

Article

Not peer-reviewed version

A GIS-Native Framework for Qualitative Place Models: Implementation and Evaluation

[Abdurauf Satoti](#)^{*} and [Alia I. Abdelmoty](#)^{*}

Posted Date: 19 September 2025

doi: 10.20944/preprints202509.1690.v1

Keywords: place models; qualitative spatial reasoning; GIS; geo-ontology; Qualitative Place Model



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

A GIS-Native Framework for Qualitative Place Models: Implementation and Evaluation

Abdurauf Satoti * and Alia I. Abdelmoty *

School of Computer Science and Informatics, Cardiff University, Cardiff, UK

* Correspondence: Satotiam@cardiff.ac.uk and AbdelmotyAI@cardiff.ac.uk

Abstract

Humans typically describe spatial location using names, hierarchies, and relative positions (e.g., *east of, inside*), yet mainstream GIS represents places primarily through geometric coordinates, rendering qualitative spatial queries computationally challenging. We introduce the *Qualitative Place Model* (QPM), a GIS-native framework that transforms standard boundary datasets and place layers into structured knowledge bases of *Qualitative Place Description* Qualitative Place Descriptions (QPDs). QPM provides a homogeneous representation whereby administrative units and physical places are treated uniformly as *Place* entities. The model materializes a compact set of local relations—hierarchical containment, directional neighbourhood, and optional proximity—that support rich inferences through sound logical operations (inverse relationships and per-predicate transitive closure). We implement QPM as an ArcGIS Pro toolbox that computes and persists QPDs within a geodatabase, with optional RDF/GeoSPARQL export for SPARQL querying. This implementation enables natural-language-style spatial queries such as "Where is x ?" or "Which places are north of x ?" within standard GIS workflows. Evaluation on Wales (UK) administrative, postal, and electoral hierarchies plus a comprehensive place layer demonstrates robust performance: QPM generated 95.8% of expected basic-place statements (52,821 places) and achieved 89.7–96.5% coverage across administrative hierarchies. All QPDs proved unique under our deterministic signature. Despite compact storage requirements, directional relations expand by more than an order of magnitude (10.6× overall expansion) under logical closure, demonstrating substantial inferential power from a minimal stored representation. QPM complements geometric GIS with an explainable qualitative layer that aligns with human spatial cognition while remaining fully operational within standard GIS environments.

Keywords: place models; qualitative spatial reasoning; GIS; geo-ontology; Qualitative Place Model

1. Introduction

Place is a multifaceted concept spanning sense of place, geographic location, and locale [1,2]. In everyday communication, humans answer *where?* questions qualitatively by combining names, hierarchies, and relative positions (e.g., "Cardiff University is east of the Wales Police Station, inside Cathays ward") [3,4]. Such descriptions rely on hierarchical containment (country → authority → ward) [5,6] and neighbourhood semantics (e.g., *inside, north of, touching*).

Mainstream Geographic Information Systems (GIS), however, model places primarily as *geometric* objects (points, lines, polygons) emphasising coordinates and boundaries [7,8]. While indispensable for precise mapping and quantitative analysis, these geometric data models inadequately support relational, human-centric place descriptions—the primary mode through which spatial knowledge is communicated [3,9]. This limitation is particularly significant because qualitative place references in natural language often lack precise or authoritative geometries (e.g., venues without published footprints, informal neighbourhoods), yet their *spatial facts* remain meaningful and actionable ("north of the station", "within Cathays").

A Geographic Information System (GIS) that represents and manipulates qualitative relations alongside geometry would enable such descriptions to be captured, queried, and reasoned over natively, rather than requiring externalization to specialized scripts or web-based knowledge graphs [4,10].

Authoritative boundary datasets (administrative, postal, electoral) inherently encode space as multi-level partitions. Conceptually, these form Discrete Local Irregular Grid Systems (DLIGS), providing robust, maintained frameworks for hierarchical containment and adjacency [11]. If these datasets could be utilized within GIS to *derive and store* qualitative relations (e.g., *inside*, *touches*, *north_of*) and link them to named places, then two significant capabilities would emerge: (i) natural-language place descriptions could be represented and queried within standard GIS workflows, and (ii) qualitative and quantitative representations could be jointly exploited—each compensating for the limitations of the other.

Qualitative Spatial Reasoning (QSR) provides calculi for topological and directional relations and has long advocated for cognitively faithful spatial representations [12,13]. Parallel research in place studies advocates *place-based GIS* that treats places as objects with attributes and relationships, rather than as isolated geometries [9]. However, these research directions often remain external to operational GIS: QSR models are frequently demonstrated on abstract datasets or external knowledge graphs, while most GIS platforms lack a homogeneous, in-platform model that (a) treats administrative units and physical places uniformly as *places*, (b) captures hierarchical and lateral semantics coherently across multiple hierarchies, and (c) implements end-to-end computation, storage, and querying of qualitative place descriptions within the GIS environment.

We present the *Qualitative Place Model* (Qualitative Place Model (QPM)), a GIS-native framework that transforms standard boundary layers and place layers into structured knowledge bases of *Qualitative Place Descriptions* (QPDs). QPM adopts a homogeneous representation: both administrative units and physical features are treated as *Place* instances, located through their unique containment paths within hierarchies and through neighbourhood relations at the same hierarchical level. The model computes and stores (i) *containment* (place $\rightarrow$ parent unit per hierarchy) and (ii) *directional neighbourhoods* (nearest neighbours in cardinal sectors for places; touching neighbours for units), then exposes these relations within the GIS environment (with optional export to RDF/GeoSPARQL for SPARQL queries).

This research makes four principal contributions to spatial information science:

1. **Homogeneous place formalism.** A unified, hierarchy-aware model where administrative units and named places share a single representational scheme supporting qualitative relations for human-centric spatial queries.
2. **Deterministic in-GIS computation.** Practical algorithms for parent assignment and directional neighbourhood computation with explicit contracts and deterministic tie-breaking, implemented within ArcGIS Pro.
3. **Compact storage with rich inference.** Minimal, deterministic stored relations expand substantially under logical inference (inverse relationships and transitive closure), enabling expressive queries without storage overhead.
4. **Empirical validation across real hierarchies.** Wales (UK) evaluation demonstrates unique QPDs, robustness across diverse hierarchical structures, and practical GIS querying support.

QPM complements geometric GIS with an explainable qualitative layer providing compact storage and substantial inferential power. The framework enables natural-language spatial queries ("Where is x?", "Which places are north of y?") within standard GIS workflows while augmenting rather than replacing geometric representations.

Section 2 reviews related work in qualitative spatial reasoning and place-based GIS. Section 3 provides the formal model definition. Section 4 presents the conceptual design and computational framework and details the GIS implementation. Section 5 reports empirical evaluation results; Section 6 discusses implications and limitations; Section 7 presents conclusions and future research directions.

2. Related Work

This section reviews approaches to qualitative modeling of place in general and in particular within GIS environments.

Qualitative place models and spatial knowledge graphs.

Research in qualitative spatial representation has developed sophisticated theoretical foundations and computational implementations that move beyond purely geometric approaches. Qualitative Spatial Reasoning (QSR) formalizes topological and directional relations while advocating naïve geography as a cognitively faithful foundation for spatial systems [13–15]. These methods align with cognitive research demonstrating that humans naturally describe places through hierarchical containment and relative positioning [3,6].

Computational implementations of these concepts have emerged primarily through place graphs and semantic web technologies. Place graphs model spatial entities as nodes connected by qualitative relations derived from textual sources or volunteered geographic information [16,17]. Chen et al. [4] demonstrate queryable place graphs capable of answering qualitative spatial questions through graph traversal algorithms. Semantic web approaches have proven particularly successful at encoding qualitative spatial relations, exemplified by the RelLoc framework [18], which represents spatial relationships as RDF triples. National mapping agencies, including the Ordnance Survey, have published comprehensive RDF models of administrative geography supporting linked data queries [19], while large-scale knowledge graphs such as KnowWhereGraph integrate diverse geospatial entities and relations [20,21].

These approaches excel at representing complex spatial relationships and supporting sophisticated reasoning capabilities. However, they typically operate on specialized graph platforms or semantic web infrastructures that remain external to the GIS environments where most spatial analysis and cartographic production occurs [22]. This separation creates barriers to adoption in operational spatial analysis contexts and limits the integration of qualitative and quantitative spatial reasoning.

Qualitative and place modeling within GIS environments.

Limited research has explored qualitative spatial modeling directly within GIS platforms, recognizing the need to bridge theoretical frameworks with operational spatial analysis tools. Almuzaini [23] developed prototype qualitative neighbourhood functionality within Quantum GIS, demonstrating the feasibility of implementing QSR concepts in open-source GIS environments. Giordano et al. [24] and Martin et al. [25] have combined quantitative and qualitative insights for analyzing narrative and user-generated spatial data, showing the complementary value of both approaches.

Place-based GIS research advocates treating places as objects with attributes and relationships rather than as isolated geometries [9], aligning with Agnew's emphasis on understanding places *in relation to* other places [1]. Recent work on homogeneous hierarchical models integrates global and local grids to support multi-scale reasoning [11], though this research focuses primarily on topological structure rather than qualitative relationships.

Despite these advances, existing GIS-based approaches remain either domain-specific, limited in scope, or focused on specific aspects of qualitative representation. Notably absent is a comprehensive, scalable framework that systematically defines places across multiple hierarchies while supporting both storage and querying of qualitative spatial relationships within standard GIS workflows.

QPM addresses these gaps by integrating qualitative spatial representation *within* rather than *alongside* GIS environments. Building on relational concepts from RelLoc [18] and aligned with place-based GIS perspectives [9], QPM provides a homogeneous framework treating administrative units and named places uniformly while supporting efficient computation, storage, and querying using standard GIS datasets and technologies. This approach enables the theoretical richness of qualitative spatial reasoning within the practical constraints of operational GIS workflows.

3. Qualitative Place Model Definition

QPM addresses three fundamental design requirements for qualitative place representation. **Homogeneous representation** provides a unified framework for representing both administrative units (e.g., districts, wards) and named places (e.g., landmarks, settlements) as instances of a single spatial entity class. **Hierarchical anchoring with lateral semantics** anchors place location through hierarchical containment while capturing lateral spatial relationships via neighbourhood relations (touching, proximity, directional). **GIS integration** supports scalable population from standard boundary datasets with seamless integration into existing GIS and ontological frameworks.

QPM adopts a homogeneous representational approach whereby both units of spatial hierarchies (e.g., administrative districts, postal areas) and named places (e.g., settlements, landmarks) are treated uniformly as *places*. Each place is spatially defined through: (a) a unique parent within a containment hierarchy, and (b) its neighbours at the same hierarchical level, connected through adjacency, proximity, and directional relations. This unified treatment ensures consistent spatial semantics across all geographic entities while accommodating geometry-appropriate relationship definitions (polygons support adjacency relations; points do not).

3.1. Formal Model Definition

QPM operates under four fundamental assumptions that ensure computational tractability and semantic consistency. **Partition completeness (A1)** requires that each hierarchical level L_i partitions the extent of level L_{i-1} without gaps or overlaps, with all units represented as valid planar polygons. **Point representation (A2)** constrains Basic Places to point geometries for computational efficiency. **Coordinate reference consistency (A3)** ensures that bearing and distance calculations use a consistent coordinate reference system (Euclidean for projected systems, geodesic for geographic systems). **Directional partitioning (A4)** requires that the set of directional sectors D partitions the angular space $(-\pi, \pi]$ into equal half-open wedges, avoiding boundary ambiguity in directional assignment. Unless otherwise specified, QPM employs 4-way directional sectors $D = \{N, E, S, W\}$ for computational simplicity while maintaining cognitive alignment with natural language directional references.

Hierarchical Structure and Place Definition

QPM represents spatial organization through a hierarchy $H = \{L_0, L_1, \dots, L_n\}$ of spatial divisions, where each level L_i consists of a set of spatial entities termed **places**. The model defines a single superclass *Place* with two principal specializations that enable homogeneous treatment of diverse geographic entities:

Unit $\sqsubseteq$ Place represents fiat polygonal regions that form components of spatial partitions at level L_i (e.g., countries, districts, wards). These administrative or postal divisions provide the hierarchical framework within which other places are located.

BasicPlace $\sqsubseteq$ Place represents named locations with point geometries (e.g., landmarks, settlements, facilities). These correspond to the places that humans typically reference in spatial discourse.

The hierarchical structure ensures complete spatial coverage through systematic partitioning. Each level L_i completely partitions its parent region according to:

$$\bigcup_{u \in L_i} u = \text{Extent}(L_{i-1}), \quad u \cap v \in \{\emptyset, \partial u \cap \partial v\}, \quad u \neq v, u, v \in L_i \quad (1)$$

This formalization ensures that every spatial unit has exactly one parent within the hierarchical structure:

$$\forall u \in L_i (i \geq 1), \exists! p \in L_{i-1} : \text{inside}(u, p) \quad (2)$$

Consequently, each place's spatial **location** is uniquely identified through two complementary mechanisms: a *containment path* $\pi(P) = (u_0, u_1, \dots, u_m)$ extending from the root unit to its immediate container, and its lateral neighbourhood relationships at the corresponding hierarchical level. This

dual characterization enables both hierarchical reasoning (through containment chains) and lateral reasoning (through neighbourhood relations).

3.2. Spatial Relations: Definitions and Computation

QPM defines five fundamental binary relations that capture essential spatial relationships between places. These relations are designed to support both local spatial reasoning (immediate neighbours and containment) and extended spatial inference through logical composition and transitivity. Each relation is defined formally and accompanied by its computational procedure to ensure reproducible implementation.

3.2.1. Hierarchical Containment

The foundational relation **inside (hierarchical containment)** establishes the unique containment hierarchy that anchors place location. This relation is both functional and transitive, ensuring that each place has exactly one parent while supporting multi-level hierarchical reasoning.

Containment relationships establish the hierarchical foundation of QPM by assigning each spatial entity to its unique parent within the hierarchy. For spatial units $u \in L_i$ and potential parents $v \in L_{i-1}$:

$$\text{inside}(u, v) \iff u \subseteq v$$

For Basic Places represented as points, containment is determined through point-in-polygon analysis:

$$\text{inside}(P, u) \iff P \in u$$

where point P is assigned to the unit u whose polygon geometry contains the point location.

For hierarchical partitions where level L_{x-1} forms complete tessellation, each unit in L_x receives unique geometric parent assignment. The algorithm computes representative interior points and employs spatial indexing for efficient point-in-polygon queries, achieving $O(|L_x| \log |L_{x-1}|)$ performance while flagging boundary conflicts for quality assurance. This computational approach ensures that every place has a well-defined position within each relevant spatial hierarchy.

3.2.2. Topological Adjacency

touches (planar adjacency) defines a symmetric relation between units sharing non-empty boundary intersections, applicable exclusively to polygonal units. This topological relation provides the foundation for more complex spatial relationships by identifying immediate spatial neighbours.

For units $u, v \in L_i$ at the same hierarchical level, where $u \neq v$:

$$\text{touches}(u, v) \iff \partial u \cap \partial v \neq \emptyset$$

This topological definition captures the intuitive notion of spatial neighbours while avoiding dependency on coordinate precision issues. The adjacency computation forms the basis for directional relationship assignment and proximity analysis. Adjacency relationships between polygonal units provide the topological foundation for lateral spatial reasoning through boundary intersection analysis.

3.2.3. Directional Orientation

QPM implements two complementary directional relations that capture spatial orientation between places. dir_d (**directional orientation**) establishes orientation relationships between neighbouring units in directional sector $d \in D$, while primaryDir_d (**primary directional choice**) provides a functional directional relation by selecting a single representative neighbour per directional sector.

The distinction between general orientation (dir_d) and primary directional choice (primaryDir_d) enables QPM to balance expressive power with storage efficiency. While dir_d can represent multiple neighbours in each direction, primaryDir_d provides a sparse, deterministic representation suitable for efficient storage and computation. For units, dir_d implies **touches** and supports inverse directional pairs

(e.g., if A is east of B, then B is west of A), enabling bidirectional spatial reasoning and consistency checking.

Computational Procedure for Units

For polygonal units, directional relationships are constrained to touching neighbours to ensure semantic consistency. The computation identifies touching candidates within each directional sector and selects a primary representative using deterministic criteria. For a unit u at level L_i , the touching candidates in directional sector d are defined as:

$$T_d(u) := \{v \in L_i \setminus \{u\} \mid \text{touches}(u, v) \wedge \text{sector}(u, v) = d\}$$

When $T_d(u) = \emptyset$, no primary neighbour is assigned for sector d , reflecting genuine boundary conditions or geometric constraints. Otherwise, a single representative is selected using lexicographic criteria that prioritize spatial significance: shared boundary length, angular alignment with sector center, centroid proximity, and stable identifiers (complete specification in Appendix A.0.0.9).

This selection process yields the primary directional assignment $\text{primaryDir}_d(u, v^*)$. Additionally, multi-valued orientation relationships are established for all touching pairs: $\text{dir}_d(u, v) \iff \text{touches}(u, v) \wedge \text{sector}(u, v) = d$, with corresponding inverse relationships $\text{dir}_{d^{-1}}(v, u)$ to ensure bidirectional consistency.

Computational Procedure for Basic Places

For point-based Basic Places, directional relationships are computed using nearest-neighbour analysis within directional sectors. This approach accommodates the absence of topological adjacency for point geometries while maintaining consistency with the overall directional framework. For basic place P and directional sector d , the candidate set includes all other Basic Places within the specified sector:

$$C_d(P) := \{Q \in PL \setminus \{P\} \mid \text{sector}(P, Q) = d\}$$

When candidates exist within sector d , the nearest neighbour is selected using distance-based criteria with deterministic tie-breaking (detailed specification in Appendix A.0.0.9). This selection yields the primary directional assignment $\text{primaryDir}_d(P, Q^*)$ while ensuring reproducible results across multiple computational runs.

3.2.4. Qualitative Proximity

near (qualitative proximity) captures qualitative notions of spatial closeness that extend beyond immediate adjacency or directional orientation. QPM implements proximity through bounded reachability in spatial neighbourhood graphs, providing a topologically grounded alternative to metric distance that remains stable across different coordinate systems and scales.

Proximity for Administrative Units

For administrative units, proximity is defined through bounded path analysis in the adjacency graph formed by touching relationships. For each hierarchy level L_i , the adjacency graph $G_i = (V_i, E_i)$ represents units as vertices ($V_i = L_i$) and touching relationships as edges ($E_i = \{\{u, v\} \subseteq L_i \mid \text{touches}(u, v)\}$). Proximity is then defined as bounded reachability within this graph:

For units $u, v \in L_i$ and hop bound $k \in \mathbb{N}$ (typically $k = 2$):

$$\text{near}_k(u, v) \iff 1 \leq \text{dist}_{G_i}(u, v) \leq k$$

where $\text{dist}_{G_i}(u, v)$ represents the shortest path distance in the adjacency graph. This definition ensures that proximity is irreflexive (units are not near themselves) while capturing units reachable within k adjacency steps. With $k = 2$, the relationship captures units that are either touching or share a common neighbour, corresponding to intuitive notions of regional proximity.

Proximity for Basic Places

Basic places, represented as points, lack the topological adjacency relationships available for polygonal units. QPM therefore defines proximity for Basic Places through two complementary approaches that accommodate different analytical requirements.

Graph-based proximity constructs a neighbourhood graph from primary directional relationships between Basic Places. For the place layer PL , the primary-neighbour graph $H = (V_H, E_H)$ has vertices $V_H = PL$ and edges derived from directional primaries:

$$E_H = \{ \{P, Q\} \subseteq PL \mid \exists d \in D : \text{primaryDir}_d(P, Q) \text{ or } \text{primaryDir}_{d^{-1}}(Q, P) \}$$

Proximity between Basic Places is then defined as bounded reachability in this graph:

$$\text{near}_k^{\text{BP}}(P, Q) \iff 1 \leq \text{dist}_H(P, Q) \leq k$$

Host-derived proximity provides an alternative approach that grounds basic place proximity in the underlying administrative tessellation:

$$\text{hostNear}_k^{(h)}(P, Q) \iff \text{near}_k^{(h)}(\text{host}_h(P), \text{host}_h(Q))$$

where $\text{host}_h(P)$ represents the base-level administrative unit containing basic place P within hierarchy h .

Computational Properties

Both proximity variants maintain essential mathematical properties that support consistent spatial reasoning. The relationships are irreflexive (places are not proximate to themselves), symmetric (proximity is bidirectional), and monotone (increasing the hop bound k never reduces the proximity set). The graph-based computation of proximity scales efficiently with spatial tessellation complexity and supports pre-computation and caching strategies. For administrative units, proximity computation requires constructing the adjacency graph ($O(|E|)$ where E represents touching relationships) followed by bounded breadth-first search. For Basic Places, constructing the primary-neighbour graph is linear in the number of directional relationships, typically $O(|D| \times |PL|)$ for $|D|$ directional sectors.

The computational procedures maintain semantic consistency across different geometric representations while accommodating the practical constraints of large-scale spatial datasets. Distance calculations employ appropriate metrics (Euclidean for projected coordinate systems, geodesic for geographic coordinate systems) to ensure spatial accuracy across diverse geographic contexts.

The QPM framework can be expressed through formal ontological axioms, shown in Table 1 that enable automated reasoning and precise semantic interpretation.

3.3. Multi-Hierarchy Support

Real-world spatial organization involves multiple, overlapping administrative and organizational systems that operate over the same geographic extent. QPM accommodates this complexity by supporting multiple spatial hierarchies $\{H^{(1)}, H^{(2)}, \dots, H^{(m)}\}$ that coexist while maintaining independent relationship structures.

3.3.1. Independent Hierarchy Management

Each hierarchy maintains its own containment and lateral relationship structures independently, ensuring that organizational differences between systems (e.g., administrative versus postal boundaries) do not create computational conflicts. Within each hierarchy $H^{(i)}$, the fundamental relationships—inside, touches, and dir_d —operate according to the formal specifications defined previously, creating coherent spatial reasoning frameworks for each organizational system.

Table 1. Formal ontological axioms for QPM. These axioms ensure semantic consistency and enable automated reasoning over spatial relationships.

Category	Axiom (Description Logic / Property)	Semantic Interpretation
<i>Containment Semantics</i>		
	$\text{Place} \sqsubseteq \forall \text{containedInUnit}.(\text{Unit} \sqcap \leq 1 \text{ belongsToHierarchy.Hierarchy})$	Each Place can be contained in multiple units, but at most one unit per hierarchy.
	inside is transitive	Containment relationships compose across hierarchical levels (if A contains B and B contains C, then A contains C).
	$\text{Unit} \sqcap \neg \exists \text{ inside. } \top$	Root units (top-level administrative divisions) have no containing parent.
<i>Adjacency and Directional Semantics (Units)</i>		
	$\text{touches} \equiv \text{touches}^{-1}$	Adjacency is symmetric: if unit A touches unit B, then unit B touches unit A.
	$\text{dir}_d \sqsubseteq \text{touches}$	Directional relationships between units require topological adjacency.
	$\text{dir}_d \equiv (\text{dir}_{d^{-1}})^{-1}$	Directional relationships have consistent inverses (e.g., east-of implies west-of in reverse).
<i>Primary Selection Constraints</i>		
	$\text{primaryDir}_d \sqsubseteq \text{dir}_d$	For units, the selected primary neighbour per sector must be directionally oriented.
	$\text{Place} \sqsubseteq (\leq 1 \text{ primaryDir}_d.\text{Place})$ (for each $d \in D$)	Each place has at most one primary neighbour per directional sector, ensuring functional relationships.
	$\text{BasicPlace} \sqsubseteq \bigcap_{d \in \{E,W,N,S\}} \leq 1 \text{ primaryDir}_d.\text{BasicPlace}$	Basic places have at most one primary neighbour per direction, with relationships constrained to other basic places.
<i>Proximity Derivation (Units)</i>		
	$\text{touches} \circ \text{touches} \sqsubseteq \text{near}_2$	Units reachable within two adjacency steps are considered proximate.
<i>Homogeneous Place Representation</i>		
	$\text{Unit} \sqsubseteq \text{Place}, \text{BasicPlace} \sqsubseteq \text{Place}$	Administrative units and named places are unified under the Place superclass.
	$\text{Place} \sqsubseteq \exists \text{ inside.Place} \sqcup \text{rootPlace}$	Every place has a containing parent except for root-level administrative units.

Implementation note: Directional relationships (dir_d) are computed for administrative units based on topological adjacency. For Basic Places (points), direction is represented through functional primaryDir_d relationships per sector. Multi-valued directional relationships over points can be derived if required for specific reasoning tasks.

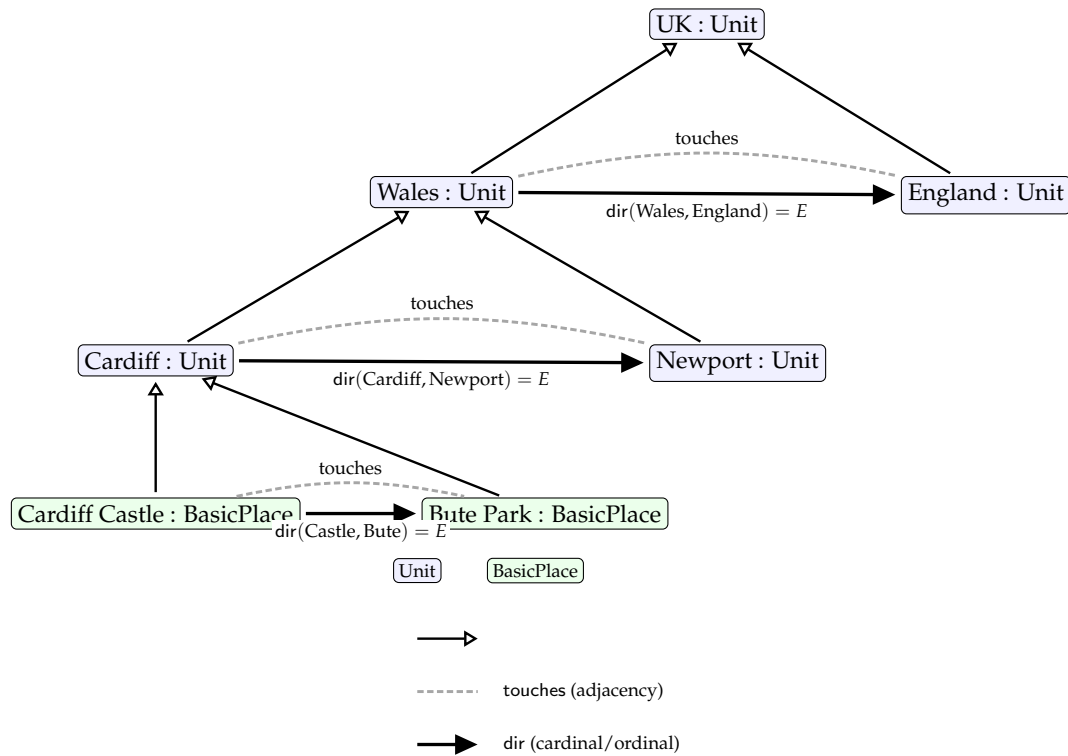


Figure 1. QPM as a homogeneous model of place. Units and Basic Places are both *Place* instances. Place location is given by a unique containment path (inside) and neighbourhood relations: touches (adjacency) and dir (direction).

This independence is essential because different spatial hierarchies often exhibit inconsistent boundary alignments and organizational principles. Administrative boundaries may follow historical precedents, while postal boundaries optimize delivery efficiency, and electoral boundaries ensure population balance. QPM's multi-hierarchy approach accommodates these differences without requiring artificial alignment or compromise.

3.3.2. Multi-Hierarchy Place Definition and Location Specification

Basic places can be simultaneously contained within units from multiple hierarchies, enabling rich, multi-dimensional place definitions that reflect the complexity of real-world spatial organization. This capability supports natural language descriptions that reference multiple organizational systems (e.g., "located in Castle ward, CF10 postal sector, and Cathays constituency").

For a basic place P and hierarchy set $\mathcal{H} = \{H^{(1)}, \dots, H^{(m)}\}$, the complete location specification is:

$$\text{location}(P) = \{(H^{(i)}, u^{(i)}) \mid P \in u^{(i)}, u^{(i)} \in H^{(i)}, H^{(i)} \in \mathcal{H}\}$$

This formalization captures the reality that places exist within multiple organizational contexts simultaneously. Each tuple $(H^{(i)}, u^{(i)})$ represents the containment of place P within unit $u^{(i)}$ of hierarchy $H^{(i)}$.

Computational Implementation

Multi-hierarchy place assignment is computed through iterative point-in-polygon analysis across all available hierarchies. For each basic place P :

1. For each hierarchy $H^{(i)} \in \mathcal{H}$:
 - (a) Identify base-level units $L_{\text{base}}^{(i)}$ in hierarchy $H^{(i)}$
 - (b) Compute $u^{(i)} = \{u \in L_{\text{base}}^{(i)} \mid P \in \text{geom}(u)\}$
 - (c) If $|u^{(i)}| = 1$, assign containment $(H^{(i)}, u^{(i)})$ to P
 - (d) If $|u^{(i)}| > 1$, flag boundary conflict and apply tie-breaking

2. Derive hierarchical containment paths for each assigned base unit
3. Store complete location specification in cross-reference table

The algorithm handles edge cases including places on boundaries between hierarchies and provides deterministic assignment through geometric precedence rules.

Query and Reasoning Capabilities

The multi-hierarchy approach enables QPM to support comprehensive spatial queries that span multiple organizational systems while respecting the independence and integrity of each hierarchy’s internal structure:

- **Cross-system queries:** "Find all places in administrative ward X that are also in postal district Y"
- **Hierarchical correspondence:** "Identify electoral constituencies that overlap with postal area Z"
- **Multi-dimensional place descriptions:** Generate complete place descriptions spanning all relevant hierarchies

Such queries require reasoning across multiple organizational systems through the cross-hierarchy relationships and multi-hierarchy place definitions, enabling sophisticated spatial analysis that reflects real-world administrative complexity.

This multi-dimensional approach to place definition aligns with human spatial cognition, where places are naturally understood through multiple, overlapping organizational and administrative contexts rather than through single, monolithic spatial frameworks.

4. QPM Design and Implementation

This section presents the practical realization of QPM through a proof-of-concept implementation within ArcGIS Pro. The implementation bridges theoretical framework with operational capabilities, demonstrating that qualitative spatial reasoning integrates into mainstream GIS workflows without sacrificing computational efficiency or analytical rigour.

4.1. System Architecture and Processing Workflow

QPM operates as an integrated ArcGIS Pro toolbox, leveraging native spatial analysis and geodatabase storage for seamless GIS compatibility. Figure 2 illustrates the processing architecture transforming raw spatial datasets into queryable qualitative knowledge bases through five coordinated phases:

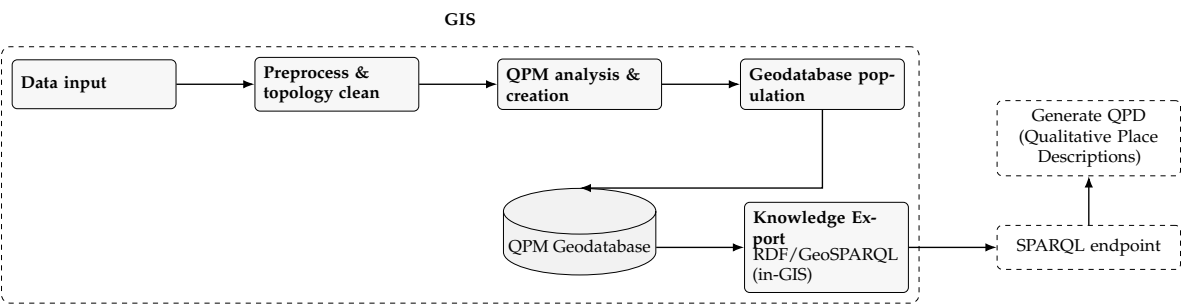


Figure 2. System architecture for QPM with consolidated processing phases inside the GIS.

Data Input accommodates standard GIS formats (shapefiles, feature classes, geodatabase tables) for boundary datasets and point-based place layers. **Preprocessing** ensures data quality through geometry validation, boundary planarization, and stable identifier generation. **QPM Analysis** systematically derives spatial relationships according to formal specifications: containment (inside) through point-in-polygon analysis, adjacency (touches) through boundary intersection, and directional relationships (primaryDir_d, dir_d) through bearing analysis and sector assignment. **Geodatabase Population** stores relationships within a custom schema supporting both spatial geometries and qualitative relationships.

Knowledge Export generates RDF/GeoSPARQL representations enabling semantic querying while maintaining native GIS accessibility.

The native GIS integration eliminates adoption barriers by enabling users to visualize geometric and qualitative representations within unified interfaces, supporting hybrid analytical approaches that leverage both precise measurement and human-centric spatial reasoning.

4.2. Implementation Framework and Methodology

The implementation adopts a research-oriented approach prioritizing algorithmic transparency and functional correctness. The technical framework leverages Python-based libraries: ArcPy for native GIS integration, GeoPandas for geometric computations, R-tree indexing for spatial query acceleration, and PyQt6 for user interface capabilities.

QPM employs a unified dual-pipeline architecture processing hierarchical units (polygons) and Basic Places (points) through parallel workflows while maintaining global consistency. The **administrative units pipeline** computes containment through geometric inclusion, materializes boundary contacts through topological intersection, selects primary directional neighbours using deterministic tie-breaking, and derives multi-valued orientation relationships with inverse consistency. The **Basic Places pipeline** assigns host units through point-in-polygon analysis, computes primary directional neighbours via nearest-in-sector analysis, and derives proximity relationships through primary-neighbour graph traversal or host unit inheritance.

Computational optimization through spatial indexing transforms geometric queries from quadratic to near-linear performance when effective R-tree indexing is employed for point-in-polygon queries and adjacency detection, while deterministic tie-breaking ensures reproducible results across multiple computational runs.

4.3. Geodatabase Schema and Data Management

The QPM schema balances efficient spatial analysis with referential integrity across qualitative relationships. Figure 3 presents the Entity-Relationship structure organizing spatial entities and relationships.

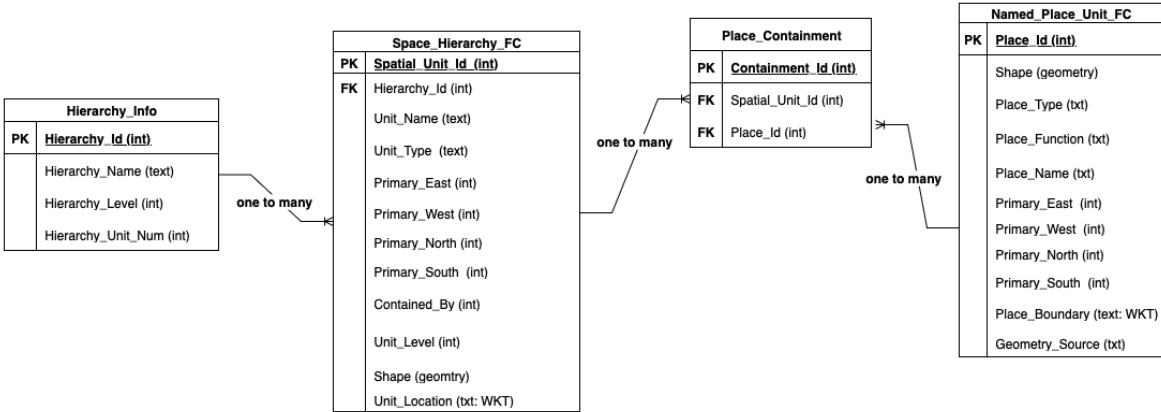


Figure 3. ER Diagram of the QPM geodatabase schema, illustrating primary entities, attributes, and relationships. Feature classes: Space_Hierarchy_FC (Units), Named_Place_Unit_FC (Basic Places). Tables: Place_Unit_Containment (host Unit per place), Hierarchy_Info (metadata). Per-sector primary neighbours are materialised as directional fields.

Core Schema Components

Primary spatial entities are stored in two feature classes: Space_Hierarchy_FC for administrative units (polygons) with attributes including Spatial_Unit_Id, Hierarchy_Id, Contained_By, and materialized directional fields (Primary_East_Unit_Id, etc.); and Named_Place_Unit_FC for Basic Places

(points) with Place_Id, Place_Name, and corresponding directional neighbour fields. This direct storage approach enables efficient directional queries without complex join operations.

Relationship management accommodates multi-hierarchy complexity through specialized tables: Place_Containment manages cross-hierarchy place-unit relationships enabling simultaneous location within different hierarchies; Hierarchy_Info stores metadata about organizational systems including hierarchy names and depth.

The schema implements strategic denormalization for frequently accessed relationships combined with appropriate indexing for efficient geometric and relational queries, ensuring common spatial queries execute efficiently while maintaining data integrity.

4.4. Algorithms and Processing

QPM employs deterministic algorithms that transform spatial datasets into structured qualitative relationships according to formal specifications (Section 3). All algorithms follow explicit contracts specifying preconditions, processing methods, and postconditions, ensuring reproducible implementation (detailed specifications in Appendix A).

Containment Computation

For hierarchical partition where level L_{x-1} forms complete tessellation, each unit in L_x receives unique geometric parent assignment. The algorithm computes representative interior points and employs spatial indexing for efficient point-in-polygon queries, achieving $O(|L_x| \log |L_{x-1}|)$ performance while flagging boundary conflicts for quality assurance.

Directional Computation

For units at the same hierarchical level, primary directional neighbours are computed in each sector while ensuring directional assertions imply topological adjacency. The algorithm groups touching neighbours by directional sector and selects primaries using lexicographic criteria: shared boundary length, angular deviation from sector mid-angle, centroid distance, and stable identifier ordering. For Basic Places, nearest-neighbour analysis within directional sectors employs distance-based selection with deterministic tie-breaking.

Performance and Edge Cases

Spatial indexing using R-tree structures transforms geometric queries from $O(n^2)$ to $O(n \log n)$ performance for containment and adjacency computations, enabling processing of datasets containing tens of thousands of entities. This improvement is achieved when spatial data exhibits reasonable geographic distribution and indexing parameters are optimized for the specific dataset characteristics.

4.5. Implementation Scope and Limitations

As a proof-of-concept implementation, QPM successfully demonstrates qualitative spatial reasoning feasibility within mainstream GIS while establishing clear operational capability boundaries.

Current implementation achieves efficient performance for moderate-scale datasets through spatial indexing optimization. Computational complexity varies across relationship types: containment achieves $O(|L_x| \log |L_{x-1}|)$ with spatial indexing; adjacency approaches near-linear performance through indexed geometric operations; primary neighbour computation scales as $O(\sum_u \deg(u))$ for units and $O(|PL| \log |PL|)$ for Basic Places with effective indexing.

QPM effectiveness depends critically on input data quality, particularly topological correctness and geometric precision. The implementation assumes administrative boundary datasets conform to partition properties, making it sensitive to gaps, overlaps, and sliver polygons. Geometric irregularities can suppress primary neighbour assignment, reducing relationship coverage and potentially compromising spatial reasoning completeness.

Current implementation exhibits tight ArcGIS Pro integration, creating dependency on proprietary software while limiting accessibility for organizations using alternative GIS platforms. The system

lacks comprehensive incremental update mechanisms for dynamic datasets, requiring complete recomputation when boundaries change.

4.6. GIS Demonstration: Dual Representation and Querying

This demonstration illustrates QPM’s operational capability to support spatial analysis through dual computational paradigms: traditional GIS geometric analysis and semantic qualitative reasoning, enabling unified analytical workflows combining coordinate-based precision with natural-language spatial reasoning expressiveness.

Figure 4 presents the QPM toolbox interface integrated within ArcGIS Pro, demonstrating seamless incorporation of qualitative spatial capabilities into standard workflows. Following execution, Figure 5 illustrates enhanced Table of Contents including original input layers and generated QPM feature classes and relationship tables, enabling parallel analysis using geometric measurements and qualitative relationships.

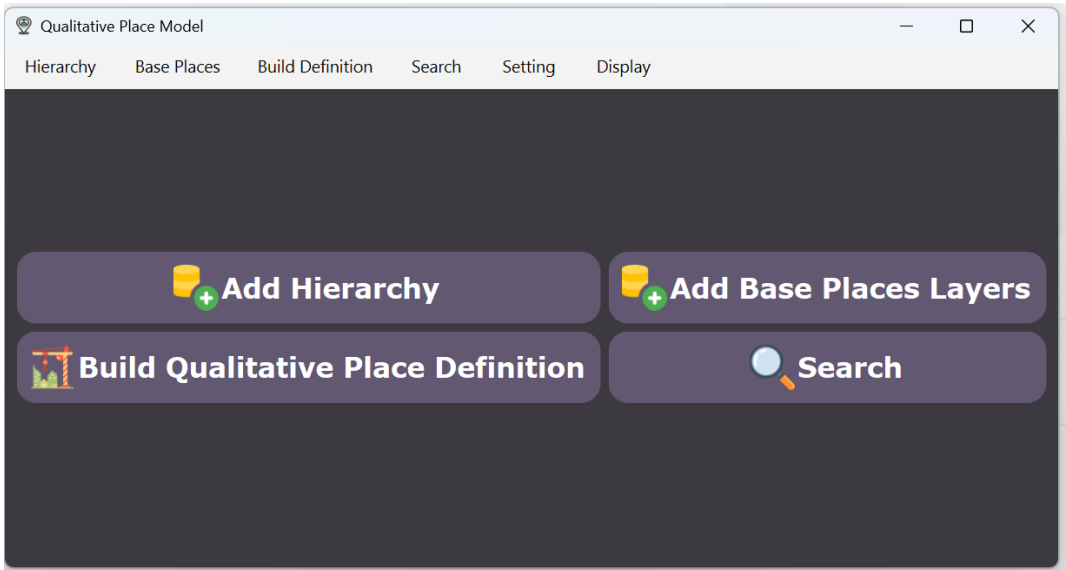


Figure 4. QPM toolbox in ArcGIS Pro: data input, model execution, and query panels.

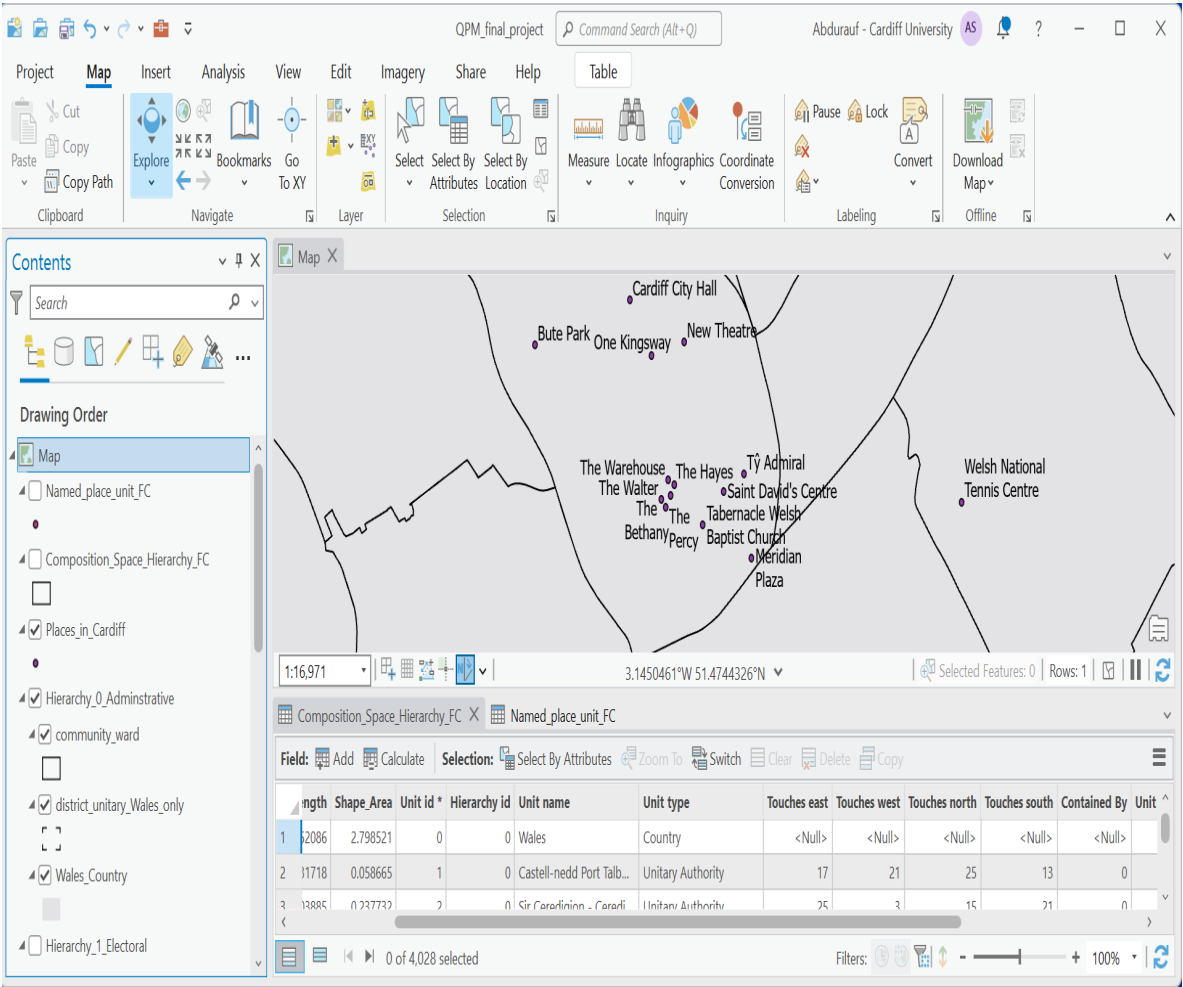


Figure 5. ArcGIS Table of Contents after QPM population: input layers (left) and generated QPM feature classes/tables (right).

A **Hierarchy Manager** component enables systematic specification of spatial layers representing hierarchical divisions with level ordering, naming conventions, and validation ensuring input datasets conform to partition assumptions. A **Place Layer Manager** component accommodates point-based datasets with attribute mapping, spatial reference validation, and schema compatibility checking. Processing execution provides real-time feedback including progress indicators, automated validation reports, and detailed logging of computational decisions and edge case handling.

Comprehensive Place Definition Demonstration

QPM is capable of generating comprehensive qualitative place descriptions combining hierarchical containment with directional neighbourhood relationships. Figure 6 presents the QPD for the place St. David’s Centre, generated in the system, illustrating multi-dimensional spatial relationships spanning multiple hierarchies while identifying primary directional neighbours within the basic place layer.

Place-Focused Queries address the fundamental question "Where is this place?" by generating comprehensive spatial descriptions that combine geometric coordinates with qualitative relationships. The following SPARQL query demonstrates how users can retrieve complete place definitions including hierarchical containment and directional neighborhoods:

Listing 1: Comprehensive Place Definition Query

```
PREFIX qpm: <http://qpm.ontology/2025#>
SELECT ?place ?placeName ?containedInUnits ?directionalNeighbors
WHERE {
  ?place qpm:place_name "Saint David’s Centre" ;
```

```
qpm:contained_by_unit ?units ;
qpm: east_of ?eastNeighbors ;
qpm:north_of ?northNeighbors .
# Additional directional and containment patterns ...
}
```

Figure 7 presents the RDF graph structure result of SparQL queries to retrieve QPD for Basic Places. Figures 8 and 9 show the same for an example spatial unit. The graphs illustrate rich relationship networks emerging from systematic spatial analysis with substantial inferential expansion through simple logical operations. The demonstration validates QPM’s capability to generate meaningful qualitative place descriptions aligning with human spatial intuitions while maintaining computational precision and reproducibility, successfully bridging theoretical qualitative spatial reasoning capabilities with practical GIS implementation.

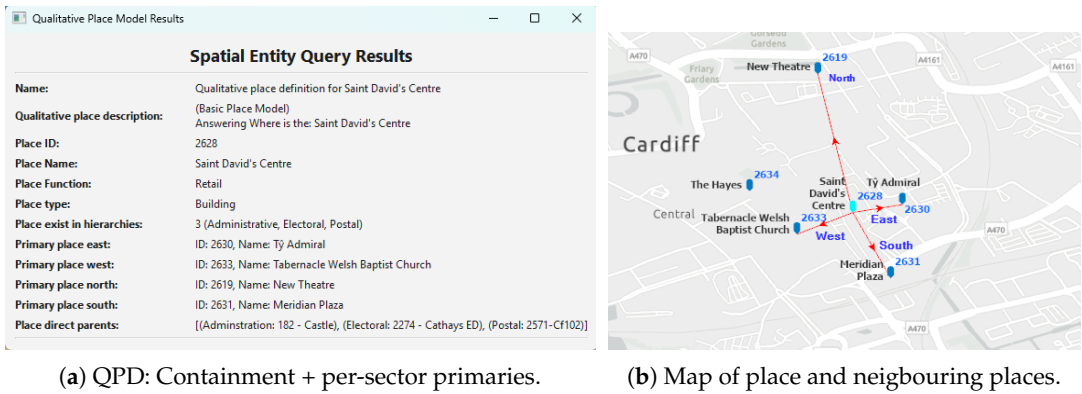


Figure 6. QPD for *St. David's Centre*; place Id: 2628.

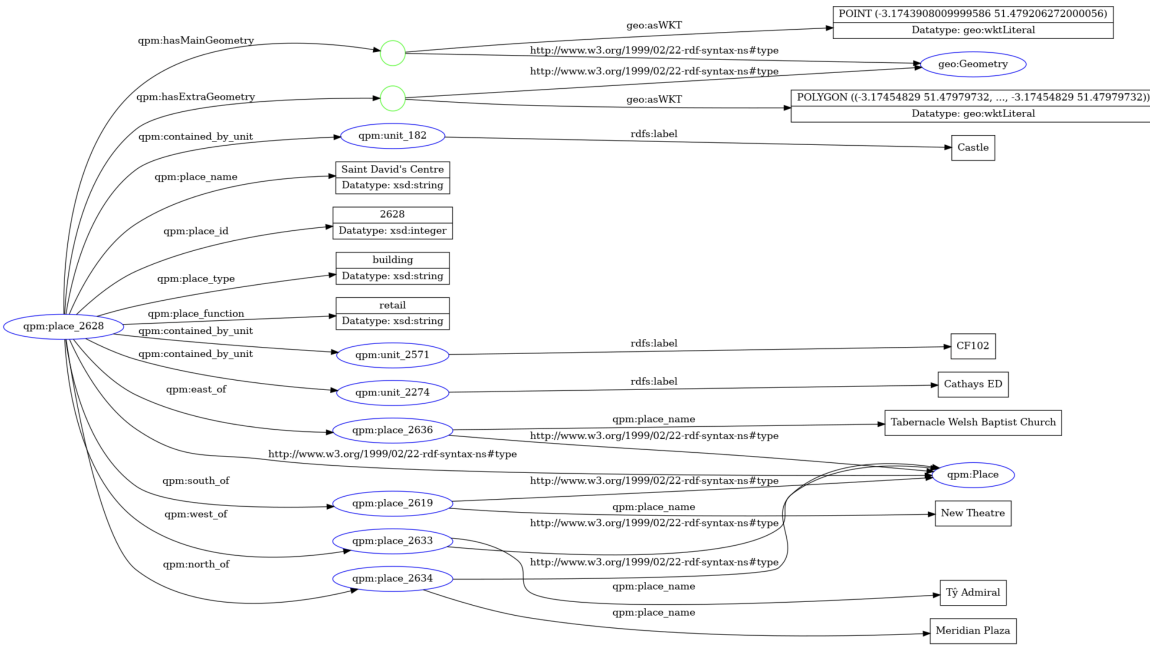


Figure 7. RDF graph for qpm:place_2628("St. David's Centre"): geometry and QPD.

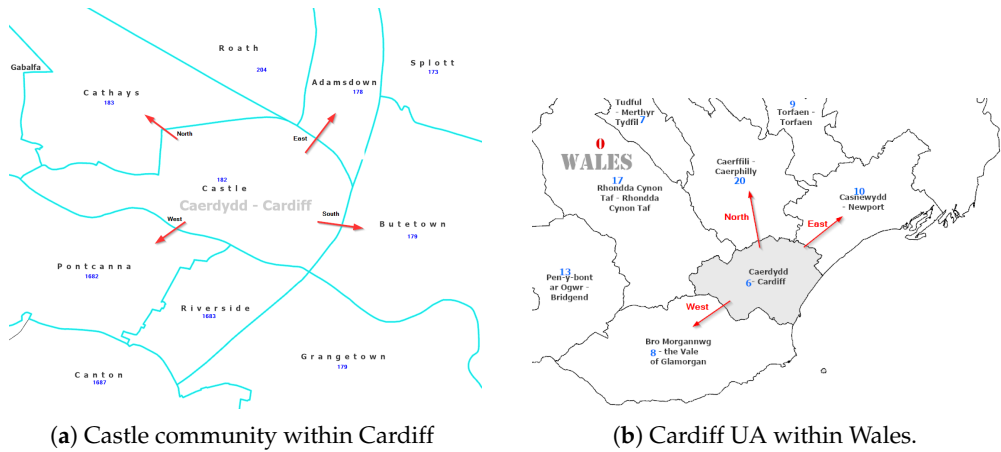
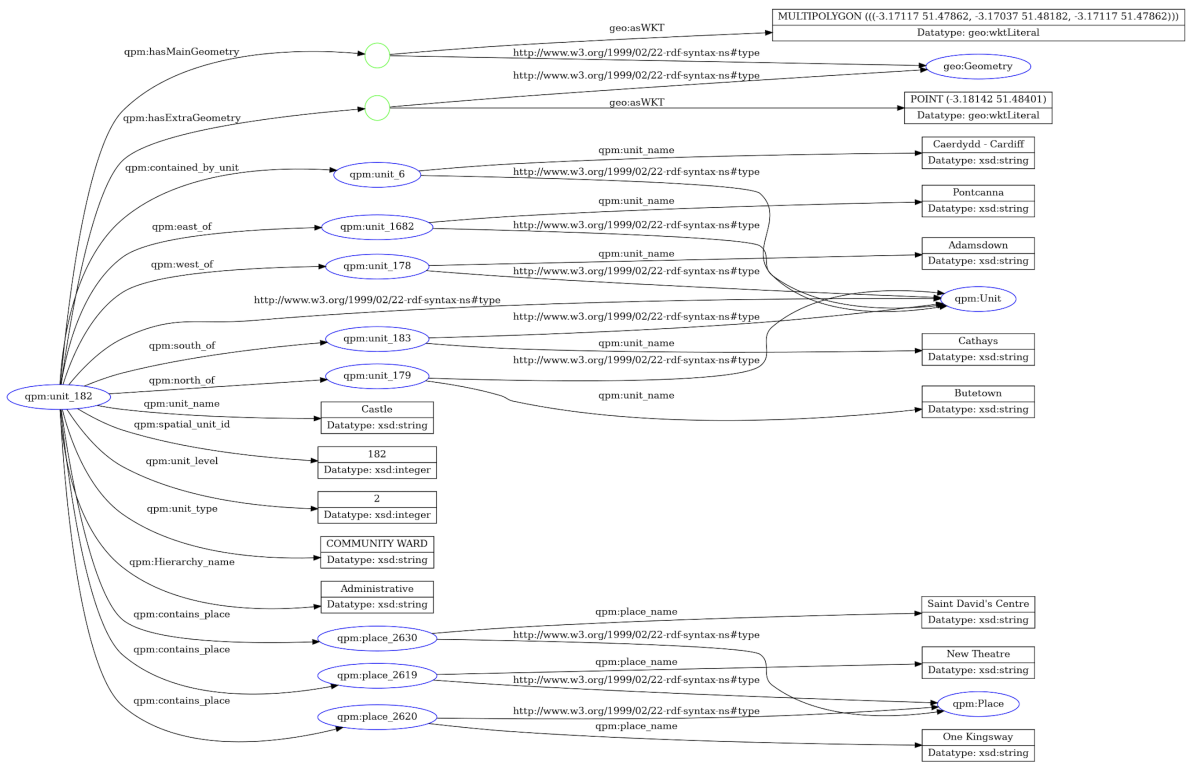


Figure 8. Spatial units in two levels in a hierarchy.



5. Evaluation of the Qualitative Place Model

We evaluate QPM along three axes: (i) **uniqueness** of Qualitative Place Descriptions (QPDs), (ii) **correctness** of the asserted relations, and (iii) **completeness** of the directional/containment graph both as *stored* and after *logical closure* (inverse & transitive inferences). We report aggregate counts and coverage ratios, and we supply procedures to reproduce the checks.

5.1. Spatial Dataset

We use open data from Ordnance Survey (Great Britain) for Wales (UK). The dataset contains three hierarchical partitions and one point layer:

- **Administrative Hierarchy:** 3 levels (Wales; 22 unitary authorities; 1,697 communities) ⇒ 1,720 units.
- **Electoral Hierarchy:** 3 levels (Wales; 5 Senedd regions; 769 electoral wards) ⇒ 775 units.
- **Postal Hierarchy:** 3 levels (8 postcode areas; 205 districts; 534 sectors) ⇒ 747 units.

- **Basic Places (PL):** 52,821 named places (e.g., St. David's Centre, New Theatre).

Hierarchy metadata reside in `Hierarchy_Info`; spatial units and places in `Space_Hierarchy_FC` and `Named_Place_Unit_FC`, respectively. Figure 10 illustrates, as an example, the administrative hierarchy divisions.

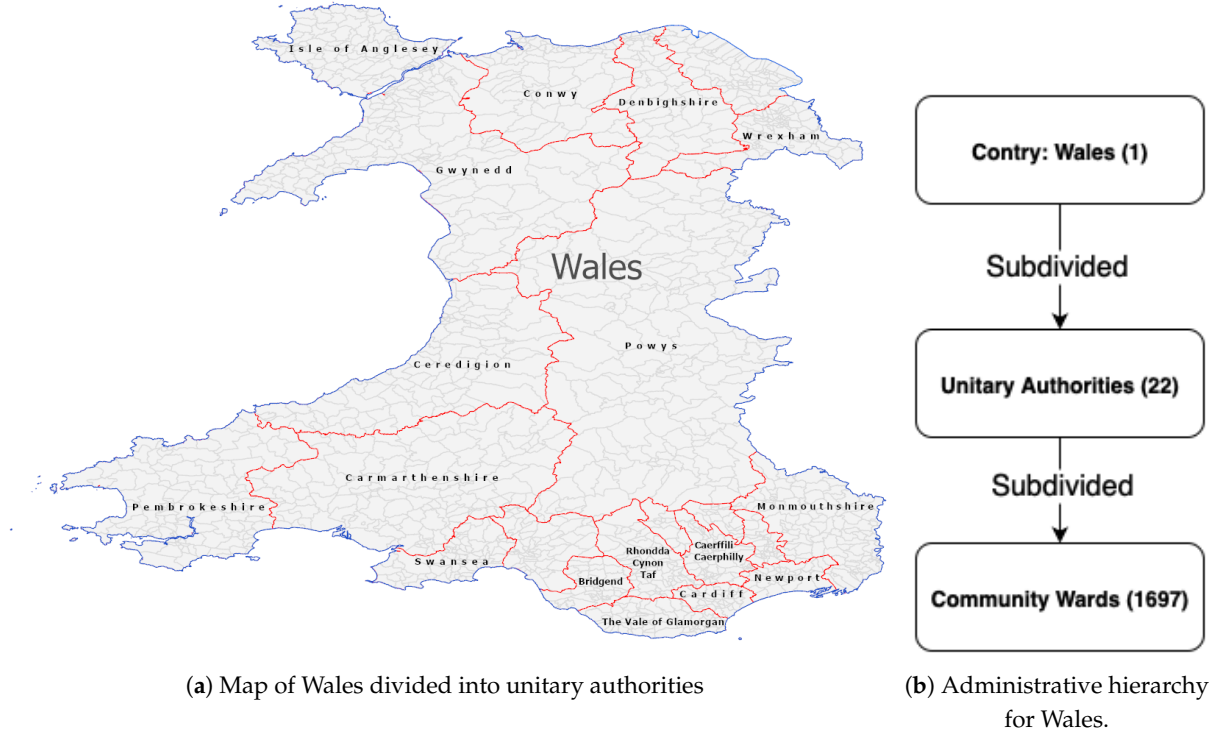


Figure 10. The Administrative hierarchy from the Wales dataset.

5.2. Evaluation Methodology

Let n be the number of entities under test (Units at a given level or Basic Places in *PL*), and let $D = \{E, W, N, S\}$. We track:

1. **Stored graph size** $|E_{\text{stored}}|$: number of asserted directional triples (east_of, west_of, north_of, south_of) plus containment.
2. **Expected primaries** (coverage baseline):

$$\mathbb{E}_{\text{prim}}(n) = n \cdot |D| + n_{\text{hasParent}}$$

i.e., at most one primary neighbour per sector per entity plus exactly one parent when applicable. For Basic Places we expect $|D| = 4$ primaries per place; for Units, boundary/top entities may reduce the achievable count in practice (e.g., no northern neighbour at the national boundary).

3. **Inverse + transitive closure counts:** given the directed graph for each predicate $p \in D$, we add inverse edges (e.g., $\text{east_of}(u, v) \Rightarrow \text{west_of}(v, u)$) and the transitive closure of p . We report new triples added by *inverse*, by *transitive*, and their union.
4. **Comparability ratio** γ : fraction of unordered pairs $\{x, y\}$ that become comparable by *some* direction after closure,

$$\gamma = \frac{|\{\{x, y\} : x \neq y, \exists d \in D : d(x, y) \vee d(y, x)\}|}{\binom{n}{2}}.$$

Uniqueness.

QPD uniqueness is tested by a canonical signature that *only* depends on qualitative content (parents and per-sector primaries), not on incidental identifiers. Two entities collide if and only if their signatures coincide.

Correctness.

Correctness checks include: (i) level/hierarchy coherence (same level for unit-unit directions), (ii) $\text{dir} \subseteq \text{touches}$ for Units, (iii) inverse pairs for Units, and (iv) absence of self-loops.

Graph completeness.

We report coverage against $\mathbb{E}_{\text{prim}}(n)$ and the size after inverse + transitive closure.

5.3. Procedures

Uniqueness test (canonical signature).

We define the canonical signature of an entity X as

$$\Sigma(X) = \left(\text{parents}(X), (\text{prim}(X, E), \text{prim}(X, W), \text{prim}(X, N), \text{prim}(X, S)) \right),$$

where $\text{parents}(X)$ is the ordered list of container IDs per hierarchy (or null where absent), and $\text{prim}(X, d)$ is the ID (or null) of the primary neighbour in sector d . Two entities have the same QPD iff their $\Sigma(\cdot)$ coincide.

Algorithm 1 QPD Uniqueness Check (hash-index)

```

1: Input: Entity set  $S$  (Units at a level or Basic Places); relations contained_by, primary*_of
2: Output: Collision report  $\mathcal{C}$  (sets of entities with identical signatures)
3:  $\text{index} \leftarrow$  empty map from signature  $\rightarrow$  list of entity IDs
4: for each  $X \in S$  do
5:    $P \leftarrow$  ordered list of parent IDs per hierarchy (use stable hierarchy order; null if none)
6:    $(e, w, n, s) \leftarrow$  IDs of primary_east_of, primary_west_of, primary_north_of, primary_south_of from  $X$  (or null)
7:    $\sigma \leftarrow$  canonical serialisation of  $(P, e, w, n, s)$  (JSON string)
8:    $\text{index}[\sigma].\text{append}(X.\text{id})$ 
9: end for
10:  $\mathcal{C} \leftarrow \{ \ell \in \text{index.values}() \mid |\ell| > 1 \}$  ▷ collisions
11: return  $\mathcal{C}$ 

```

Algorithm 2 Directional Closure and Coverage Metrics

```

1: Input: Directed graphs  $G_d = (V, E_d)$  for  $d \in D = \{E, W, N, S\}$ ; containment edges  $E_{\text{cont}}$ 
2: Output: Counts for Stored, Inverse, Transitive, Overlap, Total; comparability ratio  $\gamma$ 
3:  $n \leftarrow |V|$ ;  $E_{\text{stored}} \leftarrow E_{\text{cont}} \cup \bigcup_{d \in D} E_d$ 
4: for each  $d \in D$  do
5:    $E_{\text{inv}}^{(d)} \leftarrow \{(v, u) \mid (u, v) \in E_d\}$  ▷ inverse edges
6:    $E_{\text{tr}}^{(d)} \leftarrow$  transitive closure of  $E_d$  (e.g., BFS from each  $u$ , union of all  $(u, w)$ )
7: end for
8:  $E_{\text{inv}} \leftarrow \bigcup_d E_{\text{inv}}^{(d)}$ ;  $E_{\text{tr}} \leftarrow \bigcup_d E_{\text{tr}}^{(d)}$ 
9:  $E_{\text{total}} \leftarrow E_{\text{stored}} \cup E_{\text{inv}} \cup E_{\text{tr}}$ 
10:  $\text{overlap} \leftarrow |(E_{\text{inv}} \cap E_{\text{tr}}) \setminus E_{\text{stored}}|$ 
11:  $\text{comparables} \leftarrow |\{ \{x, y\} : x \neq y, \exists d \in D : (x, y) \in E_{\text{total}} \vee (y, x) \in E_{\text{total}} \}|$ 
12:  $\gamma \leftarrow \text{comparables} / \binom{n}{2}$ 
13: return size summaries and  $\gamma$ 

```

5.4. Results

Table 2 reports stored counts and the expected baseline $\mathbb{E}_{\text{prim}}(n) = n(|D| + 1)$ (with one less containment for each hierarchy root). For Basic Places ($n = 52,821$), we expect $5n = 264,105$ triples (4 primaries + 1 parent per place), whereas 253,102 were produced (95.8% coverage). Missing directions arise at dataset boundaries (e.g., coastal edges) or where the nearest-in-sector candidate is absent.

Table 2. Graph-based evaluation: stored RDF statements and expected baseline.

Dataset	Entities	Directional	Containment	Total Generated	Expected
Places (BPM)	52,821	200,281	52,821	253,102	264,105
Administrative Units	1,720	6,576	1,719	8,295	8,600
Electoral Units	775	2,893	773	3,666	3,875
Postal Units	747	2,612	739	3,351	3,735

All QPDs for Units and for Basic Places were tested unique via Algorithm 1 (no collisions). For Units, $\text{dir} \subseteq \text{touches}$ and inverse pairs passed; self-loops were absent. For Basic Places, primary neighbours were functional per sector by construction.

Directional closure and "complete graph" comparison.

Table 3, 4, 5 summarise, for the administrative, postal and electoral hierarchies, respectively, the growth from the stored graph to the closure graph per predicate.

Table 3. Administrative hierarchy: stored vs. inferred sizes per predicate. "Full" is the de-duplicated union of stored, inverse, and per-predicate transitive edges.

Predicate	Stored	Inverse	Transitive	Common	Full	Stored/Full (%)
qpm: east_of	1,649	1,043	26,815	76	29,431	5.60
qpm: west_of	1,658	1,034	19,758	21	22,429	7.39
qpm: north_of	1,630	970	19,164	57	21,707	7.51
qpm: south_of	1,639	961	26,394	23	28,971	5.66
qpm: contained_by	1,719	0	1,697	0	3,416	50.32
qpm: hasChildUnit	0	1,719	0	0	1,719	0.00
(sum directions)	6576	4008	92131	177	102538	6.41
(sum containment)	1719	1719	1697	0	5135	33.47

Notes. "Inverse" adds the predicate's inverse edges (e.g., $\text{west_of}^{-1} \Rightarrow \text{east_of}$) not already stored under that predicate. "Transitive" is per-predicate monotone closure (e.g., $\text{east_of} \circ \text{east_of} \Rightarrow \text{east_of}$) with de-duplication; "Common" counts edges derivable by both inverse and transitive (but not already stored). For *contained_by* in the admin hierarchy: stored = 22 + 1697 = 1719 (UA $\rightarrow$ Wales plus Community $\rightarrow$ UA); transitive new = 1697 (Community $\rightarrow$ Wales); hence full = 3416. For *hasChildUnit*, full equals the number of non-root units: 1719.

Table 4. Postal hierarchy: stored vs. inferred sizes per predicate. "Full" is the de-duplicated union of stored, inverse, and per-predicate transitive edges.

Predicate	Stored	Inverse	Transitive	Common	Full	Stored/Full (%)
qpm: east_of	667	310	3,784	33	4,728	14.1
qpm: west_of	659	318	3,449	7	4,419	14.9
qpm: north_of	637	331	4,032	19	4,981	12.8
qpm: south_of	649	319	4,633	12	5,589	11.6
qpm: contained_by	739	0	534	0	1,273	58.1
qpm: hasChildUnit	0	739	0	0	739	0.0
(sum directions)	2612	1278	15898	71	19717	13.3
(sum containment)	739	739	534	0	2012	36.7

Notes. For postal containment: stored = 205 (district $\rightarrow$ area) + 534 (sector $\rightarrow$ district) = 739; transitive new = 534 (sector $\rightarrow$ area); hence full = 1273. For *hasChildUnit*, full equals non-root units: 205 + 534 = 739.

Table 5. Electoral hierarchy: stored vs. inferred sizes per predicate. "Full" is the de-duplicated union of stored, inverse, and per-predicate transitive edges.

Predicate	Stored	Inverse	Transitive	Common	Full	Stored/Full (%)
qpm:east_of	733	394	7,326	28	8,425	8.7
qpm:west_of	729	398	6,233	18	7,342	9.9
qpm:north_of	712	377	5,260	23	6,326	11.3
qpm:south_of	719	370	7,576	18	8,647	8.3
qpm:contained_by	774	0	769	0	1,543	50.2
qpm:hasChildUnit	0	774	0	0	774	0.0
(sum directions)	2893	1539	26395	87	30740	9.4
(sum containment)	774	774	769	0	2317	33.4

Notes. For electoral containment: stored = 5 (region→Wales) +769 (ward→region) = 774; transitive new = 769 (ward→Wales); hence full = 1543. For hasChildUnit, full equals non-root units: 5 + 769 = 774.

The Stored/Full ratios in Table 6 reflects a local, single-predicate transitive closure: east_of chains only yield further east_of pairs. A richer qualitative reasoning can leverage *composition tables* for directions: if A east_of B and B north_of C, then A is constrained to be in the NE cone of C. This yields non-local inferences that cut across predicates.

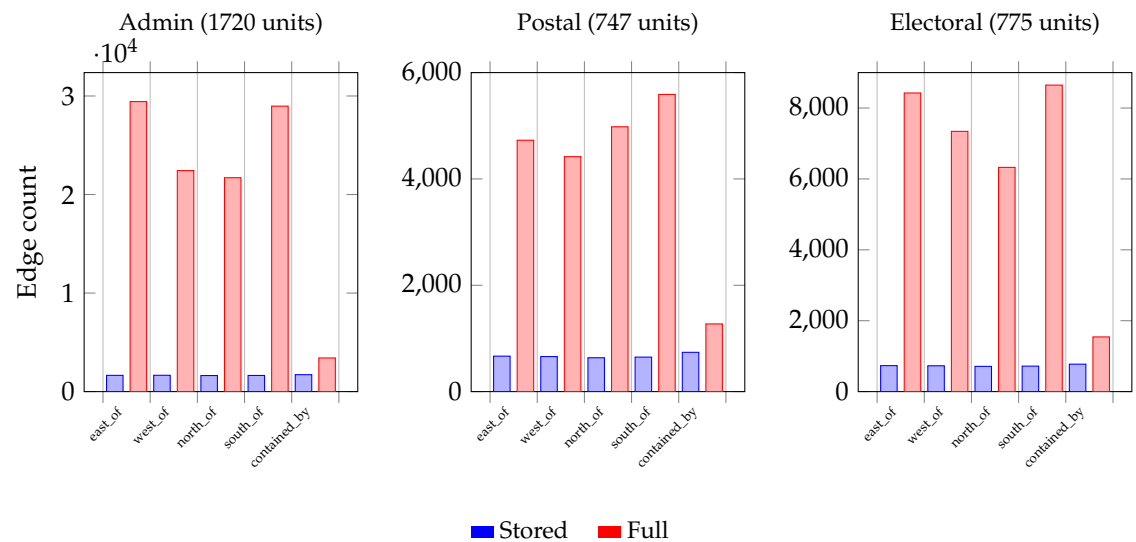


Figure 11. Stored vs. full edges per predicate across the three hierarchies. QPM stores minimal local relations that expand substantially under logical inference (inverse relationships and transitive closure), demonstrating compact storage with rich inferential power.

Table 6. Share of stored edges within full directional closure (sum over E/W/N/S).

Hierarchy	Stored	Full	Stored / Full
Administrative	6,576	102,538	6.4%
Postal	2,612	19,717	13.3%
Electoral	2,893	30,740	9.4%

6. Discussion

6.1. Empirical Validation and Model Performance

The evaluation demonstrates that QPM generates accurate Qualitative Place Descriptions across heterogeneous spatial hierarchies with high fidelity. For basic places, 95.8% of expected RDF statements were successfully generated (Table 2); for administrative units, coverage ranged from 89.7% (postal) to

96.5% (administrative hierarchies). All QPDs were confirmed unique through deterministic signature verification (Algorithm 1), while spatial relationships satisfied formal constraints: asserted directions maintained $dir \sqsubseteq touches$ for units, inverse pairs were correctly generated, and self-loops were absent.

The observed gaps in coverage arise primarily from legitimate edge effects where places near Wales' irregular boundaries naturally lack neighbors in certain directions (e.g. coastal boundaries). This pattern is consistent with the spatial structure of the test region and does not indicate systematic computational errors. The high uniqueness rate validates QPM's deterministic approach to tie-breaking and relationship selection, ensuring reproducible results across multiple computational runs.

6.2. Computational Efficiency and Inferential Power

A key design principle of QPM is the storage of minimal local relations that expand substantially under sound logical inference. This approach achieves significant computational efficiency while maintaining expressive power. Aggregating across administrative, postal, and electoral hierarchies, directional relations expand from 12,081 stored edges to 152,995 inferred edges—a 12.66× expansion with only 7.9% storage overhead. Containment relationships expand from 3,232 to 9,464 edges (2.93× expansion), yielding an overall system expansion of 10.6× from stored to fully inferred representations.

This expansion occurs through two complementary inference mechanisms: predicate inverses (generating symmetric relationships) and per-predicate monotone transitivity (extending chains of directional relationships). The substantial expansion demonstrates that QPM's sparse storage strategy does not compromise inferential capability—rather, it enables rich qualitative reasoning from a compact, easily validated foundational representation.

The computational complexity analysis (Section 4.5) indicates that relationship computation scales near-linearly with input size when spatial indexing is employed, making the approach viable for large-scale operational deployment. However, the current implementation lacks incremental update mechanisms, requiring full recomputation when spatial boundaries or place locations change.

6.3. Theoretical Contributions and Alignment

QPM contributes to spatial information theory by demonstrating that qualitative and quantitative spatial representations can be integrated within a single computational framework without sacrificing the strengths of either approach. The model aligns with Kuhn's core concepts of spatial information [26]: *location* is captured both geometrically and relationally; *neighbourhood* is explicit through topological and directional predicates; *object* is realized through feature classes; *network* emerges from containment and lateral relationships; and *granularity* is controlled through hierarchical level indices.

This integration supports Agnew's perspective that places are understood through their relations to other places [1], operationalized through primary neighbour selection and hierarchical anchoring. The approach also coheres with place-based GIS perspectives [9] by treating places as relationally-defined entities rather than isolated geometric objects.

The homogeneous treatment of administrative units and named places under a unified *Place* superclass represents a theoretical advancement over approaches that handle different spatial entity types through separate frameworks. This unification enables consistent spatial reasoning across entity types while maintaining computational efficiency.

6.4. Practical Implications and Integration

By implementing all computational steps within ArcGIS Pro—from data cleaning through relationship computation to geodatabase population—QPM maintains fidelity to standard GIS workflows while avoiding dependencies on external graph platforms or semantic web infrastructure. This integration enables immediate cartographic feedback and supports human-centric spatial queries ("Where is place x?", "Which regions are north of y?") using either native GIS tools or SPARQL endpoints.

The dual representation capability (geometric and qualitative) allows practitioners to leverage both computational paradigms within unified analytical workflows. Geometric representations support precise measurement and quantitative analysis, while qualitative representations enable

natural-language queries and cognitively aligned spatial reasoning. This complementarity addresses a significant gap in current GIS capabilities.

7. Conclusions

This paper presented the *Qualitative Place Model* (QPM), a homogeneous, GIS-native framework for representing places through qualitative spatial relationships that complement traditional geometric representations. Implemented as an ArcGIS Pro toolbox, QPM computes and stores a deterministic, compact set of local relations—hierarchical containment, directional neighbourhood, and optional proximity—that remain faithful to authoritative boundary datasets while expanding substantially under sound logical inference rules.

QPM makes four principal contributions to spatial information science: a homogeneous place formalism treating administrative units and named places uniformly; deterministic in-GIS computation with reproducible algorithms; compact storage achieving rich inferential power through $10.6\times$ expansion under logical closure; and empirical validation demonstrating robust performance across heterogeneous spatial hierarchies (95.8% coverage for Basic Places; 89.7–96.5% for administrative hierarchies). Evaluation on Welsh administrative, postal, and electoral hierarchies, together with 52,821 place entities, confirms that all Qualitative Place Descriptions are unique under deterministic signature verification.

The framework addresses a persistent disconnect between theoretical qualitative spatial reasoning and practical GIS implementation by integrating qualitative representation *within* rather than *alongside* standard workflows. This enables natural-language-style spatial queries ("Where is this place across hierarchies?", "Which regions are north of x ?") directly within mainstream GIS tools. QPM demonstrates that homogeneous, sparsely stored qualitative models can deliver substantial inferential power through simple, sound inference rules while remaining implementable at scale within operational GIS environments.

7.1. Limitations and Future Directions

Some limitations that constrain QPM's current scope are identified to suggest directions for future research. Technical limitations include the lack of incremental update mechanisms for dynamic datasets and platform dependency on ArcGIS Pro. Also, limitations arise from dependence on authoritative boundary datasets, which may be unavailable or inappropriate for some regions or applications. The deterministic tie-breaking approach, while ensuring reproducibility, may not always align with human intuitions about spatial relationships. Additionally, the current directional model uses fixed cardinal/ordinal sectors, which may be inadequate for applications requiring finer directional resolution.

Future research directions include: (1) developing cross-predicate composition tables for directional reasoning (e.g., $E \circ N \Rightarrow NE$) to enhance inferential power; (2) implementing incremental maintenance strategies for dynamic spatial datasets; (3) exploring Discrete Global Grid Systems (DGGS) as universal hierarchical substrates where authoritative partitions are unavailable; and (4) conducting cognitive evaluations to assess the alignment between QPM-generated descriptions and human spatial intuitions.

QPM bridges a significant gap between theoretical spatial reasoning capabilities and practical GIS implementation, demonstrating that qualitative spatial reasoning can be successfully integrated into mainstream GIS without sacrificing computational efficiency. By augmenting geometric representations with explainable qualitative semantics that align with natural language descriptions, QPM enhances how spatial analysts interact with geographic information. As spatial analysis becomes increasingly central to decision-making across diverse domains, frameworks combining computational efficiency with cognitive alignment will prove essential for realizing the full potential of geographic information systems in supporting human spatial reasoning and communication.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, Alia I. Abdelmoty; methodology, Alia I. Abdelmoty and Abdurauf Satoti; software, Abdurauf Satoti; validation, Alia I. Abdelmoty and Abdurauf Satoti; formal analysis, Alia I. Abdelmoty and Abdurauf Satoti; investigation, Alia I. Abdelmoty and Abdurauf Satoti; resources, Abdurauf Satoti; data curation, Abdurauf Satoti; writing—original draft preparation, Abdurauf Satoti and Alia I. Abdelmoty; writing—review and editing, Alia I. Abdelmoty; visualization, Abdurauf Satoti; supervision, Alia I. Abdelmoty; All authors have read and agreed to the published version of the manuscript.”, please turn to the [CRediT taxonomy](#) for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: “This research received no external funding”.

Data Availability Statement: The data analysed in this study are publicly available through the Ordnance Survey web site : [OS Data Hub - OpenMapLocal](#). Sample datasets, SPARQL queries and resulting RDF graphs produced by the GIS are available at GITHUB repo.

Acknowledgments: The authors utilized Anthropic’s Claude Sonnet 4 and OpenAI’s ChatGPT (GPT-5) for language editing and proofreading assistance. These tools were employed solely to enhance the readability of the manuscript; all intellectual content, analyses, and conclusions are the authors’ original work.

Conflicts of Interest: The authors have no competing interests to declare that are relevant to the content of this article.

Appendix A. Detailed Algorithms

This appendix provides detailed algorithmic specifications for computing Qualitative Place Descriptions QPDs. All algorithms follow explicit contracts specifying preconditions, processing methods, and postconditions to ensure reproducible implementation.

Algorithm 1: Containment for Spatial Units

Precondition: Level L_{x-1} forms a gap-free, overlap-free partition; point-in-polygon queries available. **Method:** For each unit $u \in L_x$, compute interior point and find unique containing parent in L_{x-1} . **Postcondition:** Each unit assigned to exactly one parent; boundary conflicts flagged.

Algorithm A1 Compute Containment for Spatial Units

```

1: Input: Hierarchies  $H$ ; levels  $L_x$  with polygon geometries
2: Output: Parent assignment  $\text{parent}(u)$  for each  $u \in L_x$  ( $x \geq 1$ )
3: for each hierarchy  $h_i \in H$  do
4:   for each level  $x = 1, \dots, M$  in  $h_i$  do
5:     Build spatial index  $I$  over  $L_{x-1}$ 
6:     for each unit  $u \in L_x$  do
7:        $q \leftarrow \text{InteriorPoint}(u)$ 
8:        $P \leftarrow \{p \in L_{x-1} \mid q \in p\}$ 
9:       if  $|P| = 1$  then
10:         $\text{parent}(u) \leftarrow \text{unique } p \in P$ 
11:       else
12:        Flag boundary conflict for  $u$ 
13:       end if
14:     end for
15:   end for
16: end for

```

Algorithm 2: Directional Relationships for Spatial Units

Precondition: Clean polygons at level L_x ; adjacency predicate $\text{touches}(u, v)$ available; directional sector function defined. **Method:** For each unit, identify touching neighbours, group by directional sector, select primary neighbour per sector using lexicographic tie-breaking. **Postcondition:** At most one neighbour per sector; directional relationships imply adjacency; inverse pairs maintained.

Algorithm A2 Compute Directional Relationships for Spatial Units

```

1: Input: Hierarchies  $H$ ; adjacency predicate  $\text{touches}$ ; sector function  $\text{sector}(u, v)$ 
2: Output: Primary directional neighbour per sector for each unit
3: for each hierarchy  $h_i \in H$  do
4:   for each level  $L_x$  in  $h_i$  do
5:     for each unit  $u \in L_x$  do
6:        $S \leftarrow$  candidate set (same parent if  $x > 0$ , else all units in  $L_x$ )
7:       for each sector  $d \in \{N, NE, E, SE, S, SW, W, NW\}$  do
8:          $T_d \leftarrow \{v \in S \mid \text{touches}(u, v) \wedge \text{sector}(u, v) = d\}$ 
9:         if  $T_d \neq \emptyset$  then
10:          Select  $v^* \in T_d$  using lexicographic tie-breaking
11:          Assign  $\text{primaryDir}_d(u) \leftarrow v^*$ 
12:         end if
13:       end for
14:     end for
15:   end for
16: end for

```

Tie-breaking Criteria

Primary neighbour selection uses lexicographic ordering: (1) maximum shared boundary length, (2) minimum angular deviation from sector bearing, (3) minimum centroid distance, (4) lexicographic identifier comparison.

Algorithm 3: Host Assignment for Basic Places

Precondition: Basic Places as points; base-level Units as polygons; point-in-polygon queries available. **Method:** Assign each Basic Place to containing Unit at base level using point-in-polygon analysis. **Postcondition:** Each Basic Place assigned to exactly one host Unit.

Algorithm A3 Assign Host Units for Basic Places

```

1: Input: Place layer  $PL$  (points); base-level Units  $L_n$  (polygons)
2: Output: Host assignment  $\text{host}(P)$  for each  $P \in PL$ 
3: Build spatial index  $I$  over base-level Units  $L_n$ 
4: for each Basic Place  $P \in PL$  do
5:    $U \leftarrow \{u \in L_n \mid P \in u\}$ 
6:   if  $|U| = 1$  then
7:      $\text{host}(P) \leftarrow$  unique  $u \in U$ 
8:   else
9:     Flag placement conflict for  $P$ 
10:  end if
11: end for

```

Algorithm 4: Directional Relationships for Basic Places

Precondition: Basic Places with assigned host Units; distance function defined; directional sectors specified. **Method:** For each Basic Place and directional sector, identify nearest Basic Place in that sector using spatial indexing. **Postcondition:** At most one primary neighbour per sector; relationships functional per sector.

Algorithm A4 Compute Directional Relationships for Basic Places

```

1: Input: Place layer  $PL$ ; distance function  $d(P, Q)$ ; sector function  $\text{sector}(P, Q)$ 
2: Output: Primary directional neighbour per sector for each place
3: Build spatial index  $I$  over  $PL$ 
4: for each Basic Place  $P \in PL$  do
5:   for each sector  $d \in \{N, NE, E, SE, S, SW, W, NW\}$  do
6:      $C \leftarrow \{Q \in PL \setminus \{P\} \mid \text{sector}(P, Q) = d\}$ 
7:     if  $C \neq \emptyset$  then
8:        $Q^* \leftarrow \arg \min_{Q \in C} d(P, Q)$ 
9:       Assign  $\text{primaryDir}_d(P) \leftarrow Q^*$ 
10:    end if
11:  end for
12: end for

```

Computational Complexity

Algorithm Performance

- **Containment (Algorithm A1):** $O(|L_x| \log |L_{x-1}|)$ per level, where $|L_x|$ is the number of units at level x and $|L_{x-1}|$ is the number of parent units
- **Directional Units (Algorithm A2):** $O(n^2)$ worst case, $O(n \log n)$ with spatial indexing optimizations
- **Host Assignment (Algorithm A3):** $O(|PL| \log |L_n|)$ with spatial indexing, where $|PL|$ is the number of Basic Places and $|L_n|$ is the number of base-level units
- **Directional Places (Algorithm A4):** $O(|PL|^2)$ worst case, optimizable with spatial partitioning

Optimization Notes

All algorithms benefit from spatial indexing (R-trees) for geometric queries. Processing can be parallelized at the unit/place level within each hierarchy. Memory requirements scale linearly with input size; disk-based processing suitable for large datasets.

Tie-breaking Criteria and Deterministic Selection

QPM employs lexicographic tie-breaking to ensure deterministic, reproducible relationship selection across multiple computational runs. These criteria are essential for generating unique QPDs and enabling consistent evaluation results.

Tie-breaking for Units (Directional Relationships)

When multiple touching neighbours exist within directional sector d for unit u (i.e., $|T_d(u)| > 1$), primary neighbour selection follows this lexicographic ordering:

1. **Maximum shared boundary length:** $\max(\text{sharedBoundaryLength}(u, v))$ - prioritizes neighbours with substantial geographic contact
2. **Minimum angular deviation from sector center:** $\min(|\text{bearing}(u, v) - \text{midAngle}(d)|)$ - selects neighbours best aligned with directional sector
3. **Minimum centroid distance:** $\min(d(\text{centroid}(u), \text{centroid}(v)))$ - prefers spatially closer neighbours
4. **Lexicographic identifier ranking:** $\min(\text{idRank}(v))$ - ensures deterministic selection using stable unit identifiers

The complete selection criterion is:

$$v^* = \arg \min_{v \in T_d(u)} \left(-\text{sharedBoundaryLength}(u, v), \right. \\ \left. |\text{bearing}(u, v) - \text{midAngle}(d)|, \right. \\ \left. d(\text{centroid}(u), \text{centroid}(v)), \right. \\ \left. \text{idRank}(v) \right) \quad (\text{A1})$$

This prioritizes spatially significant relationships while ensuring deterministic selection across multiple computational runs.

Tie-breaking for Basic Places (Directional Relationships)

For Basic Places within directional sectors where $|C_d(P)| > 1$, selection uses:

1. **Minimum distance:** $\min(d(P, Q))$ using Euclidean distance for projected coordinates or geodesic distance for geographic coordinates
2. **Lexicographic identifier ranking:** $\min(\text{idRank}(Q))$ using stable place identifiers

The selection criterion is:

$$Q^* = \arg \min_{Q \in C_d(P)} (d(P, Q), \text{idRank}(Q))$$

Boundary Conflict Resolution

When places fall exactly on boundaries between units or when geometric precision creates ambiguity:

For Containment Assignment:

1. **Largest containment area:** Assign to unit with maximum $|\text{geom}(u) \cap \text{buffer}(P, \epsilon)|$ where ϵ is a small tolerance
2. **Geometric centroid proximity:** Unit with centroid closest to place location
3. **Identifier precedence:** Lexicographically smallest unit identifier

For Multi-Hierarchy Conflicts: When a place could be assigned to multiple units across different hierarchies, each hierarchy assignment is computed independently using the above criteria, allowing places to have multiple valid containment relationships across organizational systems.

Implementation Notes

- All distance calculations use the coordinate reference system of the input data
- Identifier ranking (idRank) uses alphanumeric sorting of stable unit/place identifiers
- Bearing calculations use the centroid-to-centroid vector between spatial units
- Shared boundary length is computed using high-precision geometric intersection
- All tie-breaking decisions are logged for quality assurance and reproducibility verification

These tie-breaking rules ensure that:

1. Every entity receives exactly one primary neighbour per directional sector (when candidates exist)
2. Results are reproducible across different computational environments
3. Spatial significance is prioritized over arbitrary characteristics
4. The selection process remains stable under minor geometric perturbations

References

1. Agnew, J. Space and place. *Handbook of geographical knowledge* **2011**, 2011, 316–331.
2. Hamzei, E.; Winter, S.; Tomko, M. Place facets: a systematic literature review. *Spatial Cognition & Computation* **2020**, 20, 33–81.
3. Winter, S.; Baldwin, T.; Cavedon, L.; Stirling, L.; Kealy, A.; Duckham, M.; Rajabifard, A.; RICHTER, K.; Richter, D. Starting to talk about place. In Proceedings of the Proc. Surveying and Spatial Sciences Biennial Conference 2011. Surveying and Spatial Sciences Institute, 2011, pp. 63–73.
4. Chen, H.; Vasardani, M.; Winter, S.; Tomko, M. A graph database model for knowledge extracted from place descriptions. *ISPRS International Journal of Geo-Information* **2018**, 7, 221.
5. Richter, D.; Winter, S.; Richter, K.F.; Stirling, L. Granularity of locations referred to by place descriptions. *Computers, environment and urban systems* **2013**, 41, 88–99.
6. Freundschuh, S.; Kitchen, R. Cognitive mapping: Current theory and practice. *Professional Geographer* **2001**.
7. Aronoff, S. Geographic information systems: a management perspective **1989**.
8. Jones, L. Geographic information systems. *Essentials of Environmental Epidemiology for Health Protection: A handbook for field professionals* **2012**, p. 121.
9. Purves, R.S.; Winter, S.; Kuhn, W. Places in Information Science. *Journal of the Association for Information Science and Technology* **2019**, 70, 1173–1182. <https://doi.org/10.1002/asi.24194>.
10. Hart, G.; Dolbear, C. *Linked data: a geographic perspective*; Taylor & Francis, 2013.
11. Abdelmoty, A.I.; Satoti, A. A Homogeneous Approach to Reasoning Over Global Geographic Data. In Proceedings of the International Conference on Innovative Techniques and Applications of Artificial Intelligence. Springer, 2024, pp. 285–298.
12. Cohn, A.G.; Bennett, B.; Gooday, J.; Gotts, N.M. Qualitative spatial representation and reasoning with the region connection calculus. *geoinformatica* **1997**, 1, 275–316.
13. Egenhofer, M.J.; Mark, D.M. Naive geography. In Proceedings of the International Conference on Spatial Information Theory. Springer, 1995, pp. 1–15.
14. Egenhofer, M.J.; Franzosa, R.D. Point-set topological spatial relations. *International Journal of Geographical Information System* **1991**, 5, 161–174.
15. Freksa, C. Using orientation information for qualitative spatial reasoning. In *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*; Springer, 1992; pp. 162–178.
16. Vasardani, M.; Timpf, S.; Winter, S.; Tomko, M. From descriptions to depictions: A conceptual framework. In Proceedings of the Spatial Information Theory: 11th International Conference, COSIT 2013, Scarborough, UK, September 2–6, 2013. Proceedings 11. Springer, 2013, pp. 299–319.
17. Hamzei, E.; Chen, H.; Vasardani, M.; Tomko, M.; Winter, S. Deriving place graphs from spatial databases. *Research@ Locate* **2018**, 2087, 25–32.
18. Abdelmoty, A.I.; Al-Muzaini, K.O. A computational model of place on the linked data web. *International Journal on Advances in Software* **2016**, 9, 238–247.
19. Goodwin, J. Experiences of Using OWL at the Ordnance Survey. In Proceedings of the OWLED, 2005, Vol. 188.
20. Janowicz, K.; Hitzler, P.; Li, W.; Rehberger, D.; Schildhauer, M.; Zhu, R.; Shimizu, C.; Fisher, C.K.; Cai, L.; Mai, G.; et al. Know, Know Where, Know Where Graph: A densely connected, cross-domain knowledge graph and geo-enrichment service stack for applications in environmental intelligence. *AI Magazine* **2022**, 43, 30–39. <https://doi.org/10.1002/aaai.12043>.
21. KnowWhereGraph. About KnowWhereGraph Geo_enrichment. [KnowWhereGraph About Page](#), 2025. Accessed: 2025-01-23.
22. Longley, P.A.; Goodchild, M.F.; Maguire, D.J.; Rhind, D.W. *Geographic information science and systems*; John Wiley & Sons, 2015.
23. Al-Muzaini, K.O. Qualitative Modelling of Place Location on the Linked Data Web and GIS. PhD thesis, Cardiff University, 2017.
24. Giordano, A.; Cole, T. The limits of GIS: Towards a GIS of place. *Transactions in GIS* **2018**, 22, 664–676.

25. Martin, M.E.; Swanlund, D.; Schuurman, N. Problems with quantitative categorization: An argument for qualitative approaches. *Environment and Planning F* **2023**, *2*, 331–349.
26. Kuhn, W. Core concepts of spatial information for transdisciplinary research. *International Journal of Geographical Information Science* **2012**, *26*, 2267–2276.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.