

Article

Semantic Fusion of Health Data: Implementing a Federated Virtualized Knowledge Graph Framework Leveraging Ontop System

Abid Ali Fareedi ¹ , Stephane Gagnon ² , Ahmad Ghazawneh ^{1,3} and Raul Valverde ^{4,*} 

¹ The Academy for Information Technology (ITE), Halmstad University, 30118 Halmstad, Sweden; abid.ali@hh.se (A.A.F.); ahmad.ghazanweh@hh.se (A.G.)

² Département des Sciences Administratives, Université du Québec en Outaouais, Gatineau, QC J7X 3X7, Canada; stephane.gagnon@uqo.ca

³ School of Economics, Innovation and Technology, Kristiania University of Applied Sciences, 0153 Oslo, Norway

⁴ John Molson School of Business, Concordia University, Montreal, QC H3H 0A1, Canada

* Correspondence: raul.valverde@concordia.ca

Abstract: Data integration (DI) and semantic interoperability (SI) are critical in healthcare, enabling seamless, patient-centric data sharing across systems to meet the demand for instant, unambiguous access to health information. Federated information systems (FIS) highlight auspicious issues for seamless DI and SI stemming from diverse data sources or models. We present a hybrid ontology-based design science research engineering (ODSRE) methodology that combines design science activities with ontology engineering principles to address the above-mentioned issues. The ODSRE constructs a systematic mechanism leveraging the Ontop virtual paradigm to establish a state-of-the-art federated virtual knowledge graph framework (FVKG) embedded virtualized knowledge graph approach to mitigate the aforementioned challenges effectively. The proposed FVKG helps construct a virtualized data federation leveraging the Ontop semantic query engine that effectively resolves data bottlenecks. Using a virtualized technique, the FVKG helps to reduce data migration, ensures low latency and dynamic freshness, and facilitates real-time access while upholding integrity and coherence throughout the federation system. As a result, we suggest a customized framework for constructing ontological monolithic semantic artifacts, especially in FIS. The proposed FVKG incorporates ontology-based data access (OBDA) to build a monolithic virtualized repository that integrates various ontological-driven artifacts and ensures semantic alignments using schema mapping techniques.

Keywords: virtualized knowledge graph; ontop; federated ontology; data integration; data interoperability



Academic Editor: Paolo Bellavista

Received: 18 March 2025

Revised: 9 May 2025

Accepted: 16 May 2025

Published: 30 May 2025

Citation: Fareedi, A.A.; Gagnon, S.; Ghazawneh, A.; Valverde, R. Semantic Fusion of Health Data: Implementing a Federated Virtualized Knowledge Graph Framework Leveraging Ontop System. *Future Internet* **2025**, *17*, 245. <https://doi.org/10.3390/fi17060245>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent times, the explosion of digital data [1], tight coupling with the convenience of digitization, diverse data representations and descriptions, and varying personal preferences have led large enterprises to store vast amounts of data in multiple formats. These formats range from structured relational databases to unstructured flat files, reflecting the increasing complexity of data management and highlighting the growing complexity of data storage and integration [2,3]. The influx of global data volume is predicted to touch approximately 163 zettabytes by 2025 [4]. Large enterprises in different sectors, such as healthcare, will generate almost half of those data. According to healthcare sector

observations, most *healthcare information systems* (HISs) are patient-centric, granting patients certain privileges to access electronic health record systems, leading to the collection of vast amounts of clinical data online. However, developing integrated HIS is challenging because of *Data Integration and Semantic Interoperability* (DI&SI) issues [4,5]. DI&SI challenges create a severe bottleneck for seamless data exchange and access to other applications within or across healthcare organizations [4]. In this context, the lack of DI&SI arises from variations in data structures, limited standardization, and diverse formats, hindering data access and resulting in inefficiencies, increased costs, poor quality, scalability and performance bottlenecks, and semantic alignment challenges, as well as a lack of comprehensive virtualization frameworks constructing monolithic ontological artifacts that integrate diverse healthcare data models cohesively [4–6].

In a distributed data environment, the interconnected legacy health systems demand seamless integration across both internal and external distributed rational data sources and disparate data models that are distinct in nature, purpose, and contextual relevance [4]. Traditional approaches [7] involve transforming data for storage and distribution within a unified repository through physical consolidation, such as data warehousing (DW) using data integration tools [8]. Alternatively, disparate data can be logically merged through an integration layer [4,9]. Vendors have expanded their integration tools to enable service-based architecture, data fabrics, and middleware like GraphQL [10], enabled by graph-based solutions.

Gartner [8] identified data fabric and *Knowledge Graphs* (KGs) as key technological trends in data management frameworks. These frameworks enable a standardized query approach, establish relationships across diverse data sources and models, and introduce a shared vocabulary for data interchange and integration, such as the *Resource Description Framework* (RDF) [11].

In desktop research, data practitioners advocate for semantic technologies, particularly ontologies [12] and KGs [13], as solutions to data bottlenecks [4]. Ontologies concisely represent a structured representation of domain knowledge through a shared conceptualization with formal and explicit definitions [14]. Observations show that heterogeneous data sources can be effectively described and harmonized using ontologies, following the *Ontology-based Data Integration* (OBDI) architecture across various domains [4,15]. Therefore, Konstantinou et al. [16] published a study on implementing *Federated Ontology* (FO) to integrate disparate data models and map diverse relational data sources. Likewise, semantic-driven ontology generation aims to map a domain-centric ontology with existing databases. Schwade and Schubert [7] proposed a semi-automated database-ontology mapping approach extended from Konstantinou et al. [4,16]. A limitation of OBDI is the need to create a local ontology for each new data source to connect it to the global ontology, a process that is often complex, time-consuming, and costly [4,5]. OBDI is associated with *Ontology-based Data Access* (OBDA) [15].

The primary advantage of OBDA [17] is that it provides end users with consistent access to and a platform for querying multiple data sources, regardless of where the data are stored. It also enables querying through the *Federated Ontology* (FO) conceptual layer. The FO acts as a mediator, distributing the query to the data sources and retrieving results between entities in the data sources and the ontologies [4,7,18].

The proposed *Federated Virtual Knowledge Graph* (FVKG) framework is recognized as a promising solution to improve DI&SI bottlenecks in response to these challenges. The FVKG presents a cohesive image of disparate relational data sources and related data models through semantic technologies. In this research, we used a *Virtual Knowledge Graph* (VKG) as an instrumental tool to enable Ontop (virtual system) (<https://ontop-vkg.org/> (accessed on 21 May 2025)), providing a logical abstraction layer over different data sources

(primarily relational) and allowing for seamless querying and analysis without physically consolidating data in the distributed environment of healthcare. This research targets two questions: (i) **RQ1:** *How can ontology engineering paradigms be systematically integrated within design science research to develop data-driven artifacts?* (ii) **RQ2:** *How can data integration and semantic interoperability bottlenecks be addressed using a virtualized knowledge graph (VKG) approach leveraging the Ontop platform in federated health information systems (HIS)?*

The significant contributions of this research are to establish a connection between the potential of *Design Science Research* (DSR) and *Ontology Engineering* (OE) paradigms to address real-world problems and advance the development of modern HIS. This research integrates the principles of DSR forces with semantic web technologies, such as ontologies and KGs, to foster knowledge creation, engineering, and evolution using a hybrid methodology. We customized *Ontology-based Design Science Research Engineering* (ODSRE) and extended work from [4,19].

This research also investigates how the FVKG framework, utilizing the VKG approach, can serve as an effective tool for virtual platforms like Ontop, which powers semantic query engines to address and improve DI&SI challenges in building federated HIS. The proposed FVKG framework facilitates a unified platform as a *Global-Virtual-View* (GVV) for heterogeneous relational data sources [20], disparate models, and ontologies, overcoming data-driven bottlenecks in a distributed environment. FVKG provides a platform that minimizes data integration complexities, ensures low latency and data freshness, facilitates real-time access, and enhances scalability and performance while ensuring integrity and coherence across the federation system through a virtual approach in a distributed environment.

The FVKG also integrates OBDA to build a monolithic virtual ontological model (e.g., cardiovascular disease (CVD) model), combining various ontology-driven artifacts and ensuring semantic alignment through schema mapping techniques, especially in the context of cardiovascular disease use cases.

This research is structured into the subsequent sections: Section 2 briefly highlights the desktop research related to the Ontology-based Design Science Research (ODSR) framework in HIS, the VKG framework or ODBA, the VKG mapping and tooling ecosystem, and the data federation approach for HIS. Section 3 presents the customized methodology steps, data collection, and data analysis explaining an FVKG's anatomy and its architectural layers. Section 4 displays the experimental results. Section 5 showcases testing and evaluation procedures using DL and SPARQL query languages. Lastly, Section 6 offers a conclusion and future possibilities and directions.

2. Theoretical Background

2.1. Ontology-Based Design Science Research (ODSR) Framework in Health Information Systems

The urge for knowledge consolidation, data integration, and semantic interoperability is a crucial and essential requirement for developing modern, data-driven applications across various domains. As a result, the evolution of *Design Science Research* (DSR) is urgently required to build contemporary *Information Systems* (ISs) due to its value in integrating DSR with research in *ontology engineering* (OE) within the health IS discipline [4,19]. In scholarly literature, practitioners have established fundamental key steps for applying *Ontology Engineering* (OE) within the DSR discipline [21–23]. Reiterer et al. [23] propose the “*ontology model of DSR aspects of DSR document core ontology (DSRDCO)*” as a tool to support the model of the search and automatic summarization of DSR publications [4]. Additionally, the *Ontology-based Design Science Research* (ODSR) framework improves the understanding, implementation, and assessment of data-driven artifacts across various domains, with a particular emphasis on healthcare [4]. The ODSR framework is developed by integrating existing frameworks for DSR [24] and ontology development within the

IS discipline [25,26]. The ODSR framework illustrates an iterative process that combines DSR activities and OE to develop new kinds of modern HISs (e.g., *conversational agents*), addressing challenges related to data integration, access, semantic interoperability, and data exchange across various industries, such as healthcare [4,19]. In the healthcare sector, semantic technologies such as ontologies [27,28] and VKG [4,29] play a crucial role in tackling issues of data and semantic heterogeneity [28], facilitating knowledge integration and interoperability among heterogeneous HISs [30–32] and improving knowledge sharing issues within the healthcare domain [33].

In order to address the challenges in the healthcare domain, ontologies offer a shared understanding of domain knowledge that can be communicated across individuals and between heterogeneous, distributed systems [4] within healthcare organizations [4,34]. Ontologies provide a standardized representation of knowledge, ensuring precise and consistent data recording in a knowledge base. In healthcare, the existing medical ontologies such as the *Unified Medical Language System* (UMLS), *Guideline Interchange Format* (GLIF), *Generalized Architecture for Languages* (GALEN), *International Classification of Diseases* (ICD), and *Systematized Nomenclature of Medicine Clinical Terms* (SNOMED-CT) [4,35] serve as valuable tools to enhance access to the electronic patient record by offering a standardized common vocabulary for data access across various channels or sources [34].

2.2. The Virtual Knowledge Graph Framework

In scholarly literature, the *Virtual Knowledge Graph* (VKG) framework is also called *Ontology-based Data Access* (OBDA) [36], and its paradigms refer to an integrated data approach and data access to diverse data sources, either internal or external [4]. As a result of VKG, the rigid structure of tables in relational databases and data model structures is replaced by the flexibility of graphs that are kept virtual and are effective in encapsulating domain and contextual knowledge. The data-driven landscape requires integrating data and providing seamless access to end users that is time-consuming, cost-effective, and rigorous [29]. Various data integration tools are available [8], enabling users to integrate data from other sources using the standard relational model, which is inflexible [4]. Moreover, only a few vendors integrate data without moving or transforming data sources [29], such as Denodo [37], Dremio [38], and Teiid [39].

As a result of the VKG, three essential parameters are incorporated: First, *Data Virtualization* (DV) is a technique that assists end users by providing them with global schemas of relevant domain-oriented data, contextual knowledge, and dispersed data sources rather than individual physical data sources like *Local-As-View* (LAV) [4]. Although the federated global perspective [38,40] may still need to be fully achieved, DV enables query data without incurring storage, data movement, and retrieval time cost factors [4,29].

Secondly, the data are subsequently represented as a graph (G), where domain objects and data values serve as nodes and object characteristics as edges [4,41]. Thirdly, *Domain Knowledge* (DK) and *Contextual Knowledge* (CK) enhance the network, supporting the data by organizing concepts [4], property hierarchies, inter-domain properties, and essential attributes [40,42]. The VKG technique [4], or OBDA [43], has been extensively studied in the formal context of ontologies. Virtualization in OBDA is achieved by defining mappings between disparate data sources, models, and the domain ontology (*federated ontology*) [36].

Additionally, mapping assertions include information on how to construct identifiers for objects within the ontology, such as those that appear in replies to user queries based on the values obtained from the sources, such as *Uniform Resource Identifiers* (URIs) or *International Resource Identifiers* (IRIs) [29].

2.3. Virtual Knowledge Graph Mapping and Tooling Ecosystem

By supporting the entire life cycle of the VKG approach, various data integration systems or tools have been developed, implemented, and used by the Garner report [8]. We classified these integrated systems to support virtualization into four groups: (i) integrated systems that utilize query reformulation to answer SPARQL [44] queries over VKGs; (ii) mapping engineering tools such as PROMPT [45] for assisting mapping design; (iii) federations for accessing federated queries across several data sources; iv) query formulations tools to facilitate interaction with VKGs [4,29]. In recent times, numerous VKGs query answering systems have been developed in academia and industry [4], such as D2RQ [46], Mastro [18], Morph [47], Ontop [48], Oracle Spatial and Graph [49], Stardog [50], and Ultrawrap [51]. Utilizing these query and response systems facilitates data virtualization by industry standards and query optimization, both critical for delivering high speeds [29].

According to the practitioner's observations [4], ontologies and mapping are key classifications for developing complex artifacts, which are fundamental components of VKG systems, leading to developing domain-centric AI-powered applications (e.g., conversational agents, semantic search engines, recommended systems) in various fields. Based on these classifications and detailed analyses of how mapping techniques are represented, several mapping editors are employed, categorized as text and graphical editors [4]. Text editors handle the textual representation of mapping, typically based on R2RML [52] or an alternative system [29]. These text editors are often integrated into ontology editors like Protege or used independently, such as in the IDE Stardog Studio. In graphical editors, users define mappings by linking database schema columns with characteristics, properties, and classes from the ontology vocabulary. Notable graphical editors in this category include Map-On [53], MapVOWL [54], RMLEditor [55], and SQuaRE [29,56].

3. Methodology

3.1. Ontology-Based Design Science Research Engineering (ODSRE) Methodology

This section emphasizes the importance of integrating diverse data and knowledge and its evaluation within the *Design Science* (DS) [57] discipline, particularly highlighting the value of combining *Design Science Research Methodology* (DSRM) [24] and the *Ontology Engineering* (OE) [58] paradigms [9]. The DS methodology helps to address practical real-time cases of legacy data source integration and resolve semantic interoperability challenges for frequent access in standardized form for developing domain-centric service, especially in the healthcare industry [4]. The primary goal of merging DSR and OE is to assess new digital artifacts inspired by theoretical knowledge design [59]. We customized DSR and OE *hybrid methodology* (ODSRE) artifacts to construct an FVKG framework using the "Ontop" platform, focusing on the design and creation of *Enterprise Knowledge Graphs* (EKGs) specific to the healthcare industry. This approach supports four key themes: (i) understanding the contextual knowledge, (ii) use of legacy systems such as *Electronic Health Records* (EHRs), (iii) information sharing (*explicit knowledge*), (iv) knowledge dissemination either in tacit or implicit knowledge and access through data acquisition techniques such as modeling workshops, following steps [60] with domain experts and a focal group. Figure 1 demonstrates an ontology engineering cycle that encompasses DSR activities.

First, the *Knowledge Acquisition* (KA) phase begins by focusing on two key parameters: detailed understanding of the domain knowledge and contextual knowledge. The KA process is initiated through collaborative modeling workshop sessions with domain experts guided by specific activities [60].

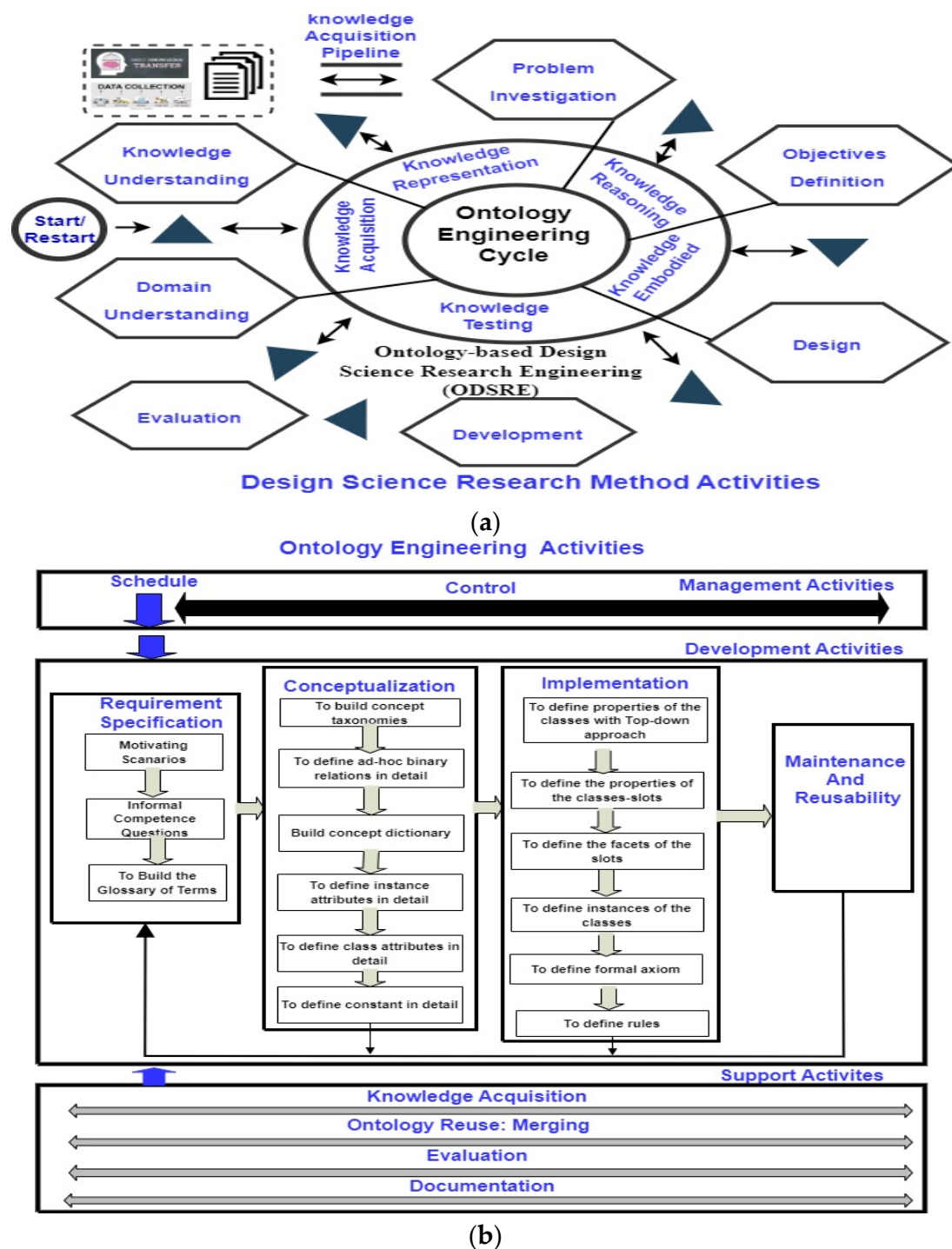


Figure 1. Hybrid methodology ontology-based design science research engineering (ODSRE). (a) The ontology engineering life cycle embedded with design science research methodology paradigms; (b) the number of activities that help to develop a customized ontology mapped to the requirements of domain contextual knowledge.

Second, in the problem investigation phase, we examine in-depth information related to data integration, access and semantic interoperability challenges. We also identify relevant theories, methods, and existing solutions to mitigate the aforementioned data-driven challenges using semantic technologies and their representation in standardized form for data exchange.

Third, in the objective definition phase, we concentrated on gathering specific requirements and defining key tasks to develop reasoning mechanisms to address particular challenges, particularly within the healthcare domain.

Fourth, the design phase focused on creating digital artifacts and outlining how the integration is to be achieved and managed in the standardized form to resolve data interoperability issues using semantic techniques such as ontologies, knowledge graphs, and virtual platforms such as Ontop.

Fifth, the development phase is dedicated to the creation of digital artifacts. Sixth, in the evaluation phase, these digital artifacts are assessed based on predefined criteria [59] grounded in design science research method activities [61]. Similarly, *Ontology Engineering* (OE) is defined as “a set of activities that link with the ontology development process, its life cycle, and the methodologies, tools and languages for building ontologies” [4,62]. In Figure 1, the OE activities are carried out using ontology development methodologies such as Methontology [63] and Tove [64,65].

3.2. Activities 1 and 2: Problem Investigation and Motivation for Heterogeneous Data Collection—Construct Artifact

The healthcare context of Cardiovascular Diseases (CVDs) plays a crucial role in this research. It aims to develop customized domain-centric ontology-driven digital artifacts (e.g., cardiovascular disease ontology) and prototypes demonstrating advanced methods for tackling data integration, access, and interoperability challenges in healthcare settings. In the data collection phase, we focus on incorporating domain-specific data related to cardiovascular diseases and embedding production rules within the ontological metadata model (OWL (<https://www.w3.org/OWL/> (accessed on 21 May 2025)) files) to enable reasoning capabilities that support healthcare professionals (HPs) and health users. A significant challenge in this phase is gathering relevant CVD data and other diverse data models (e.g., laboratory test data) from various sources and formats in a decentralized environment.

Undoubtedly, addressing data integration and interoperability issues is a trickier task, so we followed a series of steps. First, we conducted desktop research to identify the research gap and incorporated data models from data sources such as Kaggle [4,66]. Second, we created a small-scale database in a relational database such as MySQL Workbench 8.0, populating it with dummy data to observe the behavior of the data [4]. During this process, we identified existing approaches for tackling data integration, access, and interoperability challenges, particularly in federated virtual knowledge graph (FVKG) systems leveraging the Ontop platform. The Ontop platform facilitates the construction of virtual systems over distributed relational data sources and models with varying formats [67]. We also engaged in informal discussions with domain experts and industry professionals with significant expertise and industry experience in building data-driven applications and pipelines.

3.3. Activity 3: Defining the Objectives, Definition, and Relevance Data Analysis

In this section, the data analysis phase is guided by the design theory proposed by Muller and Thoring [4,59]. It follows a five-phase procedure: (i) knowledge stimulus, (ii) modeling of tacit knowledge, (iii) design of architectural artifact of federated virtual knowledge graph (FVKG), (iv) conceptual modeling to draw ontological metadata model artifacts (v), evaluation and testing using description logic (DL (<https://protegewiki.stanford.edu/wiki/DLQueryTab> (accessed on 21 May 2025)), and SPARQL (<https://www.w3.org/TR/sparql11-query/> (accessed on 21 May 2025)) queries based on competence questions (see Table 1).

Table 1. Data analysis.

Phases	Tasks	Outcomes
Knowledge stimulus (construct-level artifact)	<ul style="list-style-type: none"> Define objectives, definition, and relevance for a proposed solution and articulate research questions using desktop research. Develop a small-scale database in relational databases such as MySQL and organize others various data models and formats in different data structures such as Excel spreadsheet and CSV files. Construct <i>Competence Questions</i> (CQs) and analyze ontological frameworks and semantic web technologies to mitigate data-driven issues. Analyze mapping techniques for data integration and interoperability issues. Analyze mapping techniques and various virtual systems such as Ontop. 	<ul style="list-style-type: none"> Research data repositories. Identify numerous critical concepts related to data integration, access, and interoperability issues. Mention different design modeling tools (e.g., MS Visio, CMap (https://cmap.ihmc.us/cmaptools/cmaptools-download/) (accessed on 21 May 2025)) and ontology-related tools (e.g., Protege, Top-braid Composer (https://allegraph.com/topbraid-composer/) (accessed on 21 May 2025)), etc.). Mentioned ontological frameworks (e.g., FVKGs).
Tacit knowledge modeling (neuronal-level artifact)	<ul style="list-style-type: none"> As a knowledge engineer, draw conceptual models (ontological metadata model) in suitable tools and share them with knowledge mentors. Develop data models in diverse structure and formats using tools such as Excel, CSV, and data sources using various relational databases such as MySQL and more. 	<ul style="list-style-type: none"> Process view of the domain model based on CVO diseases context.
Architectural artifact of federated virtual knowledge graphs (FKVGs)	<ul style="list-style-type: none"> As a knowledge engineer, design layered architectural artifact based on the data acquisition, layer, federation ontology layer, mapping layer, virtual graph layer, and user application layer. 	<ul style="list-style-type: none"> Architectural design artifact.
Conceptual modeling Symbolic-level artifact (explicit knowledge)	<ul style="list-style-type: none"> As a knowledge engineer, transfer tacit knowledge (e.g., cardiovascular disease business model) into the conceptual metadata model (machine-readable format) using owl languages. Construct ontological models using ontology development methodologies. Construction of Federated Ontology (FO) metadata models using FVKG approach. Construction of federated VKGs. 	<ul style="list-style-type: none"> Conceptual meta models. Define transformation rules for developing a federated view in federated ontology (FO). Define business rules and write in SWRL (https://www.w3.org/submissions/SWRL/) (accessed on 21 May 2025)) plug-in.
Deployment, evaluation, and testing	<ul style="list-style-type: none"> Using mapping manager to map data from relational data sources to VKGs. Evaluation and testing using DL and SPARQL query languages using Ontop mapping platform and develop Ontop endpoint. 	<ul style="list-style-type: none"> Federated ontology (FO) developed in OWL. Data interoperability resolution in standardized data. Execution results to justify CQs. Execution of business rules in VKGs.

3.4. Activity 4: Construction of Federated Virtual Knowledge Graph (FVKG) Framework—Architectural Artifact

The section outlines the structure of the proposed architectural artifact, known as the Federated Virtual Knowledge Graph (FVKG) (<https://github.com/abid-fareedi/Semantic-Fusion-of-Health-Data-Implementing-a-Federated-Virtualized-Knowledge-Graph-Framework-Ontop>), also referred to as Ontology-based Data Access (OBDA) [4]. The proposed framework is categorized into five main categories: (i) data acquisition layer from different heterogeneous data sources, (ii) data federation layer, (iii) data mapping layer, (iv) federated virtual knowledge graph using Ontop virtual platform, (v) user application layer.

First, the Data Acquisition (DA) layer integrates multiple distributed internal relational data sources and external data models with varying structures and formats. The DA facilitates data transformation into their respective models, adhering to different data standards [4]. In this phase, we create a small-scale relational database deliberately targeting patient Electronic Health Records (EHRs) within the context of cardiovascular diseases (CVDs). Additionally, this layer enables the incorporation of external data models stored in various data structures such as Excel, CSV, XML, etc. It supports data population, consolidation, and distribution into a centralized data repository, ensuring the authenticity and credibility of the data. We design a small-scale database with patient-related tables for the CVD domain, such as *role*, *patient*, *medical_history*, *appointment*, *cvo_record*, *payment*, *risk_factor*, *room*, *department*, and *treatment*. Furthermore, external models (e.g., *diagnostic_Test_Results*) are retrieved from data repository platforms like Kaggle [66].

Second, the Ontology Federation (FO) layer comprises two key components: (i) diverse, distributed data models (external and internal relational databases) with varying

data structures and formats and domain-oriented ontological models (e.g., seed ontology) with domain contextual knowledge of various domains, especially healthcare. At this point, we wrote a programmable script for developing the Ontology Population and Enrichment (OPE) method to populate external data automatically or follow semi-automatically strategies to extract instances and relationships from unstructured or semi-structured data sources. The OPE method then adds them into an FO. One of the primary goals of the OPE method is to populate and enrich the ontology with real-world external big data that reflect the specific domain, especially healthcare diagnostic data. The OPE method helps to improve semantic understanding and knowledge management, enabling interoperability and improving decision-making procedures.

The FO layer also facilitates the development of an ontological metadata model tailored to specific domains, such as CVO, incorporating contextual knowledge from domain experts. Integrating the OPE method with ontological model development allows Ontology Engineering (OE) to automatically or semi-automatically create a standardized vocabulary and robust framework for seamless data integration across multiple systems and applications by enabling the FO layer to function automatically or semi-automatically.

Third, the Data Mapping (DM) layer operates across three sub-layers, with the mapping manager systematically handling three key tasks. First, it executes the SQL query to retrieve data based on the user's requested query. Second, it maps the retrieved data into RDF triples. It generates an Intermediate Query (IQ), which processes the SQL results, translates them, and structures them into the target mapping. It ensures that the same results are provided in response to SPARQL query while maintaining data movement from native sources with minimal latency and improved performance. This layer also supports the "Ontop SPARQL endpoint", allowing end users to write SPARQL queries and retrieve results stored in large triple stores like RDF-triplestore (<https://www.oxfordsemantic.tech/> (accessed on 21 May 2025)).

Fourth, an interconnected and distributed representation of domain and external knowledge makes up the Federated Virtual Knowledge Graph (FVKG). This layer follows Ontop. The Ontop system integrates data from distributed and heterogeneous sources using semantic web technologies. The functionalities of Ontop are varied and provide various features such as ontology mapping, query translation, query execution, result integration, and query answering [48]. It also follows the behavior of federation, which is highly dynamic where data in the knowledge graph are not centralized but distributed across multiple data sources, possibly located in different organizations or systems and available without data migration, with a lower latency rate and established connections and presented data in RDF standards and query data using SPARQL language across distributed data sources in real-time. The FVKG approach promotes salient features of data engineering, such as scalability, flexibility, interoperability, efficiency, and real-time insights.

Fifth, the User Application (UA) layer provides the user query endpoints for accessing federated data, enabling the construction of various services on the top of the federated knowledge graph. It also allows fulfills end user and healthcare stockholders' needs and demands in different forms, such as reports. This functionality supports healthcare practitioners in enhancing decision-making for the patient treatment process and empowers patients to improve their personalized health outcomes within the healthcare ecosystem. The FVKG architectural artifact is illustrated in Figure 2.

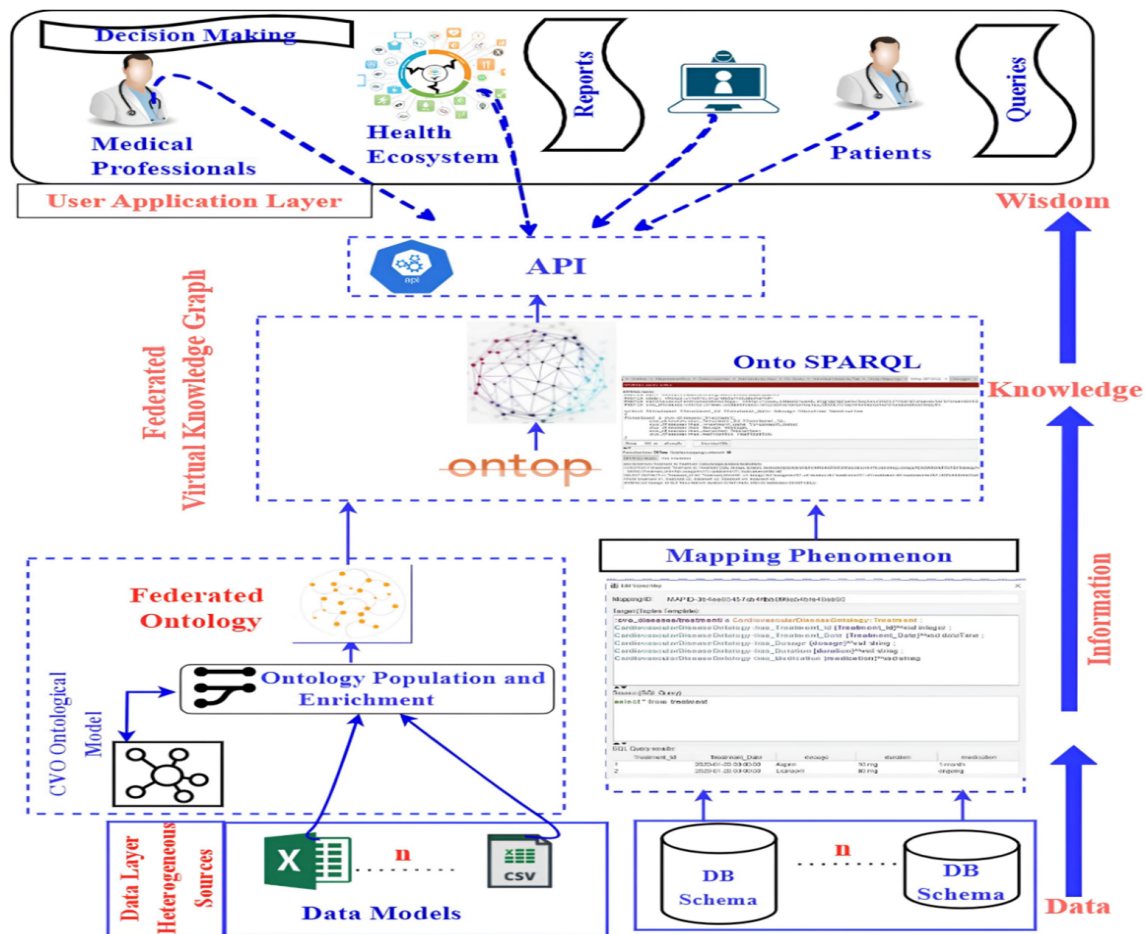


Figure 2. Anatomy of the federated virtual knowledge graph using Ontop framework.

3.4.1. Architectural Flexibility vs. Complexity

The aforementioned layered structure of FVKG (see Figure 2) allows for greater flexibility, modularity, and extensibility (e.g., easy integration of new data sources or changes in domain ontology). However, it may require careful coordination across layers (e.g., ontology design, data source integration, and mapping configuration), which can increase the initial setup effort. In this research, we developed different data-driven pipelines to construct a robust framework that deals with various data with its structure to outline flexibility and follow a modular approach to extensibility (see Section 4.2).

3.4.2. Modularization Strategies

We suggest breaking down the implementation into distinct, manageable components—such as ontology modeling, source schema analysis, mapping definition, and query optimization (see Section 4). Each can be handled using specialized tools or by domain experts, reducing the burden on a single development team.

3.4.3. Automation Tools

Mapping Generation: We used tools like Ontop's mapping editor and Karma (<https://usc-isi-i2.github.io/karma/>) (accessed on 21 May 2025)) that can semi-automatically generate R2RML [52] or OBDA mappings from relational databases based on ontology alignment.

Ontology Reuse and Alignment: Using modular, reusable ontology patterns (e.g., Ontology Design Patterns ([http://ontologydesignpatterns.org/index.php/Ontology_Design_Patterns_.org_\(ODP\)](http://ontologydesignpatterns.org/index.php/Ontology_Design_Patterns_.org_(ODP))) (accessed on 21 May 2025)) or OBO Foundry classes) can accelerate ontology construction and promote interoperability.

Continuous Integration Pipelines: Integrating the deployment process with CI/CD pipelines (e.g., using Docker (<https://www.docker.com/> (accessed on 21 May 2025)) for containerization and automated testing) can streamline updates and ensure consistent reproducibility.

3.4.4. Best Practice Recommendations

In this research, we also outline practical recommendations for managing complexity, such as using version control for mappings and ontologies, adopting collaborative modeling environments (e.g., *WebProtégé*), and leveraging template-based SPARQL query construction to simplify querying.

3.4.5. Selection of Ontop Platform Dependency

In this research, during the development, we were very reliant on the Ontop platform due to its vigilant attributes, exceptionally robust support of OBDA, its mature SPARQL-to-SQL rewriting capabilities, and its compatibility with various relational databases (e.g., Dremio, Denando, Teiid, H2, MySQL, etc.). These vibrant features make it a strong candidate for implementing the proposed FVKG framework. We also mentioned that other vendors also provide core functionalities of the principal framework, such as semantic mapping, query rewriting, and ontology alignment for the use of other OBDA tools or triplestores, including Morph-RDB, D2RQ, or RDF storage platforms like GraphDB (<https://graphdb.ontotext.com/> (accessed on 21 May 2025)) and Stardog (<https://www.stardog.com/> (accessed on 21 May 2025)) (see Section 2.3).

4. Experimental Results

4.1. Constructing a Virtual Semantic View (VSV) for Federated Ontology

Figure 3 illustrates that a virtual semantic view for generating federated ontologies (FO) requires a systematic approach explaining how semantic data integration across heterogeneous (relational) sources affects the outcome. Data consolidation is represented in the form of a triple and is expressed as “ $\lambda = (FO_D, V, Op)$ ”, consisting of the following parameters: (i) FO_D is the FO, which is generated from various data models with multiple structures and data formats (e.g., CSV spreadsheet, etc.) and domain ontology (e.g., CVD).

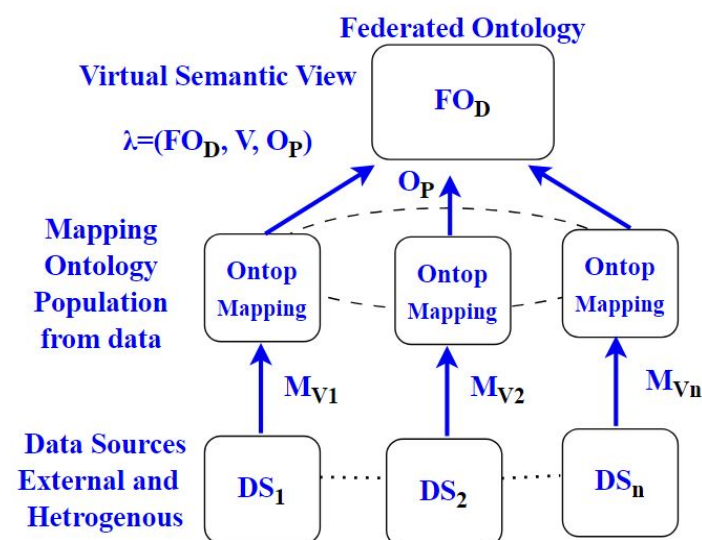


Figure 3. Virtual semantic view of federated ontology.

A shared vocabulary describes data sources in the FO generation on a unified platform; (i) The term “ V ” presents a set of local view specifications “ V_1, \dots, V_n ” that use the terms in “ FO_D ” to describe the data sources “ DS_1, \dots, DS_n ”. Ontop provides a virtual environment

that utilizes a variant of relational algebra tailored to encode SPARQL queries along the lines of the Intermediate Query (IQ) language [4,43]. The concepts in the vocabulary of “ FO_D ” that are found in “ MV_i ” are related to terms in “ S_i ” using a set of mapping rules. (ii) The term “ P ” is an ontology population process that adheres to specific mapping rules to transform relational data sources and data models into the OWL file (see Figure 3). The forthcoming section described mapping techniques for mapping heterogeneous data sources representing identical real-world entities into the FO (see Table 2). The mapping phenomenon applies to the OPE method. This method follows a defined set of mapping rules and principles [4,67].

Table 2. Relational database to ontology mapping rules.

Mapping Rules
Rule 1: Tables in databases are to be associated with OWL classes in ontology file.
Rule 2: Handling of bridge tables using constraint keys.
Rule 3: Mapping of referential integrity relationships to inheritance hierarchy.
Rule 4: Mapping of non-referential integrity columns into datatype properties.
Rule 5: Represents data type properties in the host class as Domain (D) and data type as Range (R).
Rule 6: Mapping relationships represented by referential integrity columns into object properties.
Rule 7: Representation of object property host classes as D and R.
Rule 8: All table records are mapped to individuals in ontology.
Rule 9: Database column constraints are mapped into ontology property cardinalities [68].

4.2. Domain-Oriented Ontological Model and Data Visualization Artifact

This research demonstrates the advanced development process of a focus ontology based on Cardiovascular Diseases (CVO) while capturing the contextual knowledge of healthcare units within healthcare organizations. The ontology development process follows a top-to-bottom modular approach combining, aligning, and extending predefined ontologies relevant to the CVO and healthcare domains.

Ontological Blueprint and Creation: We showcase the construction of the CVO ontology and its systematic process in Figure 4, emphasizing the sequential phases involved in building the system architecture needed to create federated IS. This phase also contributes to building FO and constituting the domain ontological model, and it consists of injecting data into the ontology OWL file using the OPE method. This section further explains the foundation for building domain ontologies, which serve specific purposes, reflect a particular domain, and define the activities to be included. We utilized mature methodologies for developing ontologies, such as Methontology [63] and Tove [64] methodologies. We used a customized approach to develop FO, integrating domain knowledge, contextual insights from domain experts, and external domain knowledge through the OPE method by fetching data from various data models and mapping into the FO. Figure 4 below showcases the ontological model taxonomies and outlines ontology visualization.

Top-to-bottom Modular Approach for Ontology Construction: In this phase, we employed various relevant ontology development methods such as Methontology and Tove. We customized certain activities of these methods to develop a robust framework for incorporating contextual knowledge regarding CVO. The customized hybrid approach helps build the FO ontological model (the model contains information mapped from various sources). The CVO ontological model proposed the foundational ontological meta-model for constructing federated information systems. It highlights a multidisciplinary application in which we

followed a modular approach for heavy-weight ontology construction with domain-centric axioms and their populations.

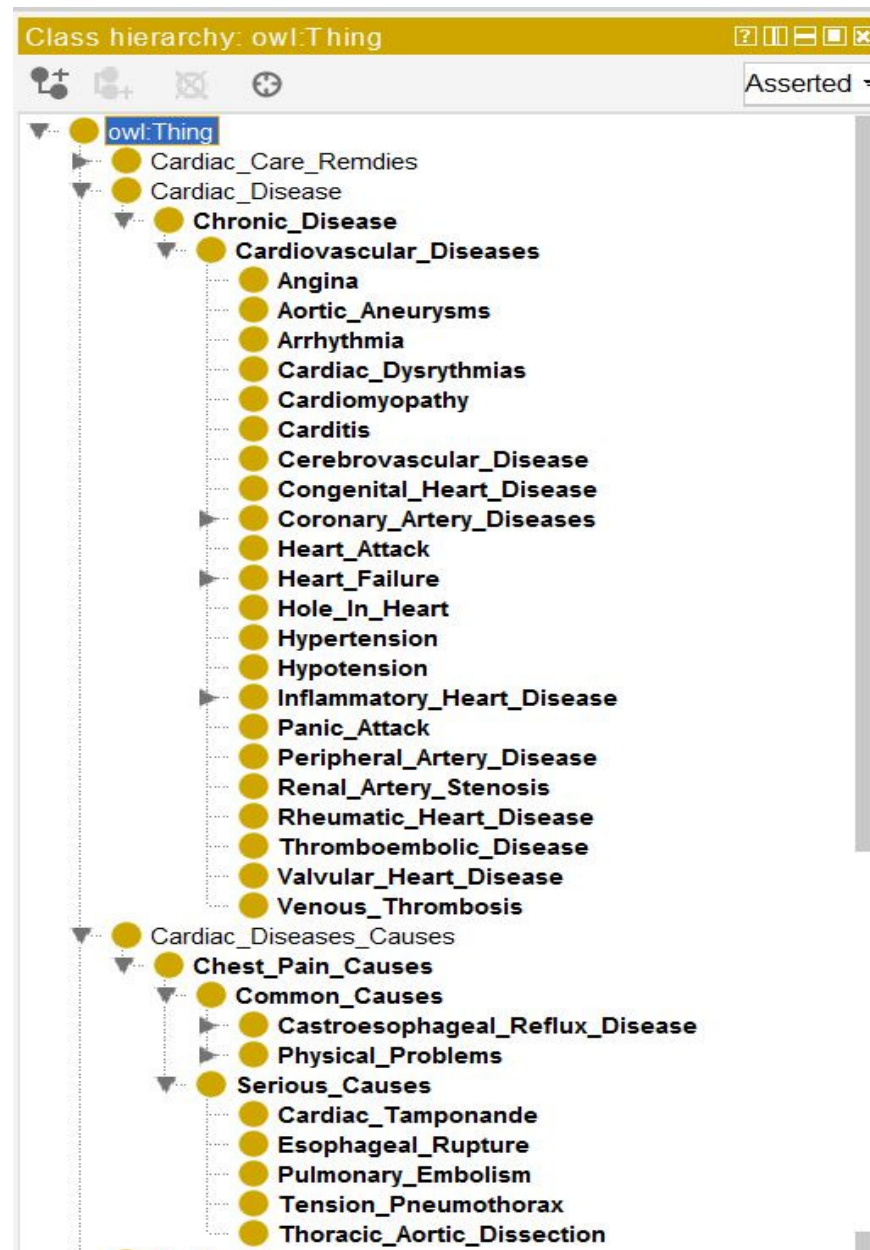


Figure 4. A chunk of Cardiovascular disease ontological metadata model and taxonomies developed in Protege.

Merging, Aligning, and Extending for Federated Ontology: In the process of FO, we demonstrate a conceptual view of merging the ontological model of CVO and contextual knowledge and mapping external knowledge from different data models using the OPE method.

At the same time, Ontop provides a platform for creating a virtual environment by integrating various internal and external relational data sources, along with knowledge consolidated in the FO. The federated knowledge is accumulated in the federated ontology, enabling virtualization capabilities. The federated ontology consolidates data results mapped from various relational databases, seed ontology, and data models without data movement or migration, resulting in a lower latency rate. The federated data are eligible for exchange with a standardized RDF format for developing different data-driven applications.

Federated Ontology (FO) Validation: In this phase, we incorporated multiple data models in different structures and formats for data validation, ensuring consistency and defining the logical reasoning of ontologies, including FO and CVO ontology, integrating them into the proposed CVO ontological metadata model. We performed validation tests to ensure ontology consistency and check the work coherence and legitimacy in the development process of ontologies. For visualization, these tests used the “*Ontology Debugger*” and “*Ontop Mapping Validate*” plugins within ontology editing tools like Protege, as illustrated in Figures 5 and 6.

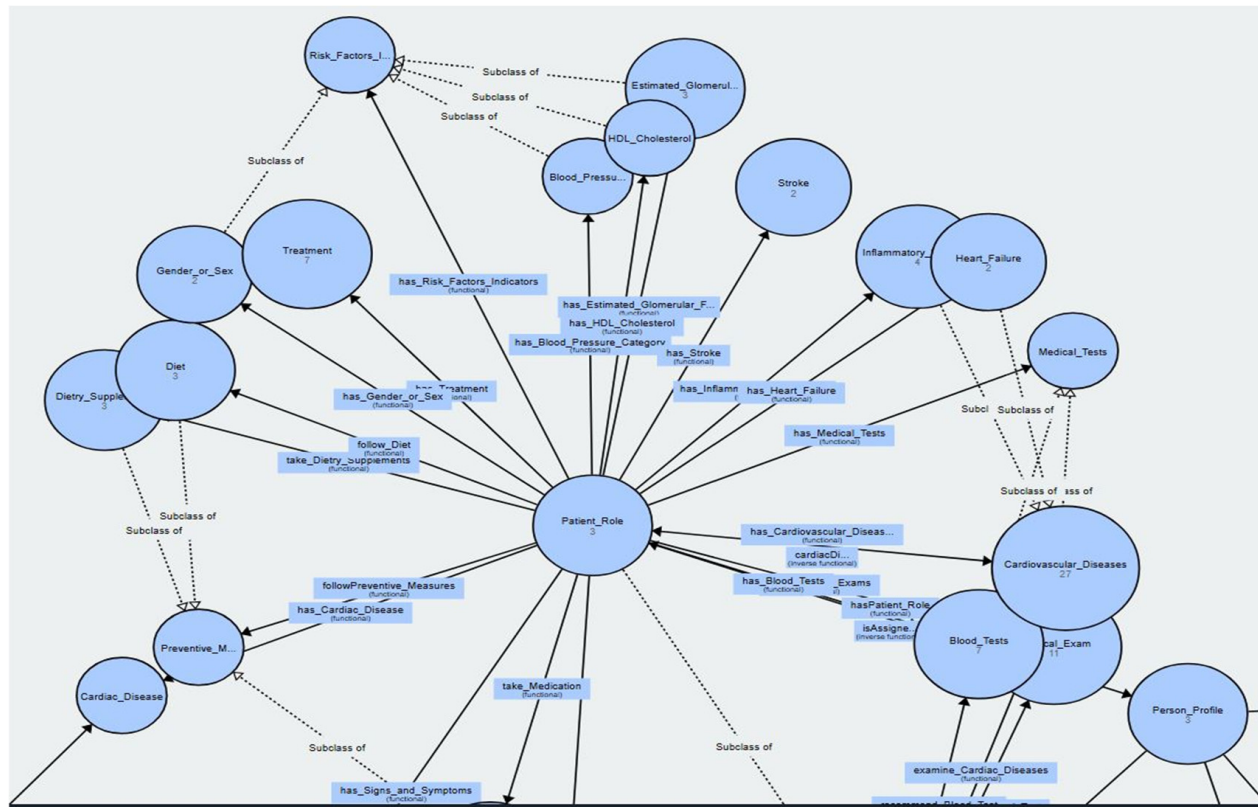
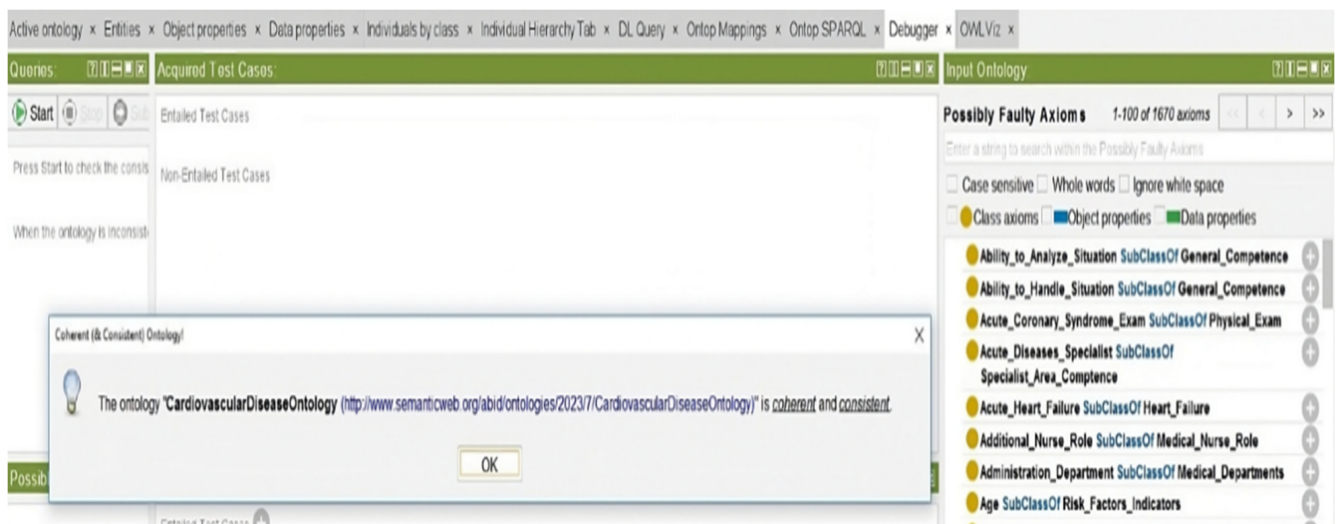


Figure 5. A chunk of Cardiovascular disease ontological model outlining visualization developed in WebProtege.

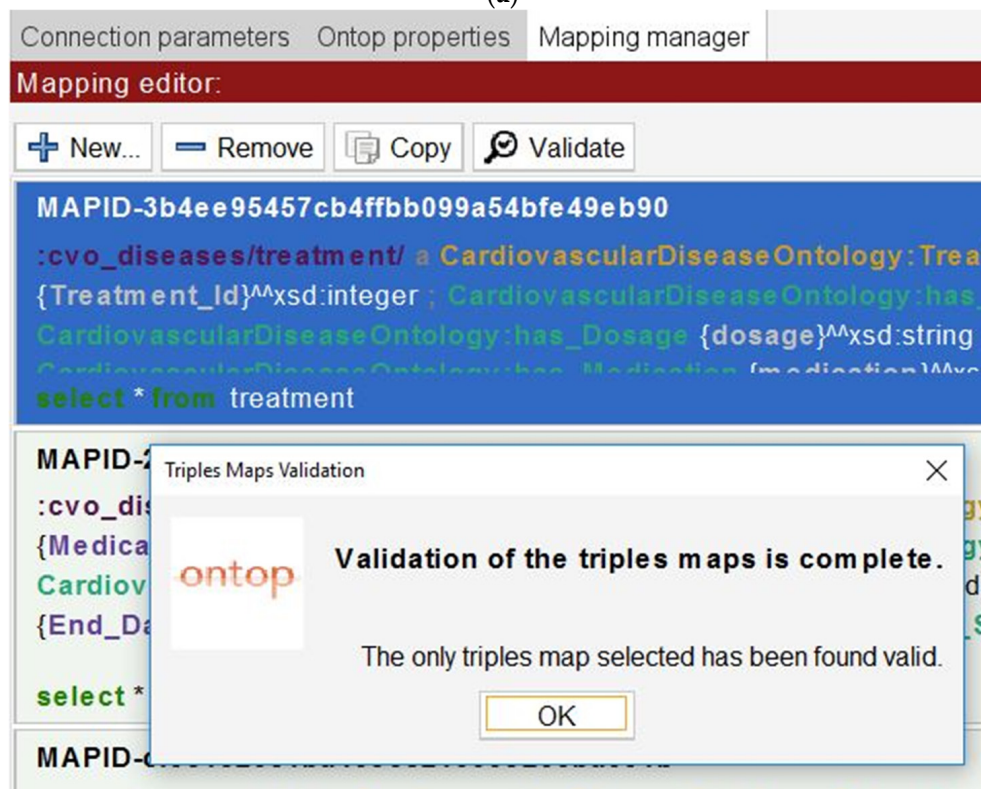
4.3. A Systematic Process for Federated Virtual Knowledge Graph (FVKG) Generation

In this section, we concentrate on the VKG approach to build a federated VKG (FVKG) framework using a virtual system like Ontop. We focused on the systematic mechanism for mapping data from various external and internal heterogeneous data sources. Many data sources in the literature are based on relational database structures. The following systematic process follows three main steps, as described in Figure 7.

The primary contribution of this research is the development of an FO that maps and extracts essential information from the input data models using the OPE method. The OPE method facilitates mapping various data models and structures (e.g., Excel Spreadsheet, CSV files) into the FO in RDF format, creating a standardized shared vocabulary and generating data harmonization to address data integration and resolving semantic interoperability issues. The FO also includes a domain ontology (CVD), contextual knowledge from the healthcare domain, and all axioms generated using the OPE method (see Figure 7).



(a)



(b)

Figure 6. Cardiovascular disease ontological model interlinking alignment, consistency, and mapping validation. (a) Federated ontology and different data models interlinking alignment and consistency test; (b) Ontop mapping validation test.

Second, in a virtual system, Ontop is dedicated to laying mapping environments using the mapping manager functionality in ontology editors like Protege, which is responsible for retrieving and mapping information from external and internally distributed data sources using the Ontop platform and employing R2RML language, which helps to retrieve information from local structured sources, enabling SQL results and mapping with SPARQL query to generate the same response for each user's query. The core functionality of this layer is to construct the federated VKG, which incorporates data from the federated ontology and fetches the data from local and external relational data sources by loading it. In this phase, we constructed a small-scale database focused on cardiovascular chronic

diseases (*cvo_diseases*) using a relational database like MySQL. A snapshot of the MySQL enhanced entity relation (EER) diagram is shown in Figure 7. Concerning the Excel sheet (external data files), We created a test data file from the published data repository Kaggle, as illustrated in Figure 7.

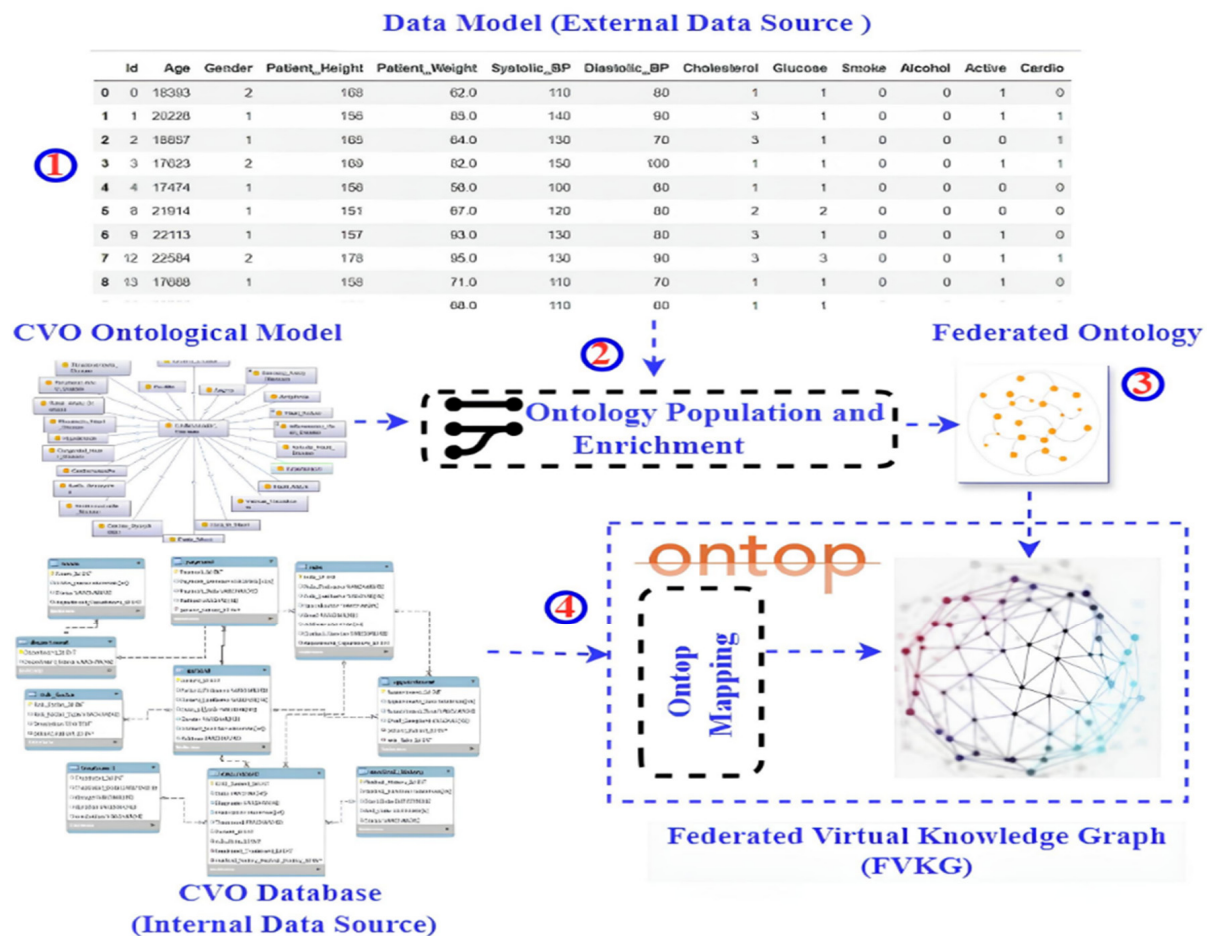


Figure 7. A systematic process of federated ontology and federated virtualized knowledge graph (VKG) generation.

Third, the federated VKG (FVKG) is built based on the FO outcomes, with results retrieved leveraging the Ontop distributed system from local or distributed data sources and showcased in FVKG. The FVKG contains the homogenized result with a standard common vocabulary for access and reuse in different contexts of health applications related to the IS landscape.

4.4. Inputs to Build VKGs: Ontology, Mappings, Queries, Databases, and Ontop

In the literature, Ontop is considered to be one of the prominent OBDA distributed systems that support all the W3C standards and recommendations related to OBDA: OWL2QL (<https://www.w3.org/TR/owl2-profiles/> (accessed on 21 May 2025)), R2RML, SPARQL, SWRL, and the OWL2QL entailment regime in SPARQL. In addition, it supports all major commercial and open-source relational databases [48].

4.4.1. Ontology

Ontop uses RDFS [69] and OWL2QL as ontology languages, guaranteeing that queries over the ontology can be rewritten into equivalent queries over the data alone. Ontop is also extended to support fragments of SWRL [70].

Example 1. The following ontology metadata captures the domain knowledge of running an example targeting the cardiovascular domain. It decides the concepts of cardiac diseases and cardiac patients with the following OWL axioms:

:Heart_Attack rdfs:subClassOf :Cardiovascular_Diseases. (1)

:Cardiovascular_Diseases rdfs:subClassOf :Chronic_Diseases. (2)

:Chronic_Diseases rdfs:subClassOf :Cardiac_Diseases. (3)

:has_Cadiac_Diseases rdfs:domain :Patient_Role. (4)

:hasCardiac_Diseases rdfs:Range :Cardiac_Disease. (5)

:hasPatient_ID a owl:DatatypeProperty. (6)

:has_Chest_Pain a owl:ObjectProperty. (7)

In the example mentioned above, classes such as :Heart_Attack, :Cardiovascular_Diseases, and :Chronic_Diseases are sub-classes of :Cardiac_Diseases (including other cardiovascular diseases). The object property :has_Cadiac_Diseases has class :Patient_Role as its domain and :Cardiac_Disease as its range. We also have a datatype property :hasPatient_ID.

4.4.2. SPARQL Queries and Mappings

Ontop mainly supports two mapping languages: the W3C RDB2RDF (<https://www.w3.org/2001/sw/wiki/RDB2RDF> (accessed on 21 May 2025)) mapping language (R2RML), a widely used standard, and the native Ontop mapping language, which is easier to learn and use to retrieve external data from external disparate sources. Ontop includes tools for converting native mappings into R2RML mapping and vice versa. Intuitively, a mapping assertion consists of a source, an SQL query retrieving values from the data sources or bases, and a target, which constructs RDF triples with values from the source [48].

Example 2. The part of the ontology in Example 1 can be populated from the database (see Figure 7). The simplified Ontop mapping syntax can be presented as follows (see Table 3):

Table 3. Ontop mapping generation performed by the mapping manager in Protege.

Prefix	
PREFIX owl: < http://www.w3.org/2002/07/owl# > PREFIX rdf: < http://www.w3.org/1999/02/22-rdf-syntax-ns# > PREFIX xml: < http://www.w3.org/XML/1998/namespace > PREFIX obda: < https://w3id.org/obda/vocabulary# > PREFIX CardiovascularDiseaseOntology: < http://www.semanticweb.org/abid/ontologies/2023/7/CardiovascularDiseaseOntology# > PREFIX cvo_diseases: < http://www.semanticweb.org/abid/ontologies/2023/7/CardiovascularDiseaseOntology# >	
Syntax of SPARQL Query	Mapping Query Generated by Ontop
select ?treatment ?Treatment_Id ?Treatment_Date ?dosage ?duration ?medication where { ?treatment a cvo_diseases:Treatment; cvo_diseases:has_Treatment_Id ?Treatment_Id; cvo_diseases:has_Treatment_Date ?Treatment_Date; cvo_diseases:has_Dosage ?dosage; cvo_diseases:has_Duration ?duration; cvo_diseases:has_Medication ?medication. }	:cvo_diseases/treatment/ a CardiovascularDiseaseOntology:Treatment; CardiovascularDiseaseOntology:has_Treatment_Id {Treatment_Id}^^xsd:integer; CardiovascularDiseaseOntology:has_Treatment_Date {Treatment_Date}^^xsd:dateTime; CardiovascularDiseaseOntology:has_Dosage {dosage}^^xsd:string; CardiovascularDiseaseOntology:has_Duration {duration}^^xsd:string; CardiovascularDiseaseOntology:has_Medication {medication}^^xsd:string.

Table 3. Cont.

Syntax of SPARQL Query	Prefix	Mapping Query Generated by Ontop
<pre> select ?medical_history ?Medical_History_Id ?Medical_Condition ?Start_Date ?End_Date ?Status where { ?medical_history a cvo_diseases:Medical_Hisotry; cvo_diseases:has_Medical_History_Id ?Medical_History_Id; cvo_diseases:has_Medical_Condition ?Medical_Condition; cvo_diseases:has_Start_Date ?Start_Date; cvo_diseases:has_End_Date ?End_Date; cvo_diseases:has_Status ?Status. } select ?role ?Role_Id ?Role_Firstname ?Role_Lastname ?Specialization ?Email ?Address ?Contact_Number where { ?role a cvo_diseases:Role; cvo_diseases:has_Role_Id ?Role_Id; cvo_diseases:has_Role_Firstname ?Role_Firstname; cvo_diseases:has_Role_Lastname ?Role_Lastname; cvo_diseases:has_Specialization ?Specialization; cvo_diseases:has_Email ?Email; cvo_diseases:has_Address ?Address; cvo_diseases:has_Contact_Number ?Contact_Number. (?Specialization="Cardiologist"). </pre>		<pre> :cvo_diseases/medical_history/ a CardiovascularDiseaseOntology:Medical_History; CardiovascularDiseaseOntology:has_Medical_History_Id {Medical_History_Id}^^xsd:integer; CardiovascularDiseaseOntology:has_Medical_Condition {Medical_Condition}^^xsd:string; CardiovascularDiseaseOntology:has_Start_Date {Start_Date}^^xsd:dateTime; CardiovascularDiseaseOntology:has_End_Date {End_Date}^^xsd:dateTime; CardiovascularDiseaseOntology:has_Status {Status}^^xsd:string. :cvo_diseases/role/ a CardiovascularDiseaseOntology:Role; CardiovascularDiseaseOntology:has_Role_Id {Role_Id}^^xsd:integer; CardiovascularDiseaseOntology:has_Role_Firstname {Role_Firstname}^^xsd:string; CardiovascularDiseaseOntology:has_Role_Lastname {Role_Lastname}^^xsd:string; CardiovascularDiseaseOntology:has_Specialization {Specialization}^^xsd:string; CardiovascularDiseaseOntology:has_Email {Email}^^xsd:string; CardiovascularDiseaseOntology:has_Address {Address}^^xsd:string; CardiovascularDiseaseOntology:has_Contact_Number {Contact_Number}^^xsd:string. </pre>

4.4.3. Databases

Ontop supports standard relational databases engines via the Java Database Connectivity (JDBC) API. These databases include all major commercial relational databases including DB2 (<https://www.ibm.com/db2>), Oracle (<https://www.oracle.com/>), MS SQL Server (<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>), and most popular open-source databases such as PostgreSQL (<https://www.postgresql.org/>), MySQL, H2 (<https://www.h2database.com/html/main.html> (accessed on 21 May 2025)), and HSL (<https://www.hsl.fi/en/hsl/open-data>). In addition, Ontop can be used with federated databases such as Teiid and Exareme (<http://madgik.github.io/exareme/> (accessed on 21 May 2025)), formally called ADP [71], to support multiple data sources including relational databases, XML, CSV, and web services [48].

4.4.4. Ontop Core

The core part of Ontop is the SPARQL engine Quest, which is responsible for rewriting SPARQL queries over the virtual RDF graph and the domain-centric federated ontology into SQL queries over the relational database (see Table 3).

4.5. Using Ontop Virtual Platform as Transformation Mapping Artifact for Data Integration and Data Interoperability Management

Developing a transformation artifact using Ontop is essential for data integration and ensuring seamless interoperability. We described domain-centric ontology (e.g., seed ontology) as representing the relevant concepts and relationships within data sources. We also map data from each source, such as individual relational databases, to the corresponding ontological concepts and build a Federated Ontology (FO). Moreover, we utilized Ontop, which establishes mapping between elements in the data sources and the concept defined in the domain ontology (e.g., CVO ontological model). Additionally, Ontop applies transformation rules using mapping languages like R2RML to convert and align data from their primitive format to the federated ontologies. These rules enable querying, transformation, and alignment with the domain ontology. The execution of this mapping phenomenon can be seen in Figure 8.

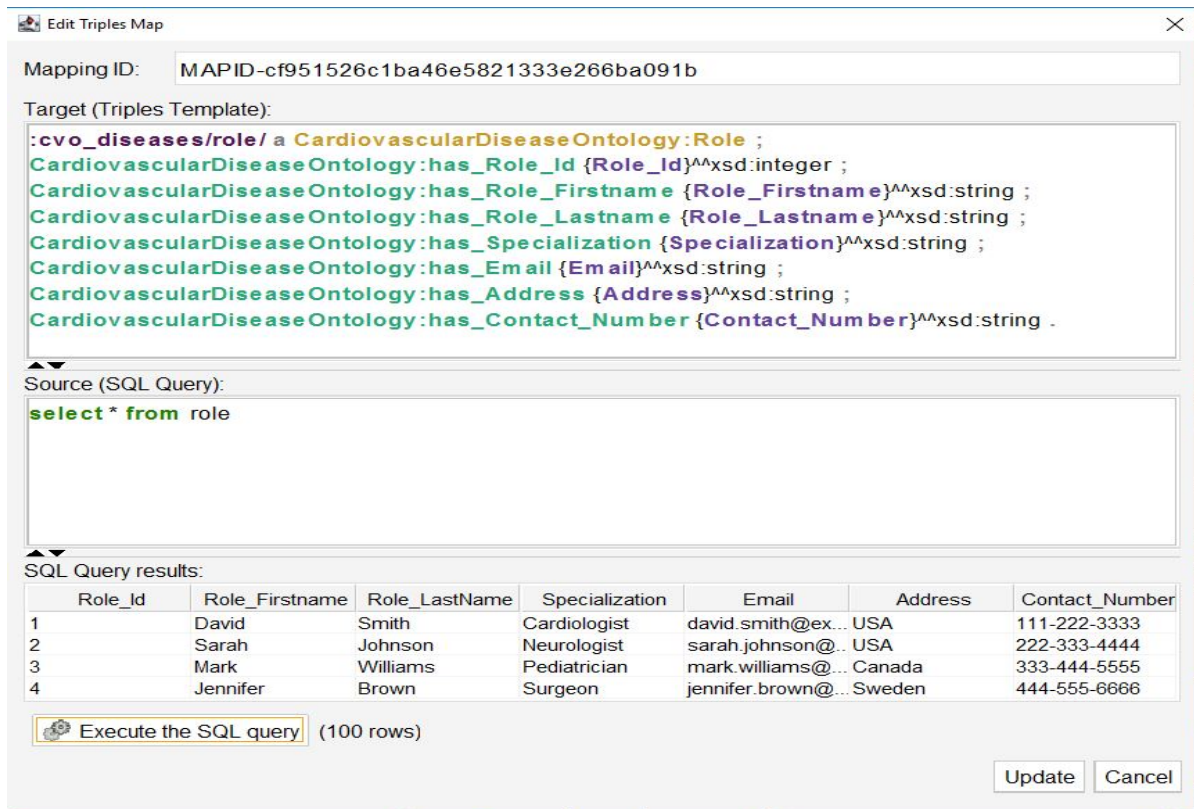


Figure 8. Transformation mapping bootstrappers: A sample of database results retrieved using the Ontop framework.

5. Evaluation and Testing

5.1. Ontological Model Testing to Verify Competence Questions Using DL in Protege

In this section, we verify the Competence Questions (CQs) by explaining a success scenario. Figure 9 demonstrates the ontological model testing the success scenario.

CQ: What roles are involved in examining cardiac diseases, performing medical tests, and recommending medicine for effective treatment?

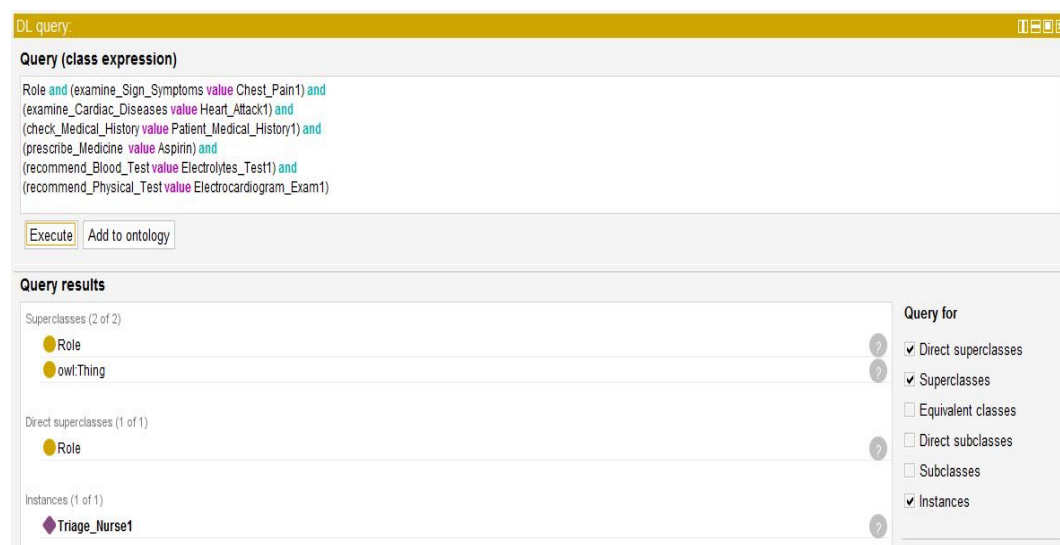


Figure 9. Cardiovascular disease ontological model testing and verification of competence questions (CQs).

Success Scenario: The Triage Nurse role (e.g., *Triage_Nurse1*) is assigned to healthcare worker “A”, who plays an active role in hospital care management and performs various tasks within the medical institution. Healthcare worker “A” is responsible for several activities, including conducting physical examinations to identify medical issues for patient treatment planning. Additionally, healthcare worker “A” possesses competencies in cultural, occupational, educational, and general areas (which are not shown for brevity) [4].

We employed two essential tools, the DL query plugin in Protege 5.6.4, to standardize metadata and validate competence-related queries. This tool is crucial for verifying CQs from the federated ontological model repository and delivering a comprehensive response regarding the success of the scenario mentioned above. Figure 9 showcases the verification and testing of CQs.

For practical validation and testing, domain experts reviewed the CQs during the information acquisition sessions with healthcare practitioners (HPs) and acquired relevant information and standard vocabularies for cardiovascular diseases (CVDs) for developing ontological meta-models for developing a robust FVKG framework.

5.2. Data Integration and Interoperability Tests Using Ontop SPARQL Endpoint to Evaluate Federated VKG

The section illustrates a state-of-the-art Ontop technique for retrieving data from the FVKG, facilitating the integration of diverse data from various structures and sources, both external and internal. Figure 10 shows the Ontop plugin’s graphical interface in Protege 5.6.4 for formulating user queries using the Ontop SPARQL endpoint. It translates the user SPARQL query into SQL query using SQL Translator and generates an Intermediate Query (IQ), which efficiently helps retrieve data from data sources and ontological models. The Ontop SPARQL endpoint is an interface provided by the Ontop platform that enables users to query virtual knowledge graphs leveraging the SPARQL query language. Ontop SPARQL serves as a mediating layer between users to submit SPARQL to the VKG generated by Ontop and allows users to query virtually with data migration from the integrated, distributed data sources in a unified manner. It supports data integration and querying across heterogeneous data sources, fostering interoperability and enabling advanced data-driven applications and landscapes.

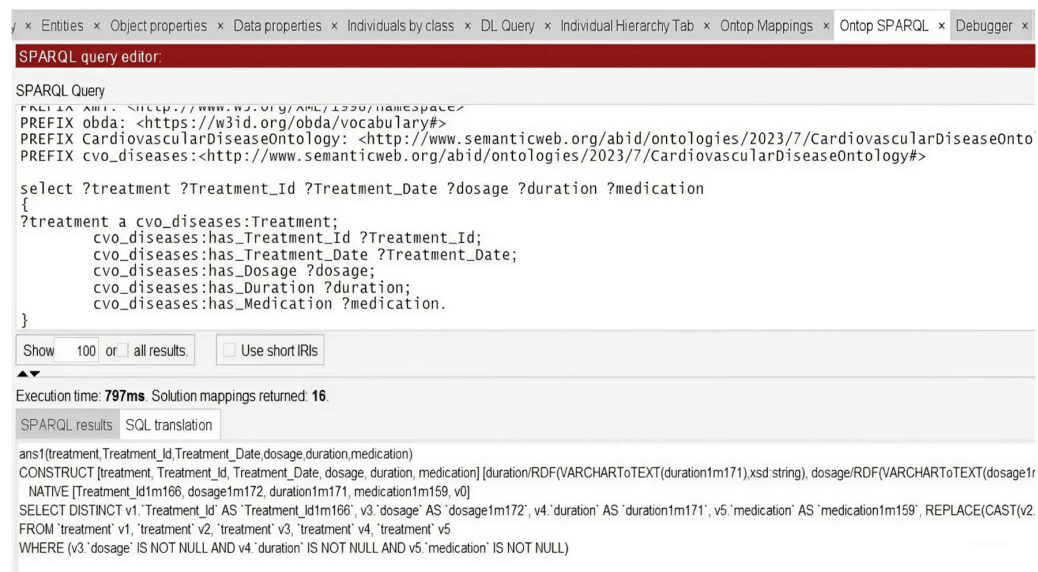


Figure 10. Evaluation of DI&I and mapping of disparate data sources; retrieval using Ontop.

Figure 10 illustrates the evaluation procedure of FVKG and data retrieval based on customized queries written by end users in a virtual environment without data migration or movement across distributed data sources. In FVKG, the virtualized approach results in lower data latency and a unified data format with a standardized vocabulary, effectively addressing data integration and interoperability challenges. This behavior of data helps for data exchange between different data-driven applications or IS within exact or other domains.

For particle validation and evaluation, we used published real-world datasets (e.g., cardiovascular disease-related data from the Kaggle data repository) to deploy a proof-of-concept system.

5.3. Comparative Synthesis with Existing Approaches: Ontop and SQL for Data Retrieval with Time over Databases

After constructing the FVKG, we tested the system's performance by executing various SPARQL queries that mapped distributed database record values and generated corresponding SQL translation values (see Table 3). We made a comparison in two steps: (i) we compared the Ontop mapping approach using the Mapping Manager plugin in Protege 5.6.4 and created the Ontop mapping language, then, the Ontop SPARQL plugin generated the SPARQL results and SPARQL translation in SQL. (ii) We tested the exact SPARQL translation generated query in MySQL Workbench 8.0 and received the same results. Table 4 highlights the mapping phenomenon of the Ontop and SQL approaches and presents an average execution time comparison of SPARQL queries and SQL queries executed using the Ontop plugin in Protege and the implementation of the virtualized knowledge graph approach to construct FVKG.

Table 4. Query execution time comparison between Ontop mapping and SQL translation.

Syntax	Query	Execution Time
PREFIX owl: < http://www.w3.org/2002/07/owl# > PREFIX rdf: < http://www.w3.org/1999/02/22-rdf-syntax-ns# > PREFIX xml: < http://www.w3.org/XML/1998/namespace > PREFIX obda: < https://w3id.org/obda/vocabulary# > PREFIX CardiovascularDiseaseOntology: < http://www.semanticweb.org/abid/ontologies/2023/7/CardiovascularDiseaseOntology# > PREFIX cvo_diseases: < http://www.semanticweb.org/abid/ontologies/2023/7/CardiovascularDiseaseOntology# >		
Ontop Mapping	select ?treatment ?Treatment_Id ?Treatment_Date ?dosage ?duration ?medication where { ?treatment a cvo_diseases:Treatment; cvo_diseases:has_Treatment_Id ?Treatment_Id; cvo_diseases:has_Treatment_Date ?Treatment_Date; cvo_diseases:has_Dosage ?dosage; cvo_diseases:has_Duration ?duration; cvo_diseases:has_Medication ?medication. FILTER(?Treatment_Id=1)}	34 ms
SQL Translation	SELECT DISTINCT v1.'Treatment_Id' AS 'Treatment_Id1m177', v3.'dosage' AS 'dosage1m184', v4.'duration' AS 'duration1m183', v5.'medication' AS 'medication1m168', REPLACE(CAST(v2.'Treatment_Date' AS CHAR(30)), ' ', 'T') AS 'v0'. FROM 'treatment' v1, 'treatment' v2, 'treatment' v3, 'treatment' v4, 'treatment' v5. WHERE ((v1.'Treatment_Id' = 1) AND v3.'dosage' IS NOT NULL AND v4.'duration' IS NOT NULL AND v5.'medication' IS NOT NULL)	0.001806 ms
Ontop Mapping	select ?role ?Role_Id ?Role_Firstname ?Role_Lastname ?Specialization ?Email ?Address ?Contact_Number where { ?role a cvo_diseases:Role; cvo_diseases:has_Role_Id ?Role_Id; cvo_diseases:has_Role_Firstname ?Role_Firstname; cvo_diseases:has_Role_Lastname ?Role_Lastname; cvo_diseases:has_Specialization ?Specialization; cvo_diseases:has_Email ?Email; cvo_diseases:has_Address ?Address; cvo_diseases:has_Contact_Number ?Contact_Number. FILTER ((?Specialization)="Cardiologist" && (?Role_Id=1) && (?Role_Firstname="Mark"))}	72 ms
SQL Translation	SELECT DISTINCT v6.'Address' AS 'Address1m156', v7.'Contact_Number' AS 'Contact_Number1m161', v5.'Email' AS 'Email1m151', v1.'Role_Id' AS 'Role_Id1m166', v3.'Role_LastName' AS 'Role_LastName1m192' FROM 'role' v1, 'role' v2, 'role' v3, 'role' v4, 'role' v5, 'role' v6, 'role' v7 WHERE ((v1.'Role_Id' = 1) AND 'Mark' = v2.'Role_Firstname' AND 'Cardiologist' = v4.'Specialization');	0.00698 ms
Ontop Mapping	select ?medical_history ?Medical_History_Id ?Medical_Condition ?Start_Date ?End_Date ?Status where { ?medical_history a cvo_diseases:Medical_History; cvo_diseases:has_Medical_History_Id ?Medical_History_Id; cvo_diseases:has_Medical_Condition ?Medical_Condition; cvo_diseases:has_Start_Date ?Start_Date; cvo_diseases:has_End_Date ?End_Date; cvo_diseases:has_Status ?Status.}	84 ms

Table 4. Cont.

Syntax	Query	Execution Time
SQL Translation	SELECT DISTINCT v2.'Medical_Condition' AS 'Medical_Condition1m159', v1.'Medical_History_Id' AS 'Medical_History_Id1m191', v5.'Status' AS 'Status1m172', REPLACE(CAST(v4.'End_Date' AS CHAR(30)),',','T') AS 'v0', REPLACE(CAST(v3.'Start_Date' AS CHAR(30)),',','T') AS 'v1' FROM 'medical_history' v1, 'medical_history' v2, 'medical_history' v3, 'medical_history' v4, 'medical_history' v5;	0.001806 ms

Figure 11 provides a comparative synthesis with existing approaches Ontop and SQL for data retrieval with time over databases. It defines their efficiency gap in query execution times during information retrieval processes. The bar chart highlights that SQL translations exhibit significantly faster execution times compared to Ontop mappings. The disparity highlights the performance metrics and the urgent need to optimize Ontop configurations to improve performance and achieve greater efficiency in data retrieval workflows. It also explains the optimization of Ontop and its configuration for performance gains in the information retrieval process over distributed data sources. The shorter execution times of the Ontop technique in terms of distributed data fetching over centralized data fetching using SQL are quite significant. The Ontop results highlight its efficacy in efficiently processing queries, showcasing its advantage over SQL in query retrieval, efficiency, and performance in distributed environments.

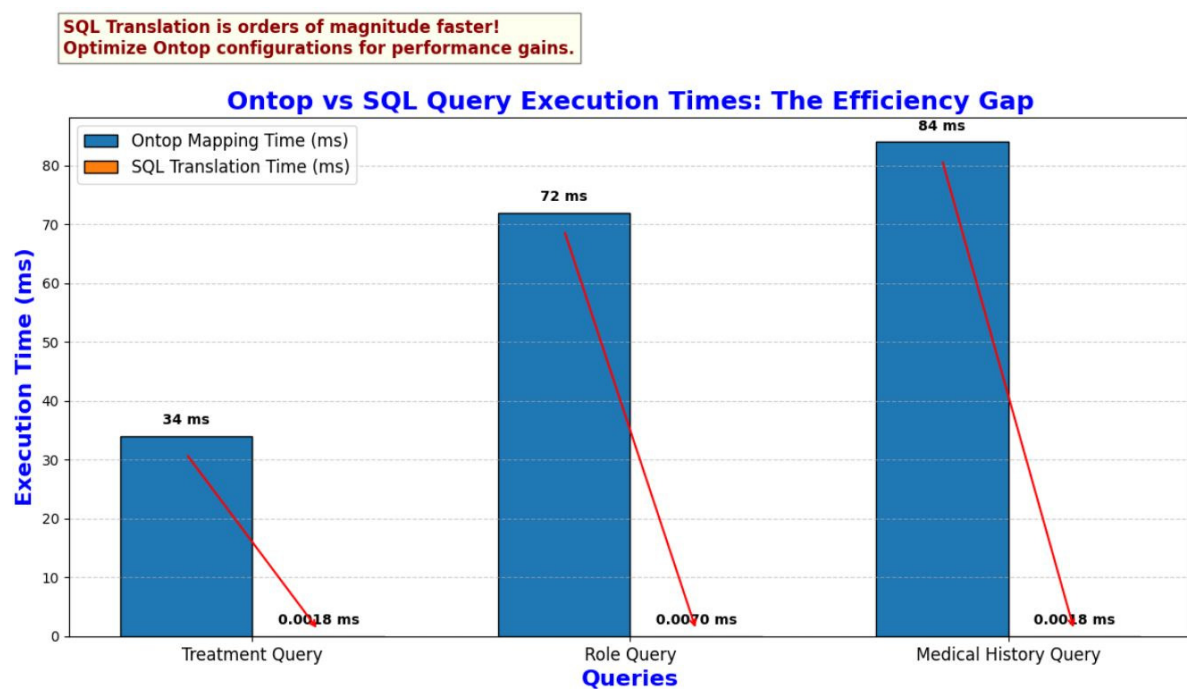


Figure 11. Bar chart illustrating the comparative synthesis of Ontop vs SQL query execution time using three sample queries.

5.4. Ontop SPARQL Endpoint with Ontop CLI

In this section, we describe setting up an Ontop SPARQL endpoint with an Ontop CLI (<https://github.com/ontop/ontop/releases/> (accessed on 21 May 2025)) environment. For this setup, we followed specific structured steps: (i) download and extract CLI; (ii) copy the database JDBC (<https://dev.mysql.com/downloads/connector/j/>) driver to Ontop_CLI_DIR; (iii) initialize the database (e.g., *MySQL*); (iv) prepare and download input files, including an OWL ontology file, a mapping file, and a database properties file, which are downloaded and organized within the input directory; (v) start the Ontop endpoint, enabling interactions with the knowledge graph data, and access the SPARQL

endpoint web interface (e.g., at <http://localhost:8080/> (accessed on 21 May 2025)). SPARQL also helps with various CQ queries executed to verify the functionality of the setup and information retrieval from the FVKG (see Figure 12). This systematic approach ensured the successful deployment of the Ontop SPARQL endpoint for data querying and exploration.

CQ: *What roles are involved in examining cardiac diseases, performing medical tests, and recommending medicine for effective treatment?*

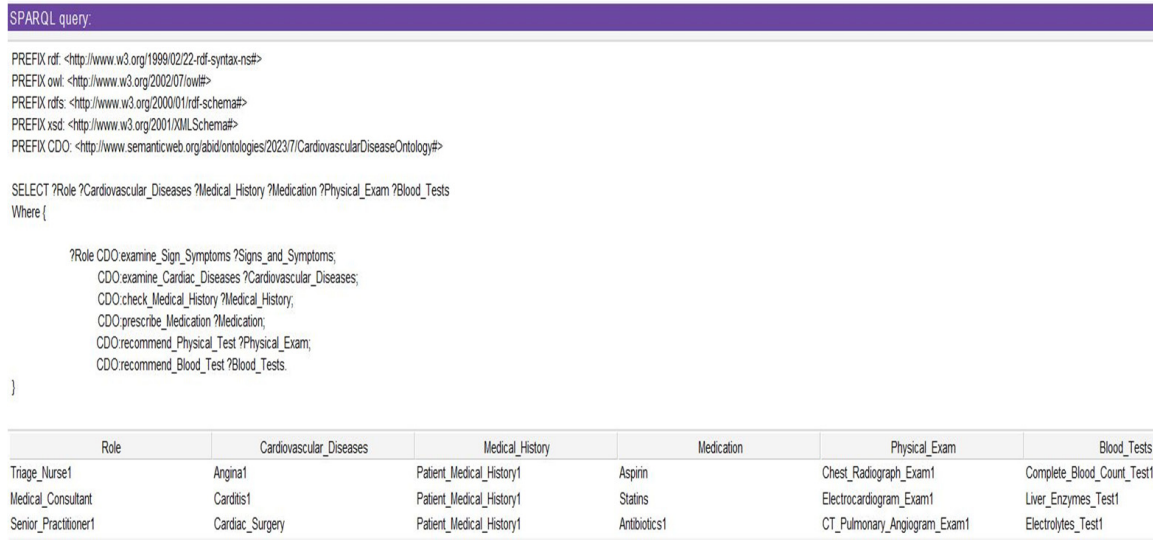


Figure 12. Ontop SPARQL endpoint for information retrieval against sample query.

The main advantages of developing a SPARQL endpoint server, particularly in the context of semantic web technologies, knowledge graphs, and data integration use cases, are as follows: (i) centralized access to structured data, (ii) semantic data querying, (iii) interoperability and data integration, (iv) supporting linked data and open data, (v) enabling advanced analytics, (vi) flexibility and extensibility, (vii) supporting federated querying and empowering developers and researchers in various domains such as healthcare (e.g., enabling semantic querying across patient records, clinical trials, and research datasets to improved interoperability and patient care).

6. Conclusions and Future Directions

This research makes significant contributions, primarily by bridging Ontology Engineering (OE) paradigms with the Design Science Research (DSR) discipline to develop digital artifacts that address real-time challenges and support the evolution of new types of IS in the healthcare domain. We introduced a customized hybrid method, Ontology-based Design Science Research Engineering (ODSRE), which integrates DSR activities with OE principals, incorporating elements from Methontology and Tove methodologies to develop ontological model digital artifacts. This contribution encourages IS practitioners to align with the software engineering perspective closely, creating new kinds of design artifacts (e.g., KGs) and leveraging *Design Science* (DS) theories for IS development across various domains, including Health Information Systems (HISs).

Second, this research demonstrates the effectiveness of utilizing the Ontop platform with the VKG approach alongside an FVKG to address Data Integration (DI) and Semantic Interoperability (SI) bottlenecks in federated HIS. The proposed framework enhances DI&SI, providing a platform as a unified virtual global view of diverse data sources and allowing for querying based on domain-centric concepts rather than underlying data structures and formats.

Additionally, FVKG minimizes data transfer, reduces latency rate, ensures data freshness, and improves overall data access efficiency and reliability. As a strategic approach to building a robust ontological model that captures domain-specific knowledge, it enables Ontology-based Data Access (OBDA) and helps in semantic alignment for ontology-driven applications. We utilized a customized hybrid framework FVKG to facilitate domain ontological model construction and data fetching and mapping using the Ontop platform from different distributed data sources or models within the healthcare domain.

We are fully motivated and looking ahead. Several areas of future research can be improved by the proposed FVKG framework and its applications in healthcare settings. As a future direction, we are fully committed to planning implementation with real-world datasets from medical institutions and deploying proposed FVKGs to see the mapping phenomenon using the Ontop platform.

Enhanced semantic mapping methods: We plan to develop advanced semantic mapping algorithms integrating Artificial Intelligence (AI) and Machine Learning (ML) techniques for improved ontology alignment and schema matching across heterogeneous healthcare data sources.

Optimized large-scale federated systems: We plan to investigate various strategies for optimizing query processing, caching, and indexing mechanisms to foster the performance of FVKG and scalability in large-scale deployments.

Automated ontology evolution and adaptation: We plan to implement dynamic ontology evolution mechanisms to automatically update and refine domain ontologies in response to changes in healthcare standards and data sources.

Hybrid AI-driven data integration: We will explore the integration of hybrid AI models to automate data harmonization, anomaly detection, and predictive analytics within the federated healthcare environment.

Interoperability with emerging healthcare standards: We plan to ensure compatibility with evolving standards such as FHIR (<https://build.fhir.org/>), SNOMED CT (<https://www.snomed.org/> (accessed on 21 May 2025)), and HL7 (<https://www.hl7.org/fhir/overview.html> (accessed on 21 May 2025)) to improve cross-platform interoperability and regulatory compliance.

User-centric knowledge retrieval interfaces: We plan to develop intuitive and intelligent user interfaces that leverage natural language processing (NLP) and conversational agents to facilitate seamless interaction with FVKG-powered AI systems.

Limitations and Future Validation

In this section, we detail how the proposed FVKG framework will be validated in future work through:

Prototype Implementation: Deploying a proof-of-concept system using real-world datasets (especially clinical or biomedical data from real-world medical institutional data depending on the use case).

Usability Feedback: Involving domain experts to assess the practicality and expressiveness of the ontological models and their mapping in real data scenarios.

Comparative Analysis: Benchmarking against existing OBDA tooling ecosystems or integration frameworks to demonstrate added value regarding semantic richness and deployment efficiency.

Author Contributions: Data curation, A.A.F.; data gathering and investigation, A.A.F.; methodology, A.A.F.; project administration, S.G. and R.V.; supervision, A.G.; validation, A.A.F.; writing—original draft, A.A.F.; writing—review editing, A.A.F. and R.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

- Hilbert, M.; Lopez, P. The World's technological capacity to store, communicate, and compute. *Inf. Sci.* **2011**, *332*, 60–65. [CrossRef]
- Gu, Z.; Corcoglioniti, F.; Lanti, D.; Mosca, A.; Xiao, G.; Xiong, J.; Calvanese, D. A systematic overview of data federation systems. *Semant. Web* **2024**, *15*, 107–165. [CrossRef]
- Reinsel, D.; Gantz, J.; Rydning, J. *The Digitization of the World from Edge to Core*, International Data Corporation; Technical Report; International Data Corporation: Framingham, MA, USA, 2018.
- Fareedi, A.A.; Ismail, M.; Ghazawneh, A.; Ahmed, S.; Bukhari, S.A.C. Empowering health data fusion: A federated virtual knowledge graph approach leveraging the Ontop platform. In Proceedings of the 16th Mediterranean Conference on Information Systems (MCIS 2024) and the 24th Conference of the Portuguese Association for Information Systems (CAPSI 2024), Porto, Portugal, 3–5 October 2024; pp. 1–17.
- Hammond, W.E.; Bailey, C.; Boucher, P.; Spohr, M.; Whitaker, P. Connecting information to improve health. *Health Aff.* **2010**, *29*, 285–290. [CrossRef]
- Gosh, B.; Scott, J.E. A grounded theory investigation into data interoperability in healthcare. In Proceedings of the Pacific Asia Conference on Information Systems (PACIS), Ho Chi Minh City, Vietnam, 11–15 July 2012; pp. 1–14.
- Schwade, F.; Schubert, P. A semantic data lake for harmonizing data from cross-platform digital work spaces using ontology-based data access. In Proceedings of the Americas Conference on Information Systems (AMCIS), Virtual, 15–17 August 2020; pp. 1–10.
- Beyer, M.A.; Thoo, E.; Zaidi, E. *Magic Quadrant for Data Integration Tools*; Technical Report ID-G00340493; Gartner: Stamford, CT, USA, 2018.
- Vasiliev, D.A.; Ghiran, A.M.; Buchmann, R.A. Data federation for a project management solution through a GraphQL middleware. In Proceedings of the 29th International Conference on Information Systems Development (ISD2021), Valencia, Spain, 8–10 September 2021; pp. 1–12.
- GraphQL. GraphQL Specifications. 2018. Available online: <https://spec.graphql.org/June2018/> (accessed on 8 May 2024).
- Cygniak, R.; Wood, D.; Lanthaler, M. RDF 1.1 Concepts and Abstract Syntax. 2014. Available online: <https://www.w3.org/TR/rdf11-concepts/> (accessed on 8 May 2024).
- Hakimpour, F.; Geppert, A. Resolving semantic heterogeneity in schema integration: An ontology based approach. In Proceedings of the Formal Ontology in Information Systems (FOIS): Collected Papers from the Second International Conference, Ogunquit, ME, USA, 17–19 October 2001; pp. 297–308.
- Grangel-Gonzalez, I.; Losch, F.; Mehdi, M.A.U. Knowledge graphs for efficient integration and access for manufacturing data. In Proceedings of the 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 10–13 September 2020; pp. 1–8.
- Studer, R.; Benjamins, V.R.; Fensel, D. Knowledge engineering: Principles and methods. *Data Knowl. Eng.* **1998**, *25*, 161–197. [CrossRef]
- Gagnon, M. Ontology-based integration of data sources. In Proceedings of the 10th International Conference on Information Fusion, Québec, QC, Canada, 9–12 July 2007; pp. 1–8.
- Konstantinou, N.; Spanos, D.E.; Mitrou, N. Ontology and database mapping: A survey of current implementations and future directions. *J. Web Eng.* **2008**, *7*, 001–024.
- Xiao, G.; Calvanese, D.; Kontchakov, R.; Lembo, D.; Poggi, A.; Rosati, R.; Zakharyashev, M. Ontology-based data access: A survey. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm, Sweden, 13–19 July 2018; pp. 1–9.
- Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; Rodriguez-Muro, M.; Rosati, R.; Ruzzi, M.; Savo, D.F. The MASTRO system for ontology-based data access. *Semant. Web* **2011**, *2*, 43–53. [CrossRef]
- Nguyen, A.; Gardner, L.; Sheridan, D. Towards ontology-based design science research for knowledge accumulation and evolution. In Proceedings of the 52nd Hawaii International Conference on System Science, Maui, HI, USA, 8–11 January 2019; pp. 1–10.
- Song, F.; Zacharewicz, G.; Chen, D. An ontology-driven framework towards building enterprise semantic integration layer. *Adv. Eng. Inform.* **2013**, *27*, 38–50. [CrossRef]
- Ostrowski, L.; Helfert, M.; Gama, N. Ontology engineering step in design science research methodology: A technique to gather and reuse knowledge. *Behav. Inf. Technol.* **2014**, *33*, 443–451. [CrossRef]

22. Reiterer, E.; Venable, J.R. Ontological support for the use of design science research results. In Proceedings of the International Conference on Design Science Research in Information Systems and Technology (DESRIST), DESRIST 2016, St. John's, NL, Canada, 24–25 May 2016; pp. 75–82.
23. Reiterer, E.; Venable, J.R.; Reiners, T. Ontological representation of design science research publications. In Proceeding of the DESRIST, Dublin, Ireland, 20–22 May 2015; pp. 125–126.
24. Hevner, A.; March, S.T.; Park, J. Design science in information systems research. *MIS Qual.* **2004**, *28*, 75–105. [CrossRef]
25. Li, Z.; Raskin, V.; Ramani, K. A methodology of engineering ontology development for information retrieval. In Proceeding of the International Conference on Engineering Design, Paris, France, 28–31 July 2007; pp. 1–12.
26. Uschold, M.; King, M. Towards a methodology for building ontologies. In Proceedings of the IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, Edinburgh, UK, 19–20 August 1995; pp. 1–15.
27. Nguyen, A.; Gardner, L.; Sheridan, D. Building an ontology of learning analytics. In Proceeding of the Pacific Asia Conference on Information Systems (PACIS), Yokohama, Japan, 26–30 June 2018; pp. 1–16.
28. Raad, J.; Cruz, C. A Survey on ontology evaluation methods. In Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, SCITEPRESS—Science and Technology Publications, Lisbon, Portugal, 12–14 November 2015; pp. 179–186.
29. Xiao, G.; Ding, L.; Cogrel, B.; Calvanese, D. Virtual knowledge graphs: An overview of systems and use cases. *Data Intell.* **2019**, *1*, 201–223. [CrossRef]
30. Jayadianti, H.; Pinto, C.S. Solving problems of data heterogeneity, semantic heterogeneity and data inequality—An Approach using ontologies. In Proceedings of the Mediterranean Conference on Information Systems (MCIS), Guimarães, Portugal, 8–10 September 2012; pp. 1–14.
31. Torab-Miandoab, A.; Samad-Soltani, T.; Jodati, A.; Rezael-Hachesu, P. Interoperability of heterogeneous health information systems: A systematic literature review. *BMC Med. Inform. Decis. Mak.* **2023**, *23*, 18. [CrossRef] [PubMed]
32. De Mello, B.H.; Rigo, S.J.; da Costa, C.A.; Righi, R.d.R.R.; Donida, B.; Bez, M.R.; Schunke, L.C. Semantic interoperability in health records standards: A systematic literature review. *Health Technol.* **2022**, *12*, 255–272. [CrossRef] [PubMed]
33. Yuwen, S.; Yang, X. Research on the clinical terminology construction based on SNOMED. In Proceeding of the Seventh International Conference on Fuzzy Systems and Knowledge Discovery, Yantai, China, 10–12 August 2010; pp. 1–5.
34. Ben Hamouda, I.; Tantan, O.C.; Boughzala, I. Towards an ontological framework for knowledge sharing in healthcare systems. In Proceedings of the Pacific Asia (PACIS), Chiayi, Taiwan, 27 June–1 July 2016; pp. 1–9.
35. Brut, M.; Al-Kukhun, D.; Péninou, A.; Canut, M.-F.; Sèdes, F. Structuration et Accès au Dossier Patient Médical Personnel: Approche par ontologie et politique d'accès XACML. In Proceedings of the 1ère édition du Symposium sur l'Ingénierie de l'Information Médicale, SIIM 2011, Toulouse, France, 9–10 June 2011.
36. Poggi, A.; Lembo, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; Rosati, R. Linking data to ontologies. *J. Data Semant.* **2008**, *10*, 133–173.
37. Denodo Technologies. Denodo Fearless Data. 2024. Available online: <https://www.denodo.com/en> (accessed on 8 May 2024).
38. Dremio Technologies. Dremio: The Unified Lakehouse Platform for Self-Service Analytics and AI. 2024. Available online: <https://www.dremio.com/> (accessed on 8 May 2024).
39. Teiid Technologies. Teiid: Cloud-Native Data Virtualization. 2024. Available online: <https://teiid.io/> (accessed on 8 May 2024).
40. Dahlberg, T.; Lagstedt, A.; Nokka, T. How to address master data complexity in information systems development—A federative approach. In Proceedings of the European Conference of Information Systems (IS2018), Portsmouth, UK, 23–28 June 2018; pp. 1–16.
41. Fletcher, G.H.L.; Hidders, J.; Larriba-Pey, J.L. *Graph Data Management, Fundamental Issues and Recent Developments*, 1st ed.; Springer: Cham, Switzerland, 2018; pp. 1–32.
42. Borgida, A.; Brachman, R.J. Conceptual Modeling with Description Logics. In *The Description Logic Handbook: Theory, Implementation and Applications*; Cambridge University Press: Cambridge, UK, 2003.
43. Xiao, G.; Kontchakov, R.; Cogrel, B.; Calvanese, D.; Botoeva, E. Efficient handling of SPARQL OPTIONAL for OBDA. In Proceeding of the ISWC, Monterey, CA, USA, 8–12 October 2018; pp. 354–373.
44. Harris, S.; Seaborne, A. SPARQL 101 Query Language. 2013. Available online: <https://www.w3.org/TR/sparql11-query/> (accessed on 8 May 2024).
45. Guarino, N.; Welty, C.A. An overview of OntoClean. In *Handbook on Ontologies, International Handbooks on Information Systems*; Staab, S., Studer, R., Eds.; Springer: Berlin, Germany, 2009.
46. Bizer, C.; Cyganiak, R. D2RQ—Lessons learned. In Proceedings of the W3C Workshop on RDF Access to Relational Databases, W3C, Cambridge, MA, USA, 25–26 October 2007.
47. Priyatna, F.; Corcho, Ó.; Sequeda, J.F. Formalisation and experiences of R2RML based SPARQL to SQL query translation using Morph. In Proceedings of the 23rd International World Wide Web Conference (WWW), ACM, Seoul, Republic of Korea, 7–11 April 2014; pp. 479–490.

48. Calvanese, D.; Cogrel, B.; Komla-Ebri, S.; Kontchakov, R.; Lanti, D.; Rezk, M.; Rodriguez-Muro, M.; Xiao, G. Ontop: Answering SPARQL queries over relational databases. *Semant. Web* **2017**, *8*, 471–487. [CrossRef]
49. Oracle. Spatial Database. 2024. Available online: <https://www.oracle.com/database/spatial/> (accessed on 8 May 2024).
50. Stardog. Virtaul Graphs in Stardog. 2024. Available online: <https://www.stardog.com/blog/virtual-graphs-in-stardog-5> (accessed on 8 May 2024).
51. Sequeda, J.F.; Miranker, D.P. Ultrawrap: SPARQL execution on relational data. *J. Web Semant.* **2013**, *22*, 19–39. [CrossRef]
52. Das, S.; Sundara, S.; Cyganiak, R. R2RML: RDB to RDF Mapping Language. 2012. Available online: <https://www.w3.org/TR/r2rml/> (accessed on 8 May 2024).
53. Sicilia, A.; Nemirovski, G.; Nolle, Á. Map-On: A web-based editor for visual ontology mapping. *Semant. Web* **2017**, *8*, 969–980. [CrossRef]
54. Heyvaert, P.; Dimou, A.; De Meester, B.; Seymoens, T.; Herregodts, A.L.; Verborgh, R.; Schuurman, D.; Mannens, E. Specification and implementation of mapping rule visualization and editing: MapVOWL and the RMLEditor. *J. Web Semant.* **2018**, *49*, 31–50. [CrossRef]
55. RMLIO Technologies. Easily Generate High-Quality Knowledge Graphs with RML.io. 2024. Available online: <https://rml.io/> (accessed on 8 May 2024).
56. Blinkiewicz, M.; Bak, J. SQuARE: A visual approach for ontology-based data access. In Proceedings of the 6th Joint International Conference on Semantic Technology (JIST2016), Singapore, 2–4 November 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 47–55.
57. Wieringa, R.J. *Design Science Methodology for Information Systems and Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2014.
58. Simpert, E.; Tempich, C.D. Ontology engineering: A reality check. In Proceedings of the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, OTM Confederated International Conferences, Montpellier, France, 29 October–3 November 2006; pp. 836–854.
59. Muller, R.M.; Thoring, K. Understanding artifact knowledge in design science: Prototype and products as knowledge repositories. In Proceedings of the Seventeenth Americas Conference on Information Systems (ACIS), Detroit, MI, USA, 4–8 August 2011; pp. 1–10.
60. Fareedi, A.A.; Tarasov, V. Modelling of the ward round process in a healthcare unit. In Proceedings of the Practice of Enterprise Modeling: 4th IFIP WG 8.1 Working Conference, PoEM, Oslo, Norway, 2–3 November 2011; pp. 223–237.
61. Prat, N.; Comyn-Wattiau, I.; Akoka, J. Artifact evaluation in information systems design science research—A holistic view. In Proceedings of the Pacific Asia Conference on Information Systems (PACIS 2014), Chengdu, China, 24–28 June 2014; pp. 1–16.
62. Gómez-Pérez, A.; Fernández-López, M.; Corcho, O. *Ontological Engineering*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 1–412.
63. Fernández-López, M.; Gómez-Pérez, A.; Juristo, N. Methontology: From Ontological Art Towards Ontological Engineering. In Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series, Stanford University, EEUU, Stanford, CA, USA, 24–26 March 1997.
64. Noy, N.; McGuinness, D.L. *Ontology Development 101*; Knowledge Systems Laboratory, Stanford University: Stanford, CA, USA, 2001.
65. Fareedi, A.A. Ontology-Based Model for the “Ward-Round” Process in Healthcare (OMWRP). Master’s Thesis, Tekniska Hogskolan I Jonkoping University, Jönköping, Sweden, 2010.
66. Ulianova, S. Cardiovascular Disease Dataset. 2019. Available online: <https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset/code> (accessed on 8 May 2024).
67. Mogotlane, D.K.; Fonou-Dombou, J.V. Automatic conversion of relational databases into ontologies: A comparative analysis of protege plug-ins performances. *Int. J. Web Semant. Technol.* **2016**, *7*, 21–40. [CrossRef]
68. Telnarova, Z. Relational database as a source of ontology creation. In Proceedings of the International Multi-Conference on Computer Science and Information Technology, Wisla, Poland, 18–20 October 2010; pp. 135–139.
69. Brickley, D.; Guha, R.V. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, World Wide Web Consortium. February 2004. Available online: <https://www.w3.org/TR/rdf-schema/> (accessed on 27 November 2024).
70. Xiao, G.; Rezk, M.; Rodriguez-Muro, M.; Calvanese, D. Rules and ontology based data access. In Proceedings of the 8th International Conference on Web Reasoning and Rule Systems (RR), Volume 8741 of Lecture Notes in Computer Science, Athens, Greece, 15–17 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 157–172.
71. Tsangaris, M.M.; Kakaletis, G.; Kllapi, H.; Papanikos, G.; Pentaris, F.; Polydoras, P.; Sitaridi, E.; Stoumpos, V.; Ioannidis, Y.E. Dataflow processing and optimization on grid and cloud infrastructures. *Bull. IEEE Comput. Soc. Tech. Comm. Data Eng.* **2009**, *32*, 67–74.

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.