to be quantified in future studies. Beyond the prototype functions demonstrated in this study, the object-oriented representation of cadastral entities provides a basis for future integration with spatial databases and land-information systems. Such integration would require explicit mechanisms for mapping library objects and their attributes to external data models and storage structures. In particular, alignment with LADM Edition II could enable the graphic and topology components of the proposed library to interact with standardized land-administration concepts and information structures [18][19] and [20].

Interoperability with contemporary geospatial ecosystems represents another important development direction. The current prototype uses DXF as its primary data-exchange mechanism, which supports compatibility with established CAD-based mapping workflows but does not by itself provide the broader interoperability required by modern distributed geospatial systems. OGC API - Features provides a standardized Web-oriented mechanism for accessing feature-based geospatial resources [23], while GeoPackage provides an open, platform-independent mechanism for storing and exchanging geospatial data [24]. Future support for such standards could extend the library beyond DXF-based exchange and facilitate integration with GIS, database, Web, and mobile environments.

Recent open-source cadastral applications also demonstrate the feasibility of combining open geospatial technologies for field and land-

information management [21], while research on three-dimensional cadastral mapping increasingly considers integration among BIM/IFC, cadastral information, crowdsourced data, and LADM [22]. These developments indicate that future cadastral software is likely to require interoperability across multiple data models, platforms, and representation environments. Accordingly, the proposed library should be viewed as a potential graphic and topology-processing component within a broader cadastral software ecosystem rather than as a complete land-administration system. This architectural choice therefore involves a practical trade-off: implementing an independent cadastral-oriented graphic core provides direct control over graphic objects and domain-specific processing, but the current prototype does not yet provide the broader interoperability, database, Web, and advanced GIS capabilities available in mature geospatial ecosystems. Future extensions may therefore include spatial-database connectivity, standardized geospatial data exchange, GIS integration, Web-based services, and three-dimensional cadastral representations. General-purpose geometry frameworks such as Open CASCADE also illustrate possible directions for extending geometric capabilities where more advanced two- and three-dimensional operations are required [25]. Such extensions, however, remain future development directions and were not implemented or evaluated in the present prototype.

4.3 Limitations and Future Work

This study has several limitations. First, the current implementation is a proof-of-concept prototype designed to validate the functional feasibility and internal integration of the proposed architecture rather than a fully operational cadastral production system. The implemented library focuses on fundamental graphic entities, cadastral-specific objects, DXF-based data exchange, and topology-processing functions. Advanced capabilities such as survey-data processing, cadastral-record management, spatial-database integration, GIS-based spatial analysis, and Web-based services were outside the scope of the present implementation.

Second, the prototype was not evaluated using large-scale cadastral datasets or quantitatively benchmarked against existing CAD/GIS platforms. Consequently, the present results do not establish comparative computational efficiency, memory performance, scalability, maintainability, reusability, or extensibility. Future evaluation should therefore employ representative cadastral datasets of different sizes and complexities and report quantitative metrics such as execution time, memory consumption, topology-processing performance, and scalability. Comparative testing against relevant CAD/GIS implementations would also provide a stronger basis for assessing the practical advantages and trade-offs of the proposed architecture.

Third, validation was limited to functional testing of the implemented prototype, and no user-oriented evaluation involving professional cadastral operators or software developers was conducted. Future studies should therefore examine usability, workflow compatibility, implementation effort, and developer experience in realistic cadastral production environments. Fourth, the present prototype relies primarily on DXF for data exchange and has not been evaluated for semantic or service-level interoperability with contemporary land-administration and geospatial standards. Future work should investigate mappings between the proposed cadastral object classes and LADM Edition II, support for open geospatial exchange mechanisms such as GeoPackage, and service-oriented access through standards such as OGC API - Features. Integration with spatial databases and modern GIS environments should also be examined.

Finally, the present implementation focuses primarily on two-dimensional cadastral mapping. Future development could extend the architecture toward three-dimensional cadastral representation, BIM/IFC integration, WebGIS, mobile applications, and cloud-based spatial-data management. These extensions should be accompanied by quantitative performance evaluation and application-level validation to determine whether the proposed architecture remains suitable as system complexity and data volume increase.

5. Conclusions

This study proposed and implemented an object-oriented graphic library architecture for specialized cadastral mapping application development. Based on the analysis of selected CAD graphic platforms and cadastral mapping requirements, the proposed architecture integrates fundamental graphic entities, cadastral-specific object classes, and geometric and topological processing functions within a unified core library. Rather than introducing new geometric or topology algorithms, the study contributes an implementation-oriented architecture that integrates established processing techniques with cadastral-specific objects and functions. Functional validation through a prototype demonstrated that the principal architectural components could be integrated and executed within a working application. The prototype supported graphic-object management, DXF data exchange, topology processing, parcel-region generation, cadastral map-frame generation,

and cadastral annotation. These results provide proof-of-concept evidence for the functional feasibility of the proposed architecture as a software foundation for specialized cadastral mapping applications. The current results do not establish comparative computational performance, scalability, reusability, maintainability, or extensibility relative to existing CAD/GIS platforms. Future work should therefore evaluate the architecture using large-scale cadastral datasets and quantitative software metrics, together with professional user evaluation. Further development should also address spatial-database integration, interoperability with LADM and open geospatial standards, GIS and Web-based integration, and three-dimensional cadastral representation.

References

[1] Aouad, G., Wu, S., Lee, A. and Onyenobi, T., (2013). *Computer Aided Design Guide for Architecture, Engineering and Construction.* Routledge, London. https://doi.org/10.4324/9780203878750.

[2] Ramirez, A., Schmidt, J. and Smith, D., (2015). *Technical Drawing 101 with AutoCAD 2016.* Stephen Schroff, The United States of America. https://doi.org/10.4324/9780203878750.

[3] Biholar, C., Carabas, I. and Manciu, F., (2018). *Topographic Works for the Achievement of a 3D Survey.* Journal of Young Scientist, Vol. 6. https://journalofyoungscientist.usamv.ro/index.php/scientific-papers/past-issues?id=545.

[4] Bohnenstiehl, K., (2002). Photogrammetry Software Review: Purchasing Photogrammetry Software Wading through a Sea of Choices. *GISVision Magazine,* 1-8. https://www.pvts.net/pdfs/ortho/photosftw_purch.pdf.

[5] Taszakowski, J., Litwin, U. and Dorož, A., (2018). Transformation of the Former Austrian Cadastre Map. Geomatics, *Landmanagement and Landscape.* 55-67. https://doi.org/10.15576/GLL/2018.1.55.

[6] Gam, N. T. H., Binh, P. D. and Thi, N. D., Application of MicroStation V8i Software and VietMap XM Software to Establish a Map of the Current State of Land Use in 2023 from the Cadastral Map of Hoang Van Thu Ward of Thai Nguyen City to Improve Land Management in the 4.0 Era. *TNU Journal of Science and Technology.* Vol. 228(07); 80-91. https://doi.org/10.34238/tnu-jst.7578.

[7] Hanh, T. H., Duong, N. T., Hoa, P. T. T. and Anh, T. V., (2024). Application of GIS to Build a Land Database in a Mountainous District of Son La Province, Vietnam. *GIS-IDEAS 2024.* https://gis-ideas.org/2024/GIS_IDEAS_e_proceedings.pdf.

[8] Raza, S., Gartner, C. and Waters, N., (2017). *Open Source, Enterprise GIS Solutions for GIS-CAD Integration in Land Surveying.* Geospatial World. https://geospatialworld.net/article/enterprise-gis-for-land-surveying/.

[9] Ismail, M. A., Muhamad Ludin, A. N. and Hosni, N., (2020). Comparative Assessment of the Unsupervised Land Use Classification by Using Proprietary GIS and Open Source Software. *IOP Conference Series: Earth and Environmental Science.* Vol. 540(1), https://doi.org/10.1088/1755-1315/540/1/012020.

[10] Fratini, S., Hine, E., Novelli, C., Roberts, H. and Floridi, L., (2024). Digital Sovereignty: A Descriptive Analysis and a Critical Evaluation of Existing Models. *Digital Society,* Vol. 3(3). https://doi.org/10.1007/s44206-024-00146-7.

[11] Shahidinejad, M. J., (2025). *Design and Implementation of a 3D Cadastral Database System (Doctoral Dissertation).* Available: University of Melbourne repository record. https://hdl.handle.net/11343/366865

[12] Babalola, S. O. and Kardam, M. S., Developing a Cadastral Information System for Part of Fadaman-Mada Area of Bauchi Metropolis for Sustainable Development. *ATBU Journal of Environmental Technology,* Vol. 4(1); 105-116.

[13] Qamilah, N., Hernandi, A., Suwardhi, D., Meilano, I., Reisa, R., and Krama, K. (2026). Voxel Based Web Visualisation for 3D Cadastre in Resource Constrained Urban Environments. *International Journal of Geoinformatics*, Vol. 22(7), 20–37. https://doi.org/10.52939/ijg.v22i7.5073.

[14] Shahidinejad, J., Kalantari, M. and Rajabifard, A., (2024). 3D Cadastral Database Systems - A Systematic Literature Review. *ISPRS International Journal of Geo-Information,* Vol. 13(1). https://doi.org/10.3390/ijgi13010030.

[15] Kalantari, M., Dinsmore, K., Urban-Karr, J., and Rajabifard, A., (2015). A Roadmap to Adopt the Land Administration Domain Model in Cadastral Information Systems. Land Use Policy, Vol. 49; 552-564. https://doi.org/10.1016/j.landusepol.2014.12.019.

[16] Esri, (2025). *ProConcepts Parcel Fabric. ArcGIS Pro 3.6 SDK for .NET, API Reference Guide.* Available: Esri ArcGIS Pro 3.6 SDK documentation.

[17] QGIS Project, (2025). PyQGIS Developer Cookbook. QGIS Documentation, Version 3.44. Available: QGIS 3.44 PyQGIS Developer Cookbook.

[18] International Organization for Standardization (ISO), (2024). Geographic Information - Land Administration Domain Model (LADM) - Part 1: Generic Conceptual Model. ISO 19152-1:2024. ISO 19152-1:2024. https://www.iso.org/obp/ui/fr/#iso:std:iso:19152:-1:ed-1:v1:en

[19] International Organization for Standardization (ISO), (2025). Geographic Information - Land Administration Domain Model (LADM) - Part 2: Land Registration. ISO 19152-2:2025. ISO 19152-2:2025. https://standards.iteh.ai/catalog/standards/cen/a74b0904-e6f8-406e-8cb1-74f723edadcd/en-iso-19152-2-2025

[20] Lemmen, C., van Oosterom, P., Kara, A. and Kalogianni, E., (2025). The Land Administration Domain Model: An Overview. *FIG Publication No. 84. International Federation of Surveyors (FIG), Copenhagen*, Denmark.

[21] Mora-Navarro, G., Femenia-Ribera, C., Terol, E. and Quiza-Neuto, C., (2025). Mobile Cadastral Application with Open-Source Software in Colombia. *ISPRS International Journal of Geo-Information,* Vol. 14(3); 96. https://doi.org/10.3390/ijgi14030096.

[22] Gkeli, M. and Potsiou, C., (2023). 3D Crowdsourced Parametric Cadastral Mapping: Pathways Integrating BIM/IFC, Crowdsourced Data and LADM. *Land Use Policy,* Vol. 131. https://doi.org/10.1016/j.landusepol.2023.106713.

[23] Open Geospatial Consortium (OGC), (2019). *OGC API - Features - Part 1: Core, Version 1.0, OGC 17-069r3*. OGC API - Features - Part 1: Core. https://docs.ogc.org/is/17-069r3/17-069r3.pdf.

[24] Open Geospatial Consortium (OGC), (2021). OGC GeoPackage Encoding Standard, Version 1.3.1, OGC 12-128r18. OGC GeoPackage Standard. https://www.geopackage.org/spec131.

[25] Open Cascade SAS, (2019). *Open CASCADE Technology 7.4.0: Overview Documentation.* Available: Open CASCADE Technology 7.4.0. https://occt3d.com/dev/doc/refman/html

[26] Bruegge, B. and Dutoit, A.H., (2010). *Object Oriented Software Engineering: Using UML Patterns and Java.* Prentice Hall.

[27] Perperidou, D. G., (2025). From the Person-Based Land Registries to the Parcel-Based Hellenic Cadastre: A Review on Securing Property Documentation, Land Administration, and Spatial Data Management in Greece. *Land,* Vol. 14(6). https://doi.org/10.3390/land14061138.

[28] Lemmen, C., (2012). *A Domain Model for Land Administration (Doctoral Dissertation).* University of Twente. Available: https://doi.org/10.3990/1.9789077029312.

[29] Hagemans, E., Unger, E.-M., Soffers, P., Wortel, T. and Lemmen, C., (2022). The New, LADM Inspired, Data Model of the Dutch Cadastral Map. *Land Use Policy,* Vol. 117. https://doi.org/10.1016/j.landusepol.2022.106074.

[30] Inan, H. I., (2015). Associating Land Use/Cover Information with Land Parcels Represented in LADM. *Land Use Policy,* Vol. 49; 626-633. https://doi.org/10.1016/j.landusepol.2015.04.015.

[31] Jamalzadeh, S. H. and Cheraghi, F., (2026). Land Guard: An Open-Source Enterprise GIS for Cadastral Parcel Mapping in Iran. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 10; 397-404. https://doi.org/10.5194/isprs-annals-X-4-W8-2025-397-2026.

[32] National Informatics Centre (NIC), (2009*). Software Requirements Specification for Cadastral Mapping Solution, Version 1.4.* Land Records Information Systems Division (LRISD), New Delhi, India.