

Article

Not peer-reviewed version

---

# Transforming FHIR into an OWL Knowledge Graph for Schema-Grounded Natural-Language Querying and Exploratory Data Analysis

---

[Steve K. Platt](#)\*, [Daniel B. Hier](#), [Borchuluun Yadamsuren](#), [Anh N. Nguyen](#), Vaughn Hartzell

Posted Date: 24 March 2026

doi: 10.20944/preprints202603.1877.v1

Keywords: FHIR; OWL; RDF; knowledge graph; ontology







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

# Transforming FHIR into an OWL Knowledge Graph for Schema-Grounded Natural-Language Querying and Exploratory Data Analysis

Steve K. Platt <sup>1,\*</sup> , Daniel B. Hier <sup>2</sup> , Borchuluun Yadamsuren <sup>1</sup> , Anh N. Nguyen <sup>1</sup>  and Vaughn Hartzell <sup>1</sup> 

<sup>1</sup> Lab for Applied Artificial Intelligence, Quinlan School of Business, Loyola University Chicago

<sup>2</sup> Department of Neurology and Rehabilitation, University of Illinois at Chicago

\* Correspondence: splatt1@luc.edu

## Abstract

FHIR was designed for transactional interoperability but is less well suited to querying and exploratory analysis because its resource-centric structure distributes meaning across deeply nested resources. To address this limitation, we transformed MIMIC-IV Demo FHIR data into an OWL-compliant knowledge graph by flattening nested elements, normalizing repeating arrays, resolving inter-resource references, and promoting frequently queried attributes to direct properties. We also aligned diagnosis and procedure codes to ICD-9-CM and ICD-10-CM terminologies and developed a schema-grounded NL2SPARQL interface for natural-language querying. Structural validation was performed with SHACL and OWL reasoning. Across a curated evaluation set, NL2SPARQL achieved a mean accuracy exceeding 95% relative to expert-authored queries. These results suggest that ontologizing FHIR can improve analytic accessibility while preserving clinically meaningful assertions.

**Keywords:** FHIR; OWL; RDF; knowledge graph; ontology

## 1. Introduction

The rapid growth of electronic health records (EHRs) has intensified the need for sharing clinical data across institutions and heterogeneous health information systems. The Fast Healthcare Interoperability Resources (FHIR) standard has emerged as the dominant framework for representing and exchanging clinical information. FHIR is optimized for transactional interoperability at the point of care, enabling structured data exchange through modular, resource-based representations.

However, FHIR was not designed for long-term archival storage or large-scale analytical workflows. Its verbose, resource-centric architecture distributes clinical meaning across networks of interlinked resources connected by reference relationships. While this design promotes flexibility and interoperability, it introduces challenges for storage efficiency, analytic accessibility, and longitudinal data analysis.

Rather than aggregating entire institutional record collections, some high-value secondary-use applications, such as specific disease phenotyping or standardized cancer reporting, rely on multi-institutional datasets that are clinically narrow but longitudinally deep [1–5]. These multi-institutional applications place heavy structural and semantic demands on FHIR data models. To track patient data over time across different sites, researchers must harmonize diverse coding practices and terminologies so that encounters, diagnoses, laboratory results, and medications are represented consistently. Consequently, downstream investigative tasks such as exploratory data analysis, cohort-level aggregation, feature engineering for machine learning, and alignment with external biomedical ontologies require explicit structural operations that modify FHIR's native transactional format. Recent surveys document a growing body of work aimed at enhancing semantic interoperability through FHIR-based architectures, including terminology normalization, ontology integration, RDF/OWL representations,

and machine learning approaches. Despite these advances, significant challenges remain in structuring FHIR-derived data for consistent ontology-aware querying and scalable secondary research.

In this paper, we define ontologization as the transformation of EHR-derived FHIR data into an OWL-compliant graph that preserves essential semantic assertions while enhancing data accessibility and enabling expressive, ontology-aware exploration. It is not our intent to build a fully axiomatized domain ontology. Rather, we create a lightweight ontologically structured knowledge graph optimized specifically for analytic accessibility. Thus, we do not attempt to satisfy the full set of formal design principles associated with foundational ontologies such as those articulated by the Open Biological and Biomedical Ontology Foundry (OBO) [6–8].

Although Health Level Seven standards (HL7) provide a normative RDF representation of FHIR [9] designed to preserve round-trip fidelity with JSON and XML exchange formats, that representation retains the native resource nesting and structural conventions of transactional interoperability. Our use of ontologization relaxes this exchange constraint to restructure FHIR-derived data for analytic accessibility and schema-grounded querying.

Classical ontology engineering [10,11] distinguishes between the ontology and the knowledge graph. The ontology defines classes, properties, and logical constraints (the TBox), whereas the knowledge graph contains assertions about individuals (the ABox) that instantiate those classes. Because our structural transformation operates primarily at this instance level, individual FHIR resources are mapped as explicit data points under lightweight class definitions and aligned to external biomedical terminologies (ICD-9-CM, ICD-10-CM). Accordingly, the resulting artifact is best characterized as a data-rich, ontologically structured knowledge graph rather than a fully axiomatized domain ontology.

In this context, ontologization emphasizes consistent typing, explicit representation of entities and relations, and principled alignment with controlled biomedical vocabularies. The objective is not to construct a comprehensive upper-level ontology, but to produce a semantically disciplined graph that preserves clinically relevant distinctions while improving analytic accessibility, interoperability, and downstream analytic utility.

Knowledge graphs expressed in RDF and enriched with OWL semantics provide a natural framework for this approach because they represent entities and relationships explicitly, support expressive query languages such as SPARQL, and enable direct linkage to external ontologies through globally resolvable identifiers. Transforming FHIR data into such graph representations increases analytical flexibility, facilitates principled terminology integration, and improves semantic queryability compared with the deeply nested, document-oriented structure of underlying FHIR resources.

An ontologized clinical knowledge graph is especially attractive because OWL introduces formal semantics that distinguish classes from individuals and object properties (relations between entities) from datatype properties (links to literal values). In graph terms, this allows inter-entity relationships to be represented explicitly as edges, while literal-valued attributes are modeled as OWL datatype assertions on nodes. In contrast, RDF alone does not enforce the formal semantic constraints made available by OWL's description logic layer. These distinctions support basic logical validation, reduce ill-formed mappings, and enable more consistent terminology alignment than RDF triples alone, even when full ontology reasoning is not employed. Although HL7 supports representing FHIR resources in RDF [9], this encoding retains deeply nested elements (requiring anonymous blank nodes), multiple scalar classes (e.g., choice[x] or polymorphic elements), and variable-length arrays (e.g., repeating elements), with the result that transforming an RDF FHIR graph into an OWL-compliant graph is technically challenging.

Our research focuses on addressing three major structural challenges within RDF FHIR that complicate its conversion into an OWL-compliant knowledge graph (Table 1).

**Table 1.** FHIR structural features that challenge the transformation into OWL-compliant knowledge graphs.

Structural feature	Implication
Deep nesting	Hierarchical data structures extend across multiple levels.
Variable-width Arrays	Unbounded repeating elements require explicit modeling.
Choice Fields	Multiple value type fields that introduce heterogeneity.

Beyond the structural challenges inherent in FHIR, secondary use of FHIR data introduces additional challenges related to scale and semantic integration. The verbose, resource-centric design of FHIR distributes clinical meaning across numerous interlinked resources and produces structurally complex graph representations even for modest cohorts. Moreover, alignment with external terminology systems such as ICD-9-CM, ICD-10-CM, and SNOMED CT remains an ongoing challenge [12]. To address these limitations, we adopt a pragmatic ontologization strategy that restructures FHIR-derived data into an OWL-compliant knowledge graph optimized for terminology integration, which ultimately enables robust clinical analysis.

However, structural re-expression alone does not guarantee practical analytic accessibility. Effective querying of OWL-compliant knowledge graphs requires both familiarity with the ontology schema and mastery of SPARQL syntax. Authoring correct SPARQL queries demands syntactic precision and a detailed understanding of classes, object properties, and datatype properties within the data model. Prior evaluations indicate that large language models exhibit degraded performance when generating structured SPARQL queries without explicit schema grounding and contextual alignment with the target knowledge graph [13,14]. Benchmark studies have shown that foundation LLMs such as GPT 3.5 and GPT 4.0 struggle to generate accurate SPARQL queries for knowledge graphs [15]. These studies suggest that semantic restructuring must be complemented by mechanisms that create a bridge between natural language and formally constrained graph query languages. We demonstrate this integrated approach using the MIMIC-IV Demo dataset as a representative validation dataset. Applying our transformation pipeline to the native FHIR data yielded two complementary graph representations:

1. A FHIR-faithful RDF graph that follows the HL7 FHIR-RDF specifications [8] and therefore preserves native nesting, references, and round-trip fidelity to JSON and XML representations; and
2. An OWL-compliant knowledge graph obtained via ontologization, optimized for concise data exploration and semantic integration.

Ontologization here refers to a set of structural transformations: flattening nested backbone elements, normalizing repeating arrays, resolving polymorphic choice fields, and replacing blank-node reference structures with explicitly named OWL object properties. Unlike the HL7-sanctioned RDF representation of FHIR, which preserves round-trip fidelity with the original FHIR resources, our OWL-compliant representation intentionally relaxes this constraint to reduce query depth and improve analytic accessibility.

To overcome the structural barriers of native FHIR for research, our core contributions are threefold:

1. *Structural Ontologization*: We present a transformative architecture that intentionally relaxes FHIR's strict round-trip constraints to prioritize analytic accessibility. By collapsing the multi-hop blank-node traversals of standard FHIR.ttl into single-step property assertions, we deliver an OWL-compliant knowledge graph optimized for complex, cohort-level exploration without sacrificing clinical fidelity.
2. *Terminology-Grounded Querying*: We integrate ICD-9-CM and ICD-10-CM terminologies directly into the graph model, elevating clinical codes from flat strings to explicit, interoperable resource instances. This enables reproducible, terminology-driven validation and cross-system consistency.

3. *Democratized Analytic Access*: We introduce a highly accurate (mean accuracy >95%) NL2SPARQL interface that effectively bridges the gap between foundation large language models and formal graph query languages. Guided directly by the OWL schema, this tool allows domain experts to execute robust exploratory analyses using natural language, significantly lowering the barrier to entry for clinical research.

Taken together, these components demonstrate that ontologization can transform FHIR-derived clinical data from a transaction-optimized exchange format into a pragmatic, analytically accessible knowledge graph.

The remainder of the paper proceeds as follows. First, we survey related approaches to FHIR query optimization. In the Methods section, we describe the end-to-end transformation pipeline from NDJSON FHIR resources to an OWL-compliant graph, followed by the architecture and operational design of the NL2SPARQL query interface. In the Results section, we characterize the structural properties of the ontologized graph and the performance of representative queries. Finally, in the Discussion section, we examine methodological trade-offs, observed limitations, and implications for future knowledge graph development.

### 1.1. Other Approaches

To contextualize our proposed framework, we survey existing approaches to FHIR query optimization and semantic integration. Several tools and frameworks address the structural and analytic challenges posed by FHIR. One prominent server-side approach is *Pathling* [12], which implements a FHIR Analytics API to support cohort selection, exploratory analysis, and data preparation directly over FHIR resources. Pathling extends the REST interface with operations such as *Import*, *Aggregate*, *Extract*, and enhanced *Search*, and it achieves scalability by leveraging Apache Spark and columnar storage (e.g., Parquet). By integrating with terminology services, it supports clinically meaningful analytics beyond purely syntactic flattening. Rosenau et al.[16] have applied this approach to extracting data from the German Portal for Medical Research Data.

In a related effort, Grimes et al. introduced SQL on FHIR [17], which defines implementation-agnostic, tabular “views” over FHIR resources using FHIR Path expressions. These views provide a standardized mapping layer from hierarchical resources to tables that can be consumed by SQL engines and machine-learning pipelines.

Complementing these server-oriented frameworks, client-side packages focus on making FHIR data practical for common data-science workflows. *FHIR-PYrate* [18] provides a Python-based pipeline for querying FHIR servers via REST and transforming hierarchical resources into Pandas DataFrames, with support for cohort-oriented querying and optional extensions (e.g., text mining). In the R ecosystem, *fhircrackr* [19] similarly retrieves resources through the Search API and converts them into rectangular tables while handling repeating elements.

A fourth strategy is explicitly graph-native. *FHIR Lens* [20] represents FHIR resources as RDF graphs and supports semantic exploration, including visualization and recursive traversal of relationships, with optional deployment as triple stores for applications such as SPARQL querying of a GraphDB knowledge graph.

Taken together, these systems illustrate four broad directions: server-side frameworks that prioritize scalable analytics and standardized access (Pathling, SQL on FHIR); client-side toolkits that flatten resources for conventional statistical and machine-learning workflows (FHIR-PYrate, fhircrackr); and graph-oriented platforms that preserve relational structure for semantic exploration (FHIR Lens). Existing approaches typically isolate their focus on analytic scalability, tabular abstraction, or structural traversal. However, the field still lacks a lightweight ontologization strategy that transforms FHIR data into OWL-compliant, terminology-integrated knowledge graphs optimized for practical semantic querying.

## 2. Methods

As the source data for this study, we utilized the MIMIC-IV Clinical Database Demo on FHIR, comprising approximately 900,000 FHIR resources in NDJSON format. This dataset served as the foundational corpus for our extract–transform–load (ETL) pipeline. Rather than modeling directly in OWL, we first constructed an intermediate FHIR-faithful RDF graph and then derived the analytic OWL graph from that representation.

### *FHIR-Faithful RDF Graph.*

As an initial transformation step, we converted the FHIR NDJSON resources into an RDF graph that preserves the native FHIR nesting structure, repeating elements, choice-type fields, and inter-resource references. This representation maintains the structural fidelity of the original FHIR specification and functions primarily as an intermediate waystation and baseline artifact for transformation validation and structural inspection, rather than as a standalone analytic model.

### *Ontologized OWL Graph.*

From the FHIR-faithful RDF graph, we then derived an OWL-compliant knowledge graph following the principles of ontology construction [10]. This transformation resolves deep nesting, variable-width arrays, and choice-type fields, enforces explicit distinctions between classes and individuals, and defines object properties and datatype properties. Selected FHIR resources were integrated with ICD-9-CM and ICD-10-CM terminologies using UMLS-derived mappings. For selected analyses, including evaluation of the NL2SPARQL interface and ontology inspection in Protégé [21], we derived task-specific projections of the ontologized OWL graph. These projections exclude high-volume resource types not required for a given task (e.g., Observation), and apply minor schema adjustments to improve tooling compatibility. They correspond to external schema views and do not constitute alternative ontologies or independent data models, but rather query-driven views of the same ontologized OWL graph [6].

### 2.1. Data Source and Conversion

We used the MIMIC-IV Clinical Database Demo on FHIR (v2.0) obtained from PhysioNet [22]. This dataset is a FHIR re-encoding of the MIMIC-IV demo database and contains 914,373 resources from 100 ICU patients admitted to Beth Israel Deaconess Medical Center in newline-delimited JSON (NDJSON) format, in which each line represents a single FHIR resource instance [23]. Each resource is uniquely identified by its logical ID.

Table 2 summarizes resource counts, including auxiliary FHIR resource classes introduced to improve analytic accessibility. In particular, the `DosageInstruction` backbone element was elevated to an auxiliary class to simplify querying and downstream analysis.

Most resource types include outgoing references to other resources (e.g., `Encounter` → `Patient`, `Observation` → `Encounter`), reflecting FHIR's graph-based relational structure. Certain resources (e.g., `Organization` and `Medication`) are primarily referenced by other resources but do not themselves contain outgoing references.

Most of the FHIR resources have special structural characteristics, including choice-type fields, variable-width arrays, and nesting structural levels (Table 3). Structural variability in the dataset was driven primarily by deep nesting, with some contribution from choice-type (`[x]`) elements and variable-width arrays. All resources include nested elements, with the maximum nesting depth ranging from 5 to 8 levels. All resource types contain variable-width arrays (repeating elements), with the maximum observed array length occurring in the `Observation` resource of 26. Choice-type elements (polymorphic fields) occurred in five resource types: `Observation`, `MedicationAdministration`, `MedicationRequest`, `Procedure`, and `Patient`.

**Table 2.** FHIR resources with at least one outgoing reference.

Class	Resources	At Least One Outgoing Reference
Organization	1	0
Location	39	39
Patient	100	100
Encounter	401	401
LocationEncounter (auxiliary)	972	972
Procedure	2,105	2,105
Condition	4,181	4,181
MedicationMix (auxiliary)	6,461	6,461
Specimen	11,817	11,817
MedicationDispense	13,455	13,455
MedicationRequest	16,366	16,366
Medication	20,075	0
DosageInstruction (auxiliary)	27,768	0
MedicationAdministration	49,635	49,635
Observation	762,633	762,633

To simplify query structure, auxiliary resources (*LocationEncounter*, *MedicationMix*, and *DosageInstruction*) were added. Only *Medication*, *Organization*, and *DosageInstruction* lacked at least one outgoing inter-resource reference.

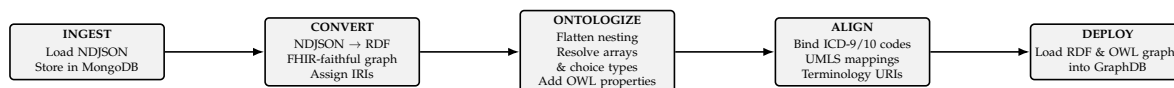
**Table 3.** Structural characteristics of FHIR resources in the FHIR MIMIC-IV Demo dataset

Type	Max. Array Len.	Max. Nesting Depth	Choice Fields
Observation	26	6	1 (4)
MedicationAdministration	1	6	1 (2)
MedicationRequest	1	8	1 (2)
MedicationDispense	1	8	0 (0)
Specimen	1	5	0 (0)
Condition	1	6	0 (0)
Procedure	1	6	1 (2)
Medication	4	6	0 (0)
Encounter	9	6	0 (0)
Patient	2	7	1 (3)
Location	1	5	0 (0)
Organization	1	6	0 (0)

**Choice Fields** reports the number of choice-type ( $[x]$ ) elements, with the number of allowed value types shown in parentheses.

## 2.2. Extract–Transform–Load Pipeline

The Extract–Transform–Load (ETL) pipeline converted FHIR resources into FHIR-faithful and OWL-compliant knowledge graphs (Figure 1).

**Figure 1.** Pipeline for transforming FHIR NDJSON resources into two graph representations.

### Step 1: Data ingestion.

Data ingestion began by preparing a dedicated environment for storing and managing the FHIR resources. MongoDB, an open-source document database, was configured as the initial repository because of its native support for JSON-like documents, efficient indexing, and suitability for hierarchical FHIR data. The downloaded FHIR dataset consisted of newline-delimited JSON (*NDJSON*) files partitioned by FHIR resource type. These partitioned files were merged into a single *NDJSON* file using standard shell commands (*cat*) prior to ingestion. The combined file was then imported into MongoDB, where each line was stored as an individual document.

Each MongoDB document represents one complete FHIR resource (e.g., Patient, Encounter, Condition), analogous to a row in a relational table while preserving the full nested structure of the original JSON object. This document-oriented storage model enables efficient iteration, indexing, and extraction of heterogeneous resources during subsequent transformation stages. Ingestion and analysis scripts are available in the project repository to support reproducibility.

*Step 2: Transformation from NDJSON to RDF.*

In the first transformation phase, FHIR NDJSON objects were converted into Resource Description Framework (RDF) triples to create an RDF representation of FHIR resources based on HL7 standards (<https://www.hl7.org/fhir/rdf.html>). Each MongoDB document corresponded to a single FHIR resource instance and contributed a set of RDF triples to the graph. The resourceType field determined the corresponding RDF class (e.g., Patient, Encounter, Condition), and the original FHIR logical identifier (id) was reused to construct stable IRIs.

PyMongo was used to retrieve the JSON documents, and a custom Python encoder mapped FHIR JSON elements to RDF predicates and serialized each resource into triples following the HL7 FHIR RDF mapping guidelines. A project-specific namespace prefix (se:) was introduced to construct compact, locally unique IRIs for resource instances; this prefix has no external semantic significance and was used solely for convenience in graph construction and SPARQL querying. The following namespace prefixes were used throughout RDF/Turtle serialization:

```
@prefix se: <http://mimic-fhir.local/ontology#> .
@prefix fhir: <http://hl7.org/fhir/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

The resulting RDF triples were loaded into GraphDB (Graphwise) to create the baseline repository, enabling SPARQL querying over the original nested FHIR structure. An example of an RDF-encoded FHIR resource is shown in Figure 2.

```
se:23a127ba-b031-5bc4-8cbc-27d56c8dfafd a fhir:Location ;
fhir:id [ fhir:v "23a127ba-b031-5bc4-8cbc-27d56c8dfafd" ] ;
fhir:status [ fhir:v "active" ] ;
fhir:name [ fhir:v "Medical/Surgical (Gynecology)" ] ;
fhir:type [
fhir:coding [
fhir:system [ fhir:v
"http://terminology.hl7.org/CodeSystem/location-physical-type"^^xsd:anyURI ] ;
fhir:code [ fhir:v "wa" ] ;
fhir:display [ fhir:v "Ward" ] ] ] ;
fhir:managingOrganization se:ee172322-118b-5716-abb-18e4c5437e15 .
```

**Figure 2.** FHIR RDF representation of a Location resource before ontologization. The location type display value ("Ward") is nested at the third structural level in this example. The resource also includes an outgoing reference to a managing Organization, illustrating inter-resource linkage in the FHIR graph.

*Step 3: Ontologization of the RDF Graph.*

The baseline FHIR-native RDF graph preserves the original nesting, repeating elements, and polymorphic ([x]) structures defined by the HL7 FHIR specification. While faithful to the exchange-oriented design of the standard, this representation requires deep path expressions and heterogeneous typing, which hinder analytic accessibility.

To improve structural clarity and support schema-guided SPARQL evaluation, we applied five systematic transformations to the baseline RDF graph: (1) flattening nested backbone elements, (2) resolving choice-type fields, (3) normalizing variable-length arrays, (4) transforming shifted dates, and

(5) introducing auxiliary structural classes. These operations simplify graph topology and enhance analytic accessibility while preserving the clinical assertions encoded in the source resources.

(1) *Flattening nested backbone elements.* In native FHIR, clinical meaning is often encoded within deeply nested hierarchical structures (Figure 2). In the RDF representation, these structures require multi-hop traversals through blank nodes. To improve accessibility, nested elements terminating in literal values were promoted to direct OWL datatype properties attached to the resource instance. This transformation collapses multi-hop paths into single-step assertions while preserving original codes and display values.

In particular, deeply nested triples encoding diagnostic codes (e.g., ICD-10 code and display values) were promoted to top-level datatype properties without nesting, as illustrated in Figure 3. In this transformation, hierarchical elements such as `Condition.code.coding.code` and `Condition.code.coding.display` are mapped to flattened properties (e.g., `flat:icd10_code`, `flat:icd10_display`), eliminating intermediate blank nodes and reducing query depth. Coding system URIs were not retained as separate properties; instead, semantic meaning was preserved through normalized code and display fields.

(a) Native nested FHIR Condition

```
{
  "resourceType": "Condition",
  "id": "c0298f4a-70f6-5357-9ae4-ec1fbc91640",
  "identifier": [{"value": "26739864-3-E78.5"}],
  "category": [
    { "coding": [{"code": "encounter-diagnosis"}] }
  ],
  "code": {
    "coding": [
      {
        "system": ".../diagnosis-icd10",
        "code": "E78.5",
        "display": "Hyperlipidemia"
      }
    ]
  },
  "subject": {
    "reference": "Patient/f77a5b72-65fd-5b20-8cef-6b6e4791265"
  },
  "encounter": {
    "reference": "Encounter/5754127d-83e5-5d67-b2a6-529ed781610e"
  }
}
```

(b) Flattened OWL Condition

```
se:c0298f4a-70f6-5357-9ae4-ec1fbc91640
  a flat:Condition ;
  flat:id "c0298f4a-70f6-5357-9ae4-ec1fbc91640" ;
  flat:identifier "26739864-3-E78.5" ;
  flat:category_code "encounter-diagnosis" ;
  flat:icd10_code "E78.5" ;
  flat:icd10_display "Hyperlipidemia" ;
  flat:encounter_reference se:5754127d-83e5-5d67-b2a6-529ed781610e ;
  flat:patient_reference se:f77a5b72-65fd-5b20-8cef-6b6e4791265 .
```

**Figure 3.** Side-by-side comparison of a native FHIR Condition resource (JSON excerpt) and its flattened OWL/RDF representation for a Condition resource with an ICD-10-CM code E78.5 (hyperlipidemia).

(2) *Resolving choice-type fields.* FHIR choice-type (`[x]`) elements permit multiple datatypes within a single structural position. These were transformed into explicitly typed OWL datatype properties, enforcing datatype semantics at the schema level and eliminating ambiguity during query generation. For example:

valueQuantity → flat:value\_quantity  
 valueString → flat:value\_string  
 valueCodeableConcept → flattened code/display properties

(3) *Normalizing variable-length arrays.* Repeating elements were normalized using two complementary strategies to stabilize graph structure for querying. Literal arrays were represented as repeated datatype property assertions. Structured arrays (e.g., `Encounter.location`) were externalized as instances of auxiliary classes linking the parent resource to the referenced resource. This preserves relational structure while simplifying query patterns.

For example, across 415 Encounters, the number of distinct locations per encounter varied between 1 and 9 (Table 4). To resolve this variability, each `Encounter.location` element was represented

as an instance of the auxiliary class `LocationEncounter`, producing a consistent relational pattern independent of array cardinality:

`Encounter → LocationEncounter → Location`

**Table 4.** Distribution of the number of `Encounter.location` elements per encounter in the demo dataset (N=415).

Locations per encounter	Count	%
1	153	36.9
2	106	25.5
3	65	15.7
4	45	10.8
5	23	5.5
6	14	3.4
7	4	1.0
9	5	1.2
<b>Total</b>	<b>415</b>	<b>100.0</b>

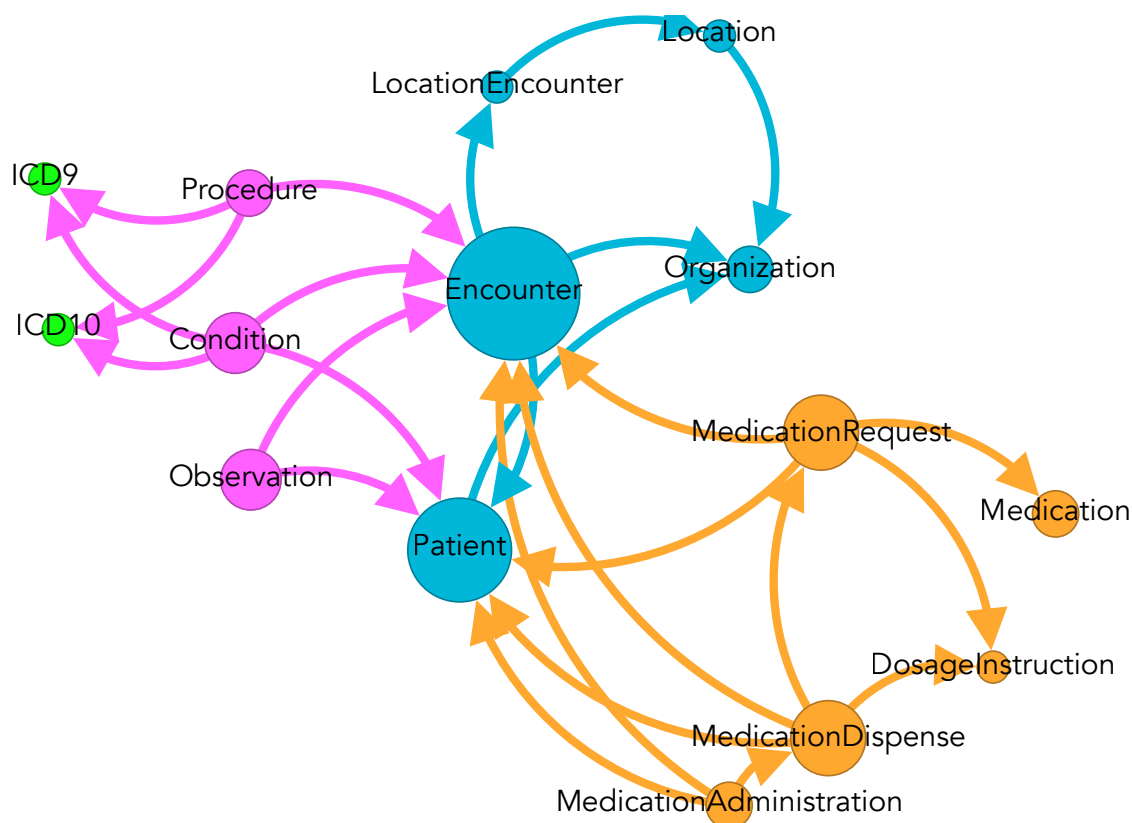
(4) *Transforming shifted dates.* The MIMIC-IV FHIR dataset applies systematic date shifting for de-identification, rendering absolute timestamps non-comparable across patients. However, temporal relationships within individual patients are preserved, and date arithmetic within a single patient remains valid. Birth dates were converted into derived age-at-encounter values, represented as datatype properties. Absolute calendar time was not retained in analytic queries.

(5) *Introducing auxiliary structural classes.* To simplify SPARQL queries and reduce structural complexity, auxiliary classes were introduced to externalize nested relational groupings. These included: `LocationEncounter`, `MedicationMix`, and `DosageInstruction`.

These derived classes preserve the connectivity of the clinical graph while enabling more direct and consistent query patterns, as illustrated in Figure 4. The structural and semantic impact of these transformations is summarized in Table 5, including an approximately 53% reduction in graph edges and an 80% reduction in graph nodes.

**Table 5.** Comparison of the FHIR-faithful RDF graph and the ontologized OWL graph. Ontologization reduces structural complexity, decreases node and edge counts, and introduces explicit OWL datatype and object properties to support schema-driven querying.

Category	FHIR-faithful RDF	Ontologized OWL
Structure	Nested	Flattened/ontologized
Nodes	20,782,214	4,252,079
Edges	33,133,045	15,582,681
Classes (declared)	12	15
OWL Datatype Properties	0	103
OWL Object Properties	0	16
FHIR-faithful representation	Yes	Transformed
OWL-compliant	No	Yes
RDF-compliant	Yes	Yes



**Figure 4.** OWL schema (TBox) of the ontologized FHIR graph. Core FHIR resource classes are extended with auxiliary classes to reduce SPARQL query complexity, and ICD-9-CM and ICD-10-CM are modeled as OWL classes to support terminology-grounded binding to Procedure and Condition. Edges represent outgoing OWL object properties derived from FHIR inter-resource references; arrows are unidirectional and denote domain-to-range relationships. Edge colors indicate the domain class of each property (blue = Patient, Encounter, Location; violet = Procedure, Observation, Condition; orange = Medication; lime = ICD9/ICD10). Node size is proportional to incoming references.

*Step 4: Terminology Integration and Alignment.* Diagnosis and procedure codes were extracted from the original FHIR system and code fields and represented as resolvable URIs under dedicated namespace prefixes corresponding to ICD-9-CM, ICD-10-CM, and ICD-10-PCS. The original FHIR system URIs were not retained explicitly; instead, their semantic role was preserved through assignment to the appropriate terminology namespace. No cross-terminology normalization was performed, and codes were preserved exactly as encoded in the source dataset. In the ontologized graph, ICD codes are modeled as identifiable resources rather than opaque string literals.

Each coded clinical assertion (e.g., a Condition or Procedure) is linked via an OWL object property to the corresponding ICD code instance (see Figure 4). This transformation promotes codes from literal values to first-class, queryable entities within the graph, enabling explicit linkage between clinical assertions and standardized terminologies. Terminology integration supports two primary capabilities:

1. *Code-label validation.* Linking codes to explicit ICD resources enables internal consistency checks between stored codes and associated human-readable labels.
2. *Terminology-grounded querying.* Queries may reference standardized code identifiers directly, rather than relying on string matching of literal fields. This improves query robustness and reproducibility across datasets that share the same coding systems.

Although the source FHIR resources contained limited references to SNOMED CT, terminology coverage for SNOMED CT in the MIMIC-IV demo dataset was minimal (only a single SNOMED-coded concept was observed in the Encounter resource). Because this limited representation did not materially influence analytic structure or downstream querying within this dataset, SNOMED

CT was not incorporated into the present knowledge graph. This decision reflects characteristics of the demo corpus rather than a broader judgment regarding SNOMED CT. In contemporary production EHR environments, SNOMED CT frequently serves as a primary clinical terminology for problem lists and structured findings and is therefore likely to play a central role in future FHIR-based ontology integrations. Following terminology integration, the OWL knowledge graph supports natural-language-driven SPARQL queries over ICD-aligned clinical concepts.

*Step 5: Deployment and Use of Graph Representations.* The transformation pipeline yields two complementary graph representations: a FHIR-faithful RDF graph that preserves the native nested resource structure, and an ontologized OWL graph that resolves nesting, arrays, and choice-type fields to support concise, schema-explicit querying.

Both graphs were deployed in GraphDB alongside aligned terminology graphs (ICD-9-CM, ICD-10-CM, and ICD-10-PCS). The FHIR-faithful RDF graph serves as a structural reference representation for fidelity verification, whereas the ontologized OWL graph serves as the primary analytic representation for SPARQL-based querying.

For selected analyses, task-specific projections of the ontologized OWL graph were generated. High-volume resource types—particularly `Observation` (over 800,000 instances)—substantially increase ABox size and degrade performance in ontology inspection tools such as Protégé, as well as during iterative NL2SPARQL evaluation. To address this, trimmed analytic views were created by excluding selected high-volume resource types when required for tool compatibility.

These projections do not alter the underlying schema or modeling conventions; rather, they represent analytic slices of the canonical ontologized graph. All reported SPARQL queries were executed against the graph representation appropriate to the task under evaluation. Unless otherwise specified, analytic queries and NL2SPARQL results are based on the ontologized OWL graph or its task-specific projections.

### 2.3. Schema-Grounded NL2SPARQL Query Interface

To enable natural-language interaction with the ontologized knowledge graph, we developed a schema-grounded NL2SPARQL interface that converts user questions into executable SPARQL queries aligned with the OWL data model. Here, “schema-grounded” denotes that query generation is constrained by the ontology’s classes, properties, and relationships. The system uses a two-stage workflow—ontology-guided query planning followed by schema-constrained SPARQL generation—implemented through the OpenAI GPT-4o API.

#### *Stage 1: Ontology-guided query planning.*

Given a natural-language question, GPT-4o identifies target ontology classes (e.g., `Patient`, `Encounter`, `Observation`) corresponding to the entities referenced in the query. Starting from these classes, the system performs a bounded breadth-first search (BFS) over the object-property graph to identify valid class paths (i.e., join chains). Conditioned on the selected classes and paths, GPT-4o selects relevant datatype and object properties and applies filters, aggregations, or temporal constraints implied by the question. The OWL schema is provided to the model as enumerated classes and properties, and the resulting query plan is validated against the schema prior to SPARQL generation.

#### *Stage 2: Constrained SPARQL generation.*

From the validated query plan, GPT-4o constructs a SPARQL template encoding the selected classes, paths, properties, and query constraints. This template, together with the natural-language question and explicit construction rules, is then used to generate the final SPARQL query.

Generation is restricted to ontology-defined classes and properties, and the model is required to preserve the join structure specified in the template. While large language models can often generate syntactically valid SPARQL directly, this intermediate templating step enforces adherence to the ontology schema and stabilizes query structure.

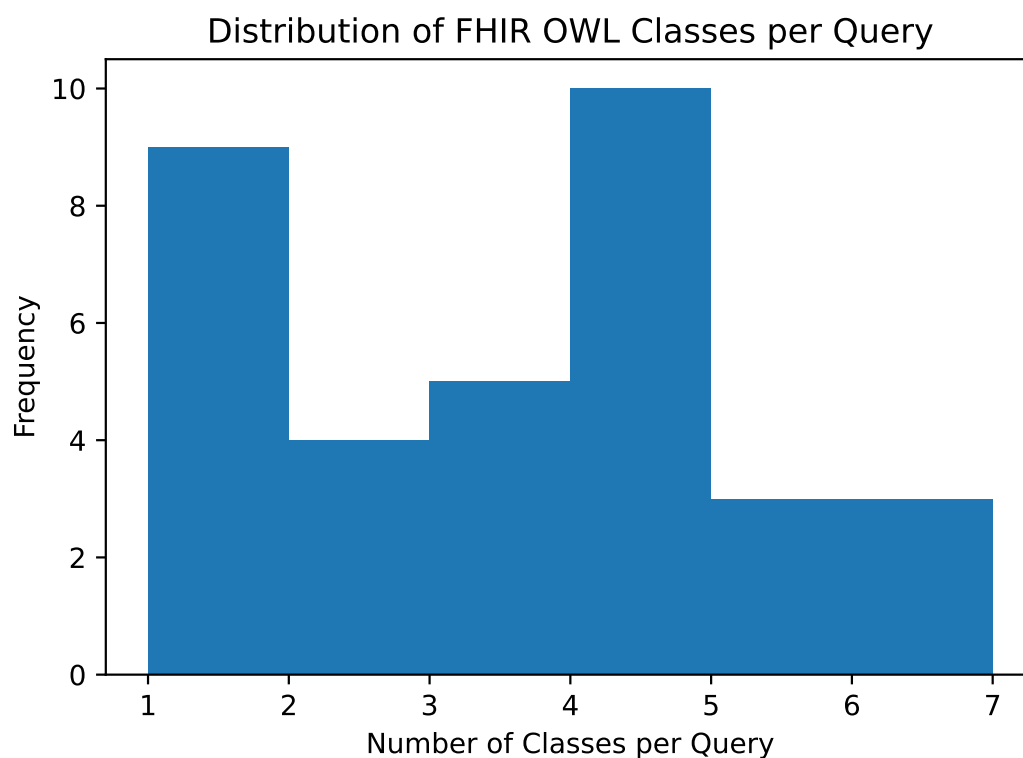
Separating ontology-aware planning from template-based realization reduces hallucinated predicates, invalid joins, and property mismatches, while enabling clearer separation of semantic (planning) and syntactic (generation) errors.

A curated set of 34 natural-language test queries was developed to evaluate the NL2SPARQL interface across common analytic tasks, including entity counts, attribute-based filtering, multi-entity joins, and terminology constraints (e.g., ICD-9/ICD-10 groupings). A representative multi-resource query was:

*Among patients with a diagnosis of essential hypertension (I10), what is the count of procedures whose category is Invasive Lines?*

To characterize structural complexity, each query was analyzed based on the number of distinct FHIR resource classes required to construct the corresponding SPARQL query. Across the 34 canonical queries, the number of required classes ranged from 1 to 7, with most queries involving 3–5 classes (Figure 5).

For each query, results generated through NL2SPARQL were compared with gold-standard outputs derived from domain-expert SPARQL queries. Outputs were scored as correct or incorrect based on exact result agreement. Each query was executed ten times at three temperatures (0.0, 0.01, 0.1), and mean accuracy and standard deviation were computed across runs. The full set of evaluation queries is available in the project's GitHub repository.



**Figure 5.** Distribution of the number of FHIR resource classes required per query across the 34 NL2SPARQL test queries. Queries vary in complexity, with most involving 3–5 classes and some requiring up to seven, corresponding to multi-step joins in the ontology graph.

#### 2.4. Validation of the Ontologized Graph

The ontologized knowledge graph was validated for structural correctness, semantic fidelity, and logical consistency at both the instance (ABox) and schema (TBox) levels.

For instance-level validation, SHACL shape constraints in GraphDB were used to assess RDF triple conformance to the flattened OWL schema. Shapes were defined for core clinical entities (Patient, Encounter, Condition, Observation, Organization, Location) and enforced property occurrence, datatype constraints, and referential integrity. Detected inconsistencies—such as untyped literals, mismatched date formats, or missing coded values—were corrected through explicit datatype alignment (`xsd:boolean`, `xsd:date`, `xsd:dateTime`) and restoration of missing reference codes. Following correction, all shapes passed validation.

Schema-level validation was performed in Protégé [21] using the ELK, Hermit, and Pellet reasoners. No unsatisfiable classes or logical inconsistencies were detected. Inferred class memberships and property constraints were consistent with the asserted model, confirming that the OWL-compliant graph supports coherent reasoning across multiple reasoning engines.

### 3. Results

The MIMIC-IV on FHIR demo dataset [23] comprised 100 de-identified patients, 12 FHIR resource types, and 914,373 FHIR resources. Our resource count was slightly lower than that reported by Bennett et al. (915,281); this discrepancy was not further investigated. Structural conformance to the FHIR specification was confirmed using the HAPI-FHIR validator.

The NDJSON export was converted to RDF following HL7 guidelines and then transformed into an OWL-compliant graph (Table 5). The OWL transformation reduced graph size and structural complexity by eliminating intermediate blank nodes while preserving clinically relevant entities and inter-resource relationships. Nested elements, variable-length arrays, and choice-type fields were mapped to explicit OWL classes, object properties, and datatype properties.

Ontologization simplified SPARQL authoring by replacing deeply nested FHIR-RDF paths (e.g., `Condition.code.coding.code`) with single-hop predicates in the OWL representation (e.g., `flat:icd10_code`). To illustrate this effect, we evaluated the query:

*How many unique patients have at least one diagnosis of essential hypertension (ICD-10 I10)?*

In the ontologized graph, this query can be expressed concisely as:

```
SELECT (COUNT(DISTINCT ?pat) AS ?n_patients) WHERE {
  ?cond a flat:Condition ;
        flat:icd10_code "I10" ;
        flat:patient_reference ?pat .}
```

In contrast, the FHIR-faithful RDF representation requires multiple predicate hops through intermediate nodes, for example:

```
SELECT (COUNT(DISTINCT ?patientRef) AS ?n_patients) WHERE {
  ?cond rdf:type fhir:Condition .
  ?cond fhir:code ?codeNode .
  ?codeNode fhir:coding ?codingNode .
  ?codingNode fhir:code ?codeValue .
  ?codeValue fhir:v "I10" .
  ?cond fhir:subject ?subjNode .
  ?subjNode fhir:reference ?refNode .
  ?refNode fhir:v ?patientRef .}
```

This expansion increases query length and structural complexity by requiring traversal of nested coding and reference structures.

The NL2SPARQL interface achieved high accuracy. Across 10 independent runs of the 34-query evaluation set at temperature 0, the NL2SPARQL interface achieved a mean accuracy of 95.0%, with minimal variability (range: 94.1%–97.1%). This consistency indicates that query correctness is primarily determined by schema grounding rather than stochastic variation.

### 4. Discussion

Our results show that lightweight ontologization can transform FHIR-derived clinical data from an exchange-oriented format into an OWL-compliant knowledge graph that preserves semantic fidelity while substantially improving analytic accessibility. This directly addresses the gap identified in our

review of prior approaches, positioning the strategy as a principled bridge between exchange-oriented interoperability standards and graph-based analytic infrastructures. In doing so, we intentionally relax strict round-trippability to the original FHIR exchange format in order to obtain simpler, schema-explicit querying for secondary analysis.

In our instantiation, we retain the 12 original FHIR resource types from the MIMIC-IV FHIR demo and introduce two terminology classes (ICD-9-CM, ICD-10-CM) plus three auxiliary structural classes (*MedicationMix*, *LocationEncounter*, *DosageInstruction*) to simplify query patterns without expanding the conceptual scope of the model. The transformation follows established ontology design principles [10] but does not aim to satisfy formal OBO Foundry criteria [7,8,24,25]. The result is best viewed as an ontologically structured knowledge graph rather than a fully axiomatized domain ontology: its TBox is deliberately lightweight—twelve resource classes without an explicit `rdfs:subClassOf` hierarchy—while its ABox is instance-dense ( $\approx 914,000$  individuals). This reflects a purpose-driven scope: we define only those classes and properties needed for consistent typing, explicit representation of entities and relations, and principled alignment with controlled biomedical vocabularies.

The central contrast is with `fhir.ttl`, the normative HL7 OWL ontology for representing FHIR resources in RDF while preserving round-trip fidelity to JSON and XML. Although both representations are OWL-compliant, they embody different modeling commitments. Because `fhir.ttl` must maintain exchange reversibility, it preserves native FHIR nesting—the main source of query complexity—including deep backbone hierarchies, variable-width repeating arrays, and polymorphic choice fields that require multi-hop blank-node traversal. Our strategy explicitly relaxes this round-trip constraint and restructures the graph to prioritize analytic tractability: blank-node reference structures are replaced with named OWL object properties, and frequently queried attributes are promoted to datatype properties, collapsing multi-hop traversals into single-step assertions. Structural conformance of the transformed graph was verified using SHACL constraints and OWL reasoning.

This approach is distinct from prior efforts that improve FHIR access through server-side APIs (e.g., Pathling, SQL on FHIR), client-side tabular transformations (e.g., FHIR-PYrate, `fhircrackr`), or direct RDF visualization (e.g., FHIR Lens). Rather than flattening resources into tables or preserving nested RDF structure, we restructure FHIR data into an OWL-compliant knowledge graph in which inter-resource references are explicit object properties, scalar attributes are typed datatype assertions, and diagnosis and procedure codes are first-class entities aligned to ICD-9-CM and ICD-10-CM. This supports terminology-grounded querying and principled alignment with external biomedical vocabularies—capabilities not readily afforded by tabular or purely exchange-oriented representations. Das and Hussey [26] have similarly argued that FHIR lacks a formal semantic model and proposed OWL2-based ontologization as a necessary remediation for continuity-of-care interoperability; our approach shares this motivation but pursues a lightweight strategy optimized for analytic queryability and schema-grounded natural-language querying over a single-institution dataset rather than comprehensive cross-institutional coverage.

Representing FHIR resources in OWL makes class membership and terminology linkages formally explicit and shifts emphasis from data extraction to semantic structuring. OWL's description logic layer introduces semantic constraints absent from RDF alone—distinguishing classes from individuals and object from datatype properties—thereby reducing ill-formed mappings and improving terminology alignment even when full reasoning is not invoked. The two representations produced by our pipeline serve complementary roles: the FHIR-faithful RDF graph preserves exchange fidelity and round-trip compatibility with NDJSON, whereas the ontologized OWL graph prioritizes analytic accessibility and semantic integration.

A major barrier to ontology-based querying is the dual expertise required in SPARQL and schema understanding. The schema-grounded NL2SPARQL interface addresses this barrier through a two-stage process that separates semantic query planning from syntactic realization, enabling a large language model to generate executable SPARQL queries constrained by the OWL schema rather than

unconstrained text generation. This architecture preserves formal precision while allowing domain experts to query the graph without mastering SPARQL syntax. Evaluation showed translation accuracy above 90%, with mean accuracy of approximately 95% across repeated runs.

An important practical consideration is the reproducibility and portability of the transformation pipeline. The ontologization procedure operates entirely on standard FHIR NDJSON resources and produces RDF/OWL graphs using widely supported semantic web technologies, including RDF, OWL, SHACL, and SPARQL. Because the transformation rules are resource-centric rather than dataset-specific, the approach can in principle be applied to other FHIR repositories beyond the MIMIC-IV demo dataset. This design supports reproducible generation of knowledge graphs from FHIR data and facilitates integration with existing semantic web tooling, graph databases, and downstream analytic workflows.

#### 4.1. Limitations and Future Work

Several limitations warrant consideration.

First, FHIR's resource-centric design produces highly instance-dense graphs. Even the 100-patient demo dataset generated more than 900,000 resources, over 800,000 of which were `Observation` instances. While manageable at this scale, the rapid growth in node counts produced by larger datasets may require more robust graph database infrastructure and optimized storage and indexing strategies.

Second, the ontologized graph combines a relatively small TBox (12 native classes and 3 auxiliary classes) with a large ABox (approximately 914,000 individuals). Description logic (DL) reasoners are generally optimized for class-level inference rather than large-scale instance reasoning; accordingly, analytic workflows relied primarily on SPARQL queries rather than exhaustive DL reasoning.

Third, terminology heterogeneity limits cross-resource harmonization. Although codes were modeled as first-class entities aligned to ICD-9-CM and ICD-10-CM, we did not perform cross-system normalization. Clinical concepts therefore remain distributed across ICD-9-CM, ICD-10-CM, ICD-10-PCS, NDC, and proprietary identifiers. Future work could introduce systematic harmonization, for example through crosswalk strategies that align legacy coding systems to ICD-10-CM or by mapping diagnoses and procedures to SNOMED CT, laboratory observations to LOINC, and medications to RxNorm.

In the current implementation, external terminologies are represented as flat code systems, with codes modeled as individuals rather than hierarchical classes. This modeling approach reflects implementation decisions that prioritized simplicity and scalability over formal ontology structure. Future extensions could represent these terminologies as axiomatized class hierarchies linked via `rdfs:subClassOf`, enabling hierarchy-aware querying and subsumption-based reasoning.

The NL2SPARQL evaluation used a modest test set under controlled API-based conditions. While this approach enables reproducible benchmarking, broader evaluation across larger query corpora and interactive conversational settings will be needed to assess robustness and real-world applicability.

Finally, the round-trip trade-off underlying this work is explicit and intentional. While the FHIR-faithful RDF graph remains fully round-trippable to NDJSON, the OWL-compliant representation introduces structural transformations that enhance queryability but are not strictly reversible to the original exchange format. This deliberate sacrifice of round-trip fidelity in favor of analytic accessibility represents the central design decision of the ontologization strategy presented here.

## 5. Conclusions

FHIR is optimized for transactional interoperability rather than exploratory analysis. Although HL7 provides a normative RDF serialization and associated OWL ontology that maintain round-trip fidelity with other exchange formats, this representation preserves native FHIR nesting and does not directly address structural features—deep backbone hierarchies, polymorphic elements, and repeating arrays—that complicate analytic querying. Because RDF and OWL must remain round-trippable with XML and JSON, architectural restructuring that would simplify analytic queries is intentionally constrained.

Relaxing this round-trippability requirement allows FHIR resources to be re-expressed, via lightweight ontologization, as an OWL-compliant knowledge graph that preserves semantic fidelity while improving queryability and schema clarity. Combined with a schema-grounded natural-language query interface, this approach enables reliable exploratory analysis of FHIR-derived clinical data without requiring SPARQL expertise and without sacrificing relational or terminological richness, providing a principled bridge between exchange-oriented interoperability standards and graph-based analytic infrastructures.

**Author Contributions:** Conception: SKP, DBH. Methodology: VH, DBH, SKP. Project administration and funding: SKP. Software: VH and DBH. Data curation: VH, AN. Data analysis: VH, DBH, SKP, AH, BY. Writing—original draft: BY. Writing—review and editing: BY, SKP, VH, AN, DBH.

**Funding:** This work was supported in part by the National Science Foundation under Award Number 2423235.

**Institutional Review Board Statement:** Institutional Review Board review was not required for this study.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The datasets and python programs used in this study can be found in the project GitHub repository at <https://github.com/vaughnh2022/nl2sparql>

**Acknowledgments:** This study used the MIMIC-IV Clinical Database Demo on FHIR dataset (version 2.1.0) obtained from PhysioNet [23,27,28].

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Mangalampalli, B.M.; Kolla, T. FHIR-Based Interoperability Frameworks For Real-Time Healthcare Data Exchange: Architecture Patterns And Performance Optimization. *International Journal Of Advances in Signal and Image Sciences* **2026**, pp. 1514–1536. <https://doi.org/10.29284/bmx0rg85>.
2. Gruendner, J.; Gulden, C.; Kampf, M.; Mate, S.; Prokosch, H.U.; Zierk, J. A framework for criteria-based selection and processing of fast healthcare interoperability resources (FHIR) data for statistical analysis: design and implementation study. *JMIR Medical Informatics* **2021**, *9*, e25645. <https://doi.org/10.2196/25645>.
3. Gruendner, J.; Deppenwiese, N.; Folz, M.; Köhler, T.; Kroll, B.; Prokosch, H.U.; Rosenau, L.; Rühle, M.; Scheidl, M.A.; Schüttler, C.; et al. The architecture of a feasibility query portal for distributed COVID-19 Fast Healthcare Interoperability Resources (FHIR) patient data repositories: design and implementation study. *JMIR Medical Informatics* **2022**, *10*, e36709. <https://doi.org/10.2196/36709>.
4. Vorisek, C.N.; Lehne, M.; Klopfenstein, S.A.I.; Mayer, P.J.; Bartschke, A.; Haese, T.; Thun, S. Fast healthcare interoperability resources (FHIR) for interoperability in health research: systematic review. *JMIR Medical Informatics* **2022**, *10*, e35724.
5. Löbe, M.; Draeger, C.; Strübing, A.; Palm, J.; Meineke, F.A.; Winter, A. Pitfalls in Analyzing FHIR Data from Different University Hospitals. In Proceedings of the GMDS, 2023, pp. 146–151. <https://doi.org/10.3233/shti230706>.
6. Smith, B. Beyond concepts: ontology as reality representation. In Proceedings of the Third International Conference on Formal Ontology in Information Systems (FOIS 2004). IOS Press, Amsterdam, 2004, pp. 73–84.
7. Smith, B.; Ashburner, M.; Rosse, C.; Bard, J.; Bug, W.; Ceusters, W.; Goldberg, L.J.; Eilbeck, K.; Ireland, A.; Mungall, C.J.; et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology* **2007**, *25*, 1251–1255. <https://doi.org/10.1038/nbt1346>.
8. Jackson, R.; Matentzoglou, N.; Overton, J.A.; Vita, R.; Balhoff, J.P.; Buttigieg, P.L.; Carbon, S.; Courtot, M.; Diehl, A.D.; Dooley, D.M.; et al. OBO Foundry in 2021: operationalizing open data principles to evaluate ontologies. *Database* **2021**, *2021*, baab069. <https://doi.org/10.1093/database/baab069>.
9. Prud'hommeaux, E.; Collins, J.; Booth, D.; Peterson, K.J.; Solbrig, H.R.; Jiang, G. Development of a FHIR RDF data transformation and validation framework and its evaluation. *Journal of Biomedical Informatics* **2021**, *117*, 103755. <https://doi.org/10.1016/j.jbi.2021.103755>.
10. Noy, N.F.; McGuinness, D.L.; et al. Ontology development 101: A guide to creating your first ontology. *Stanford University* **2001**.
11. Kendall, E.F.; McGuinness, D.L. *Ontology Engineering*; Morgan & Claypool Publishers, 2019.

12. Grimes, J.; Szul, P.; Metke-Jimenez, A.; Lawley, M.; Loi, K. Pathling: analytics on FHIR. *Journal of Biomedical Semantics* **2022**, *13*, 23. <https://doi.org/0.1186/s13326-022-00277-1>.
13. Lee, G.; Bach, E.; Yang, E.; Pollard, T.; Johnson, A.; Choi, E.; Jia, Y.; Lee, J.H. FHIR-AgentBench: Benchmarking LLM Agents for Realistic Interoperable EHR Question Answering. *Proceedings of Machine Learning Research* **2025**, *297*, 1–19. Machine Learning for Health (ML4H) 2025.
14. Idrissi-Yaghir, A.; Arzideh, K.; Schäfer, H.; Eryilmaz, B.; Bahn, M.; Wen, Y.; Borys, K.; Hartmann, E.; Schmidt, C.; Pelka, O.; et al. Using a Diverse Test Suite to Assess Large Language Models on Fast Health Care Interoperability Resources Knowledge: Comparative Analysis. *Journal of Medical Internet Research* **2025**, *27*, e73540. <https://doi.org/10.2196/73540>.
15. Hernandez-Camero, I.V.; Garcia-Lopez, E.; Garcia-Cabot, A.; Caro-Alvaro, S. Context-aware few-shot learning SPARQL query generation from natural language on an aviation knowledge graph. *Machine Learning and Knowledge Extraction* **2025**, *7*, 52. <https://doi.org/10.21203/rs.3.rs-4977169/v1>.
16. Rosenau, L.; Gruendner, J.; Behrend, P.; Triefenbach, L.; Kurscheidt, M.; Majeed, R.; Prokosch, H.U.; Ingenerf, J. From Feasibility to Insight: Piloting Feature Extraction from FHIR Cohorts to Advance Clinical Research. *Research Square* **2024**. <https://doi.org/10.21203/rs.3.rs-4977169/v1>.
17. Grimes, J.; Brush, R.; Rhyzhikov, N.; Szul, P.; Mandel, J.; Gottlieb, D.; Grieve, G.; Sadjad, B.; Sanyal, A. SQL on FHIR-Tabular views of FHIR data using FHIRPath. *npj Digital Medicine* **2025**, *8*, 342. <https://doi.org/10.1007/978-3-031-79486-5>.
18. Hosch, R.; Baldini, G.; Parmar, V.; Borys, K.; Koitka, S.; Engelke, M.; Arzideh, K.; Ulrich, M.; Nensa, F. FHIR-PYrate: a data science friendly Python package to query FHIR servers. *BMC Health Services Research* **2023**, *23*, 734. <https://doi.org/10.1186/s12913-023-09498-1>.
19. Palm, J.; Meineke, F.A.; Przybilla, J.; Peschel, T. “fhircrackr”: an R package unlocking fast Healthcare Interoperability resources for statistical analysis. *Applied Clinical Informatics* **2023**, *14*, 054–064. <https://doi.org/10.1055/s-0042-1760436>.
20. Tomaszuk, D.; Smajevic, A.; Sagi, T.; Hose, K. FHIR Lens: A Graph-Based Approach to Semantic EHR Exploration. *2025 IEEE 38th International Symposium on Computer-Based Medical Systems (CBMS)* **2025**, pp. 1–6. <https://doi.org/10.1109/cbms65348.2025.00133>.
21. Tudorache, T.; Noy, N.F.; Tu, S.; Musen, M.A. Supporting collaborative ontology development in Protégé. In *Proceedings of the International Semantic Web Conference*. Springer, 2008, pp. 17–32.
22. Moody, G.B.; Mark, R.G.; Goldberger, A.L. PhysioNet: Physiologic signals, time series and related open source software for basic, clinical, and applied research. In *Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 8327–8330. <https://doi.org/10.1109/iembs.2011.6092053>.
23. Bennett, A.M.; Ulrich, H.; Van Damme, P.; Wiedekopf, J.; Johnson, A.E. MIMIC-IV on FHIR: converting a decade of in-patient data into an exchangeable, interoperable format. *Journal of the American Medical Informatics Association* **2023**, *30*, 718–725. <https://doi.org/10.1093/jamia/ocad002>.
24. Smith, B.; Brochhausen, M. Putting biomedical ontologies to work. *Methods of Information in Medicine* **2010**, *49*, 135–140.
25. Overton, J.A.; Jackson, R.C.; Matentzoglou, N.; Duncan, W.D.; Vita, R.; Harris, N.L.; Mungall, C.J.; Peters, B. COB: A Core Ontology for Biology and Biomedicine. In *Proceedings of the ICBO*, 2022, p. 1.
26. Das, S.; Hussey, P. HL7-FHIR-based ContSys formal ontology for enabling continuity of care data interoperability. *Journal of Personalized Medicine* **2023**, *13*, 1024. <https://doi.org/10.3390/jpm13071024>.
27. Goldberger, A.L.; Amaral, L.A.; Glass, L.; Hausdorff, J.M.; Ivanov, P.C.; Mark, R.G.; Mietus, J.E.; Moody, G.B.; Peng, C.K.; Stanley, H.E. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation* **2000**, *101*, e215–e220.
28. Bennett, A.; Ulrich, H.; Wiedekopf, J.; Szul, P.; Grimes, J.; Johnson, A. MIMIC-IV Clinical Database Demo on FHIR. *PhysioNet* **2025**. Version 2.1.0.

**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.