Semantic Representation of Road Infrastructure Information

tianjiao zhao
Marquette University

SEMANTIC REPRESENTATION OF ROAD INFRASTRUCTURE INFORMATION

by

Tianjiao Zhao

A Dissertation submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy

Milwaukee, Wisconsin

August 2022

ABSTRACT

SEMANTIC REPRESENTATION OF ROAD INFRASTRUCTURE INFORMATION

Tianjiao Zhao

Marquette University, 2022

A multi-sectoral collaboration completes a successful transportation infrastructure project. The cooperation involves designers, contractors, operators, users, government agencies, and maintenance staffs. Throughout the project's life cycle, a huge amount of data is generated and stored in various sectors. Therefore, an efficient information cross-sector exchange approach is necessary. Additionally, the World Wide Web is ubiquitous and enmeshed with multiple business processes. Therefore, it is imperative to represent business information in a format that improves information exchange as well as automated processing of business data. Ideally, road data scattered across different information sources, such as design software, geographic information systems (GIS), cost estimating software, and maintenance and repair databases can be shared across the Internet. However, the reality is the information in each transportation sector is created and updated separately. Moreover, the project's data is stored in various formats, such as text document, pdf, XML, and relational database. Different systems, file formats, technologies, and semantics hinder the smooth data exchange and systems interoperability throughout the road project's lifecycle (van Nederveen et al. 2015). Therefore, a new data modeling approach is required to facilitate automatic data integration.

This dissertation proposes a novel approach to road infrastructure projects using the Semantic Web technology. The SW technology provides a modeling framework for representing various road data sources, such as design documents and GIS. A vocabulary is developed in this study to represent all the information involved in the modeling framework. The data structured by SW technology creates a knowledge base. This knowledge base can take advantage of machine processing, facilitate interoperability among distributed systems, and allow domain users to loosely and on-demand integrate several geographically, organizationally, or temporally distributed sources of information. This extendable data model enables domain engineers to complete a domain knowledge base and keep it up to date through the project's lifecycle independently for each road-related domain. This study focuses on streamlining the integration of distributed road infrastructure information provided by road designers, estimators, schedulers, and GIS. The information stored in knowledge bases can be queried with Simple Protocol and Resource Description Framework Query Language (SPARQL) endpoints or semantic web services

ACKNOWLEDGMENT

Tianjiao Zhao

First and foremost, I am extremely grateful to my advisors, Dr. Saeed Karshenas, Dr. Yong Bai, and Dr. Sheikh Iqbal Ahamed, without whom this would not have been possible. I would like to thank them for their invaluable advice, continuous support, and patience during my Ph.D. degree. I owe my deepest gratitude to them all.

Among all the individuals who have contributed to completing this work, I would like to sincerely thank my advisor, Dr. Saeed Karshenas, for his tutelage, understanding, patience, and encouragement. His immense knowledge and ample experience benefited me profoundly throughout my academic research. Particularly, when working on the Road Shared ontology, I struggled to structure various concepts due to my lack of practical experience in engineering projects. During that time, his ideas, encouragement, and considerable patience were essential for me to complete this part of the work finally. Besides that, I would like to thank him for always being there for me throughout my Ph.D. study at Marquette.

I would also like to thank Dr. Yong Bai for his valuable suggestions and support for my learning in the field of scheduling. The lab teaching section was a valuable experience for me, and my teaching assistant task could not have been completed without his help.

My appreciation also goes to my family and friends for their tremendous understanding and encouragement that supported me throughout my studies.

Finally, I would like to express my gratitude to Marquette University for providing me with teaching assistant work and an excellent atmosphere for my research work. Without it, it would have been impossible for me to complete my study in the past few years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

As a critical national asset, road infrastructure provides the foundation for public transport and logistics. A successful road infrastructure project is completed and operated by a multi-sectoral collaboration. Throughout its entire lifecycle, the main participants include designers, contractors, suppliers, operators, users, government agencies, and maintenance staff. Figure 1 provides a general view of the main stages a road project goes through in its' lifecycle.

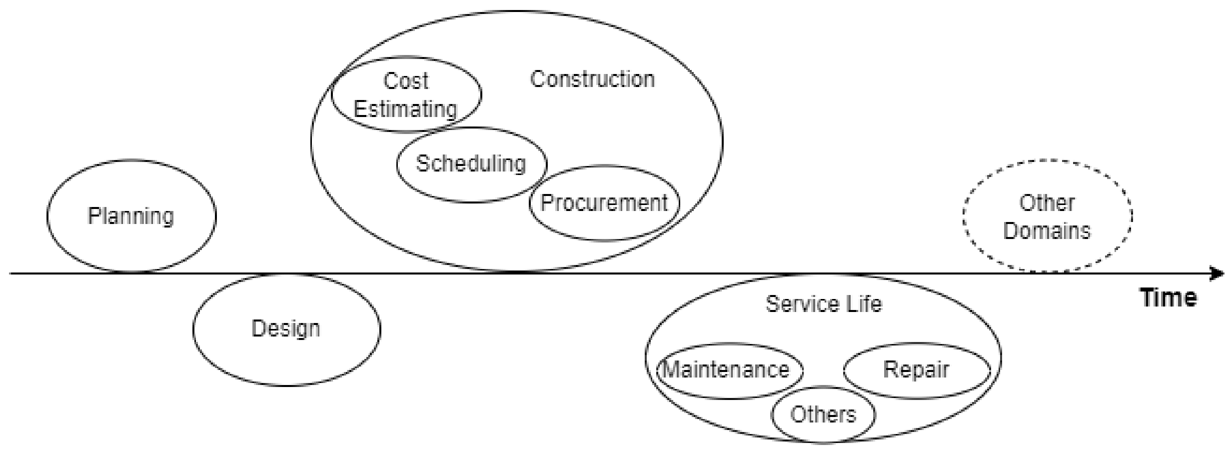


Figure 1 Main stages throughout the lifecycle of a road project

Each project stage involves one or more road-related domains. Each domain generates domain-specific information that the others could require. For example, a new maintenance project requires data from the design domain, cost information from the construction domain, traffic information from the operation domain, and maintenance history from the maintenance domain. Figure 2 presents the potential information exchanging demands among these sectors. Considering the high demand for information sharing among domains, an efficient data exchange and integration approach among domains is necessary.

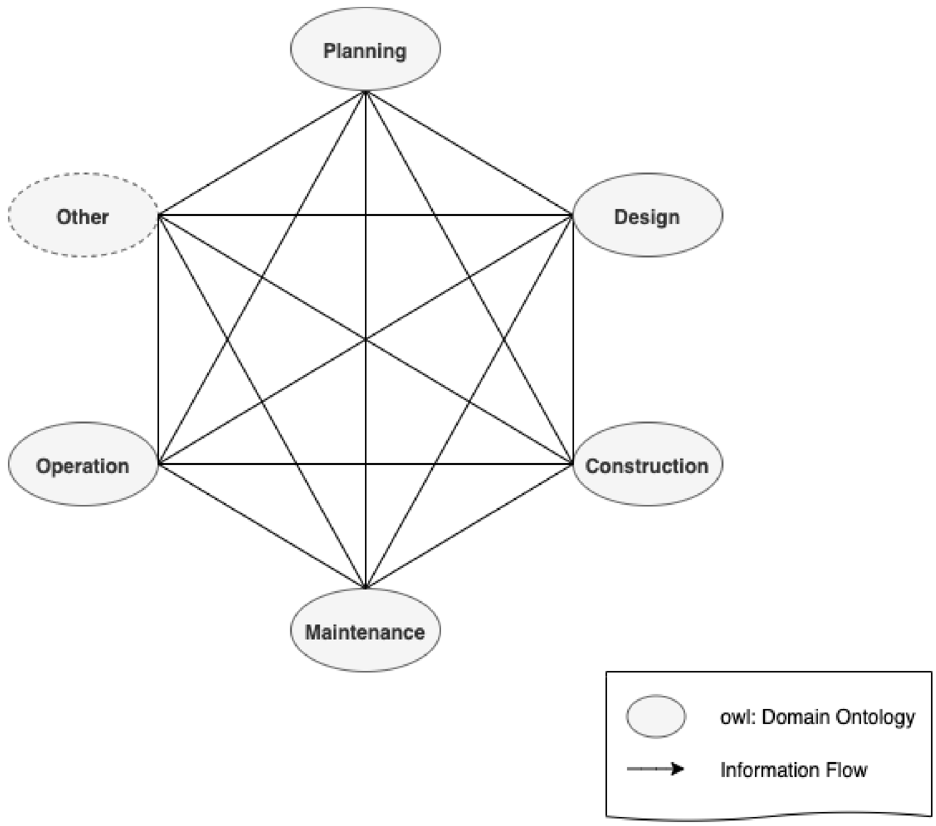


Figure 2 A general view of information exchanging demands among various domains

Additionally, the application of the World Wide Web is currently ubiquitous and entangled with various business processes. Road infrastructure engineering is no exception. Smart transportation and smart logistics have emerged as new trends, with greater emphasis on digitizing road infrastructure information. As the volume of freight and traffic data grows, more and more road infrastructure-related sectors, including highway transportation, logistics distribution, urban planning, and designing, have adopted the digitalized work mode. Both industry practitioners and users have increasingly gotten accustomed to using electronic devices to create, gather, process, and share information. Subsequently, the difficulty in processing massive amounts of engineering data has turned into the main challenge for many construction companies to carry out refined management. When a construction company cannot quickly and accurately obtain data to support resource planning, they can only rely on empirical decision-making, which can easily lead to errors in decision-making and cause losses. Therefore, it is imperative to represent road infrastructure information in a machine-processable format to aid the automated processing of the massive data.

However, the road infrastructure data is generated and updated separately by each domain involved in a project. For example, design documents are governed by designers, the local transportation sector manages sensor-generated data, and the maintenance and repair data are stored in the maintenance sector's databases. Additionally, given the different storage methods, the data is often stored in various formats, such as text document, pdf, XML, and table in a relational database. Throughout the lifecycle of a road infrastructure project, different systems, file formats, technologies, and semantics impede the smooth exchange of information and system interoperability (van Nederveen et al. 2015). As a result, cross-domain data acquisition is time-consuming and labor-intensive, and the timeliness of the data acquired is difficult to ensure.

Subsequently, the current road infrastructure projects require new technology to streamline the integration of road infrastructure lifecycle information across distributed sources. To address this need, this study uses the Semantic Web technology as a modeling framework for representing various sources of transportation feature information. It defines data in a format streamlining machine-to-machine data exchange and automated processing. Consequently, engineers from diverse road infrastructure-related domains can access the most up-to-date data from other domains and independently update their own knowledge base. Furthermore, with the aid of computers, human involvement will be considerably reduced, particularly in repetitive procedures.

To better prepare for the novel technique proposed in this study, the following sections give some basic knowledge on road data (Section 1.1) and Semantic Web technologies (Section 1.2). The organization of this paper will be provided in Section 1.3.

1.1 Road Data

1.1.1 Type of Road Data

Each domain participating in a road project generates domain-specific road data during one or more project stages. There are several examples.

1. The design domain defines the road elements and their properties, such as cross-section elements (assemblies) and the geometry features of the road (road alignment).
2. The construction domain can be further divided into several specialized groups, such as scheduling and estimating. The scheduling domain determines the road elements' construction sequence and schedules. The estimating domain assigns resource costs as well as other related construction expenses.
3. The pavement condition information is collected and processed for maintenance planning in the maintenance domain.

Each road domain creates specific domain data in its own taxonomy following the domain convention. All these types of domain data together describe the properties of the road elements, actors involved, resources needed, road location, and changes made to the road throughout its lifecycle.

1.1.2 Dynamic Segmentation

The road is a linear, repetitive transportation feature. Throughout the length of a road, its properties keep changing. A road segment is a section of a road where at least one property does not change. That is, each segment has a specific uniform property along the segment length. When different properties are used as the segmentation criteria, they form different road segments. For example, the segmentation for the number of lanes is different from that of the pavement condition. No matter the segmentation criteria used, the collection of segments constitutes the road.

To handle a wide range of segmentation criteria, dynamic segmentation (DynSeg) emerged (Cadkin 2002). DynSeg is a method for associating multiple sets of attributes (properties) with any portion of a linear feature (Cadkin 2002). For example, properties such as the number of lanes, pavement condition, and pavement material can be independently attached to the same portion of a road simultaneously. See Figure 3 for an example. DynSeg facilitates the representation of properties in combination and improves the efficiency of roads' multi-property analysis.

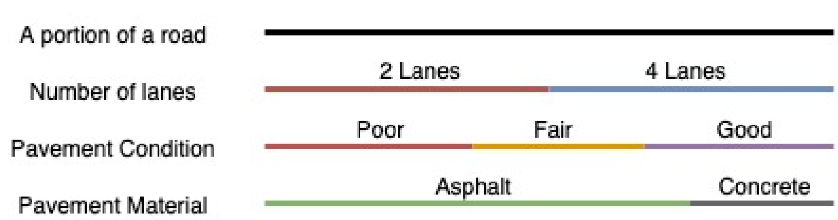


Figure 3 An Example of Dynamic Segmentation

1.1.3 Time

Time is a significant consideration interweaving with road projects. It contains temporal concepts and describes the temporal properties of resources in a road project (W3C 2017). The main temporal concepts on the top level are time position, time duration, referencing systems, and time zone. Time position can be used to record the start date of a construction activity, and time duration can be applied to record the duration of it. Before the time position and time duration can be defined, the referencing system information must be specified. Examples of the referencing systems include the temporal reference system (TRS), which specifies the calendar used, such as the Gregorian, Unix-time, and geologic time. Another concept on the first level is time zone, which specifies the amount by which the local time is offset from the Universal time code (UTC).

1.1.4 Location

Accurately locating the road and its related properties are significant for all the domains involved in the road project to apply domain-specific properties to the road. Therefore, the location information keeps being reused by each domain throughout the road's lifecycle.

Different referencing systems with different representations usually handle the location information. The Spatial Referencing System (SRS) and the Linear Referencing System (LRS) are the two most widely used location referencing systems. The SRS is the real-world coordinate system, which can describe the location information of all sorts of features, like a road, while the LRS uses a known point as a reference to find a specific position along the road. The known point is defined in a known datum (NCHRP 1974). The following sections provide a brief description of the SRS and LRS.

1.1.4.1 SRS

SRS mainly includes three sub-categories of the coordinate systems, i.e., the Geographic coordinate system (GCS), the Projection coordinate system (PCS)[1], and the Vertical coordinate system (VCS)[2]. In SRS, a point is referenced by its horizontal and vertical coordinates. The horizontal coordinate can be taken from the GCS as latitude and longitude values or from the PCS as x- and y- or Easting and Northing coordinate values. The vertical value can be taken from the VCS as a z-, height, or depth coordinate value. In the geographic software like ArcGIS, PCS and GCS can easily be converted to each other through a projection conversion tool and related toolkits, which benefits the cross-processing of geographic information in different coordinate systems. One thing worth mentioning is that the VCS could not be defined on a dataset without GCS or PCS (ESRI 2006). Some examples of the GCS are WGS84 and Beijing54 (based on different coordinate centers); while Universal Transverse Mercator (UTM) and Gauss Kruger (based on different projection methods) are examples of the PCS.

1.1.4.2 LRS

The linear referencing system stores and maintains location information for road elements or events that occur within a transportation network (Miller and Shaw 2001),

---

[1] In the United States, the State Plane Coordinate System (SPCS) is commonly used as the PCS (Miller and Shaw 2001).
[2] Commonly used vertical-specific datums in North America are the National Geodetic Vertical Datum of 1929 (NGVD29) and the North American Vertical Datum of 1988 (NAVD88).

like road widening and pavement repairs. It is used for many reasons; for example, (1) it requires fewer data to record locations within linear features, and (2) it is the basis of road segmentation (ESRI 2007). Several linear referencing methods (LRM) are used to express LRS location (Scarponcini 2005), for example:

1. Absolute LRM measures from the start of the linear element.
2. Relative LRM measures from the closest marker.
3. Interpolative LRM uses computers to interpolate one unknown location between two known locations along a route, representing the unknown location as a proportion between them.

The main referencing methods are listed in Table 1.

Table 1 Main types of referencing method

LRM Type	Referencing Method
Absolute LRMs	Mile Point; Kilo Point
Relative LRMs	Mile Post; Kilo Post
Interpolative LRMs	Percent (0-100); Normalized
Others	Addressing

Different LRMs offer users various options; however, different expression formats make it difficult to combine information. To integrate the information expressed by multiple LRMs, the National Cooperative Highway Research Project (NCHPR) has made significant progress by proposing a conceptual model, the NCHRP Project 20-27 model (NCHRP 1974; Vonderohe et al. 1997), to standardize linear referencing terminology (Scarponcini 2005). According to the NCHRP Project 20-27 model, the anchor points (known locations along the road) and anchor sections (physical roadways of known length) constitute the linear datum. Nodes and links that are located based on the linear datum constitute topological networks. On top of networks, LRMs are used to measure along linear elements. Finally, along the linear elements represented by LRMs, an event can be located by specifying its distance expression. Therefore, the formalized location expression includes three parameters: referencing method (implied by referencing point used), linear element, and distance expression. Table 3 shows one example for an absolute LRM as Point 1 and an example for relative LRM as Point 2.

Table 2 Examples of location expression with LRMs

	Location Expression		
	Referencing Method	Linear Element	Distance Expression
Point 1	mile point	Rd01	2
Point 2	reference post	Rd01	2+.50

One thing worth mentioning is that, in practice scenarios, many current modified models based on NCHRP Project 20-27 model do not strictly follow the four-level construct. They may skip or combine one or two levels as needed. For example, Minnesota DOT cuts the network level for a topologically complete linear datum (Ross et al. 2002).

Additionally, some researchers have attempted to expand and refine the NCHRP Project 20-27 model. There are two traditional representative models: (1) to facilitate the sharing of geographic information systems used by transportation agencies (GIS-T), Dueker and Butler proposed an enterprise-level data model for the sharing of digital road map databases within and among transportation organizations (Dueker and Butler 1997); and (2) Koncz and Adams stepped further and integrated temporal dimension data into the spatial dimension data to create the Multi-Dimensional Multi-Modal Transportation Location Referencing System (MDLRS) model (Koncz and Adams 2002).

The above data models provide a clear view of the relationships among transportation-feature-related concepts. However, both the GIS-T Enterprise model and the MDLRS model only support specific kinds of information's cross-domain integration, and their machine-processing capabilities are limited.

1.2 Semantic Web Technologies

The Semantic Web is "a web of data that can be processed directly and indirectly by machines" (Berners-Lee et al. 2001). It enables domain knowledge to be explicitly defined, captured, and formalized (Motik et al. 2005). Inherited from the Semantic Web, Semantic Web technologies will significantly aid in the comprehension of concepts across domains, data processing automation, and different systems' interoperability. These benefits coincide with our vision of sharing road information across domains.

With the Semantic Web technologies, data is represented in a graph form, and scattered sources of information can interoperate over the Internet. The data model behind the Semantic Web is the Resource Description Framework (RDF). RDF represents information in a labeled graph form by using subject-predicate-object triples (see Table 3 and Figure 4 for an example of triples). Since the triples can only be represented in one way, the information expressed by triples can be easily understood and processed by computers. Additionally, in RDF graphs, each node can be merged directly with another node sharing the same web identifiers (Allemang and Hendler 2011). The web identifiers used in the Semantic Web are Uniform Resource Identifiers (URI) (W3C 2005), which can globally identify the data items and their interrelations. The RDF and data URIs form a "global information space" for the interlinked data (Cyganiak and Jentzsch 2010), with which there is no need for programming to integrate data physically. It also allows a certain degree of variability of viewpoints from different sources (Allemang and Hendler 2011). Figure 5 shows an example of merging two different domain models in RDF.

Table 3 Triples Examples

Subject	Predicate	Object
Road01	hasID	Rd01
Road01	hasSegment	Seg01

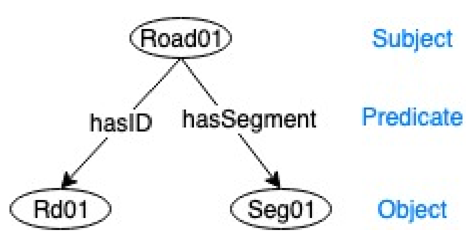


Figure 4 Example graph of triples

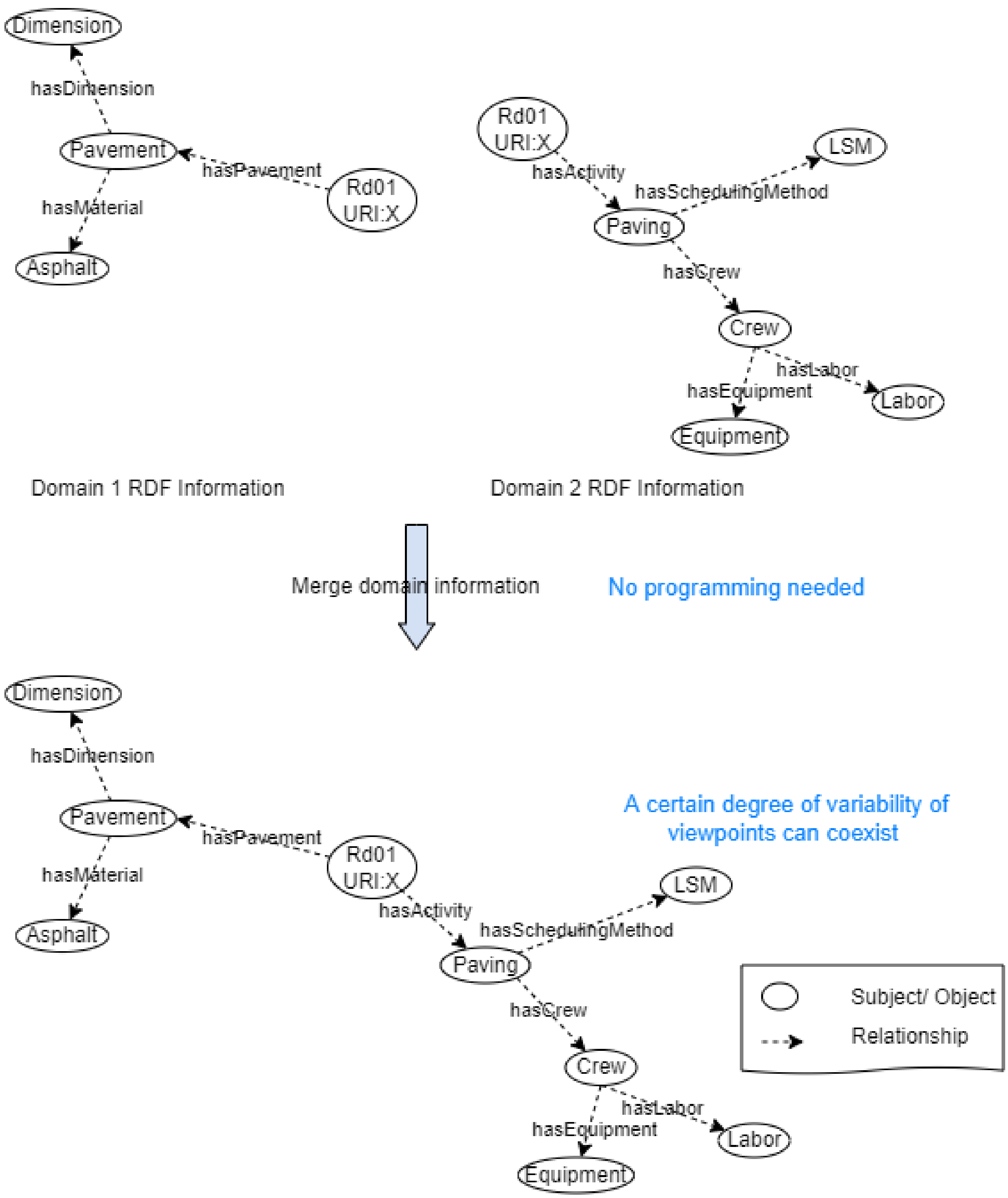


Figure 5 An example of merging two different domains models in RDF

Based on RDF, Semantic Web also provides several modeling languages with higher expressivity. Examples include RDFs, which describe the commonality and variability of basic concepts; OWL, which allows detailed constraints between classes, entities, and properties; and OWL 2, which adds language primitives to support the richer expressiveness required (Allemang and Hendler 2011). These languages may be represented in syntaxes such as Turtle, RDFa, RDF/XML, and N-Triples.

To present and organize domain knowledge, the Semantic Web uses ontologies. According to Motik et al., "ontology is an explicit and formal specification of a conceptualization" (2005). It defines the domain concepts and the relationships among them based on the consensus of the domain experts in a hierarchical form. The ontology is also created using any of the languages mentioned above.

There are tools that can be used to develop an ontology. Ontology editors are widely used for developing ontologies, such as OntoEdit (Sure 2003) and Protégé (Stanford University 2015). Protégé is widely used because it is freely available, it supports RDF/OWL, it is a pluggable system, and it contains many sample ontologies.

1.3 Organization of Dissertation

This dissertation is generally organized into 7 chapters. Following the present introductory chapter, Chapter 2 will analyze the current status of data storage and the challenges of data integration from distributed sources.

Chapters 3, 4, and 5 define the required road-related ontologies, the methodologies for creating them, and their corresponding knowledge bases developed.

Chapter 6 explores application scenarios for the data models developed.

Chapter 7 summarizes and concludes the dissertation. Recommendations for future work will also be given in this chapter.

CHAPTER 2: INTEGRATING DISTRIBUTED SOURCES OF ROAD INFRASTRUCTURE DATA

Road data come from all the domains involved in a road project throughout the road's lifespan. It could be a 3D road model developed by designers, a cost estimating spreadsheet produced by estimators, work item information generated by contractors, or a road surveillance video provided by the Department of Transportation (DOT). Each domain uses a specific set of data storage and processing method, either according to its own needs or following traditional conventions. These diverse types of data generated in heterogeneous systems are usually stored in different formats. Various data formats do make information exchange across domains difficult.

Additionally, the problem is more complicated because the definition, naming, and classification of the same concept in each field may differ. For example, in the cost estimating domain, an assembly refers to the combination of all necessary workitems to complete a unit of work, while in the design domain, an assembly refers to the collection of elements in a road cross-section such as lanes, shoulders, and ditches. The variations in terminology and semantics in various road knowledge domains make information integration difficult.

The following sections of this chapter mainly discuss: (1) the status quo of the primary data storage methods in different road-related domains and their limitations (Section 2.1, Section 2.2, and Section 2.3), (2) the primary data exchange technologies and its limitations (Section 2.4), and (3) the current application of the Semantic Web technology in road information modeling and a review of the related literature (Section

2.5).

2.1 Road Design Data Storage Methods

Road design data is vital to departments responsible for roadway construction, operation, maintenance, and asset management (Maier et al. 2017). It primarily outlines the geometric shape of the road model, such as the centerline and cross-section structure, as well as the materials to be utilized. This data is typically diverse and involves a considerable number of parameters. According to respective conventions, various locations, organizations, and software categorize and name road elements differently. In the design program Open Road, for example, the basic structure blocks exhibited in the cross-section view are referred to as templates, but in Civil 3D, they are called assemblies. The difference in semantics makes the integration of road design data challenging.

Additionally, road design data can be collected from various sources, including 2D drawings, 3D road models, design reports, and project documents. As a result, road design data can be recorded in a variety of formats, such as drawing, ASCII point file, Geography Markup Language (GML), and text (Autodesk 2021). Integrating data from disparate forms is often tricky. Furthermore, rather than being published online, road data, particularly the Road model, is typically stored locally in a Computer-Aided Design and Drafting (CADD) system (Maier et al. 2017). CADD is a locally installed computer program used to create road plans. AutoCAD, Civil 3D, MX Road / Open Road, and CARD/1 are all popular examples of the CADD developed for road projects. Once the design is finished, the data is typically exported in the default drawing format and saved on the user's computer, making it difficult to be integrated with other road data over the

Internet.

2.2 Location Data Stored in Relational Databases

Location data, also known as spatial data, is usually stored in spatial databases, which deal with storing, indexing, analyzing, and querying the spatial data (Mamoulis 2011). There are several methods for storing geographical data, including relational databases and non-relational databases. Oracle and Microsoft SQL Server are well-known examples of relational databases, while MongoDB is an example of a non-relational database. However, the storage technique and architecture differ depending on the data type. Vector data, for example, employs points, lines, and polygons (areas) to represent real-world features in maps, whereas raster data is composed of pixels (also referred to as grid cells) (Zeiler 1999). The integration of road data is a challenge due to massive data volumes and disparate storage systems and formats.

Additionally, relational databases are historically not designed for integrating with other systems (Reed 2006). The relational database has certain inherent disadvantages in terms of information integration. First, it is not published on the internet. As a result, several levels of access permissions are required. A relational database, on the other hand, is a sort of database that organizes data into tables. Once the data model and data relationship are established, it becomes difficult to modify, which some researchers refer to as the model rigidity of relational databases (Bergman 2009). Furthermore, some highly complicated queries are required to establish links across databases to integrate them (Kuchibhotla et al. 2009; Alexander 2013), which considerably increases the difficulty of integrating relational databases. As a result, integrating location information stored in relational databases remains a challenging task.

The difficulties in integrating the location information stored in a database stem not only from the relational database's inherent disadvantage in integrating data but also from the features of the location data. The present general relational database management systems (RDBMS) are ineffective when dealing with unstructured and semi-structured data, such as GIS data (Amirian et al. 2013). Various solutions have emerged to improve the database management systems (DBMS) ability to manage spatial data: MapInfo's SpatialWare, ESRI's ArcSDE, Oracle's Oracle Spatial, IBM's DB2 Spatial Extender, and Informix's Spatial DataBlade. As a leading partner in the GIS field, ESRI further developed the Geodatabase. The Geodatabase, which is designed for spatial data storage, is a collection of diverse types of geographic datasets stored in a common file system folder (ESRI 2008). Many researchers, however, discovered Geodatabases' limitations: (1) they lack sufficient 3D data modeling and data processing tools. Processing surface and volume models will necessitate the creation of new 3D geodatabases, not to mention 4D models that include temporal characteristics (Breunig and Zlatanova 2011); and (2) it is difficult to integrate spatial and non-spatial data, and spatial mapping is required to convert non-spatial data into spatial data during conversion (Egenhofer, M. 1994). As a result, achieving data exchange and interoperability within or between Geodatabases remains difficult.

2.3 Construction Data Storage

The construction domain is divided into various subdomains, including scheduling, estimating, and procurement domains. This study only focuses on scheduling and cost estimating subdomains. Both subdomains use various data storage methods, such as saving data in text documents, word table form, Excel spreadsheet, or scheduling/estimating software files (such as Oracle Primavera P6 and Sage Timberline). Domain engineers break down the project into activities, assign the resources needed for each activity, and then manually enter the data into a table form or software. Professional scheduling/estimating software can help with resource quantity calculations. However, due to the sheer vast volumes of data, this manual approach makes project management labor-intensive and time-consuming. Furthermore, piecing together the data stored in multiple software applications typically necessitates additional proprietary applications (Akinyemi et al. 2018).

2.4 Data Exchange Method - Extensible Markup Language (XML)

Standardized data exchange techniques will result in more normalized data interoperability workflows, reducing the possibility of human mistakes (Maier et al. 2017). Extensible Markup Language (XML), ISO 10303-28 (STEP-XML), Construction IT Alliance eXchange (CITAX), Construction Operations Building Information Exchange (COBie), and CityGML are some examples of AEC-related typical representations (Niknam 2015). Among these approaches, XML, as a versatile text format, is extensively used and extended for data sharing over the Internet (W3C 2016), particularly for transportation data (Ziering, 2007). It offers the mark-up file format, which serves as the foundation for schemas such as LandXML, TransXML, InfraGML, ifcXML, and AecXML (Maier et al. 2017). LandXML (LandXML 2017) is the most significant format for roads. It covers road concepts such as surface, point, alignment, cross-sections, design speeds, and pipes and structures (Lefler 2010).

However, integrating information from XML files has several hurdles and constraints. First, because XML files lack semantics and merely consist of data encoding and parsing, semantic heterogeneity is one of the significant difficulties in integrating XML file information (Chung and Mah 1995).

Second, because XML allows for the expression of data in a variety of ways (Sequeda 2012; Berners-Lee 1998; Niknam 2015), reading an XML file necessitates the usage of specialized programming code (Cambridge 2015). As a result, this approach complicates information exchange by involving a large amount of human effort and time.

Third, merging XML data from two sources necessitates copying data from both sources into a new document (Niknam 2015). That is, each XML file integration procedure generates a new integrated file. As a result, not only will the number of files expand rapidly as the number of information integrations grows, but data duplication will also occur.

Furthermore, while many software products now support the XML file format, a considerable amount of data is still stored in relational databases. Integrating data between XML files and relational databases is more complicated and necessitates using specialist data interaction tools such as SAX, DOM, and Oracle XSU (Vittori 2001).

Some researchers sought to solve the data exchange issues by introducing additional mediators and wrappers. The Transportation Extensible Markup Language (TransXML) framework was developed in order to enhance data interchange among participants in infrastructure-related projects (Ziering et al. 2007). It defines schemas for data serialization in a variety of business applications. Some researchers have attempted to employ a variety of algorithms, such as libSyD, to map the data source schema to the TransXML schema (Collins et al. 2002).

Overall, the XML file format indeed remarkably facilitates data exchange on the

World Wide Web. However, integrating information across XML files and between XML files and relational databases is still challenging.

2.5 Data Modeling and Storage with The Semantic Web Technologies

So far, researchers have developed several solutions to facilitate information sharing and interoperability. For example, various mediators, such as data distribution service (DDS) (Yim et al. 2017), have been created to enable information exchange between applications that allow other applications to connect directly with them (Nesi et al. 2016). However, most of these methods only focus on information exchange within a specific domain. Information exchange across domains requires an additional level of significant coordination between the domains (Costin 2016). Additionally, the process of understanding and mapping data created from other sources is a labor-intensive, costly, and time-consuming process. Therefore, it is also important to format the data in a machine-readable manner to streamline the information integration process.

2.5.1 Data Modeling Methods with Semantic Web Technologies

An ontology is a standard paradigm in computer and information sciences that consists of an agreed-upon glossary and the constraints that exist amongst them (Keet 2018). It is developed for researchers who need to share information in a domain (Noy and McGuinness 2001). An ontology categorizes the concepts in a knowledge domain into representational primitives, typically including classes, their properties, and class relationships (Gruber 2001). Three methods can be used to integrate information across ontologies:

1. Experts develop a complete single ontology that covers all knowledge

domains involved in a project.

2. Each domain independently develops its own ontology, and then domain experts map them to each other (Segaran et al. 2009).
3. Each domain develops its own ontology by extending a shared (foundation) ontology (Fisher et al. 2011).

In the first approach, all the related domains are considered as a whole and a complete single ontology is developed for a project. Considering the vast amount of knowledge from various infrastructure subdomains and their diversity (like dynamic segmentation based on different criteria in different domains), a road ontology will ultimately be too big and too complex to implement and maintain. Therefore, the consensus is that no single model can fully encompass all knowledge in a given domain of interest (Gruber 1995). Additionally, given the fact that there is continuous variability of road-related information (such as the ever-deteriorating pavement and the constantly emerging new technologies and devices), flexibility and scalability are important factors for a road ontology. Otherwise, it will be challenging to maintain such a complex system for rewriting and modifying the ontology during every update. Therefore, a single ontology is clearly not practical because of its size.

In the second approach, each domain develops a domain ontology entirely according to its own criteria. It remains independent when maintained or updated. However, in this approach, sharing information among the various domains requires mapping or alignment of domain ontologies, which is the process of discovering correspondences between concepts in two distinct domain ontologies (Segaran et al. 2009). That means each slight difference must be dealt with via relationships to tell the computer how to map between them. If the difference is simply about naming, it is easy to declare whether they are the same or not via assertions. However, if they have different classifications, mapping the two ontologies could be a big problem. Therefore, this second approach can be challenging to integrate because of the large number of mapping efforts required.

The third method is a hybrid of the first and the second methods. It requires a shared ontology to provide a common vocabulary for concepts maintained in different domain ontologies (Fisher et al. 2011). Then the domain ontologies can be developed by extending the shared ontology. This architecture gives the third method the advantages of both the first and second methods. First, the shared ontology delegates unique concepts defining tasks to each separate domain. Unlike the first method, the shared ontology is neither cumbersome, complicated, nor difficult to understand, and it does not need to be frequently modified for minor changes. These benefits significantly reduce the difficulty of maintaining the model while adding flexibility simultaneously. Secondly, the concepts are made consistently by confining the domain to a set of common concepts. This method allows each domain to flexibly develop its own domain ontology without being affected by unexpected data mapping problems like the second method (Antoniou et al. 2012). Therefore, the third method overcomes the problems in the first two methods, and it is the method used in this study.

To create ontologies, several methods have been put forward, such as Grüninger & Fox (Grüninger and Fox 1995), Gruber (Gruber 1995), Uschold and King (Uschold and King 1995), KACTUS (Schreiber et al. 1995), METHONTOLOGY (Fernández-López et al. 1997), Noy and McGuinness (Noy and McGuinness 2001), DILIGENT

(Pinto et al. 2004), On-To-Knowledge (Sure et al. 2004), and NeOn (Suárez-Figueroa et al. 2011; Suárez-Figueroa et al. 2012) methodologies.

This study adopts the NeOn methodology. Instead of prescribing a rigid workflow, NeOn methodology is a scenario-based methodology that emphasizes the reuse of ontological and non-ontological resources, ontology re-engineering and merging, and taking collaboration and dynamism into account. (Suárez-Figueroa et al. 2011). In order to answer a general problem, the NeOn methodology identifies a set of nine flexible scenarios for the problem. These scenarios allow for collaboratively developing ontologies and ontology networks in distributed environments. Every scenario is broken down into different processes and activities (Suárez-Figueroa 2012). The following provides a brief explanation of the 9 NeOn methodology scenarios.

Scenario 1: defines the ontology development specifications, purpose, scope, and the implementation language used for the ontology. In this study, Protégé (Stanford University 2015) is used as the ontology editor and Resource Description Framework (RDF) (W3C 2014) and Web Ontology Language (OWL) (W3C Standard 2015) are used as the ontology implementation languages.

Scenario 2: requires the reusing and reengineering of non-ontological resources. Non-ontological resources usually refer to the published documents in a domain. For example, this study uses non-ontological resources such as AASHTO specifications (AASHTO 2011) and Construction Scheduling Manual codes (NJDOT 2013).

Scenario 3: reuses existing ontological resources. This study reuses QUDT ontology (Hodgson et al. 2011) and time ontology (W3C 2017). The QUDT ontology provides a standardized and consistent glossary for science and engineering fields to express quantities and units of measurements (Hodgson et al. 2011).

Scenario 4: reuses and reengineers existing ontological resources. In this study, this scenario is not applicable.

Scenario 5: reuses and merges existing ontological resources. In this study, this scenario is not applicable.

Scenario 6: reuses, merges, and reengineers existing ontological resources. In this study, this scenario is not applicable.

Scenario 7: reuses ontology design patterns. In this study, this scenario is not applicable.

Scenario 8: restructures ontological resources. This study will develop domain ontologies by extending the shared ontology to constitute the road ontology network.

Scenario 9: localizes ontological resources. In this scenario, ontology engineers are required to tailor an existing ontology to other linguistic and cultural communities to achieve a multilingual ontology, which does not apply to this study.

2.5.2 Road Knowledge base

To store information, a knowledge base is created based on ontologies. A domain knowledge base is a repository for information collection, organization, and sharing (Noy and McGuinness 2001). Simply put, an ontology along with road instance data comprises the road knowledge base. Because distributed knowledge bases over the Internet can share data, a domain engineer can therefore access other domain knowledge bases for cross-domain information integration.

After the information and instances have been deposited and organized in a knowledge base, users should be able to access the data for extracting, modifying, and reasoning using a query language. The query language tailored for the Semantic Web is known as Simple Protocol and Resource Description Framework Query Language (SPARQL) (Prud'Hommeaux and Seaborne 2008). SPARQL is a powerful query language that enables a web-integrated query using triple patterns, conjunctions, disjunctions, and optional patterns. Results of SPARQL queries vary in form, and can be ordered, limited, and offset in number (W3C 2008). Additionally, since SPARQL uses standard web technologies, knowledge bases that are created as SPARQL endpoints can be directly queried (Niknam and Karshenas 2015).

2.5.3 Existing Road-related Ontologies

Researchers have tried to introduce Semantic Web technologies (SW) for organizing transportation feature information. The following sections provide a review of current literature on the applications of the semantic web technologies in highway construction and several models using the SW technologies in infrastructure construction.

2.5.3.1 HiOnto model

El-Diraby and Kashif (2005) developed the first highway construction ontology—HiOnto. HiOnto tries to present a full ontological description of the six main concepts in highway construction projects, i.e., project, process, product, actor, resources, and technical topics. It claims to have built a 4,000-term glossary, including 2,800 unique concepts, 281 processes, 384 highway-related products, 117 highway-specific actors, and 441 application-level resources. Within HiOnto, subdomain ontologies span three levels of abstraction: domain, application, and users (Guarino 1997; Uschold and Jasper 1999). However, the HiOnto ontology is limited in three ways: (1) it is confined to highway construction; (2) its information exchanges with other lifecycle phases (such as scheduling and estimating) still require a large amount of human effort; and (3) given the thousands of terms used in the ontology, it is hard to use it without a well-classified glossary.

2.5.3.2 A Collaborative Portal Model

El Gohary and El-Diraby (2010a) proposed a prototype portal for integrating infrastructure construction processes collaboratively. The portal can form temporary virtual organizations for projects and project stakeholders to work together. It is founded on three main theses: a process-centered approach, knowledge-based systems, and semi-automated human-savvy portals. The process-centered approach is crucial for organizing domain knowledge and supporting stakeholders' access to the product, actor, and resource properties; the knowledge-based systems are crucial to enabling formal knowledge representation and exchange; and the semi-automated human-savvy portals are essential for aiding human communication. Additionally, the six main layers of the portal include storage, access, communication, interoperability, service, and presentation. Every two adjacent layers form a level, i.e., data level, kernel level, and client level. Among them, the kernel level consists of six modules for the representation, merging, navigation, and exploration of the process knowledge. The following sections describe the IC-PRO-Onto and the Onto-Integrator related to the collaborative portal. IC-PRO-Onto is for the representation of the process knowledge, while the Onto-Integrator is for ontology merging.

El-Gohary and El-Diraby (2010b) later proposed an Infrastructure and Construction PRO-cess Ontology (IC-PRO-Onto) for the portal. The IC-PRO-Onto is the ontology developed for the collaborative portal for process information representation and integration within the infrastructure construction domain. It represents domain knowledge with five concepts: entity, constraint, attribute, modality, and family, whereby an entity can be a project, action, actor, product, resource, or mechanism. Additionally, the process life cycle is represented by five stages: initiating stage, planning stage, execution stage, monitoring and control stage, and closing stage. Each process is made up of a set of subprocesses, which are composed of activities. Each activity is composed of tasks. The IC-PRO-Onto models' processes are divided into four categories: (1) core processes, (2) management processes, (3) knowledge integration processes, and (4) support processes. Core processes are product-specific procedures that create a project's primary products. Management processes enable core processes and ensure the design and construction follow project objectives. Knowledge integration processes extensively and formally embed key concepts, knowledge, and experience into a project throughout its life cycle. Support processes, such as administration, support other processes (El-Gohary and El-Diraby 2010b). Together, the prototype collaborative portal and IC-PRO-Onto begin developing an approach for knowledge exchange and process integration within infrastructure projects.

El-Gohary and El-Diraby (2011) later developed an ontology merger called Onto-Integrator to deal with heterogeneous and distributed ontologies within infrastructure domains in the prototype. The Onto-Integrator uses semantic similarity comparison methods, extensions of Relational Concept Analysis, and a heuristic approach for merging concept taxonomies, relations, and axioms (El-Gohary and El-Diraby 2011, Pauwels et al. 2017). The most significant contribution of this work is the merging of axioms, which was not supported by previous tools.

However, some problems are still unsolved to achieve efficient cross-domain information exchange with the portal and its related ontologies. First, the IC-PRO-Onto is not developed in a domain-specific modular architecture. That means information from a specific domain may be scattered across several big categories or processes, which is not convenient for domain experts to add or maintain domain information. Second, as Törmä (2013) notes, the integration approach proposed in the Onto-Integrator continues to rely on manual work for information exchange in receiving tasks, reducing the efficiency of information integration. Finally, in practical scenarios, most of these ontologies and the portal stay at the conceptual level and have not been fully refined or validated. Therefore, these current semantic web models are not practical for cross-domain information sharing in the infrastructure industry.

2.5.3.3 Modular Domain Models

The architecture of a road data model can be created more flexibly by using a domain-specific modular approach. One example is to create a shared ontology first, and each domain then develops its own ontology separately by extending the shared ontology. Domain experts can then access other domain information via the shared ontology. Several ontologies have been developed for buildings using this approach. For example, BIM shared ontology (Karshenas and Niknam 2013), BIM design ontology (Karshenas and Niknam 2013), BIM estimating ontology (Niknam and Karshenas 2015), and BIM scheduling ontology (Niknam and Karshenas 2016).

CHAPTER 3: ROAD LOCATION ONTOLOGY

3.1 Challenging and Requirements

Thanks to the evolution of advanced positioning and cartography technologies, highly accurate geographic location information is accessible with the help of satellite navigation systems. However, determining how to fully utilize this information to meet the demands of various users for query, analysis, and output of geographic information has become a new challenge. There are several major challenges:

1. Geographic information is defined and classified differently by users from different fields. The understanding and architecture of location information will be greatly influenced by the culture, language, and application environment.
2. Different users tend to use different methods for storing and querying geographic information. Some users, for example, query when they have latitude and longitude coordinates handy, while others try to query by place name and yet others directly select graphics on the map. How to connect these various types of geographic data is a pressing issue that must be addressed.
3. Different users have different requirements on the accuracy and response speed of geographic information. For example, when it comes to transportation, we only need a two-dimensional route diagram for subway or ferry routes; however, when it comes to drone flight routes, we must consider the elevation information of nearby architectures. There are also different levels of accuracy required, such as the truck driver not needing to be shown

the minor road, whereas the tourist may miss the booked homestay if the driveway is not accurately depicted on the map. In general, higher accuracy necessitates a higher data density, whereas a larger amount of information will significantly slow down data processing speed. As a result, how to balance the information accuracy and information processing speed has become a new challenge.

Considering these challenges, this study employs the Semantic Web technologies to develop a location ontology that is extendable, machine-processable, allows annotation, and supports data integration and exchange across domains and data formats. The scope of this study will include general localization methods, but more importantly, it will serve the requirements for road-related location information from all road-related domains.

The information related to the location of the road, as well as the location of features and properties attached to it, is represented by the Road Location ontology. In more detail, the Road Location ontology allows for assigning a road segment's location properties. The Location ontology uses 3D coordinates or a linear referencing method to represent the start and end of a road or a road segment. The Road Location ontology is created in Protégé software (Stanford University 2015), and the prefix RLO has been assigned to it. The location ontology encompasses all three currently popular modes of location representation: absolute mode, relative mode, and address mode. The absolute location representation mode represents the spatial referencing system (SRS), the relative location representation mode represents the linear referencing system (LRS), and the conventional address names is represented by address location representation mode, which is not discussed in this dissertation.

The Road Location ontology is developed as follows.

- **Purpose:** the Road Location ontology is created to provide a conceptual model for providing location information for points along a road.
- **Scope:** The scope is limited to providing the point location expression with absolute and relative methods. The absolute method defines the point location with 3D coordinates, while the relative method defines the location of a point with the distance measured from a known point (a point defined with 3D coordinates).
- **Implementation language:** the Road Location ontology is implemented in RDF/OWL language.
- **Intended end-users:** engineers from different road knowledge domains that need to assign information to a road, examples include: storing crash accident information or assigning speed limit to a road segment.
- **Intended use:** assigning information to a point or a road segment.

To globally identify the concept, property, and relationship in the Semantic Web, a unique ID should be assigned to each. In this study, the Internationalized Resource Identifier (IRI) is used. The IRI is created by adding the identity of each concept, property, and relationship to the IRI of the ontology. For example, the IRI of the Road Location ontology is http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology# and the prefix assigned to it is RLO. Thus, the IRI of the Point001 is presented as RLO:Point001 or http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Point001. Protégé, the ontology editor used in this study, automatically generates the IRI for each class, property, and ontology.

Additionally, several non-ontological resources are referred to in this study: (1) for SRS: ISO 19107 (2001), ISO 19111 (2002), ISO 19115 (2002), and Esri Maps (ESRI 2006); and (2) for LRS: ISO 19133 (2003), NCHRP 20-27 model (NCHRP 1974, Vonderohe et al. 1997), extended NCHRP 20-27 model (Adams et al. 2001).

Finally, the QUDT ontology (Hodgson et al. 2011), which establishes a standardized and consistent vocabulary for expressing units of measurement for scientific and technical terms (Niknam and Karshenas 2017; Hodgson et al. 2011), is reused in the development of the Road Location ontology.

3.2 Road Location Ontology

Figure 6 presents the top view of the Road Location ontology. There are three common ways to represent location information: absolute mode, relative mode, and address. The address mode is typically associated with a comprehensively defined postal system, which is not suitable for the precise road locating system discussed in this study. As a result, the following sections of this dissertation only discuss the absolute mode (Section 3.3) and the relative mode (Section 3.4).

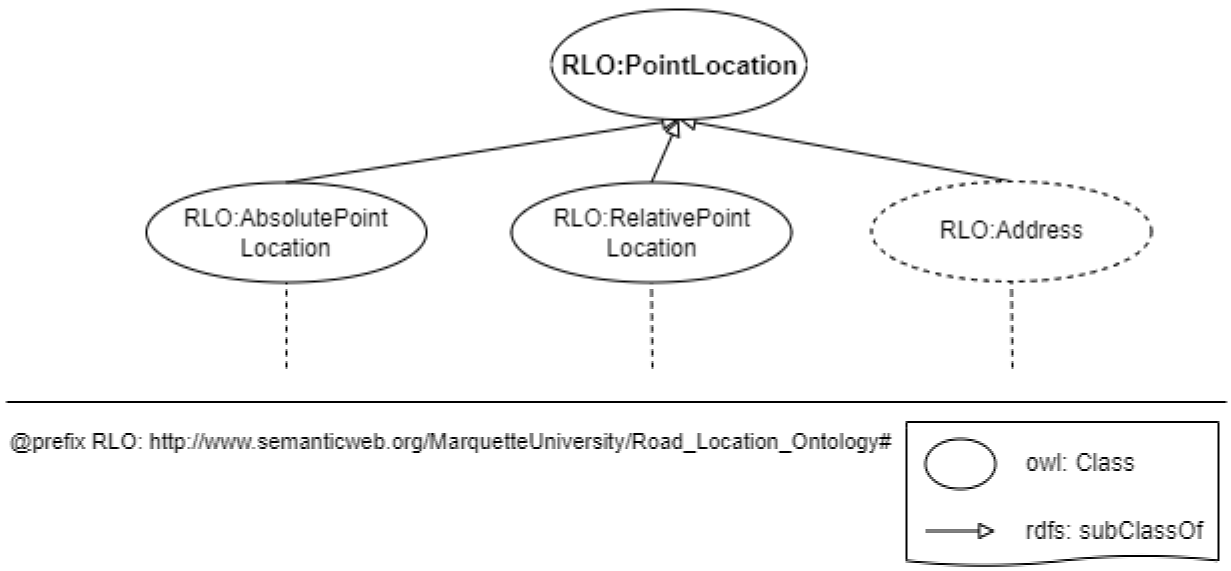


Figure 6 Top view of the Road Location ontology

3.3 Absolute Location

To specify a location in three-dimensional space, both horizontal and vertical coordinates are required, referring to the horizontal coordinate system (HCS) and vertical coordinate system (VCS), respectively. SRS is made up of HCS and VCS.

The HCS includes the geographic coordinate system (GCS) and the projected coordinate system (PCS). As a result, four types of absolute point location representations are formed: horizontal position, planar position, 3D geographic position (combination of horizontal position and vertical position), and 3D projected position (combination of planar position and vertical position). Figure 7 shows the architecture of the top-level concepts defined in the absolute point location ontology. For the sake of brevity, not all classes and relationships are depicted.

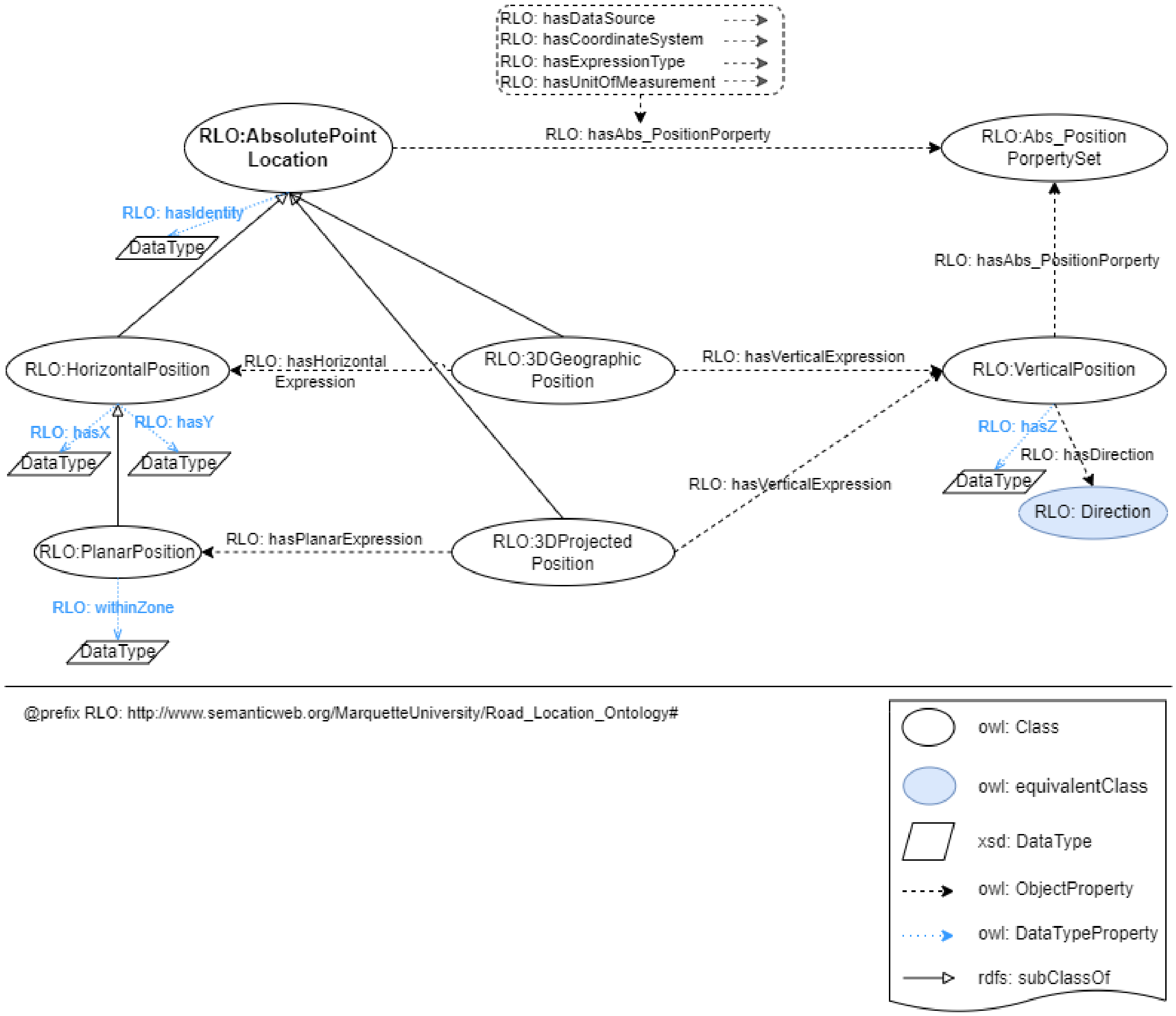


Figure 7 Top view of the absolute point location ontology architecture

The class RLO:Direction is developed as an enumerated class. An enumerated class, also known as an equivalent class in Protégé (Stanford University 2015), is an anonymous class that explicitly lists all of its individuals (Horridge et al. 2009). Users can pick up the defined instances on demand rather than enter the names by themselves. Here, the instances RLO:upward and RLO:downward are created as equivalent classes of the class RLO:Direction.

According to ESRI (ESRI 2006), a GCS includes an angular unit of measure, a prime meridian, and a datum; a PCS includes a map projection, a set of projection parameters that customize the map projection for a particular location, and a linear unit of measure; and a VCS includes a unit of measure, a datum, and a representation of direction (ESRI 2006). The inherited system information described above is preset parameters of the coordinate systems which are not defined in the Road Location ontology.

The properties collected in the property set of the absolute position include data sources, coordinate systems, expression types, and the unit of measurement used in the ontology. Among these properties, classes of data source, subclasses of the coordinate system, and expression types are created as enumerated classes. For example, equivalent classes of HorizontalExpressionType include limited types: DecimalDegrees_DD, DegreesMinutesSeconds_DMS, and DegreesDecimalMinutes_DDM; equivalent classes of DataSource mainly include GPS, RemoteSensor, and SiteSurvey; while equivalent classes of VerticalCoordinateSystem(VCS) mainly include NGVD29, EGM96, and EGM20008. Users simply select the coordinate system from the equivalent class list when using this ontology. Figure 8 shows the architecture of the property set.

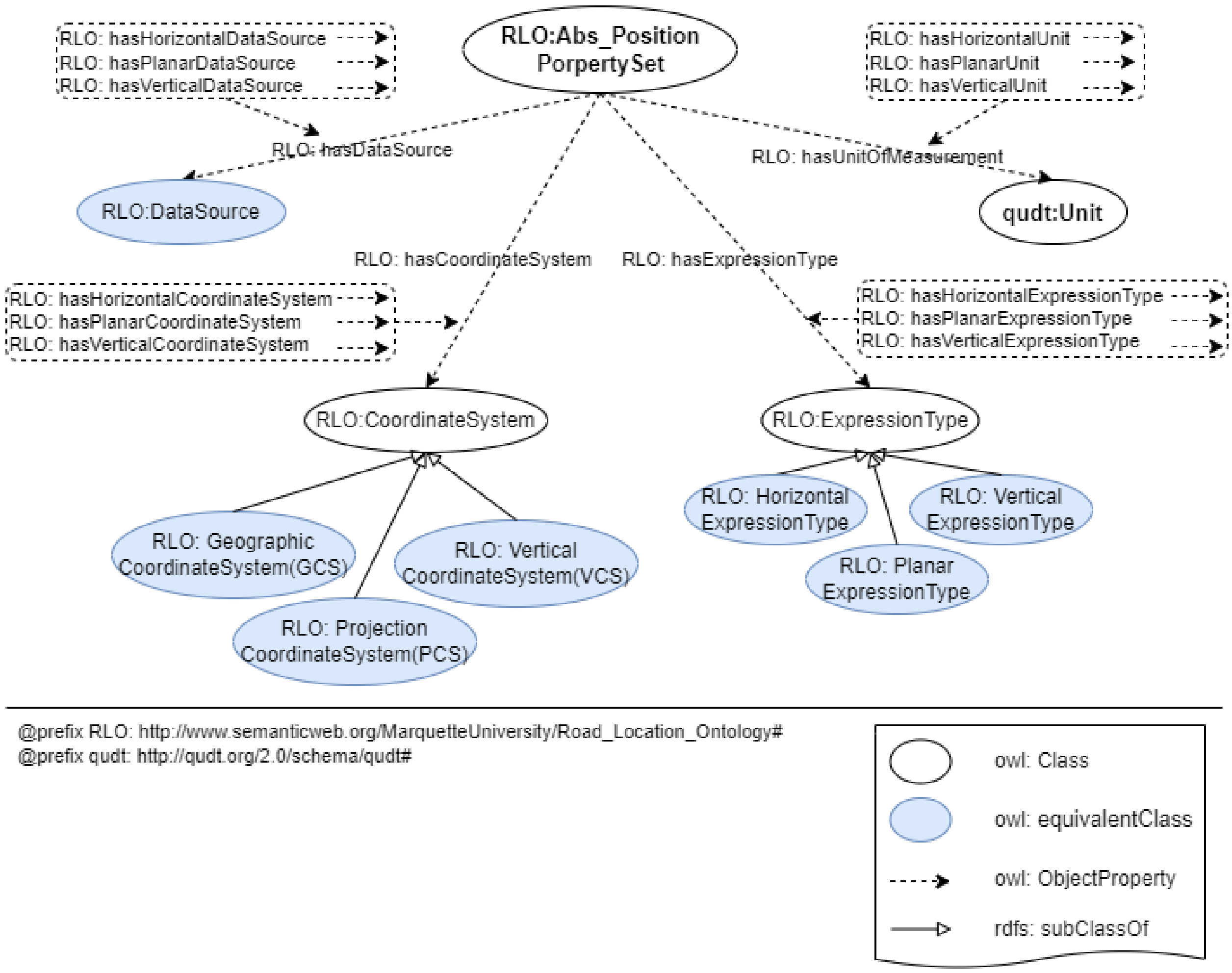


Figure 8 Architecture of the RLO:AbsolutePositionPropertySet

Figure 9 provides an absolute point location example of Point001 that is defined using GCS. This example employs the WGS84 system, collects data from GPS, and adopts the Decimal Degrees (DD)[3] expression type for the point's horizontal location expression. Similarly, this example employs the WGS84 system, collects data from the GPS, adopts the elevation expression type, and specifies the upward direction[4] for the

[3] Three main GCS information expression types are Decimal Degrees (DD), Degrees Minutes Seconds (DMS), and Degrees Decimal Minutes (DDM).
[4] Generally, the default positive direction for height and elevation expression type is upward while, for depth expression type, it is the opposite (ESRI 2006).

point's vertical location expression. The IRI of each primary class and properties are listed at the bottom of the figure.

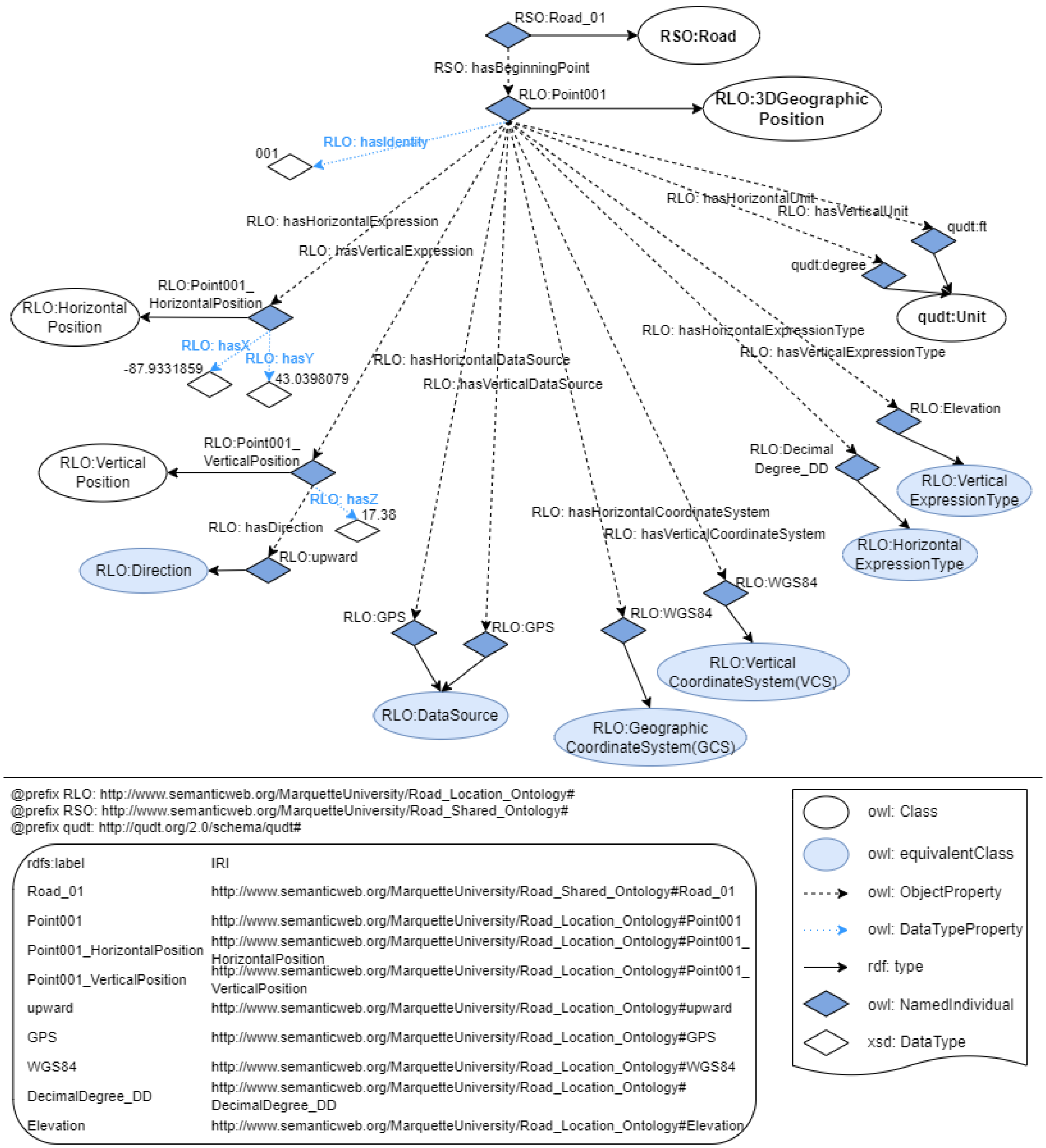


Figure 9 A 3D geographic position point location example

Appendix 1 provides an example of the absolute point location of the Point001 represented with the PCS properties. This example employs the NAD83 system, specifies the zone of the NAD83 system, collects data from site survey records, and adopts Northing and Easting[5] expression type for the point's planar location expression. It uses the NAD83 system, collects data from site survey records, adopts depth expression type, and correspondingly specifies the downward direction for the point's vertical location expression.

3.4 Relative Location

There are several methods for locating a point with a relative location. As mentioned in Chapter 1, the LRS is the one exactly tailored for the location expression of linear objects, which is a good fit for the topic of this study. The other common positioning methods such as mobile positioning will not be discussed in this dissertation.

LRS is widely used as a less-data-intensive way for specifying the location of elements, events, or segments along with a physical linear transportation feature (ISO 19133 2003, Scarponcini 2005). Designers, for example, typically use stationing to specify assemblies' location, whereas safety officers prefer to use reference markers to record the location of a traffic accident. As mentioned in Chapter 1, position identity, linear element, linear referencing method (LRM), reference markers, one or more anchor points, one or more anchor sections, and distances are key concepts used by LRS to specify a location (ISO 19133 2003, Scarponcini 2005). When using an offset measure, the offset variable should also be included in the position expression. Figure 10 depicts

[5] Usually, in PCS, geodetic locations on the surface of the earth are designated as eastings and northings, or x and y) in the planar system (ESRI 2006).

the LRS's architecture defined in the relative point location ontology. The explanation of the ontology architecture is as follows.

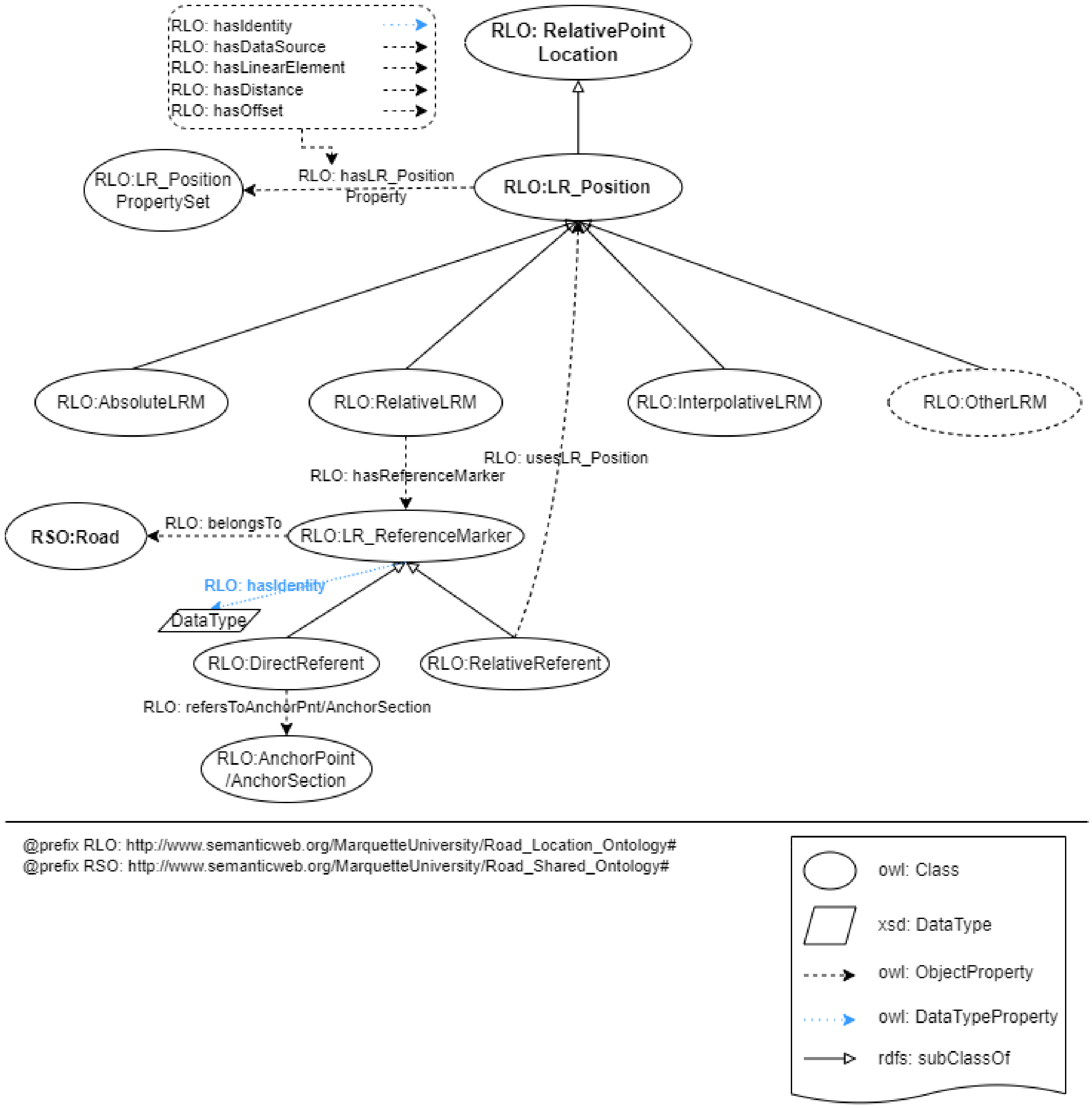


Figure 10 A partial view of relative location ontology architecture

Due to the multiple forms of construction and maintenance data, there is no one sort of location reference mechanism that can independently support a comprehensive asset management system. Typically, one or more LRM kinds are employed on-demand by different sectors. To represent the linear referencing position (LR Position expression), the four forms of LRM, absolute LRM, relative LRM, interpolative LRM, and some other LRMs, commonly employ distinct expressions. However, the structure of the aforementioned LR_position expressions is formed by a uniform relative base expression (Scarponcini 2005). The components of the four expression types are presented schematically in Table 4. The relative base expression is the formalized location expression discussed in Chapter 1 and displayed in Table 2. The special reference marker concept and related properties are illustrated in Figure 10, and the following provides a more detailed description of each concept.

Table 4 LR Position Expression of four LRM types

LRM Type	LRM Subtype	LR Position Expression
Absolute LRMs	Mile Point	LR_PositionPropertySet
	Kilo Point	
Relative LRMs	Mile Post	LR_PositionPropertySet + Reference Marker
	Reference Post	
	County Post	
	Intersection Offset	
Interpolative LRMs	Percentage	LR_PositionPropertySet (values range: 0~100)
	Normalized	LR_PositionPropertySet (values range: 0~1)
Other LRMs	Address	Address

- **Position Property Set:** collects the data source, the linear element that the point position is measured by, the distance measured from the anchor points/reference markers, the distance unit, and, if necessary, the offset.
- **Data source**: similar to the absolute point location, examples of the data source can be GPS, site survey, and remote sensor systems.
- **Linear element**: the identity of the linear element, such as the identity of the road or the segment, should be specified.
- **Distance**: depending on whatever LRM is used, there are several sorts of expression: (1) when the mile point is used, for example, distance 1.5 means the point is 1.5 miles from the origin of the specified linear element; (2) when the milepost is used, for example, distance 1+.50 means the point is 0.5 miles from milepost 1; and (3) when percentage LRM is used, the distance value 50 means the point is 50 percent of the entire length of the linear element from its' start point; while normalized LRM is employed, the situation is similar, except that the numbers range from 0 to 1.
- **Offset**: offset expression includes the offset reference, offset direction, offset value, and unit of measurement. While centerline, edge of travel, and curb are all equivalent classes of class RLO:OffsetReference. In terms of positive offset direction, right is typically the default (ISO 19133 2003). Figure 11 provides the architecture of distance- and offset-related concepts and relationships. Equivalent classes are employed here for defining the classes RLO:OffsetReference and RLO:OffsetDirection. Please see ISO

19133 for further information on offset references.

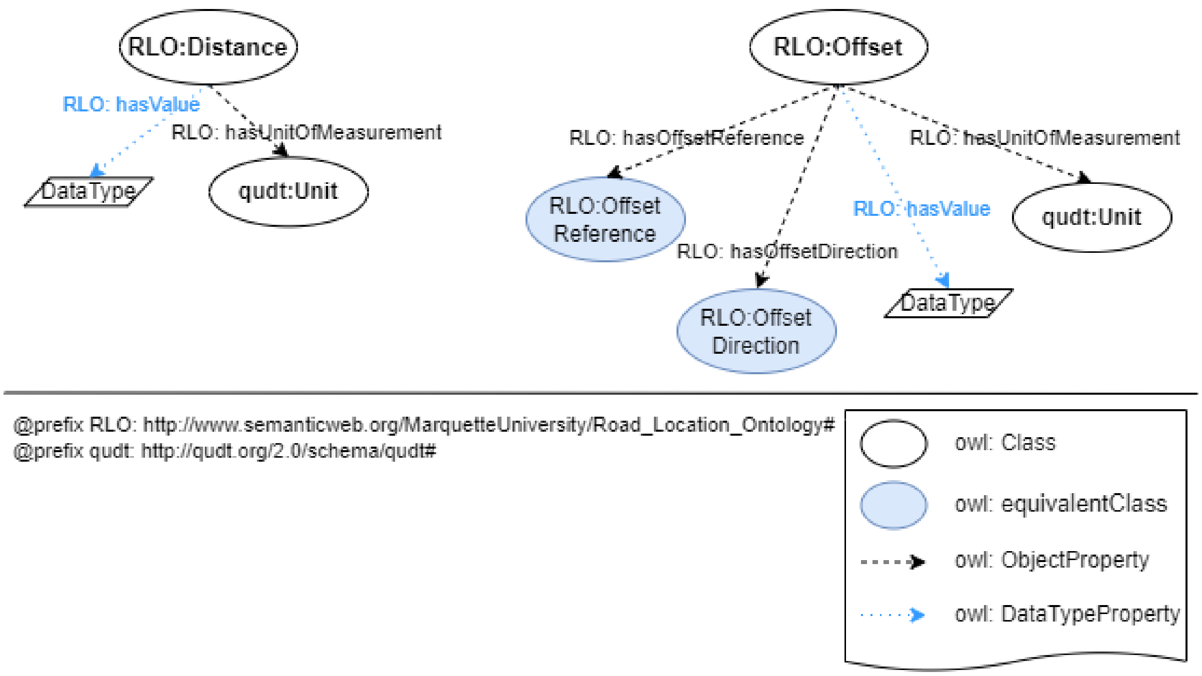


Figure 11 Architecture of RLO:Distance and RLO:Offset

More information regarding the anchor point and reference point can be found here for a better understanding of the reference marker used in LRS. The anchor and reference points are often used in LRS-related documents. Anchor points are typically conceptual points, such as the intersection of two road centerlines, whereas reference points are frequently referred to as real, easily located things (Dueker and Butler 1997). Furthermore, reference points are commonly defined in a geographic datum such as NAD83. This study makes no distinction between the anchor point and the reference point, and it also incorporates user-customized reference markers into the class AnchorPoint. The "conventional anchor point" complies to the ISO standard (ISO 19133 2003), which takes a geometry of type point, as stated in the ISO 19107 geometry (GM) package, as a parameter. The GM point is specified by a set of coordinate values in a coordinate reference system. In this study, the anchor point refers to a point with known absolute location values. Figure 12 depicts the relationships between the classes RLO:AnchorPoint and RLO:AnchorSection, as well as associated concepts.

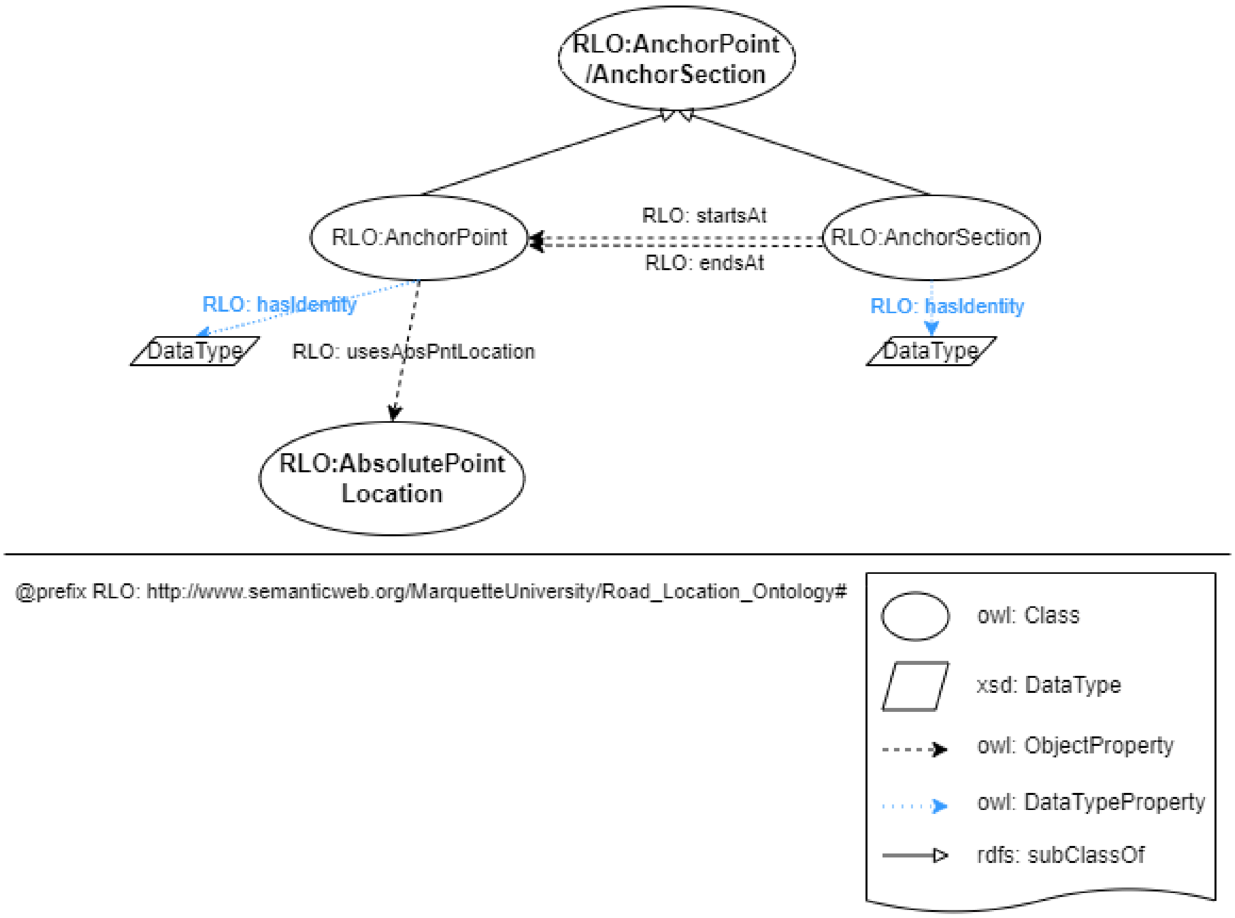


Figure 12 Architecture of RLO:AnchorPoint and RLO:AnchorSection

Furthermore, the LRS allows users to customize reference markers on demand. The reference marker can be defined by referring to a specific anchor point or another reference marker that has already been defined.

The following provides two examples of PointX that is defined using LRS with Absolute LRM and Relative LRM, respectively. Figure 13 provides a schematic top view of the road, Road_01. The unknown Point X is marked with red circle. The reference marker used in the Absolute LRM (mile point) and that used in the Relative LRM (mile post) are marked in the figure as well.

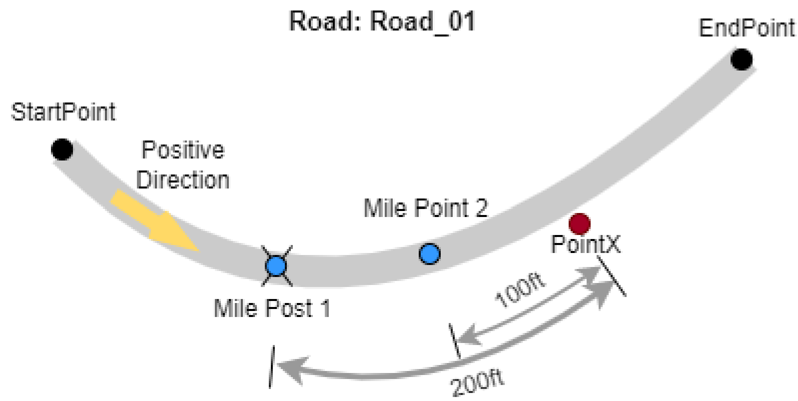
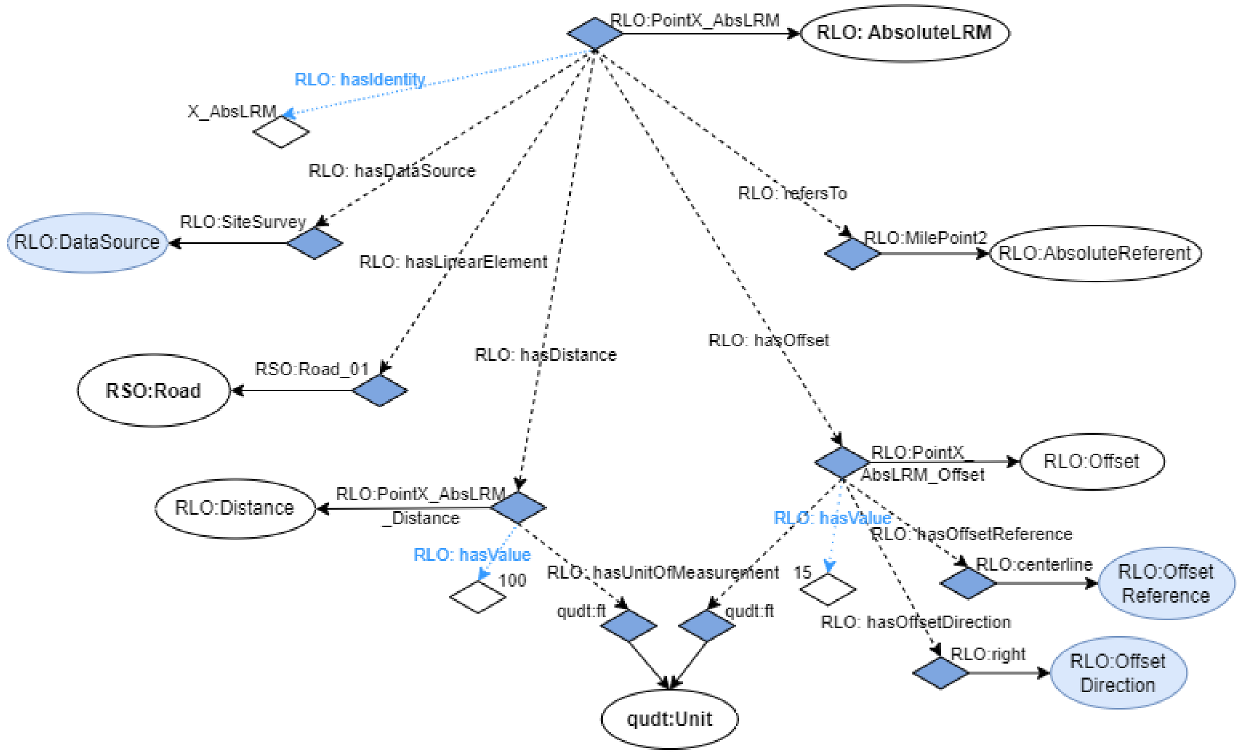


Figure 13 A schematic top view of Road_01

Figure 14 provides an absolute LRM representation of the PointX and Figure 15 provides a relative LRM representation of the PointX. The IRI of each primary class and properties are listed at the bottom of the figures.



@prefix RLO: http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#
@prefix RSO: http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
@prefix qudt: http://qudt.org/2.0/schema/qudt#

rdfs:label	IRI
PointX_AbsLRM	http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#PointX-AbsLRM
SiteSurvey	http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#SiteSurvey
Road_01	http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Road_01
PointX_AbsLRM _Distance	http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#PointX_AbsLRM _Distance
right	http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#right
centerline	http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#centerline
PointX_AbsLRM _Offset	http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#PointX_AbsLRM _Offset
MilePoint2	http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#MilePoint2

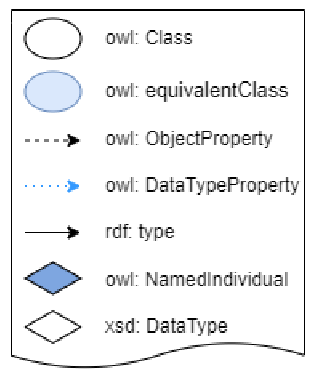


Figure 14 An example of the Absolute LRM expression of Point X

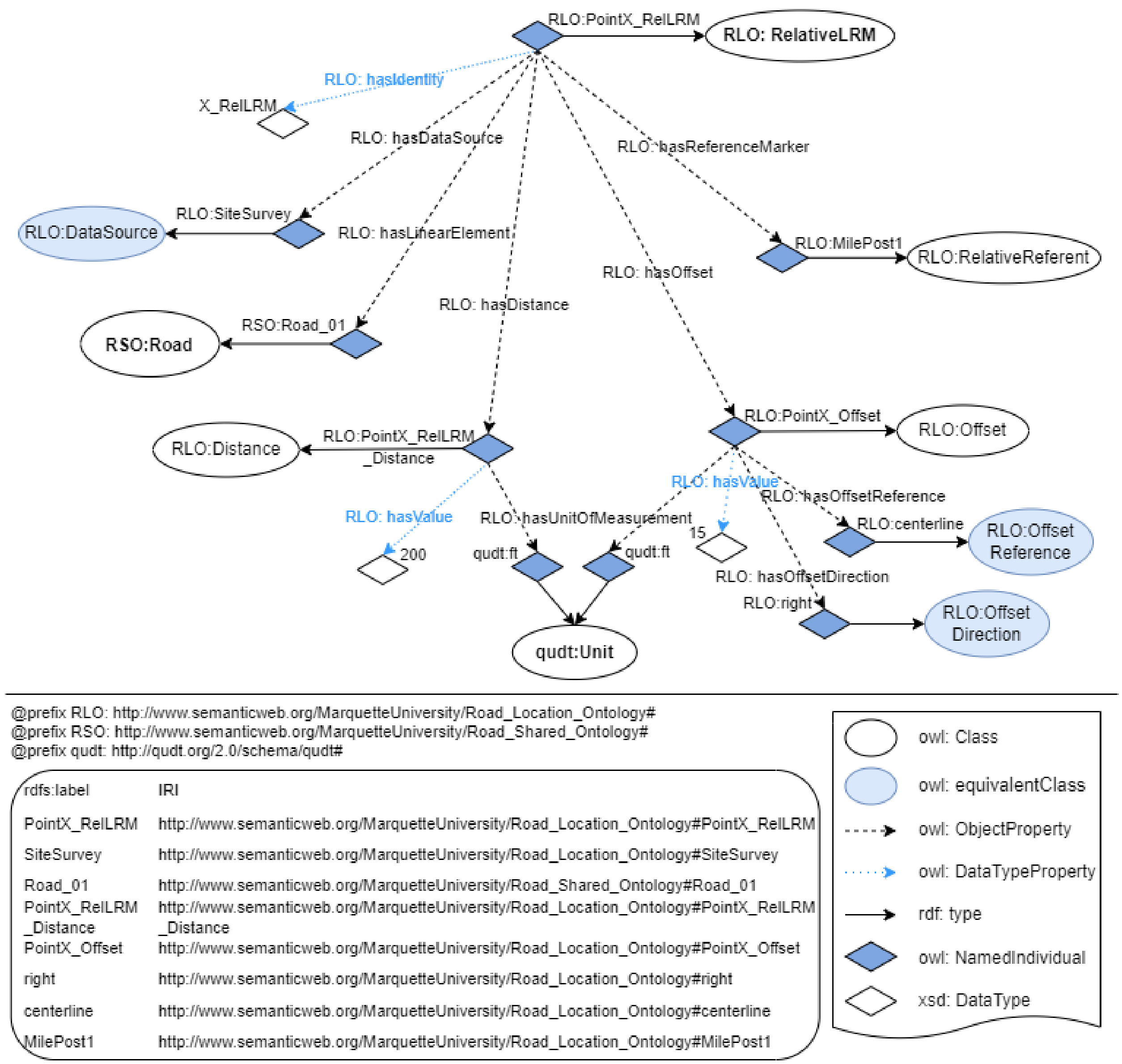


Figure 15 An example of the Relative LRM expression of Point X

Figure 16 depicts the Road Location Ontology implementation in Protégé software (Stanford University 2015). The left panel displays the concepts (classes), the center panel displays the object properties, and the right panel displays the data type properties.

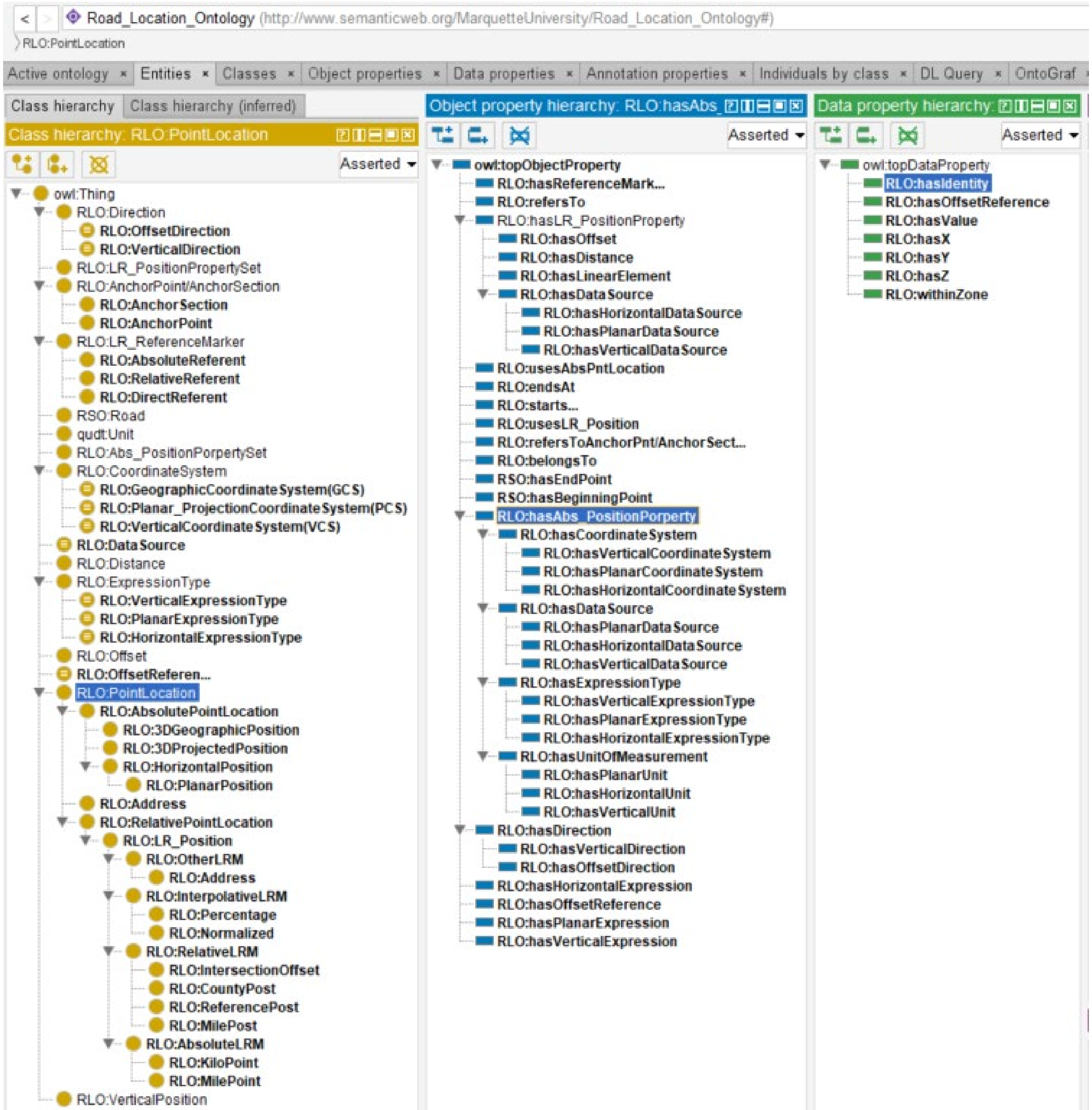


Figure 16 The Road Location ontology implemented in Protégé

CHAPTER 4: ROAD SHARED ONTOLOGY ARCHITECTURE

This chapter provides the methodology for developing a shared ontology for the road infrastructure and the knowledge base created based on it. Section 4.1 discusses the general requirements of creating a road ontology, Section 4.2 presents a general view of the shared ontology and explains the concepts it contains, and Section 4.3 provides the developed knowledge base for the shared ontology.

4.1 Road Infrastructure Ontology

A road infrastructure project, as mentioned in Chapters 1 and 2, involves a wide variety of data. The data is stored in multiple storage methods and formats based on each domain's requirements and conventions. Presently, for example, the location data is collected from sources such as digital maps, remote sensors, field surveying reports, and GIS. GIS data is mainly stored in relational databases (e.g., Oracle Spatial) or a non-relational Geodatabase; design data is generated in a 3D modeling platform (e.g., Autodesk Civil 3D) and stored in a format such as dwg, landXML, or tables in a relational database (e.g., ODBC, OJBC); and scheduling data and estimating data are created in a form or software and then stored in a text document, word table form, Excel spreadsheet, or a scheduling/estimating software (such as Oracle Primavera P6 and Sage Timberline). Integrating road information across these various systems, formats, and domain knowledge bases presents a major challenge. In this study, the author argues that the method of establishing ontology that employs Semantic Web technology is the most promising solution for the information integrating challenge.

Establishing a single ontology that covers all knowledge domains involved in a road’s life span is obviously not feasible (O'Leary 1997). This study proposes a modular road ontology to exchange information across domains instead of creating complex mappings between domain ontologies. The Road Shared ontology serves as a “semantic bridge” when all related road domains are mapped to the shared ontology instead of each other to exchange domain information (Mascardi et al. 2009; Niknam 2015). Domain data is organized and maintained by domain experts. Engineers from one domain can access information stored in other domains via this semantic bridge. This chapter aims at the semantic representation of the kernel road concepts that are shared by multiple road-related domains. One example of a modular ontology is provided below to develop a shared ontology that various road project domains can reuse for creating domain ontologies. The shared road ontology defines the framework architecture consisting of those concepts.

Figure 17 shows how a shared ontology can act as the “semantic bridge” and how these domain ontologies fit together to provide a modular architecture for a road ontology.

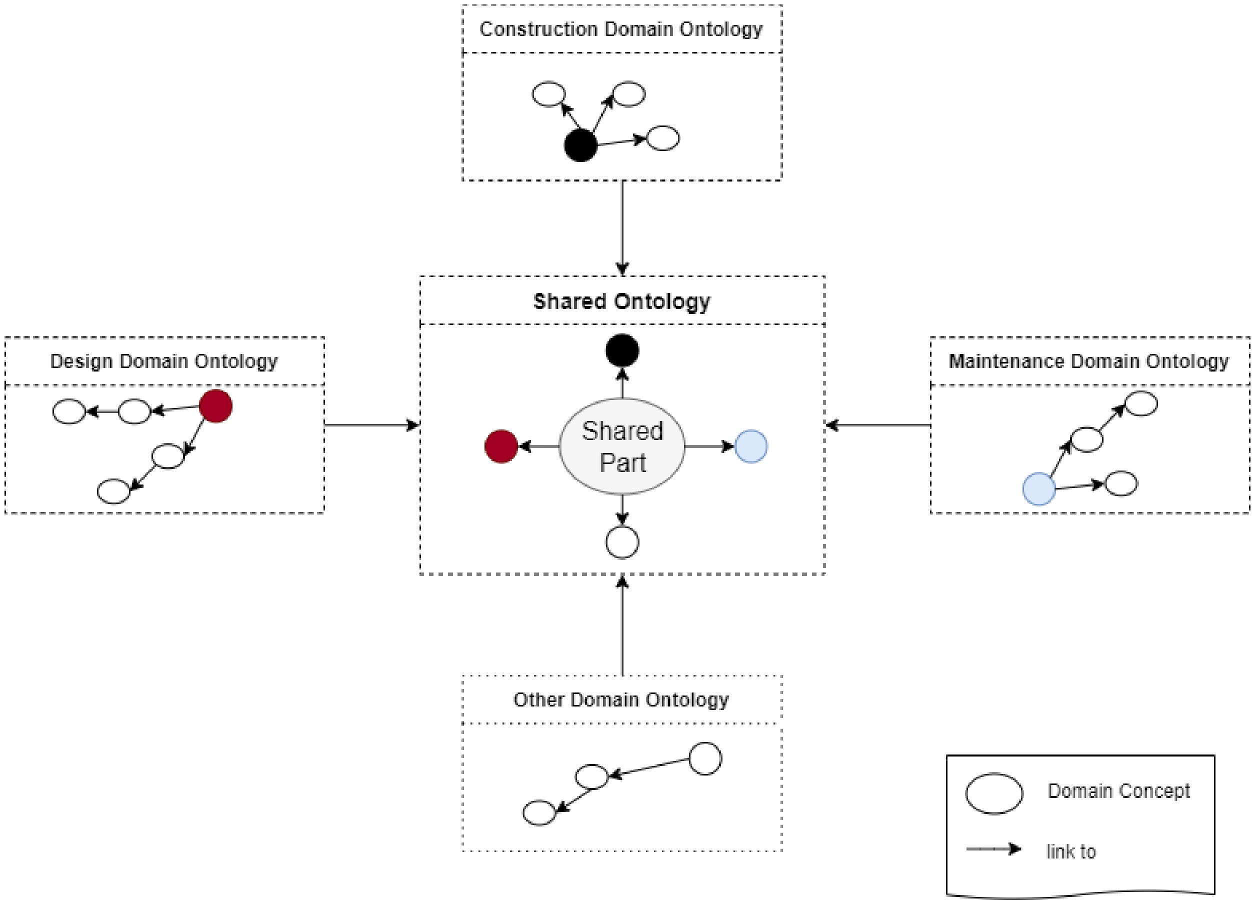


Figure 17 A schematic diagram of road information integration using a road shared ontology

This chapter deals only with the development of the Road Shared ontology and the following Chapter 5 will introduce the development of domain ontologies. The methodology used to create the Road Shared ontology and the concepts contained therein are explained as follows.

4.2 Road Shared Ontology

To better understand the structure of a road, Figure 18 provides a simple schematic view of a piece of road as an example. The concepts contained in the figure will be discussed in detail in the following section.

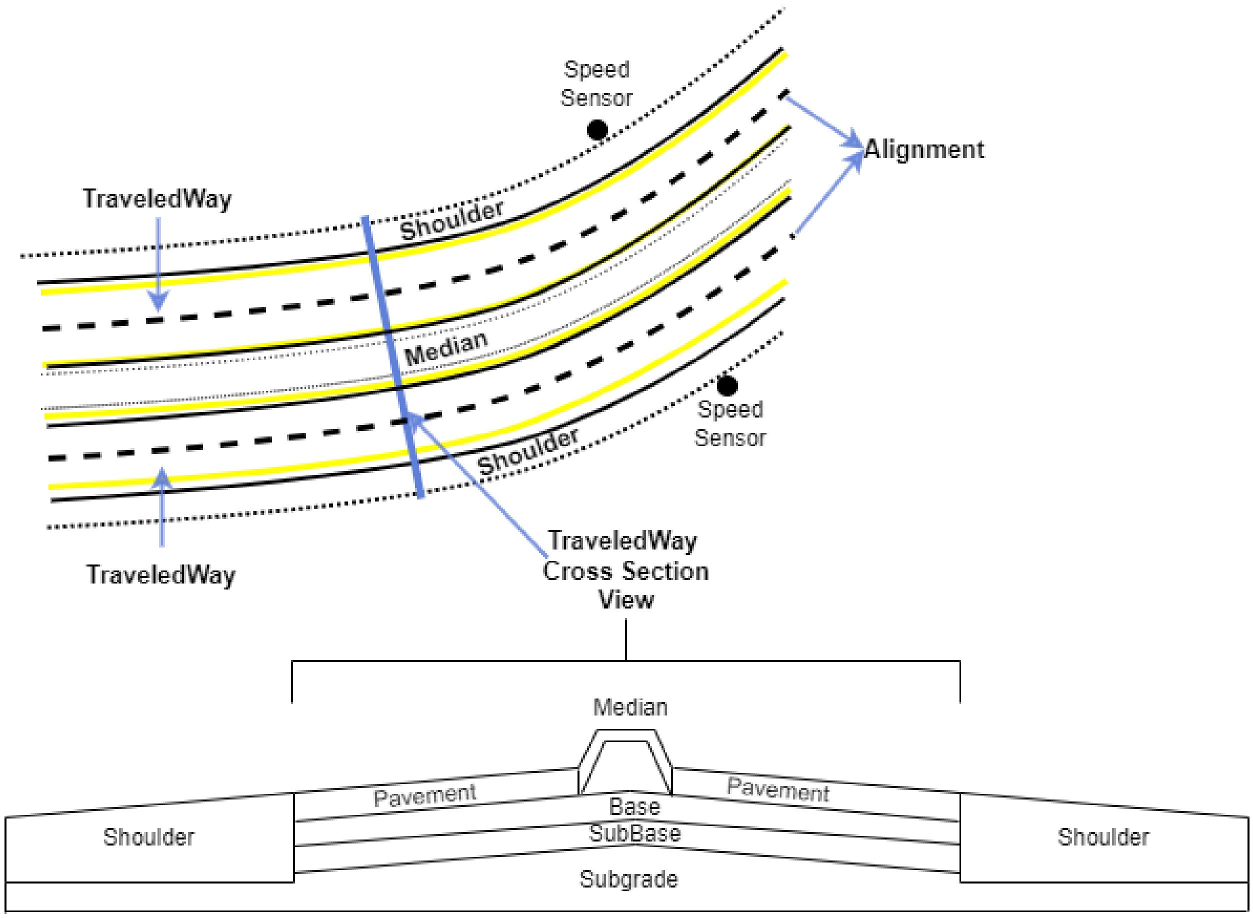


Figure 18 A schematic view of the structure of a road

A well-developed shared ontology should be able to describe the fundamental concepts shared among many domains and serve as the bridge for specialized information into the nuanced concepts and well-approved vocabularies of specific domains (Doerr et al. 2003). That is, concepts that are widely reused by road-related domains should be defined in the Road Shared ontology, along with the relationships among them. Next, road domain experts can extend the shared ontology to develop their own domain-specific ontologies to organize domain information. In turn, the domain information maintained by each domain adds domain-specific properties to the Road Shared ontology.

For example, in the design domain, designers add road design properties such as material and dimensions; in the construction domain, contractors add construction properties such as schedule and cost; in the operation domain, new properties are added to a road element when the road is repaired. In a word, all the top-level concepts defined in each domain should be defined in the Road Shared ontology, serving as the starting point for expanding to various domains.

Specifications of the Road Shared ontology is developed as follows.

- **Purpose:** The Road Shared ontology is developed as a fundamental conceptual knowledge model of road infrastructure information.
- **Scope:** the scope of the Road Shared ontology is limited to the main concepts used in the design, construction, and maintenance domains.
- **Implementation language:** the Road Shared ontology is implemented in RDF/OWL language.
- **Intended end-users:** design, construction, and maintenance domain engineers.
- **Intended use:** provides a “semantic bridge” for (1) integrating and exchanging road-related information and (2) serving as a start point for various road-related domains to create their own domain ontologies.

Subsequently, several non-ontological resources are referred to in this study: AASHTO specifications (AASHTO 2011) and Civil 3D Developer's Guide (Autodesk 2021). Also, a time ontology (Ontology URI: http://www.w3.org/2006/time#) is used to provide temporal information (W3C 2017).

The time concept specifies the temporal data related to the road information. This study will reuse the time ontology from the World Wide Web Consortium (W3C) (W3C

2017), which is an OWL-2 DL ontology. Prefix time is assigned to the Time ontology in this study.

According to the W3C (2017), five major classes in the time ontology support an explicit description of the temporal information of a specific event, entity, or activity. The five classes include temporal reference system (TRS), time zone, day-of-week, temporal position, and temporal duration. The temporal position is the common superclass which indicates the TRS, the temporal position, and the datetime in use. Additionally, the temporal duration mainly specifies the time duration and the set of temporal units in use.

In this study, the Road Location ontology developed in Chapter 3 is reused in the Road Shared ontology to provide point location information along a road. The prefix RLO is assigned to the Road Location ontology in Protégé.

Figure 19 shows the main conceptual architecture of the Road Shared ontology. The prefix RSO is assigned to it. Since it’s hard to display all the concepts in one figure, only the main concepts and relations are displayed in Figure 19. For example, under the assembly, Figure 19 only expands the Traveled-Way subassembly as an example to show its subdivided architecture. For more details, the RDF/XML file of this ontology is provided in Appendix 2. The Road Shared ontology includes concepts such as jurisdiction, phase, segment, assembly, subassembly, alignment, event, device, traffic, location, and time. The following section provides a more detailed description of these core concepts and relations among them.

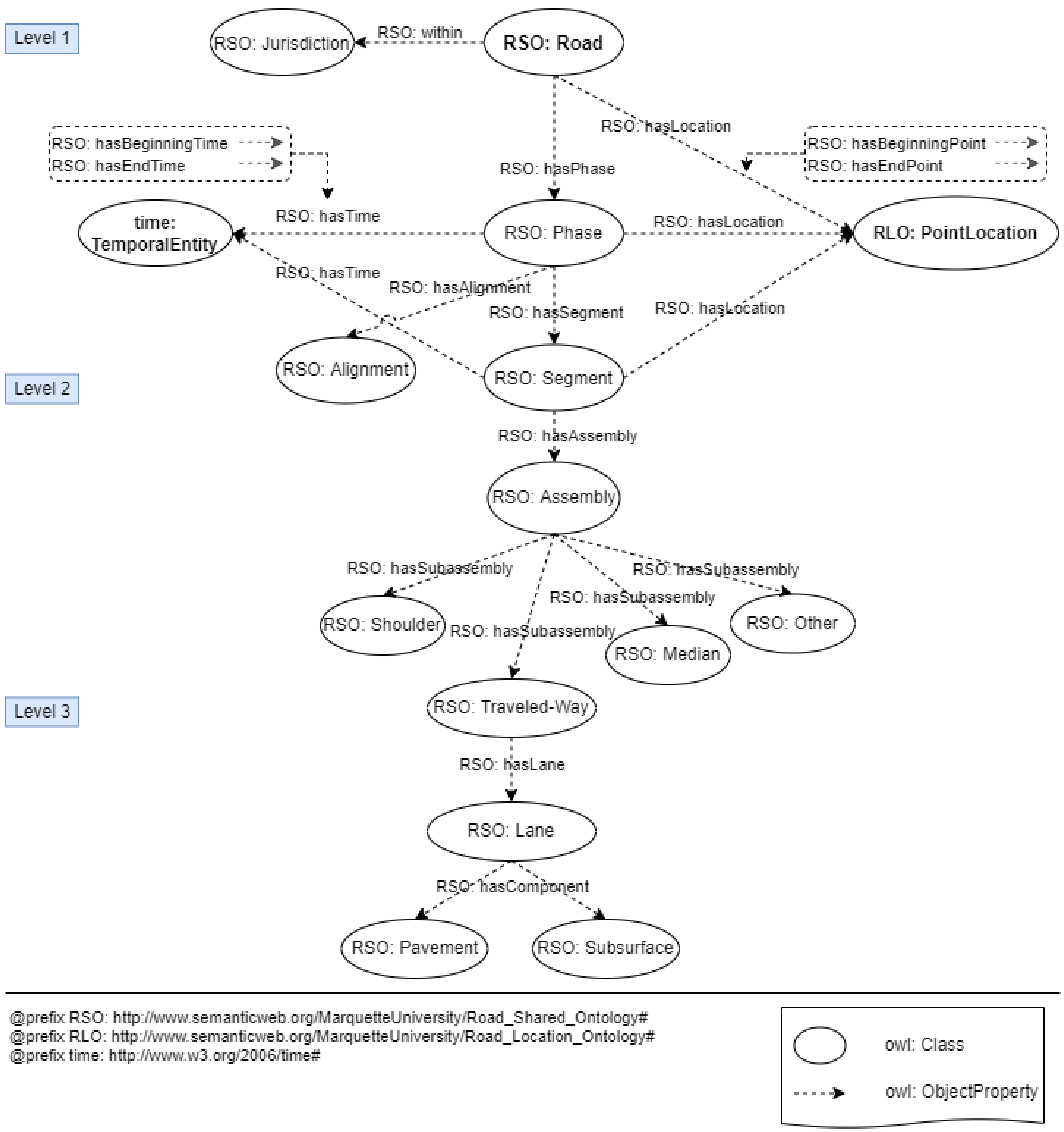


Figure 19 A general view of the Road Shared ontology architecture

To globally identify each concept, property, and the relationships among them in the Semantic Web, a unique ID is assigned to the concept. As explained in the development of the Road Location ontology, in this study, the IRI that is generated in

Protégé for each ontology, class, and property is served as the unique identifier. The full IRI of the Road Shared ontology is http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#. Since the prefix RSO is assigned to the Road Shared ontology, all the concepts, properties, and relationships defined within it are prefixed with RSO. Again, the IRI is generated by adding the ID to the prefix of the ontology. For example, the class Road's IRI is RSO:Road, which is equal to the following IRI: http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Road. A list of the ontology IRIs and their corresponding labels is shown at the bottom of Figure 19.

To explain the road infrastructure concepts defined in the Road Shared ontology more clearly, the architecture of the ontology is divided into 3 levels: (1) concepts related to the road or segments of the road, (2) road assemblies and subassemblies that compose a segment of a road, and (3) components that make up the subassemblies. The following provides a brief description of the concepts defined in the ontology from the top down.

4.2.1 Level 1 Classes

Level 1 defines project-related concepts at the top level, such as road, road identity, jurisdiction, project phase, events that may happen along the road, traffic, and devices installed along the road. Devices include objects such as traffic lights, cameras, and speed sensors. Figure 20 shows the architecture of Level 1.

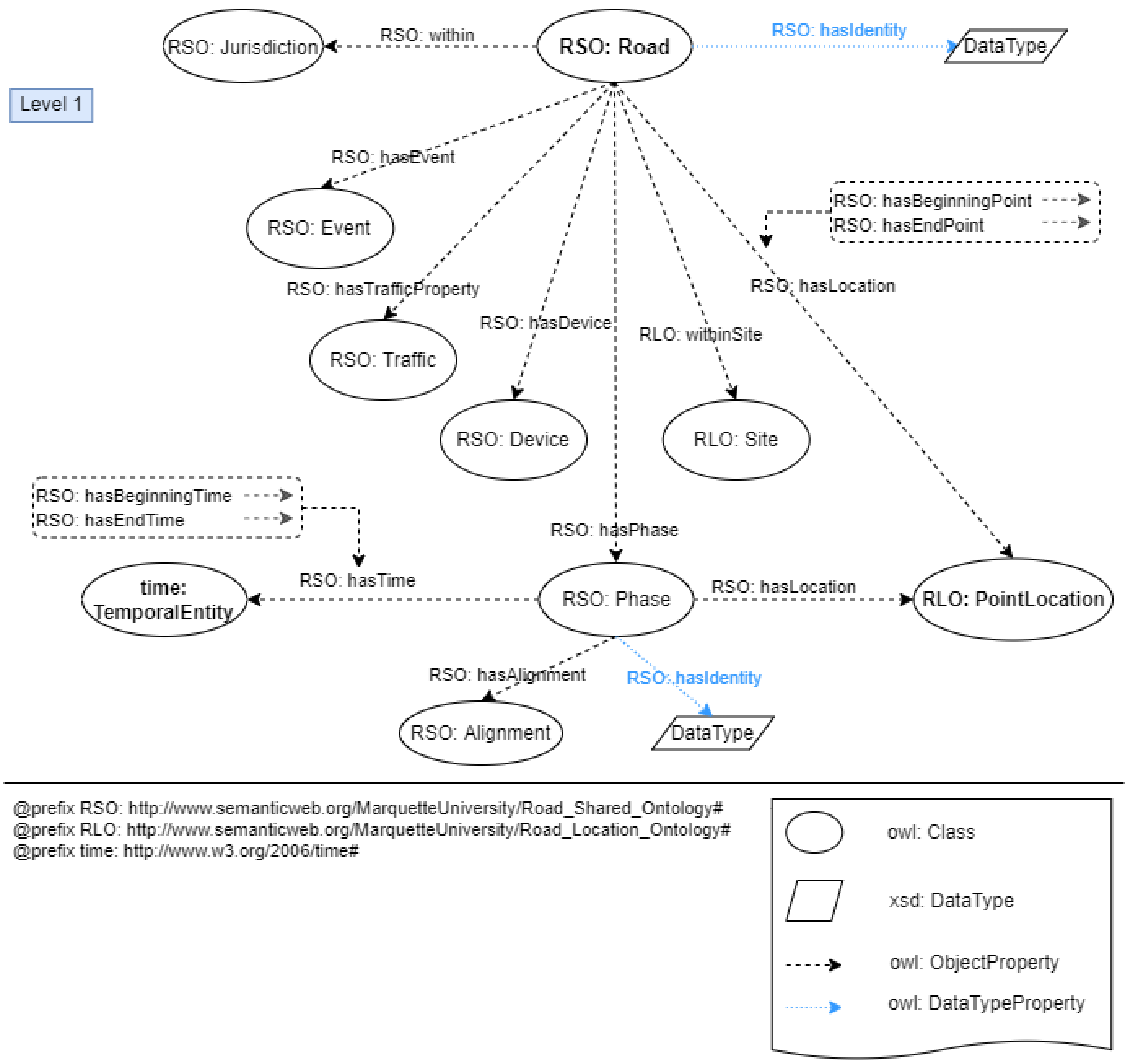


Figure 20 The Architecture of the Road Shared ontology Level 1

4.2.1.1 Jurisdiction

Regarding policies, each road should be administered under a specific legal body, usually is the local or state Department of Transportation (DOT). That is, at the same time, a road project typically only has one fixed specific legal body. In the Road Shared ontology, class RSO:Jurisdiction is placed on the first level, directly connected to class RSO:Road. The information about these legal authorities that designate roads and their names are identified in the RSO:Jurisdiction class.

4.2.1.2 Phase

Due to constraints such as budget, resource availability, policy, and environment, a lengthy road construction project can take many years and won't be built all at once. Thus, a big complex road network project is usually planned into several phases. Resources involved like the crew, machinery plant, and contractor may be totally different. Subsequently, phase is a kind of road segmentation mechanism. Class RSO:Phase defines the alignment and phase-related information throughout the design, construction, operation, and maintenance stages of the road. In addition, time information and location information are required to accurately record the temporal and location beginning and end information of a road phase, as discussed previously.

Following that, there are more details on the alignment. In appearance, alignment is typically a linear combination of lines, curves, and spirals, which provides geometrical information and defines the shape of a road. Figure 21 details the road alignment types defined in the ontology. Alignment is either horizontal or vertical. The vertical alignment is also known as a profile. Designers typically use the original ground profile to show the elevation change of the existing ground and the design profile to show that for the expected case. The horizontal and vertical alignment can be divided into straight and curve line types. For straight alignment, the computation is much simpler with fewer parameters. More details related to the parameters will be defined by designers and included in the design domain ontology (Chapter 5). Usually, creating and defining a horizontal alignment on the map is one of the first steps in roadway design.

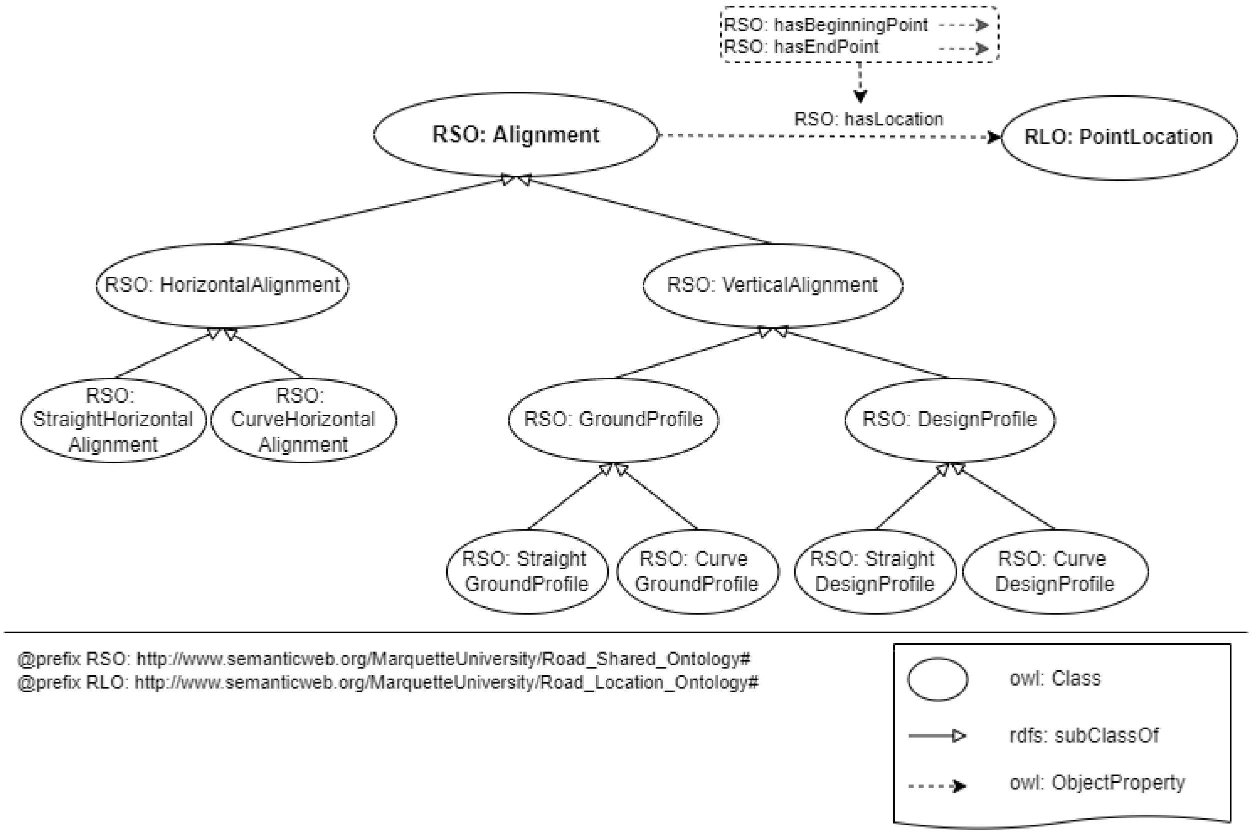


Figure 21 Top level concepts of alignment ontology

4.2.1.3 Event

Class RSO:Event represents occurrences along the road. An event must be assigned to a segment or a point along the road. Event class specifies the planned or unscheduled occurrences that happen on a road. For example, a crash accident at a spot or the close of a piece of road due to bad weather. An event can be typically divided into three types: a point event, a linear event, and an area event (Kenneth et al. 2000). A point event can be a traffic crash, a linear event may refer to a traffic jam section, and an area event can refer to a close area caused by construction activities or bad weather.

4.2.1.4 Traffic

Class RSO:Traffic defines the transportation properties along the road, including but not limited to traffic congestion, traffic flow, travel modes, and speed test. There are two main types of existing transportation-related ontologies. One type is created from the perspective of a traveler user, for example, the complete information about the nearest bus stop to a particular place or the nearest parking slots (Nandini and Shahi 2019). The other type is developed basically for public transit information, like the Public Transit Ontology (Megan Katsumi 2016). In this study, however, the class Traffic is meant to only focus on the traffic operation information (such as speed test and traffic congestion).

4.2.1.5 Device

With the advent and development of the Smart City, Intelligent Transportation Systems (ITS), and the Internet of Things (IoT), an increasing number of advanced monitoring and management devices will be installed on transportation networks. Examples include all assets installed along a road, such as light posts, traffic cameras, traffic speed and density detectors, and ramp control signals. Class RSO:Device specifies the device’s information such as location, installation date, device model, utilization time, and the road that a device is installed on.

4.2.1.6 Site

Sometimes, the site refers to the area where a road is located. Usually, only for a piece of road located on the site. A site may contain several parcels, which are typically used to represent real estate, such as parking lots in a subdivision (Autodesk 2021).

Planners and designers commonly use the site to make comprehensive planning for a specific area.

4.2.2 Level 2 Classes

Level 2 defines road elements or properties that are specified for a section of a road. The major concepts include the segment, time, location, and assembly. Figure 22 presents the architecture of Level 2.

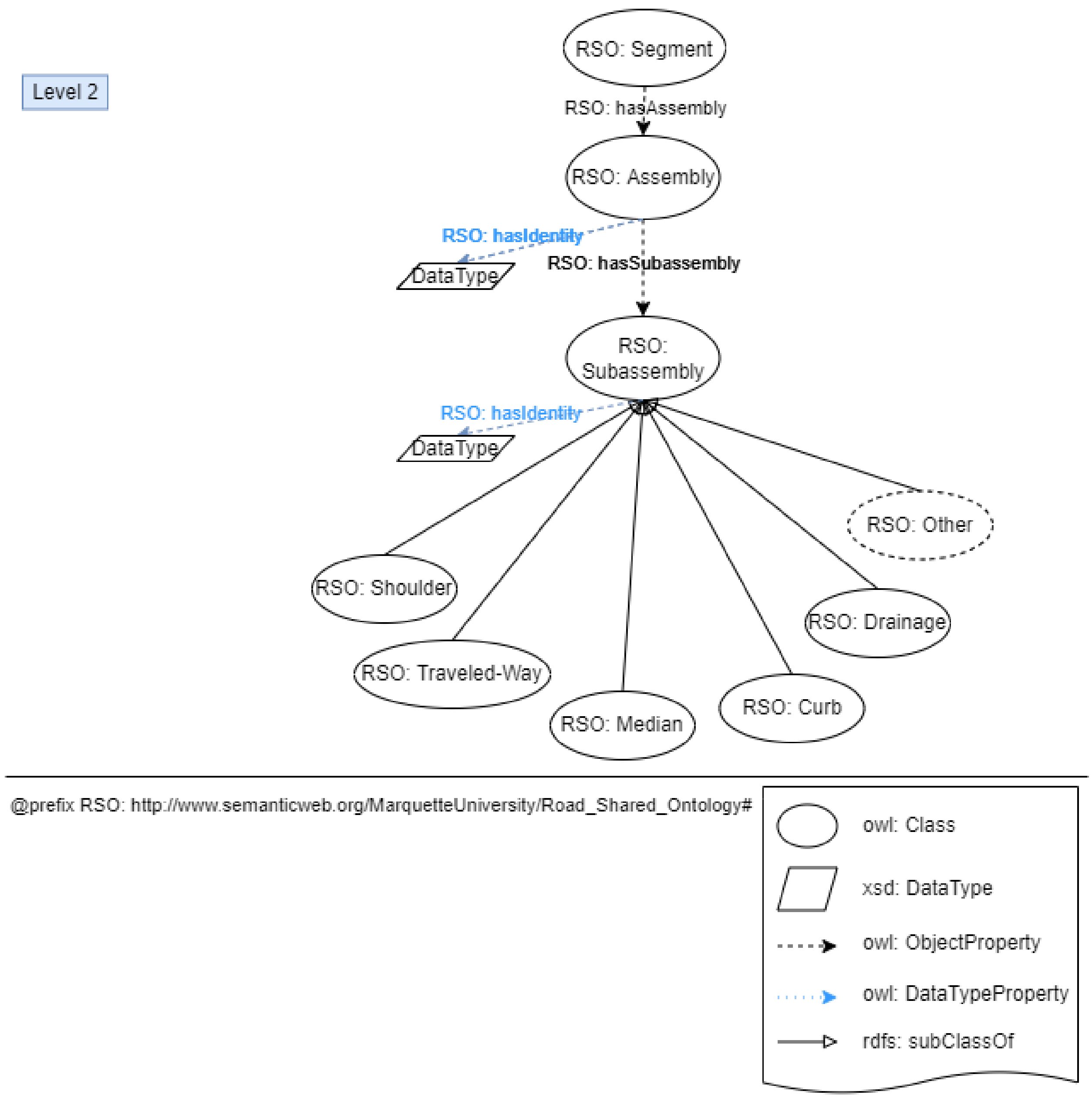


Figure 22 The Architecture of the Road Shared ontology Level 2

4.2.2.1 Segment

Road properties are assigned to class RSO:Segment on the second level instead of the class RSO:Road in Level 1. As mentioned in Section 1.1.2 (Chapter 1), a segment refers to the part of a road where the road assembly (assembly is explained in the following section) does not change. A road can have as many segmentations as the number of assemblies along the road. Since a road's cross-section elements do not stay the same, the segmentation method must be very flexible to accommodate the continuous changes in the road cross-section. For example, the condition of road pavement changes as maintenance and repair work is done. Therefore, the segmentation that represents a road's pavement condition changes over time. Assigning properties to class RSO:Segment can immensely improve adaptation to the frequently changing features of road properties and facilitate subsequent analysis and processing of road property information.

4.2.2.2 Assembly & Subassembly

Assemblies form the basic structural blocks of a 3D corridor model. The cross-sectional views created for a road model show the assemblies along the road. Figure 18 shows one example of an assembly that is composed of several subassemblies, including the traveled-way, median, and shoulder. When a single or a set of assemblies is applied along a horizontal alignment, a 3D corridor is created.

As aforementioned, each assembly is composed of several subassemblies. A subassembly can be further divided into components. For example, the subassembly traveled-way can be divided into several lanes. A lane can be further divided into pavement and subsurface. All these subassemblies and components combine to ensure a safe and efficient road project. For the sake of continuity, this dissertation follows the AASHTO standard terms and design criteria for defining the road elements for subassemblies classification. All the subassemblies used to create the 3D road model that is used in this study are taken from a Civil 3D WisDOT add-in, which is also developed according to AASHTO specifications. A list of the subassemblies is provided in Table 5 (FDM 2019; AASHTO 2011; Findley 2016). This list is not meant to be exhaustive. Different terminology, countries' classification methods, cultural languages, or new construction techniques could introduce new subassemblies. For the sake of abbreviation, only several subassemblies are listed in Figure 22.

Table 5 Subassemblies classification

Traveled-way
Outer Separation
Frontage Roads
Ramps
Grade-Separated Pedestrian Crossings
Shoulders
Subgrade Course
Surface Course
Curb
Median
Flush Median
Curbed Median
Paved Shoulder
Sidewalks
Bike Lanes
Parking Lanes
Passing Lanes
Foreslope

Backslope
Pavement
Base Course
Sub-base Course
Terrace
Roadside Barriers
Fencing
Median Barriers
Crash Cushions
Noise Barriers
Park-and-ride facilities
Bus turnouts
Bicycle Facilities
Rumble Strips
Clear Zones
Lateral Offset
Crown
Embankment
Retaining walls
Channelization
Ground Covers

4.2.3 Level 3 Classes

Level 3 further defines the components of each subassembly. The complexity of each subassembly's components varies according to its characteristics. The traveled-way subassembly, for example, can be further subdivided into one or more lanes. Each lane is composed of pavement and subsurface components, which can be further subdivided into different layers. Some subassemblies, on the other hand, are relatively simple, such as the unpaved shoulder, which is no longer subdivided into components. Figure 23 provides a general architecture of the Level 3. Again, all classes and relations are not shown. In Figure 23, only the traveled-way subassembly is expanded into its pavement and subsurface components.

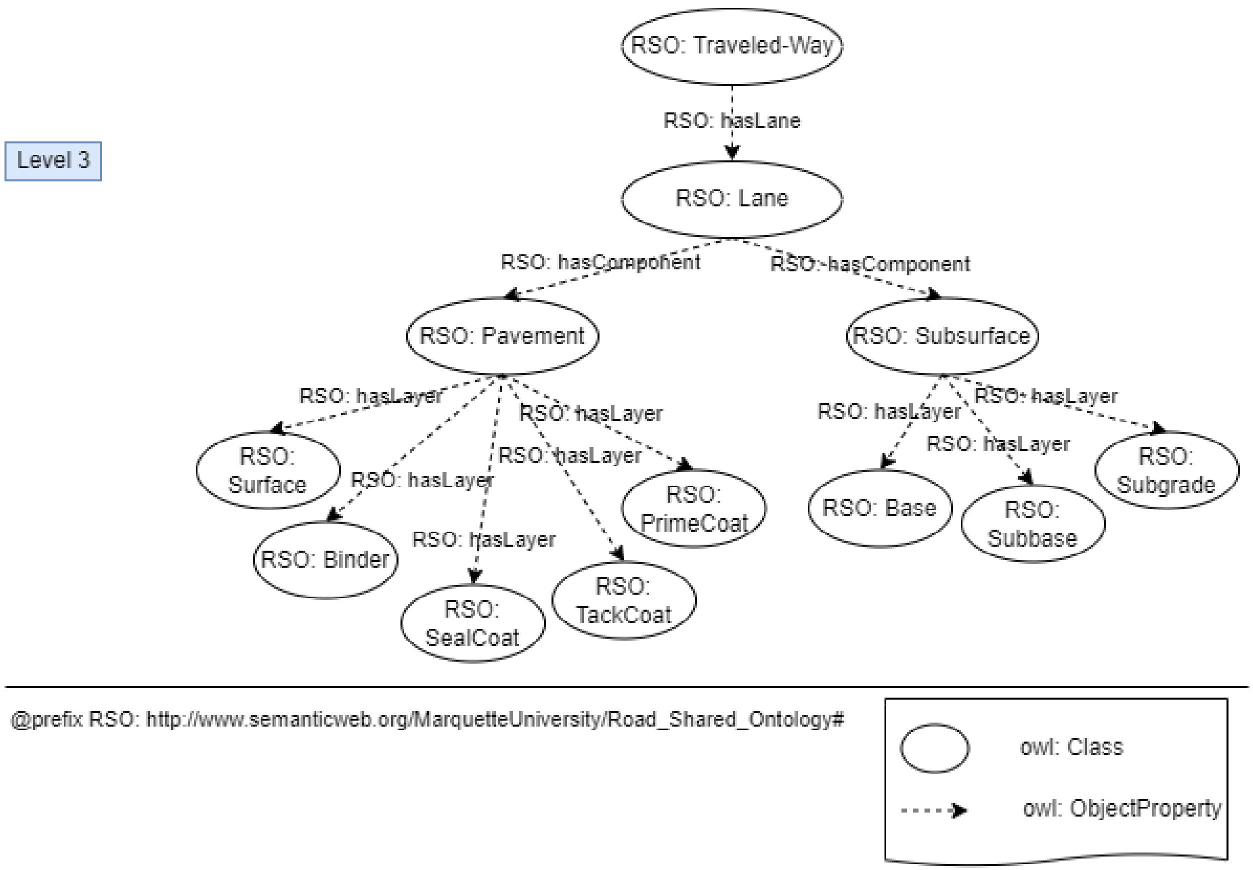


Figure 23 The Architecture of the Road Shared ontology Level 3

A pavement may have several optional user-definable layers such as surface, binder, seal coat, and prime coat; while the subsurface can be generally divided into a base course, a subbase course, and a subgrade course. In this study, all these structural layers and components' characteristics follow AASHTO specifications (AASHTO 2011).

4.3 The Road Shared Ontology Knowledge Base

A knowledge base is a repository of information created using an ontology to gather, organize, and share domain information (Noy and McGuinness 2001). Simply stated, a domain ontology, along with its instance data, is referred to as the knowledge base of a specific domain. All the knowledge bases together form the road information repository to manage all road-related project information. The ontologies developed in this study will be created and implemented in RDF/OWL format. This chapter creates a shared knowledge base by applying the Road Shared ontology to a road project. The following provides an example to illustrate the knowledge base for a road project example, which is called Road_01. In this example, the Road Shared ontology, and its instances are coded in RDF/OWL language in Protégé software (Stanford University 2015). The prefix RSO is assigned to it. The author also reused the Time and the Road Location ontologies by directly importing their IRIs into the ontology, and the prefixes time and RLO are assigned to them, respectively.

The Road model was created with Civil 3D. Figure 24 provides the general view of the road. The example road is 971.24 ft in length. It is divided into two segments and is planned to be constructed within one phase. The original ground geographical information was downloaded from the Wisconsin Department of Transportation (WisDOT) website. The subassemblies used to create this road model are standard subassemblies provided by the Civil 3D WisDOT add-in.

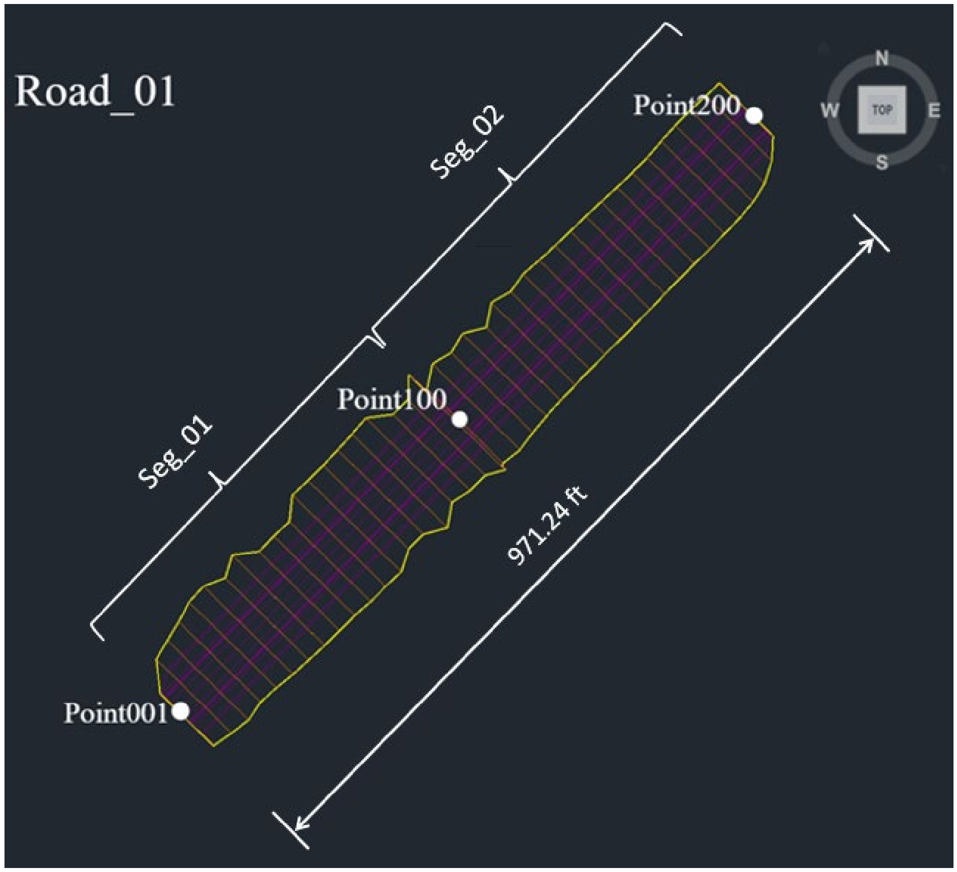


Figure 24 A general view of the Road_01 3D model

Figures 25 and 26 show schematic views of two parts of the road Road_01 knowledge base. They illustrate how instances of phases, segments, assemblies, subassemblies, alignments, time, and location of the road Road_01are defined using the Road Shared ontology vocabulary. Since the Road Shared ontology is prefixed with RSO, the IRI of each instance, property, and relationship is created by adding the identity to the IRI of the ontology. For example, the IRI of Road_01 can be presented as RSO: Road_01instead of the following IRI: http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Road_01. A list of the IRIs of all the concepts included in this shared knowledge base and their corresponding labels are shown at the bottom of each figure.

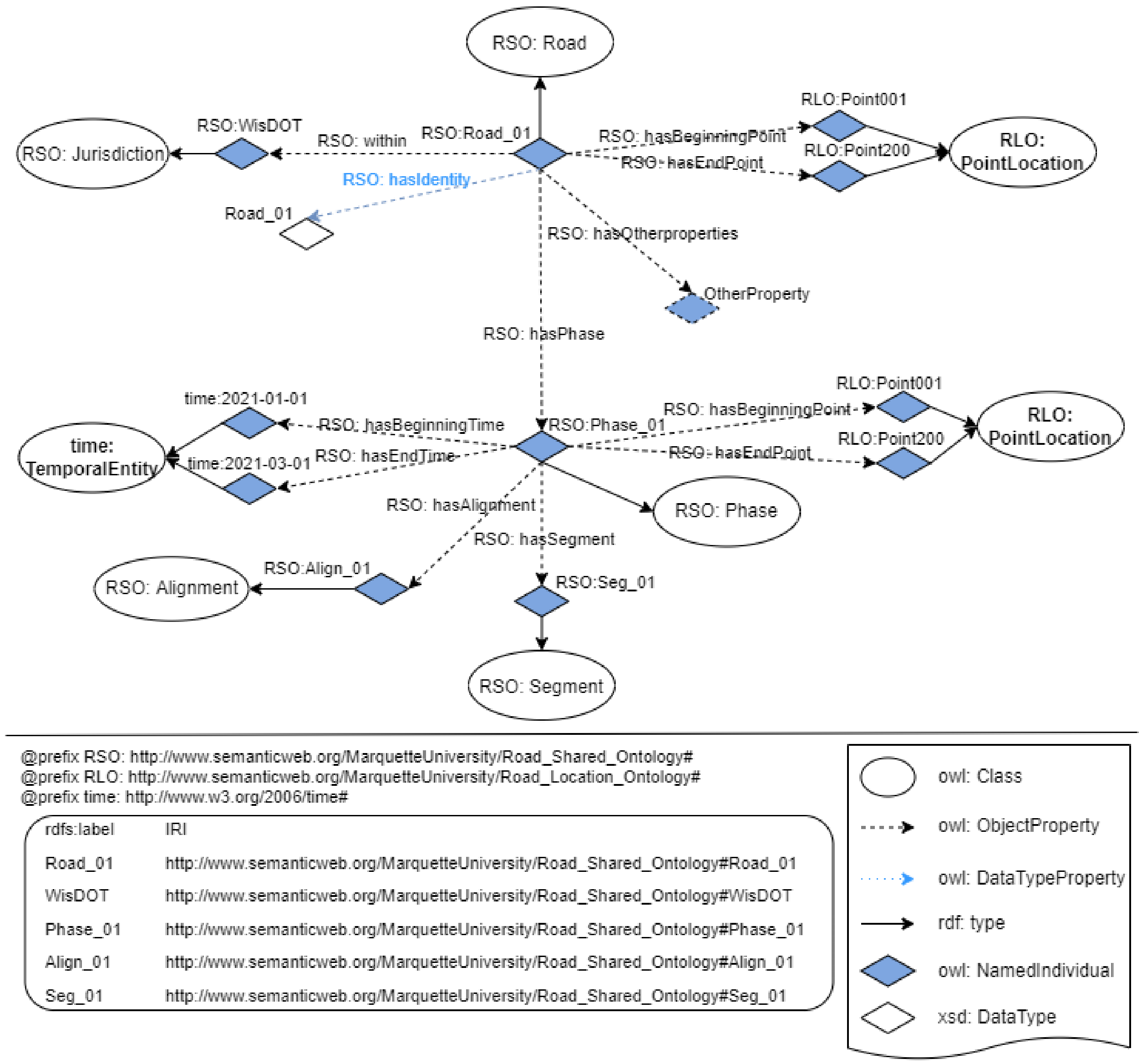


Figure 25 A general view of the Road_01 stored in the road shared knowledge base

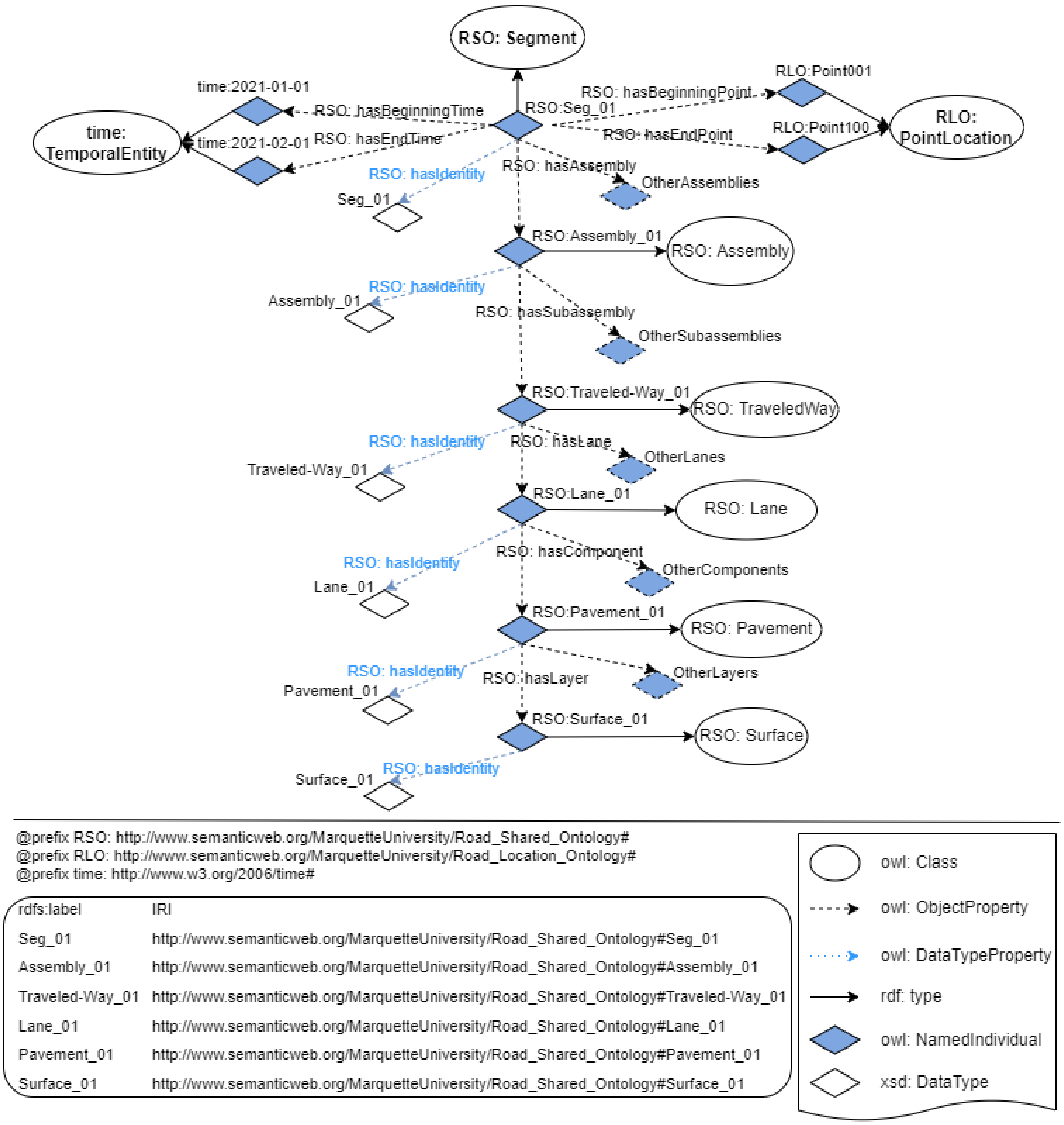


Figure 26 A general view of Seg_01 stored in the road shared knowledge base

Since an ontology not only defines concepts but also the relationships among them, the road shared knowledge base should not only represent the instances and their properties but the relationships among them as well. These relationships are named object property in OWL. In Figures 25 and 26, the object property relationships are distinguished from the data type properties with different types of dashed lines. For example, the road Road_01 has data type relationships such as the jurisdiction name, WisDOT, and the phase name, Phase 1, as well as object property relationships such as RSO:hasAlignment and RSO:hasAssembly.

The implementation of the Road Shared ontology in Protégé (Stanford University 2015) is shown in Figure 27. The left panel shows the concepts (classes), the middle panel shows the object properties, and the right panel shows the data type properties defined in the ontology.

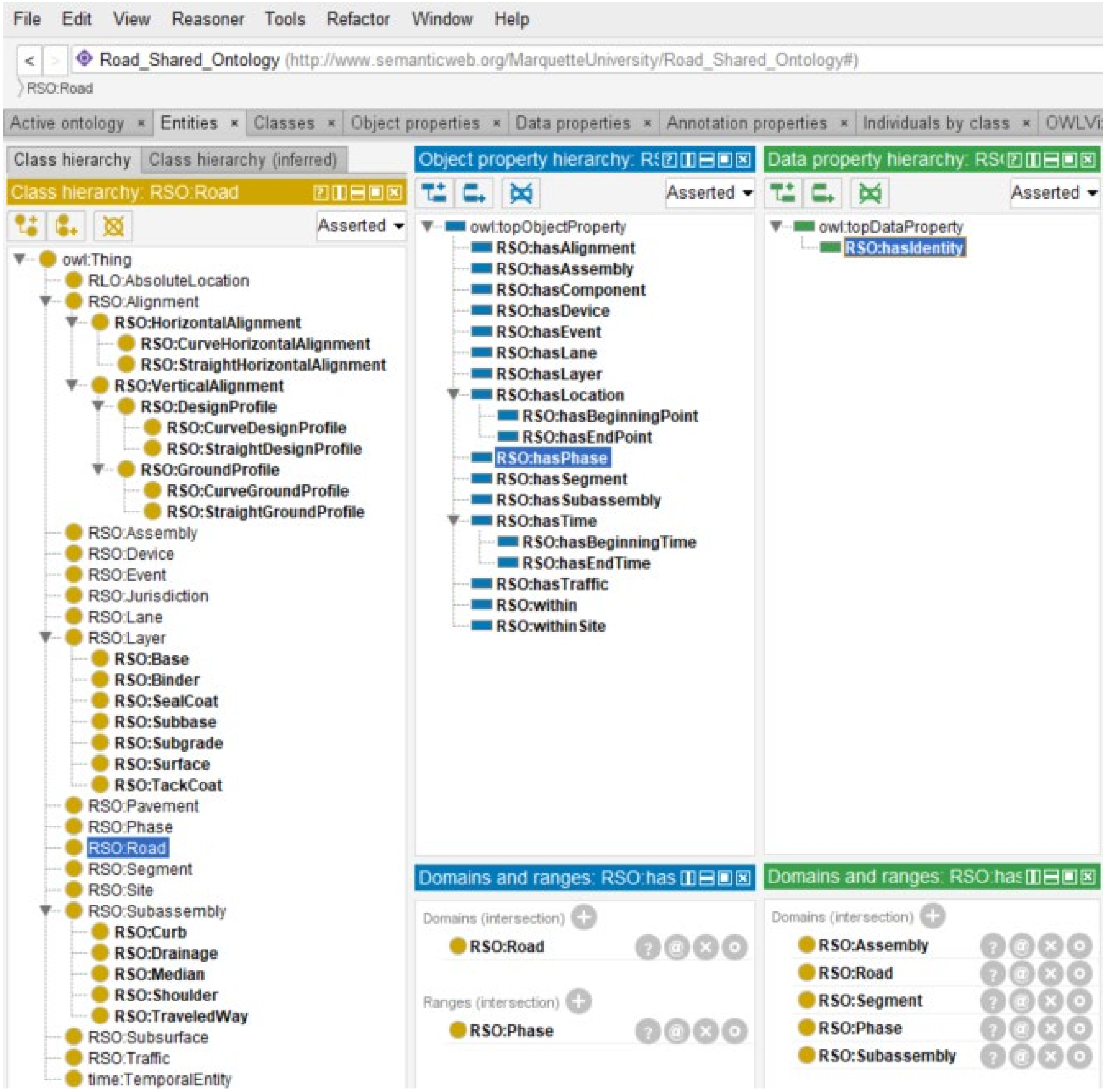


Figure 27 The Road Shared ontology implemented in Protégé

CHAPTER 5: DOMAIN ONTOLOGIES

The design domain and construction domain face similar challenges in data integration. The integration and sharing of information across domains have been hindered by heterogeneous data formats, widely divergent storage methods, and labor-intensive information management processes. Furthermore, data changes are frequent because a construction company is influenced by market and on-site conditions. For example, when market prices fluctuate significantly or construction progress is forced to be halted due to bad weather, cost estimators and schedulers must change a large amount of data. Considering these facts, machine-processable and constantly updated Semantic Web technologies are ideal for developing data models for road-related information.

By extending the shared ontology, various domain ontologies can be created. The road's domain-specific properties will be added to respective entities in each domain ontology. A domain ontology aims to provide an expandable and practical representation of the domain's shared knowledge, rather than exhaustively cataloging all the concepts within this domain (El-Diraby 2013). Put simply, a domain ontology is a conceptualized representation of knowledge in a domain that has been organized by domain experts. Design, construction, maintenance, and operation ontologies are examples of road domain ontologies. This study will be restricted to the creation of design and construction ontologies. The architecture of the Road Design ontology, Road Cost Estimating ontology, and Road Scheduling ontology developed in this study are described in Sections 5.1, 5.2, and 5.3, respectively.

5.1 Road Design Ontology

Design domain ontology is created by extending the subassembly and alignment entities defined in the Road Shared ontology by adding design domain properties to the entities.

The ontology requirements for the Road Design ontology are as follows:

- **Purpose:** the Road Design ontology is developed as a fundamental conceptual knowledge model of road design information.
- **Scope:** the scope of the Road Design ontology is limited to the alignment and subassembly concepts. The subassembly concepts were defined in Chapter 4. Alignment defines the shape and coordinates of the centerline of a road.
- **Implementation language:** the Road Design ontology is implemented in RDF/OWL language.
- **Intended end-users:** designers.
- **Intended use:** creating road design knowledge bases.

Some of the non-ontological resources used in this study include AASHTO specifications (AASHTO 2011) and the Civil 3D Developer's Guide (Autodesk 2021).

The QUDT (Hodgson et al. 2011) ontology will be reused to represent units of measurement, and the FreeClassOWL ontology (Ontology Engineering Group 2015) will be reused to classify materials.

5.1.1 Subassembly

A subassembly is a component that must be completed to construct a road’s assembly. Some are simple and represent one workitem. For example, the road curb and noise barriers. Some subassemblies are complex and consist of several workitems. For example, the traveled-way subassembly is composed of lanes, which can be divided into pavement and subsurface, and each of them is composed of several layers. The Road Design ontology collects specific knowledge related to the design details of each subassembly, such as the dimensions and the material that designers specify for the subassembly. The design ontology provides an explicit conceptual model for the design properties of the subassembly defined in the Road Shared Ontology. Figure 28 depicts the architecture of the subassembly part of the Road Design Ontology. Due to space constraints, not all classes and relations are illustrated. The ontology will be described briefly in the sections that follow.

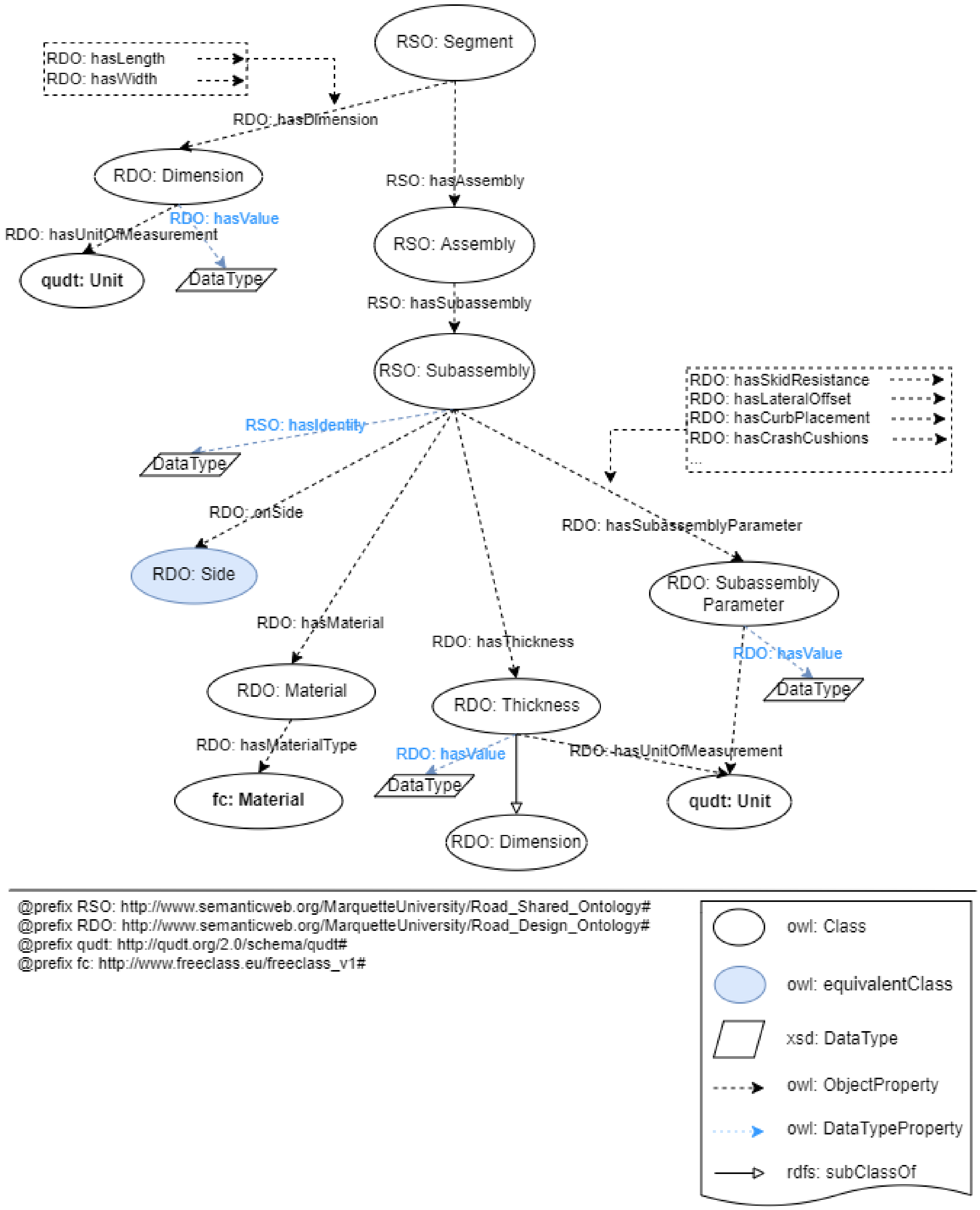


Figure 28 Top level architecture of the design ontology

Most local transportation authorities use design criteria and specifications derived from the American Association of State Highway and Transportation Officials (AASHTO) (FDM 2019; AASHTO 2011), though design code standards developed by each authority vary depending on their requirements and actual situations. All assembly and subassembly terms used in this study adhere to the AASHTO classification system, which was discussed in Chapter 4. In addition to customizing a brand-new subassembly, some built-in standard subassemblies are available in Civil 3D. Some DOT departments also developed subassembly add-ins for Civil 3D, which are created based on their local specifications.

In this study, the prefix RDO is assigned to the Road Design ontology, which equals its full IRI: http://www.semanticweb.org/MarquetteUniversity/Road_Design_Ontology#. Since the Road Design ontology is extended from the Road Shared ontology, the Road Shared ontology is reused in the design domain as the start point. The related ontology IRIs are listed in Figures 27 and 28. Since the Road Design ontology is prefixed with RDO, the IRI of each class, property, and relationship is created by adding the identity to the IRI of the ontology. For example, the IRI for a subassembly shoulder can be represented as RDO:Shoulder instead of the full IRI which is http://www.semanticweb.org/MarquetteUniversity/Road_Design_Ontology#Shoulder.

The dimension and material of a subassembly are defined in the design ontology. The class RDO:Dimension defines the information such as width, depth, and length. The RDO:Material defines the type of material used for each subassembly. In this study, the Free Class OWL ontology (Ontology Engineering Group 2015) is reused to provide material-related information. The Free Class OWL ontology is derived from the free classification standard freeClass (freeClass 2022) to describe building materials and services (Ontology Engineering Group 2015). The IRI for the Free Class OWL ontology is http://www.freeclass.eu/freeclass_v1.owl and is prefixed with fc. Since a subassembly can have more than one subdivided components, as a result, one or more materials are attached to each subassembly.

5.1.2 Alignment

As discussed in Chapter 4, alignment provides geometrical information for the road centerline. The Road Shared ontology defines the top-level architecture of the alignment ontology, which is shown in Figure 21. Road Design ontology specifies the design details for the alignment. It provides an explicit conceptual model for the alignments' properties and relationships.

The alignment typically represents the road's geometrical information with 3D coordinates, like LandXML (LandXML.org 2017), or a set of parameters plus the 3D coordinates of the beginning point (Kavanagh and Glenn 1992). The former method stores the coordinates of points along a road, while the latter method uses the geometric parameters to compute point location based on the 3D coordinates of the beginning point. Obviously, the latter method stores the road's geometrical information in a more compact manner. Thus, in this study, the author employs the parameters method.

Figures 29 and 30 show the architecture of the alignment ontology. Figure 29 depicts the primary parameters and their fundamental relationships used to define the straight and curve type horizontal alignments. For example, the straight horizontal alignment can be defined with the parameters of length and direction angle. The Road

Design ontology includes both the azimuth and bearing forms of direction angle.

Figure 30 depicts the primary parameters and their relationships for defining straight and curve ground and design profiles, respectively. For example, the straight type profile can be defined with start height and start gradient parameters. Table 6 provides a list of the main alignment computation parameters and their corresponding abbreviations.

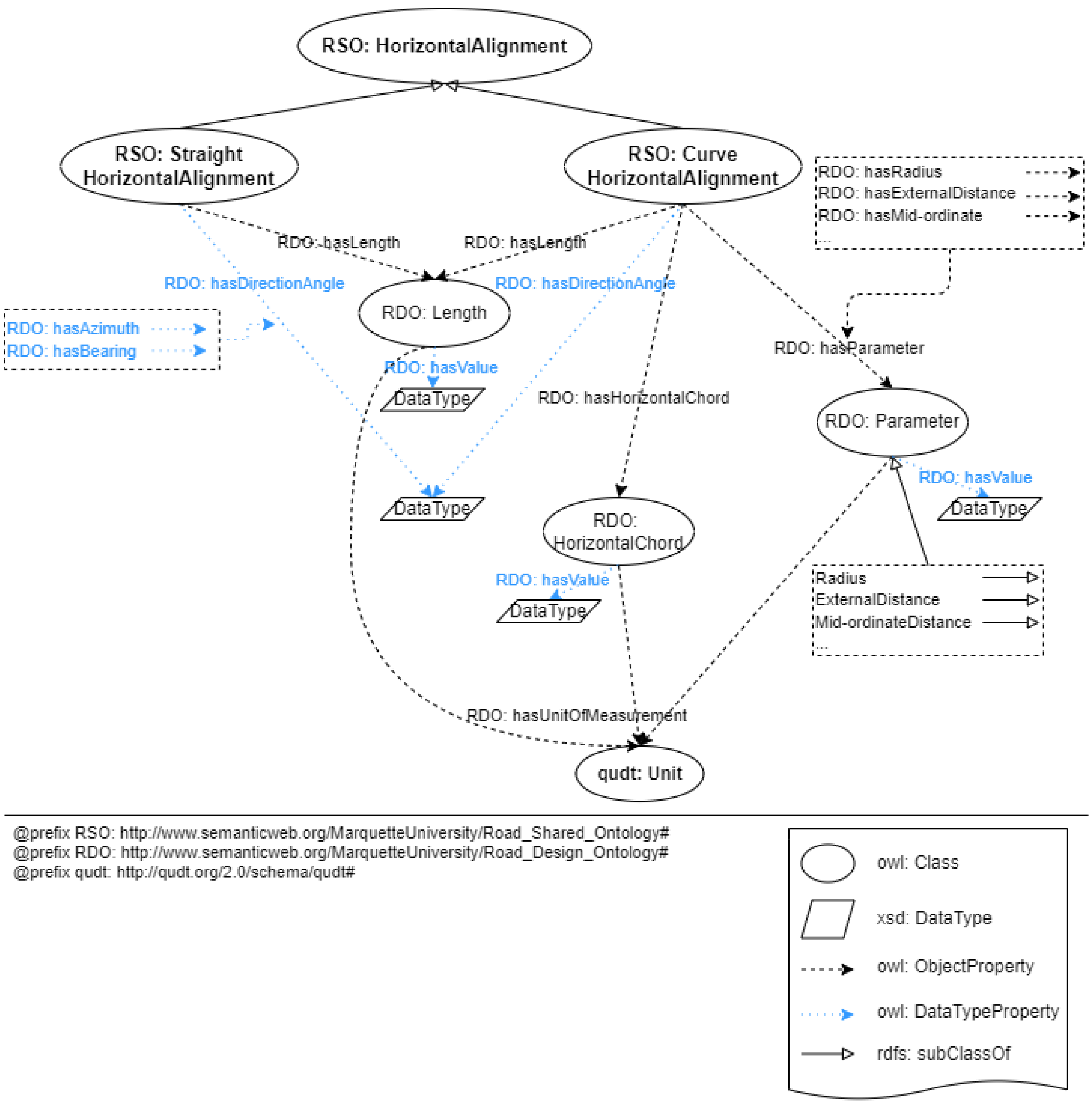


Figure 29 A general view of horizontal alignment ontology architecture

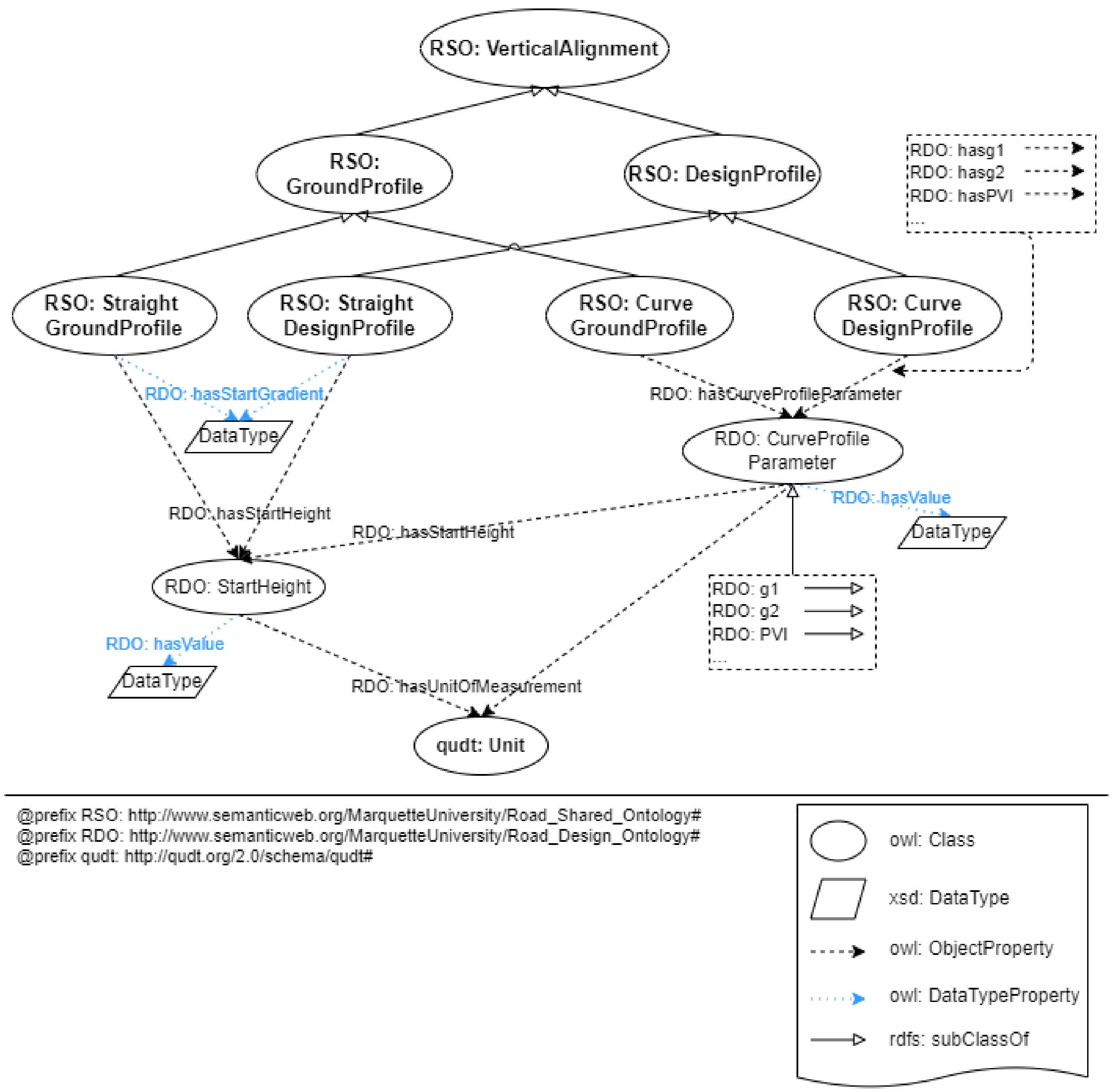


Figure 30 A general view of vertical alignment ontology architecture

Table 6 A sample of curve type alignment parameters (Kavanagh and Glenn 1992)

Alignment Parameters	Abbreviation
Horizontal Alignment Parameters	
Point of Tangent Intersection	PI
Beginning of Curve	BC

End of Curve	EC
Radius	R
Long Chord	C
Mid-Ordinate	M
External Distance	E
Tangent Length	T
Length of Curve (or projection of the vertical curve onto a horizontal surface)	L
Deflection Angle (central angle of the curve in degrees)	Δ
Vertical Alignment Parameters	
Slope (percent) of the lower chaingage grade line	g1
Slope (percent) of the higher chaingage grade line	g2
Algebraic change in slope direction	A
Distance from the PVC to the high/low Point	x
Beginning of the Vertical Curve	BVC
End of the Vertical Curve	EVC
Point of intersection of the two adjacent grade lines	PVI
Point of vertical curvature	PVC
Point of vertical tangency	PVT
Horizontal distance required to effect a 1% change in slope on the vertical curve, K=L/A	K

5.1.3 Design Knowledge Base

Figure 31 provides a general architecture of an assembly employed in one segment of the Road_01, which is stored in the Road Design knowledge base. The example assembly is composed of subassemblies: multilayer Traveled-Way (Traveled-Way_01), multilayer shoulders (Shoulder_01), and standard type daylight (daylight_01). The Traveled-Way_01 has two lanes on each side of the road, and both lanes have a width of 12 ft. Only the Lane_01 is expanded as an example in Figure 32. The Lane_02 can be represented in the same way. The magnified structure of the lane and thickness of each layer is shown at the bottom of Figure 31. Figure 32 expands the lane with a pavement component example, Pavement_01. The other components such as the base and subbase can be represented in the same way. The pavement can be constructed with multiple layers, such as surface layer, binder layer, and seal coat layer. Here, only expand the Surface_01 as an example. The material used to construct the Surface_01 is ready-mixed concrete, which is coded as 12-05-05-05 in freeClass classification system (freeClass 2022) and the other layers can be represented in the same way.

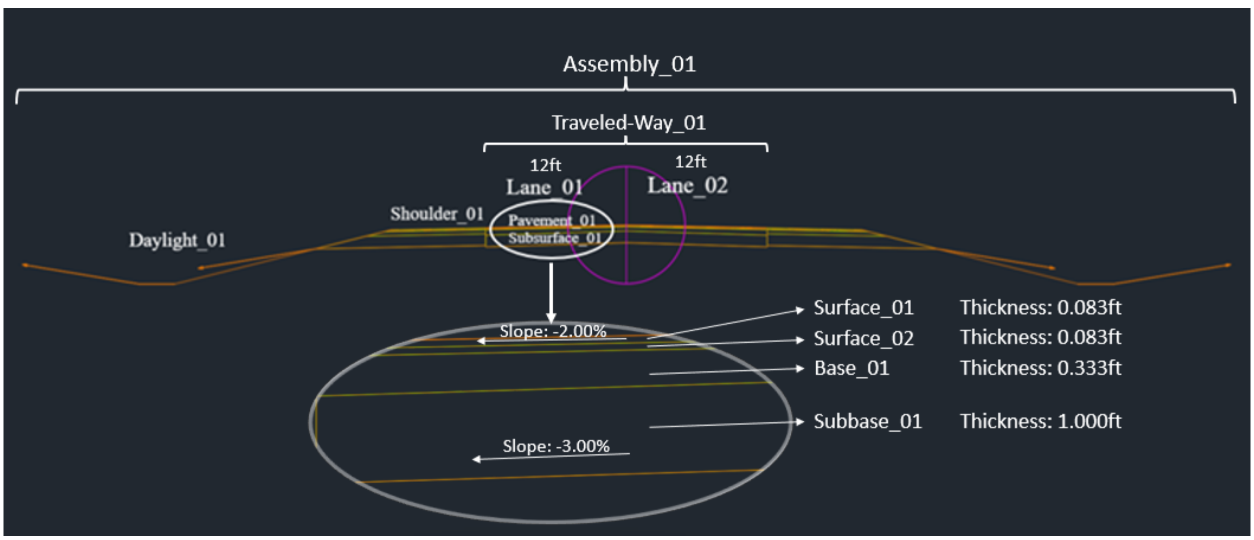


Figure 31 An assembly example created in the Road_01 model

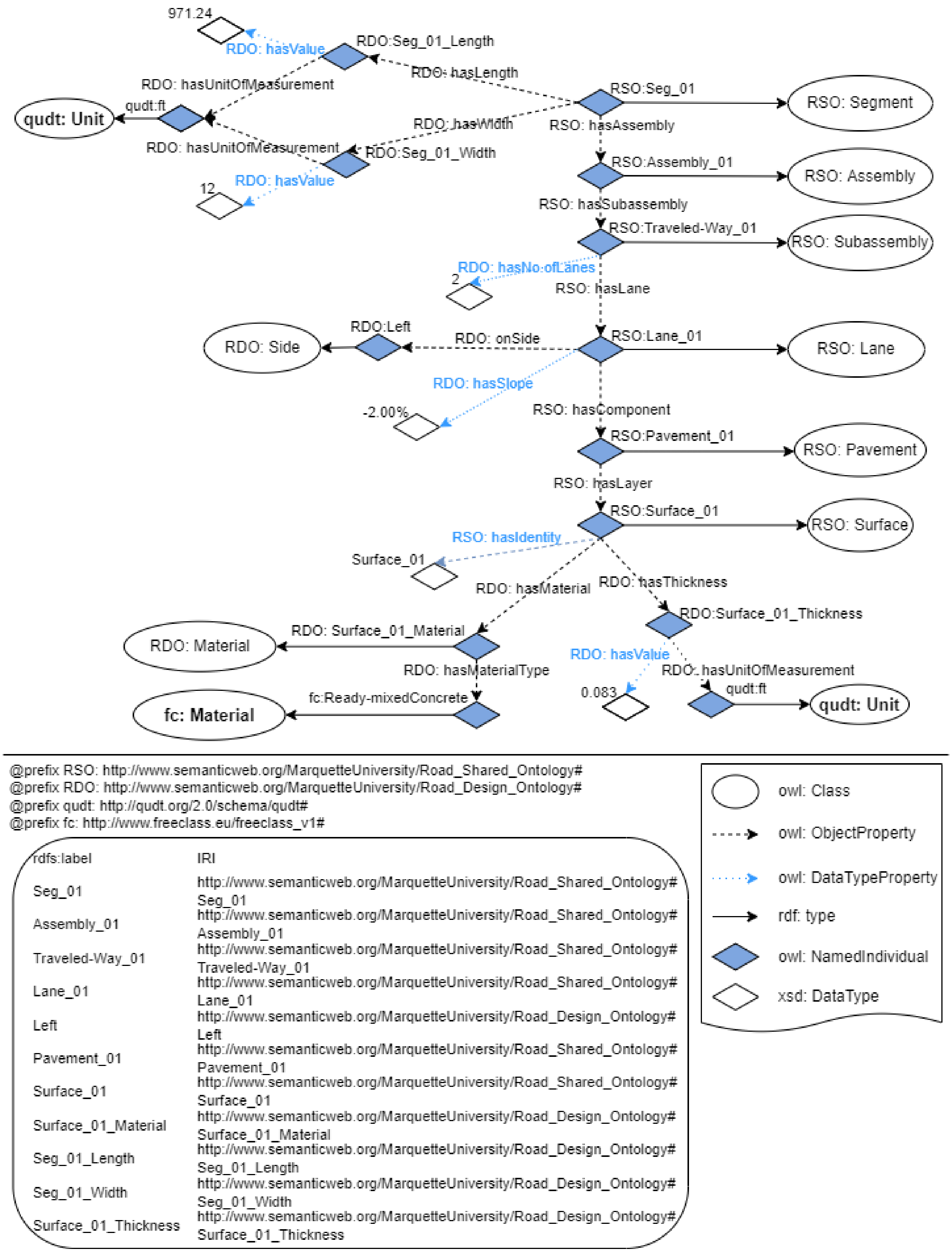


Figure 32 Surface_01 example from the road design knowledge base

Given that each alignment usually involves a different collection of parameters due to varying levels of complexity, a knowledge base example for a typical road alignment is not given here.

The implementation of the Road Design ontology in Protégé (Stanford University 2015) is shown in Figure 33. The left panel shows the concepts (classes), the middle panel shows the object properties, and the right panel shows the data type properties defined in the Road Design ontology.

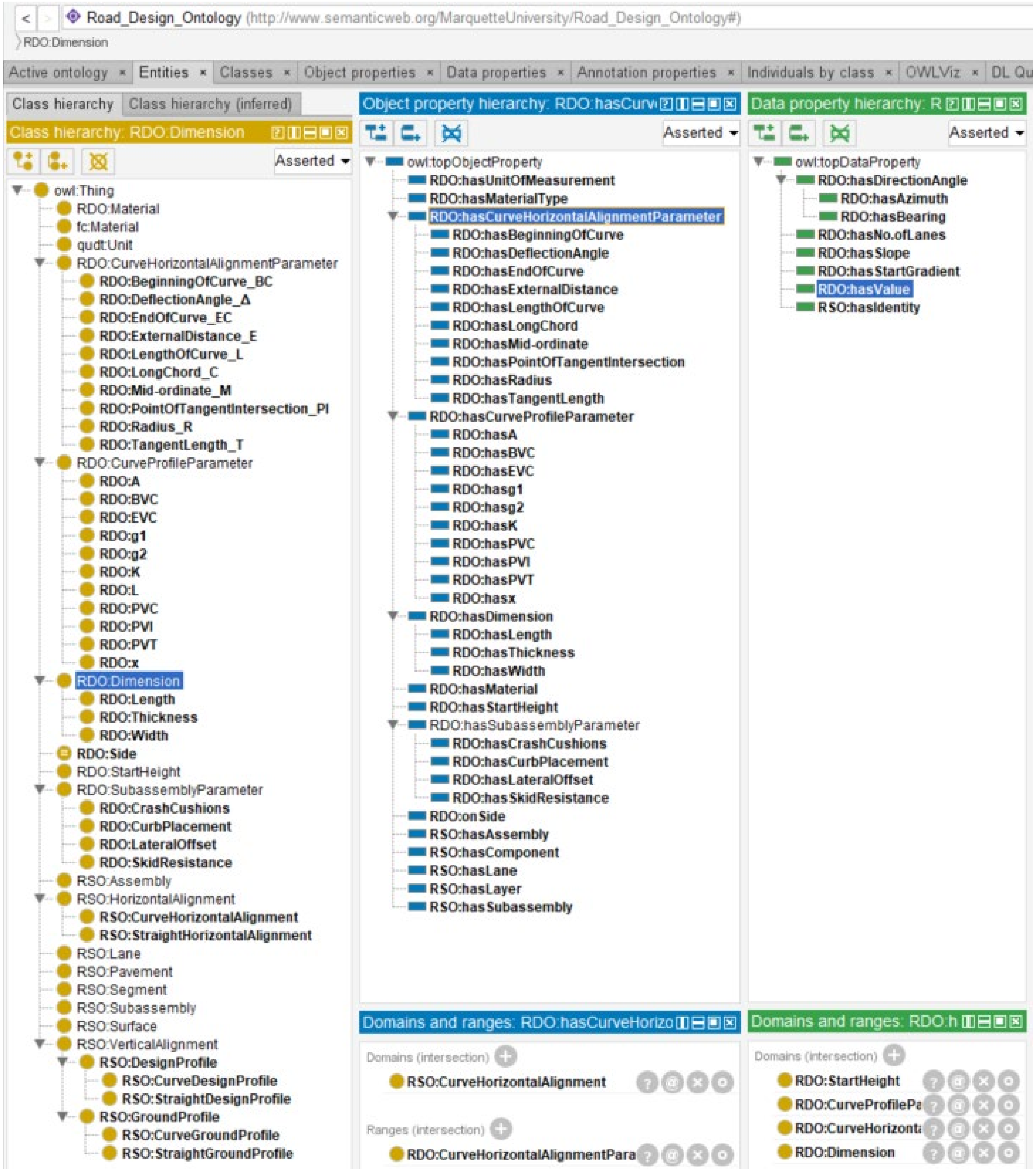


Figure 33 The Road Design ontology implemented in Protégé

5.2 Road Cost Estimating Ontology

Figure 34 shows the top level of construction domain ontology. This section will focus on the development of the Road Cost Estimating ontology. The Road Scheduling ontology will be explained in the following Section 5.3. The development of ontologies for the other construction-related domains, such as the procurement domain, is not included in this study.

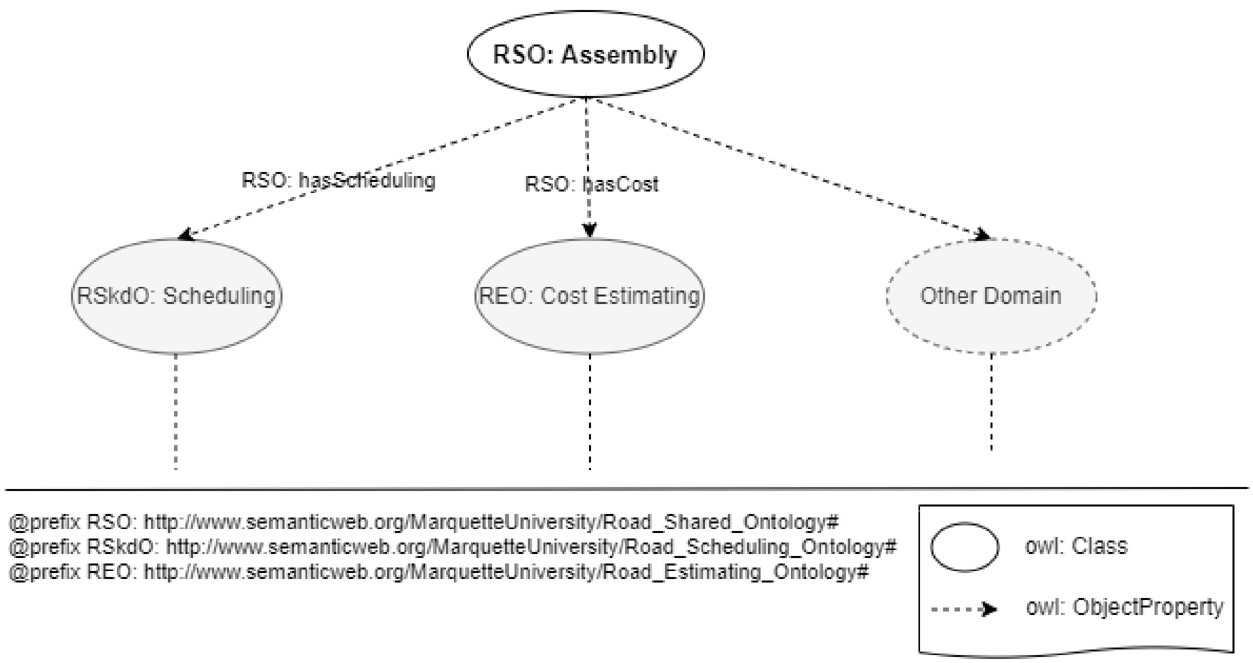


Figure 34 A general view of the construction ontology architecture

Cost estimating is a critical sub-domain in the construction domain and is the primary manifestation of a road project's economic properties. Throughout the life cycle of a road project, the construction domain relies on data from other domains such as design, scheduling, procurement, and maintenance. Semantic web technology has been used to model building construction cost data (Niknam and Karshenas 2015). This study will concentrate on road projects and employ Semantic Web technology to develop a specific road cost ontology to provide a semantic model for the road assemblies defined in the Road Shared ontology.

The ontology requirements for the Road Cost Estimating ontology are as follows:

- **Purpose:** the Road Cost Estimating ontology is developed as a conceptual knowledge model for road cost estimating information.
- **Scope:** the scope of the Road Estimating ontology is limited to the construction cost data of a road's assemblies.
- **Implementation language:** the Road Estimating ontology is implemented in RDF/OWL language.
- **Intended end-users:** cost estimating knowledge developers.
- **Intended use:** creating a road construction cost estimating knowledge base.

Currently, there is no semantic model for representing road construction cost data. The assembly-related concepts in the Road Estimating ontology are classified and organized according to the AASHTO (AASHTO 2011) classification system.

Additionally, the QUDT (Hodgson et al. 2011) ontology is reused to represent units of measurement, the FreeClassOWL ontology (Ontology Engineering Group 2015) is reused to classify materials, and the organization ontology (Ontology URI: http://www.w3.org/ns/org#) developed by W3C is reused to represent the organization of the responsible party involved in conducting a work item (W3C 2014).

In this study, the prefix REO has been assigned to the Road Cost Estimating ontology, which is the abbreviation for the full IRI: http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology#. Figure

35 shows the architecture of the Road Cost Estimating ontology.

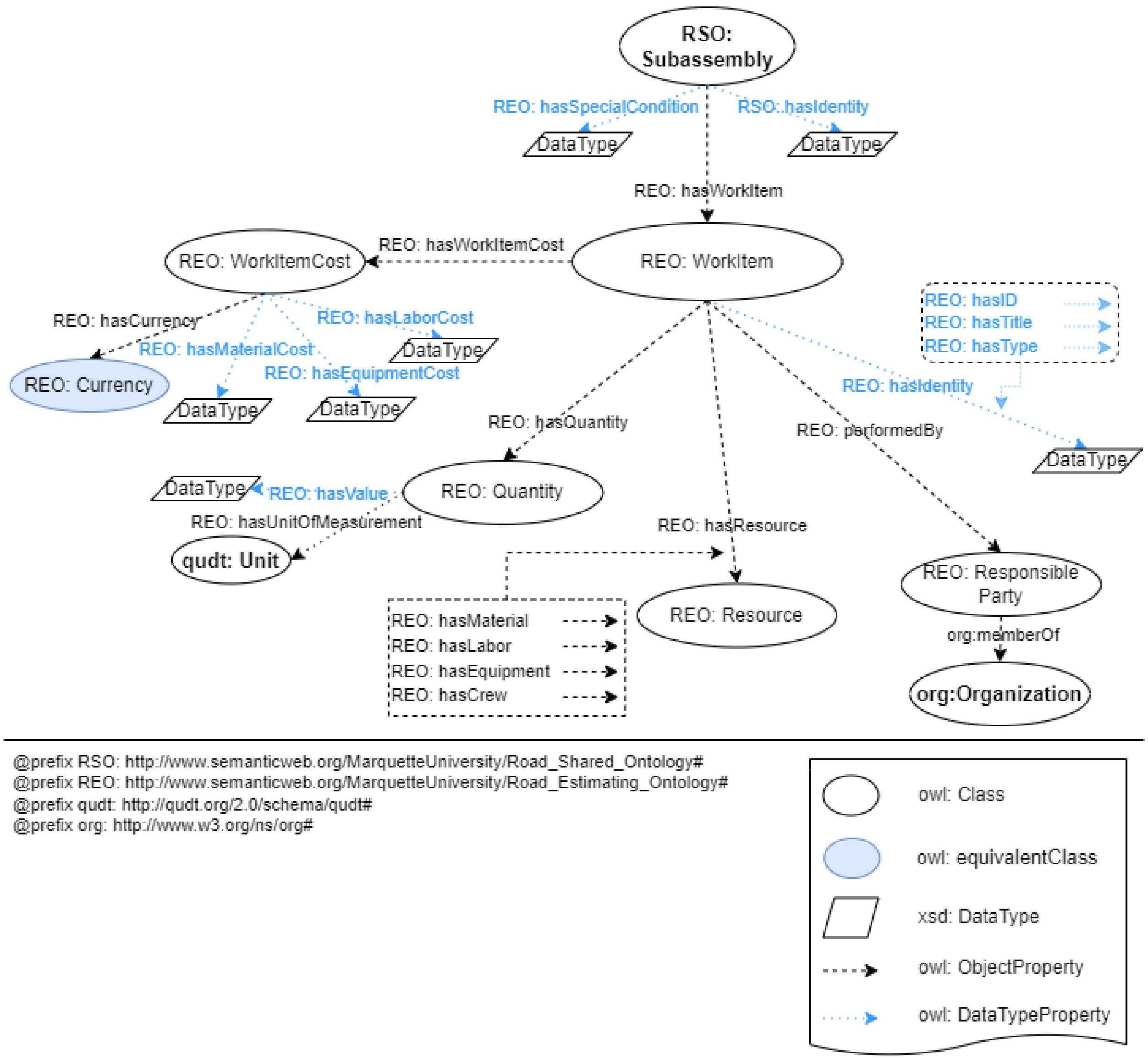


Figure 35 A general view of the architecture of the Road Cost Estimating ontology

As aforementioned, some subassemblies can have more than one component and multiple layers. In this case, each component and layer, rather than only the subassembly, should specify its own work items as well. The subassembly's type and identity are defined in the Road Shared ontology.

The Road Cost Estimating ontology adds the properties such as a list of work items, work item cost, work item quantity, resources involved, work responsible party, and special conditions. These concepts are explained as follows:

- **Identity**: each work item should be specified with its type, ID, and title.
- **Special conditions:** refer to the job conditions that can affect work item production rate, such as budget and weather conditions.
- **Work Item:** the list of the work items is obtained from estimating references such as RSMeans reference books (RSMeans 2015).
- **WorkItemCost**: defines the estimated cost of a work item. The cost should specify the value and the currency used.
- **Resource**: the resource types include material, labor, equipment, and crew. The Resource class defines the resource type and resource unit price. The Crew class also specifies the production rate, which is obtained from the road scheduling ontology. Figure 36 depicts the resource ontology's architecture.
- **Quantity**: the quantity is calculated based on the dimension data obtained from the design domain. It defines the work item quantity with the value and the unit of measurement.
- **Organization**: defines the responsible parties involved in the construction of a work item, such as the contractors, material suppliers, and the inspecting organizations. The organization ontology (Ontology URI: http://www.w3.org/ns/org#) developed by W3C is reused here (W3C 2014).

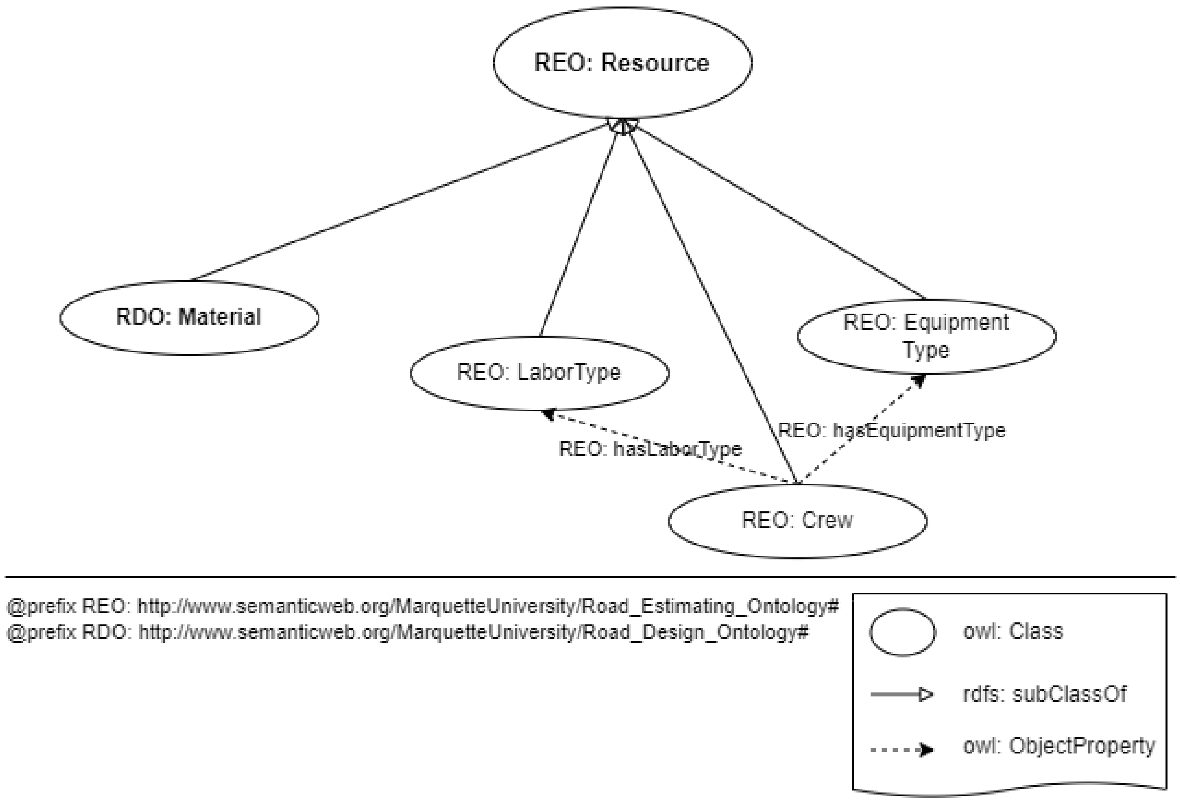


Figure 36 Architecture of the resource ontology

As shown in Figure 36, labor, equipment, and material are subclasses of resources. Some work only involves labor, such as painting the traffic signs; while others involve labor and equipment, such as hauling concrete. The group of labor and equipment used to construct a work item is called a crew, which is also classified as a subclass of class REO:Resource. The architecture of each of them are shown in the following figures from Figure 37 to Figure 40.

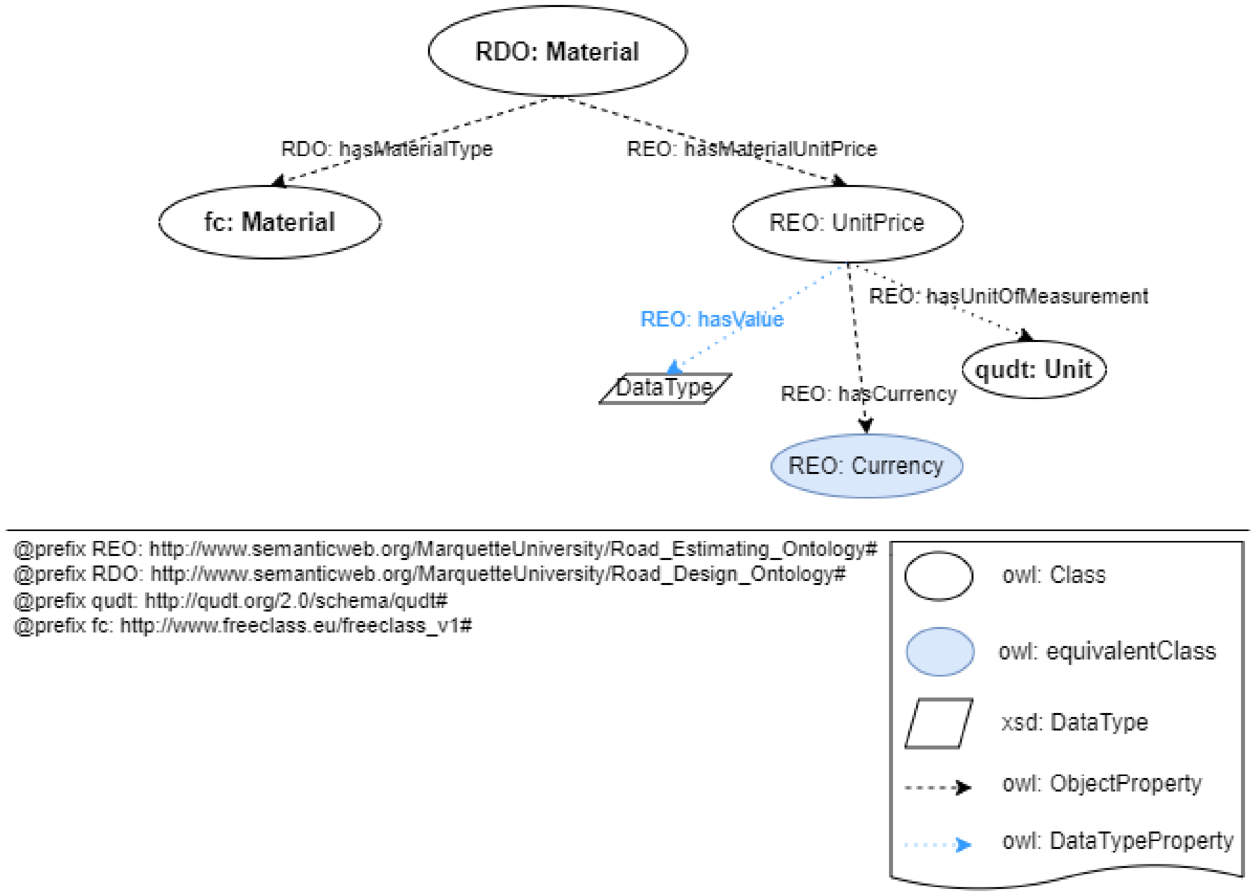


Figure 37 Architecture of the material ontology

The material ontology defines the material type and the material unit price. Here, the FreeClassOWL ontology (Ontology Engineering Group 2015) is reused to classify the material type. The material unit price is defined by the value, the currency used, and the unit of measurement. In this study, the currency class is developed as an enumerated class, which is also known as the equivalent class in the Protégé (Stanford University 2015). Users can pick up the currency on demand rather than enter the currency name by themselves. A set of common currencies, such as the US dollar (USD), the Yuan (CNY), and the Japanese Yen (JPY), have been developed as instances of the currency class in the material ontology.

The labor ontology defines the specialty and the unit price of the labor. Figure 38 provides the semantic representation of the architecture of the labor ontology. The specialty of the labor is created as an equivalent class (such as laborer, carpenter, and equipment operator) to provide preset options for users.

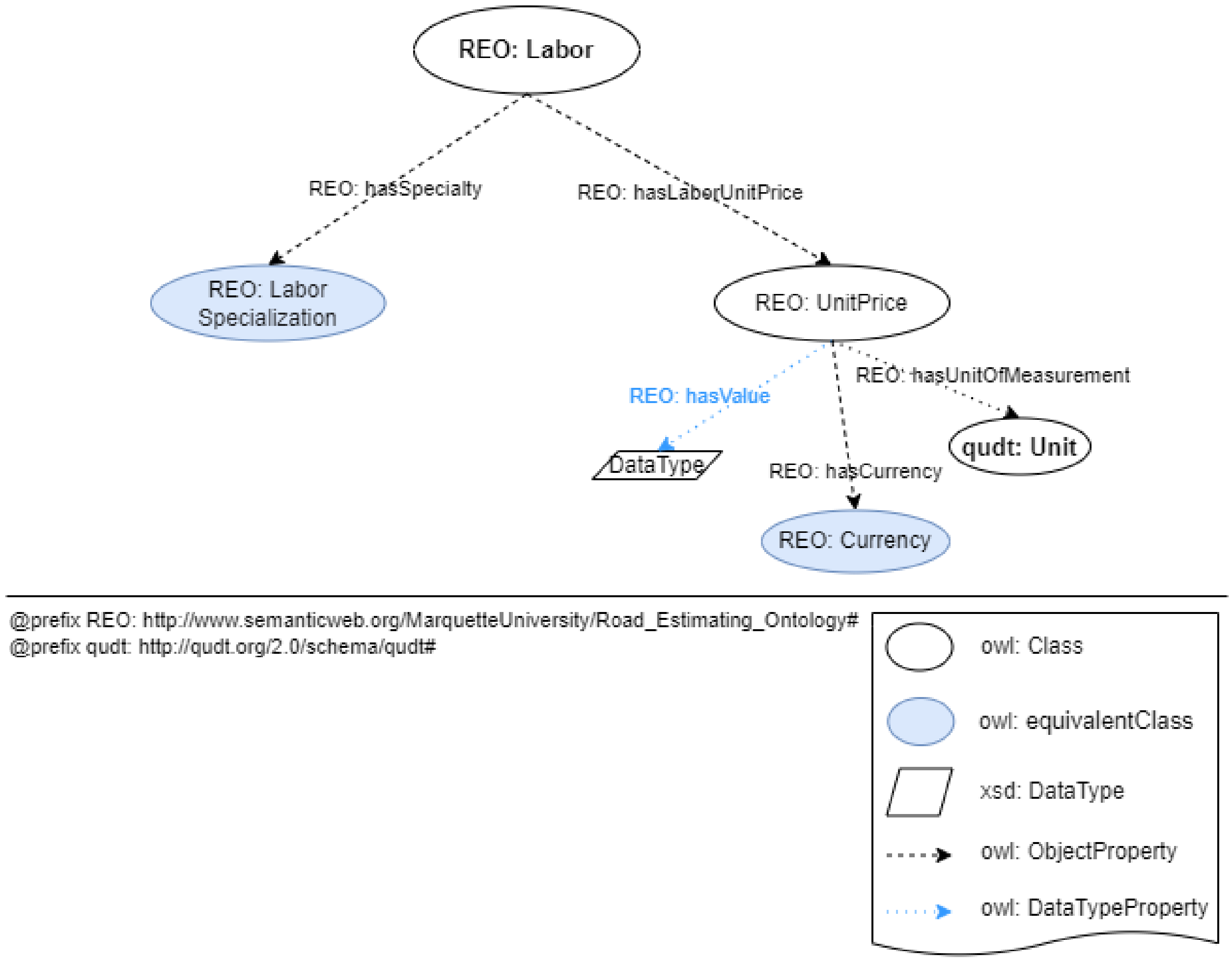


Figure 38 Architecture of the labor ontology

Similarly, the equipment ontology defines the equipment type and the unit price of the equipment. The equipment type is defined with two data type properties: the equipment model and the equipment manufacture. Figure 39 provides the semantic representation of the architecture of the equipment ontology.

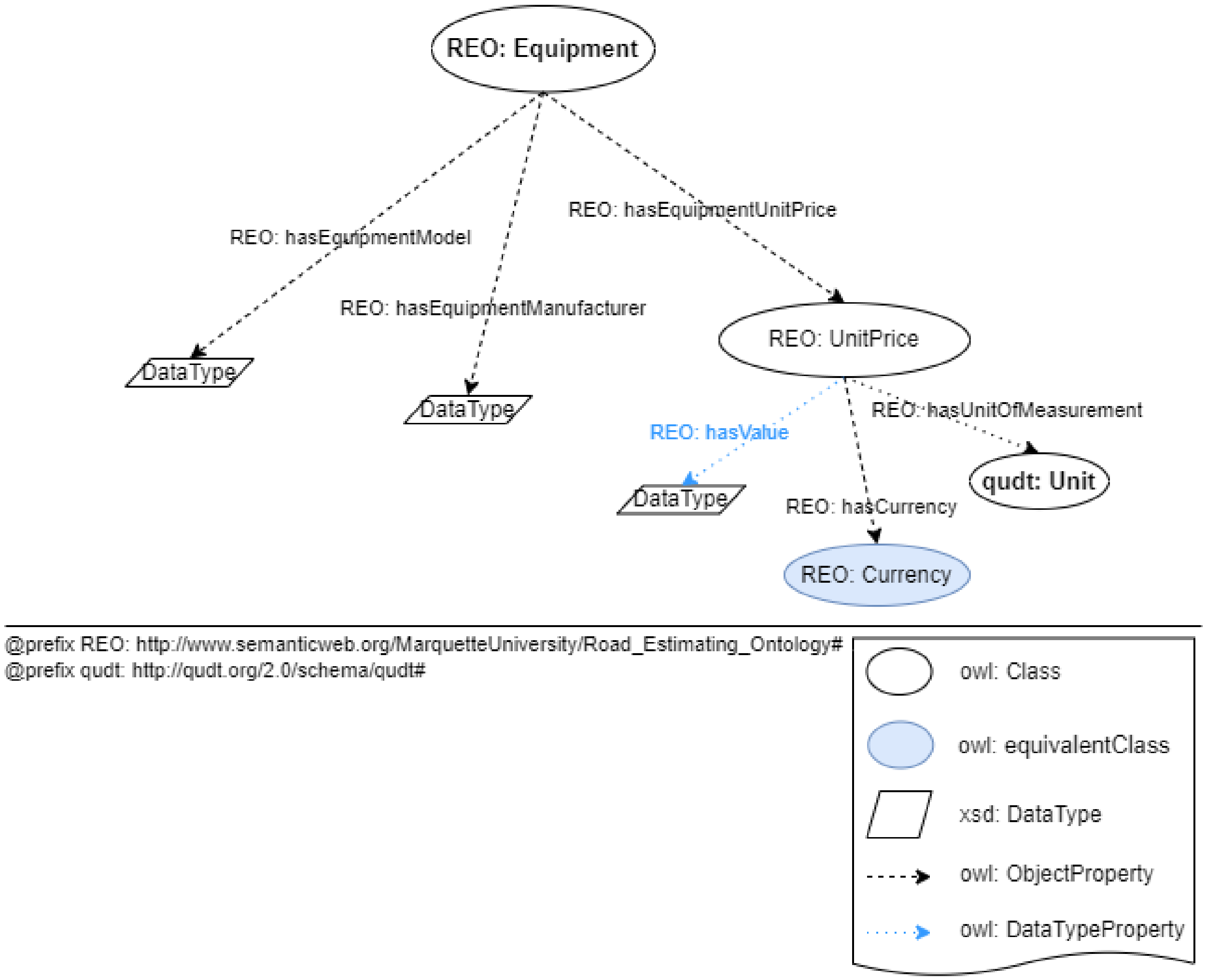


Figure 39 Architecture of the equipment ontology

The number of each type of labor and equipment required for a job is defined in the crew ontology. The crew ontology also defines the unit cost of labor and equipment and the production rate of the crew. The unit labor cost of a crew is the total unit cost of all types of labor. For example, suppose there are 2 laborers and 2 carpenters in a crew. In that case, the unit labor cost of this crew is the summary of the unit price of the laborer multiplied by 2 and the carpenter multiplied by 2. Likely, the unit equipment cost of a crew is the total unit cost of all types of equipment. Figure 40 provides the architecture of the crew ontology.

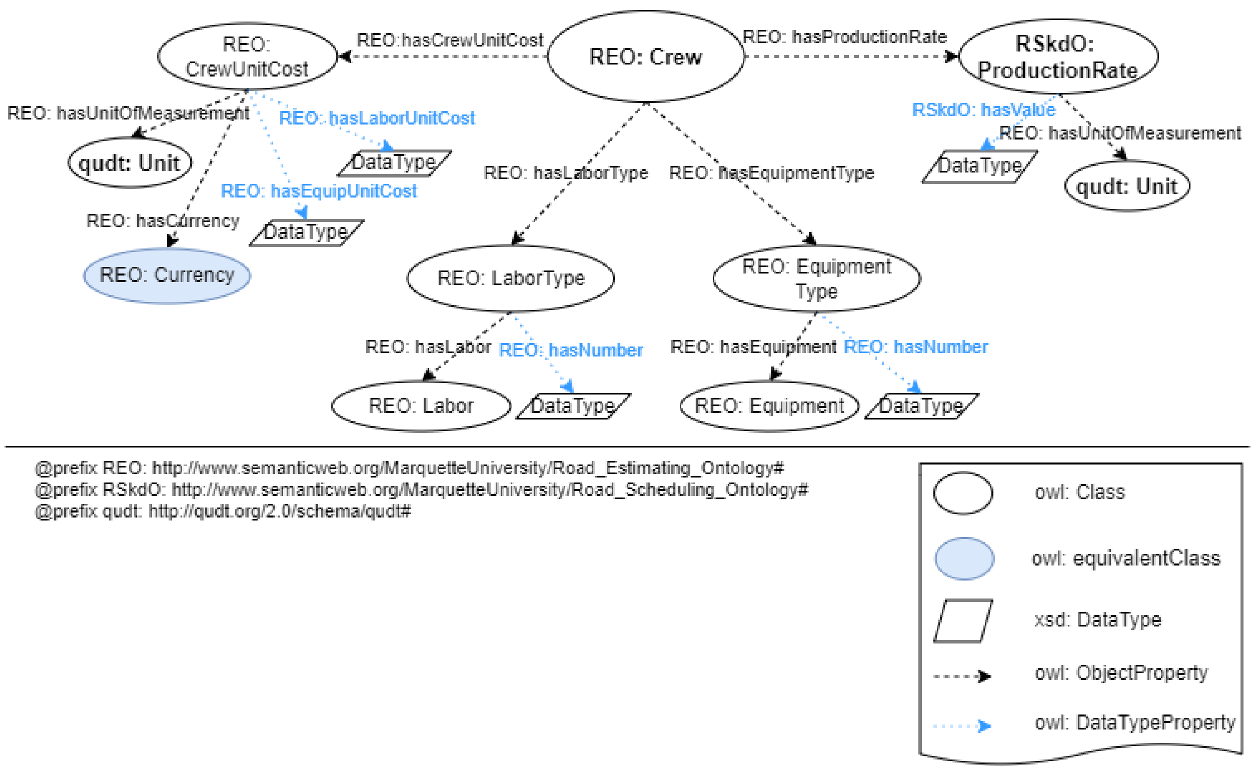


Figure 40 The architecture of the crew ontology

The following is a semantic representation example of the construction of Surface_01, which is the example developed in the design domain, and created in the Road Cost Estimating knowledge base. The type of material is fc:12-05-05-05, ready-mixed concrete, defined by designers. The unit of measurement used is in cubic feet, and the currency used is USD. The QUDT ontology (Hodgson et al. 2011) is reused to provide the unit of measurement. Figure 41 provides the semantic representation of the material used for Surface_01.

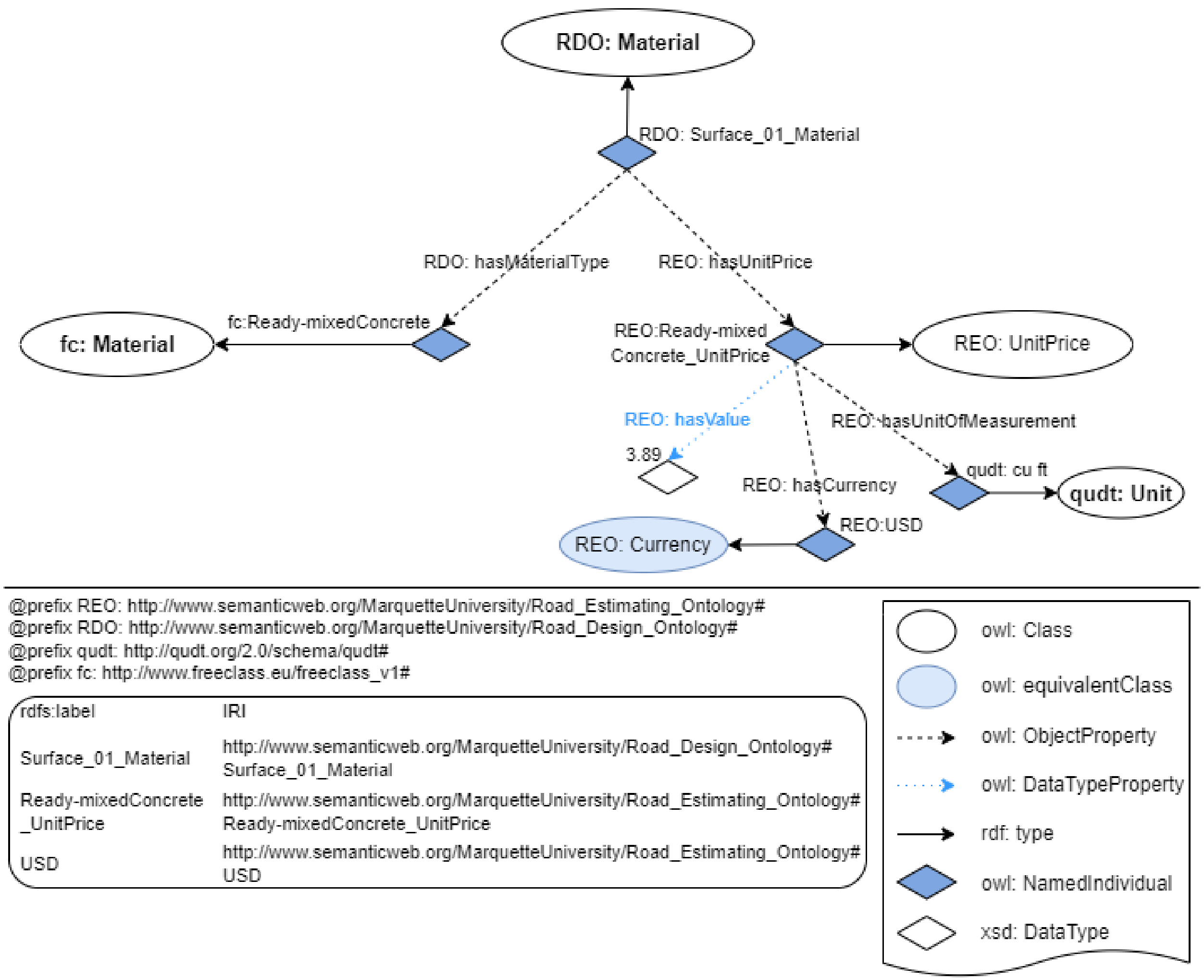
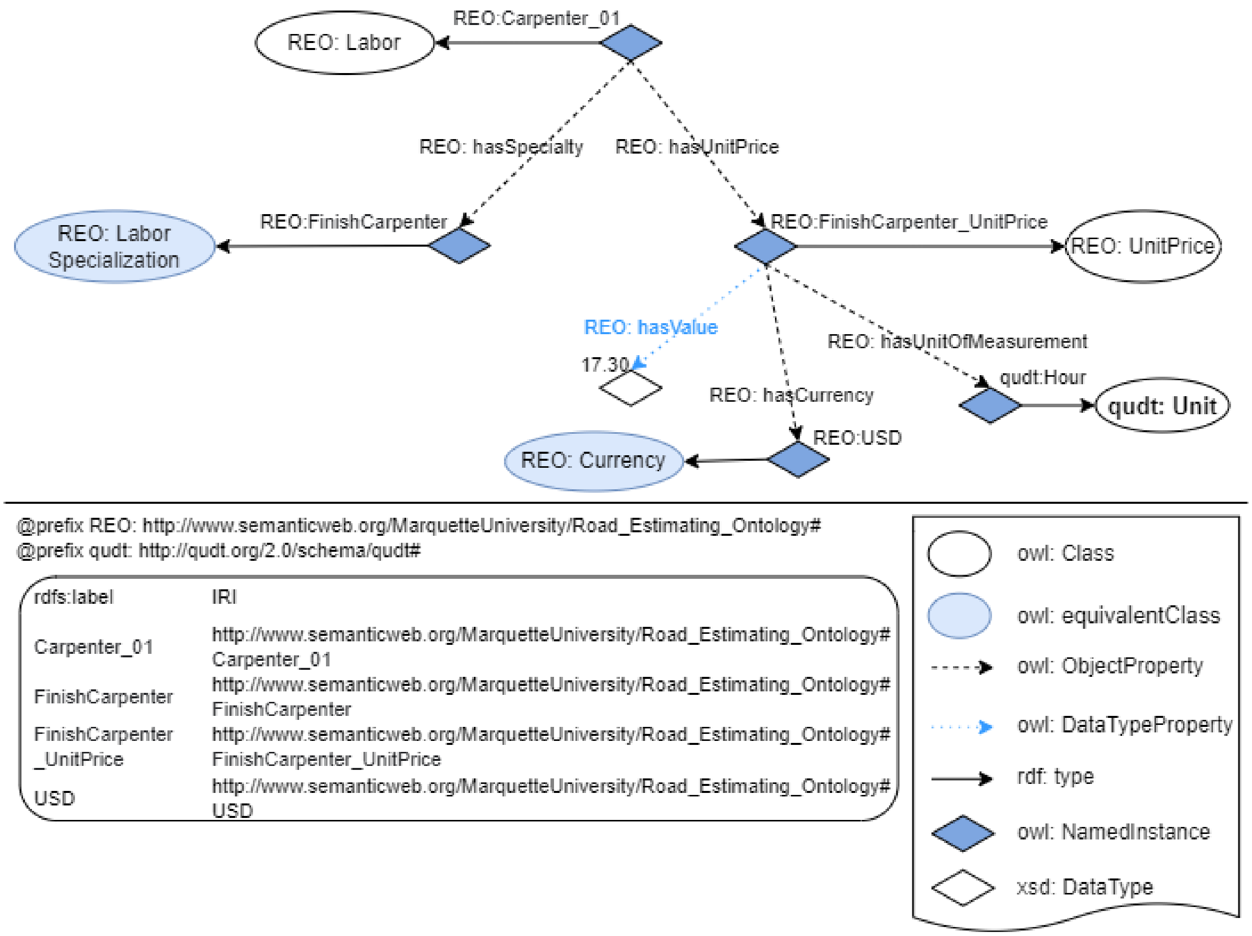


Figure 41 A material example stored in the road cost estimating knowledge base

Figures 42 and 43 provide the semantic representation of the labor and equipment involved in the construction work of Surface_01. Figure 42 shows the properties of the carpenters (Carpenter_01) involved in a crew (Crew B-26). The specialty of the Carpenter_01 is finish carpenter and the unit price of each carpenter is $17.3 per hour. Similarly, Figure 43 shows the properties of the truss screed equipment (TrussScreed _01) involved in crew B-26. The TrussScreed _01 is ZPL-300y model made by Hiking Machinery company. The unit price of each truss screed equipment is $20 per hour.



@prefix REO: http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology#
@prefix qudt: http://qudt.org/2.0/schema/qudt#

rdfs:label	IRI
Carpenter_01	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# Carpenter_01
FinishCarpenter	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# FinishCarpenter
FinishCarpenter _UnitPrice	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# FinishCarpenter_UnitPrice
USD	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# USD

Figure 42 A carpenter example stored in the road cost estimating knowledge base

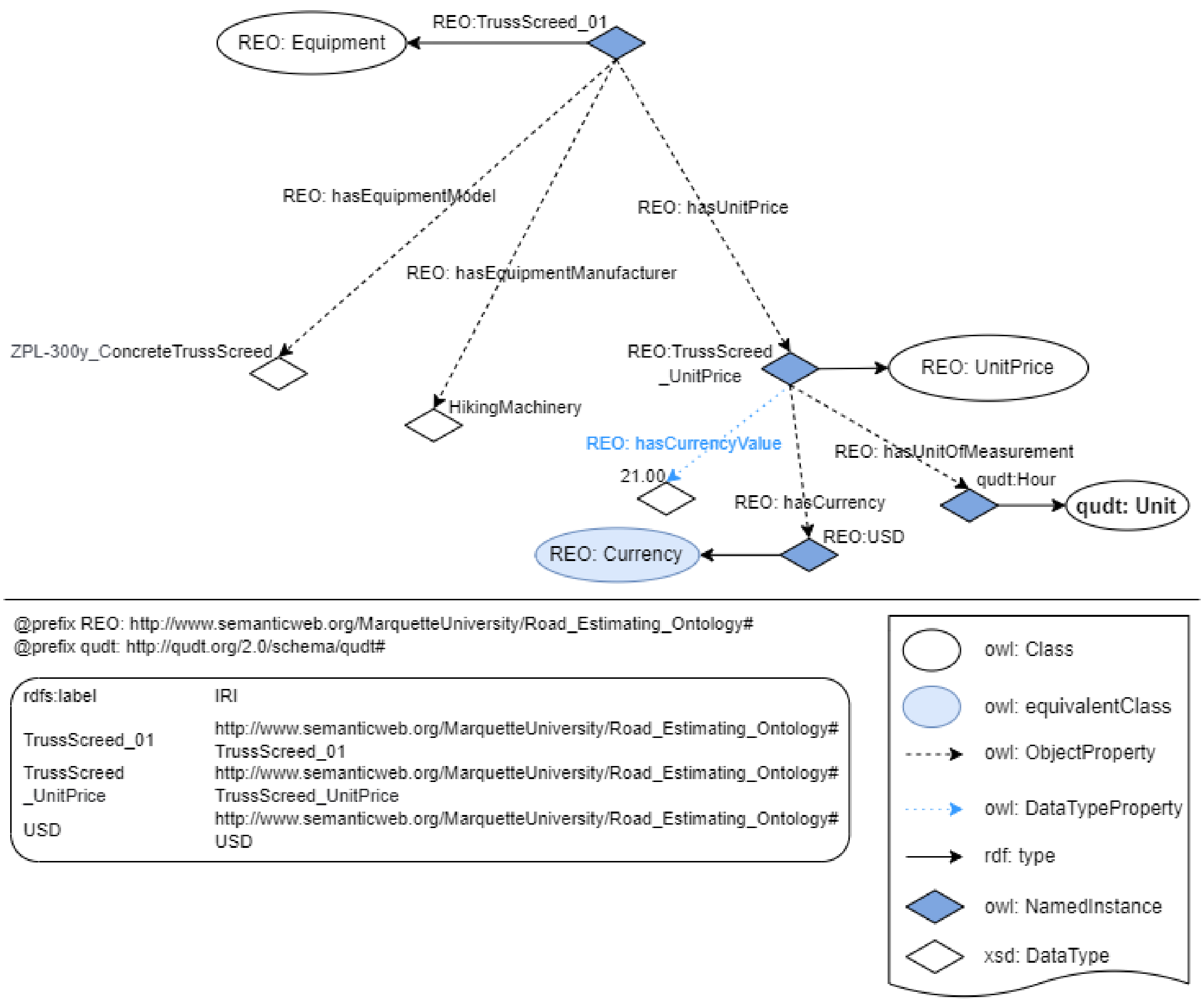
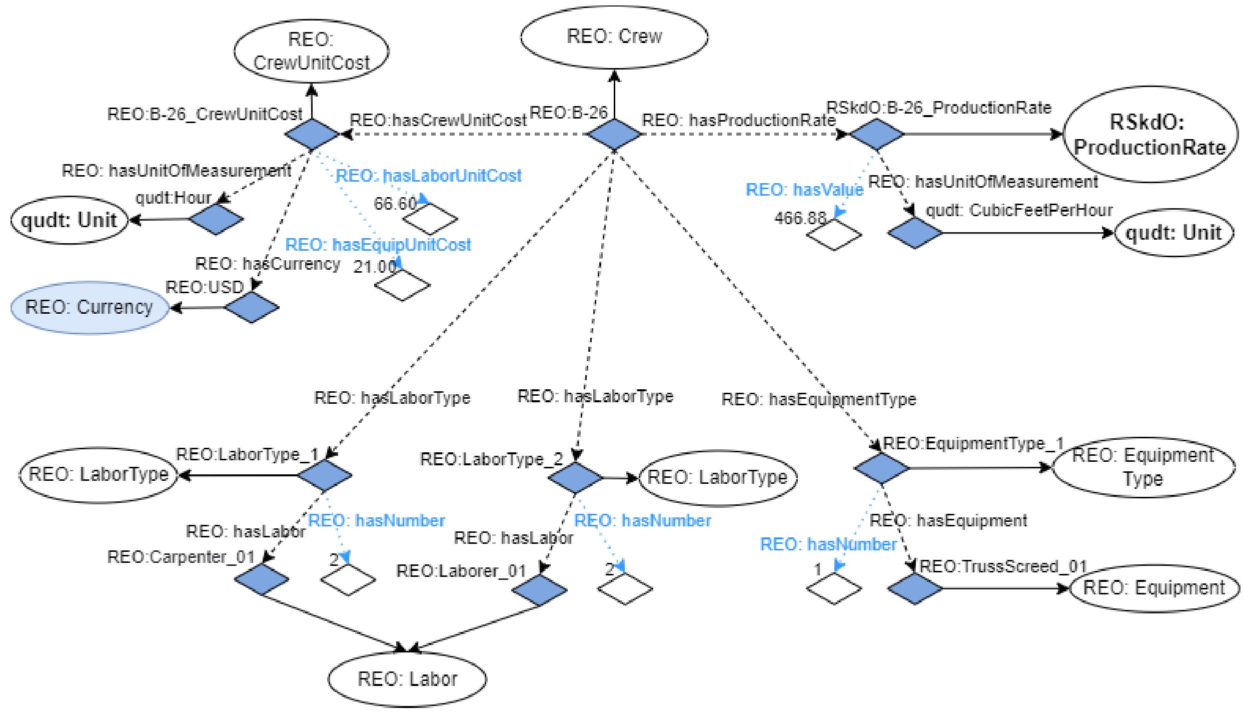


Figure 43 A truss screed example stored in the road cost estimating knowledge base

Figure 44 shows the properties of the crew B-26. There are 2 Carpenter_01 type carpenters, 2 Laborer_01 type laborers, and 1 TrussScreed _01 type truss creed in the crew.



@prefix REO: http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology#
@prefix RSkdO: http://www.semanticweb.org/MarquetteUniversity/Road_Scheduling_Ontology#
@prefix qudt: http://qudt.org/2.0/schema/qudt#

rdfs:label	IRI
B-26	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# B-26
B-26_CrewUnitCost	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# B-26_CrewUnitCost
LaborType_1	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# LaborType_1
LaborType_2	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# LaborType_2
Laborer_01	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# Laborer_01
Carpenter_01	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# Carpenter_01
EquipmentType_1	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# EquipmentType_1
TrussScreed_01	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# TrussScreed_01
B-26_ProductionRate	http://www.semanticweb.org/MarquetteUniversity/Road_Scheduling_Ontology# B-26_ProductionRate
USD	http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology# USD

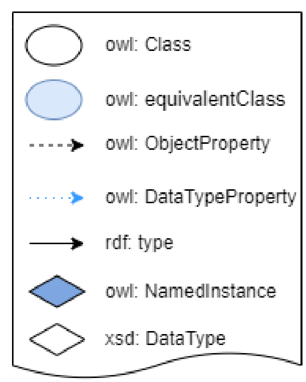


Figure 44 A crew example stored in the road cost estimating knowledge base

Figure 45 provides the top-level concepts involved in the work item, Surface_01_Construction. In the cost estimating domain, the work item Surface_01_Construction is defined with ID, work item cost, work item quantity, resources, and the responsible party involved. The ID of the work item is WorkItem_01. The contractor is xyz Company, which is assigned the prefix xyz for its full IRI:

http://www.semanticweb.org/xyz_Company#. The working condition is rainy weather.

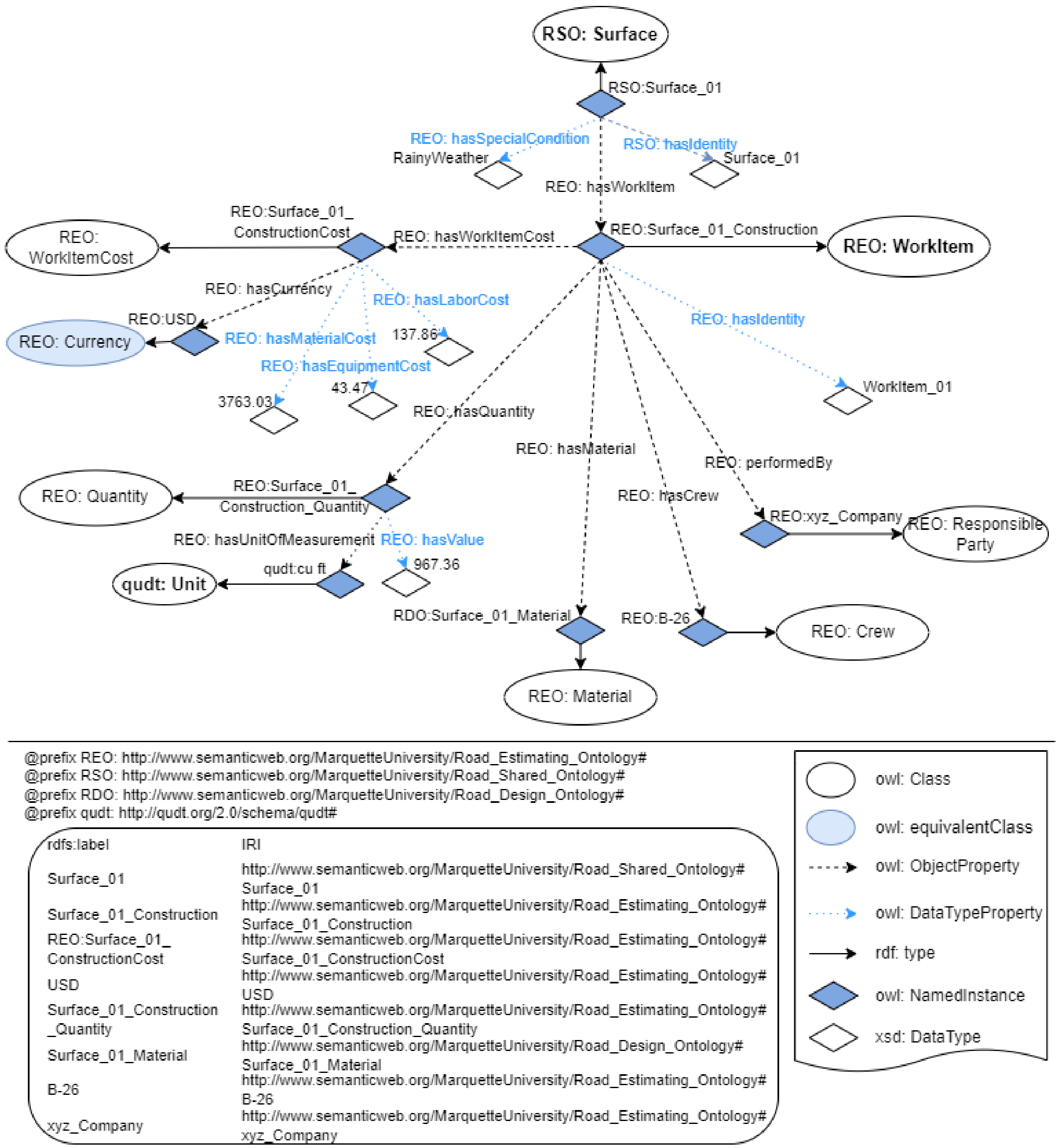


Figure 45 An example for Surface_01_Construction work item

The Road Cost Estimating ontology developed in this study is coded in the

Protégé software (Stanford University 2015). The Protégé user interface allows users to open ontologies available across the web and reuse them when developing a new ontology. The implementation of the Road Cost Estimating ontology in Protégé is shown in Figure 46. It is a screenshot of the realization of the estimating ontology. The left panel shows the concepts (classes), the middle panel shows the object properties, and the right panel shows the data type properties defined in the ontology. The lower parts of the figure show the properties of the selected class, object property, and the data type property, respectively.

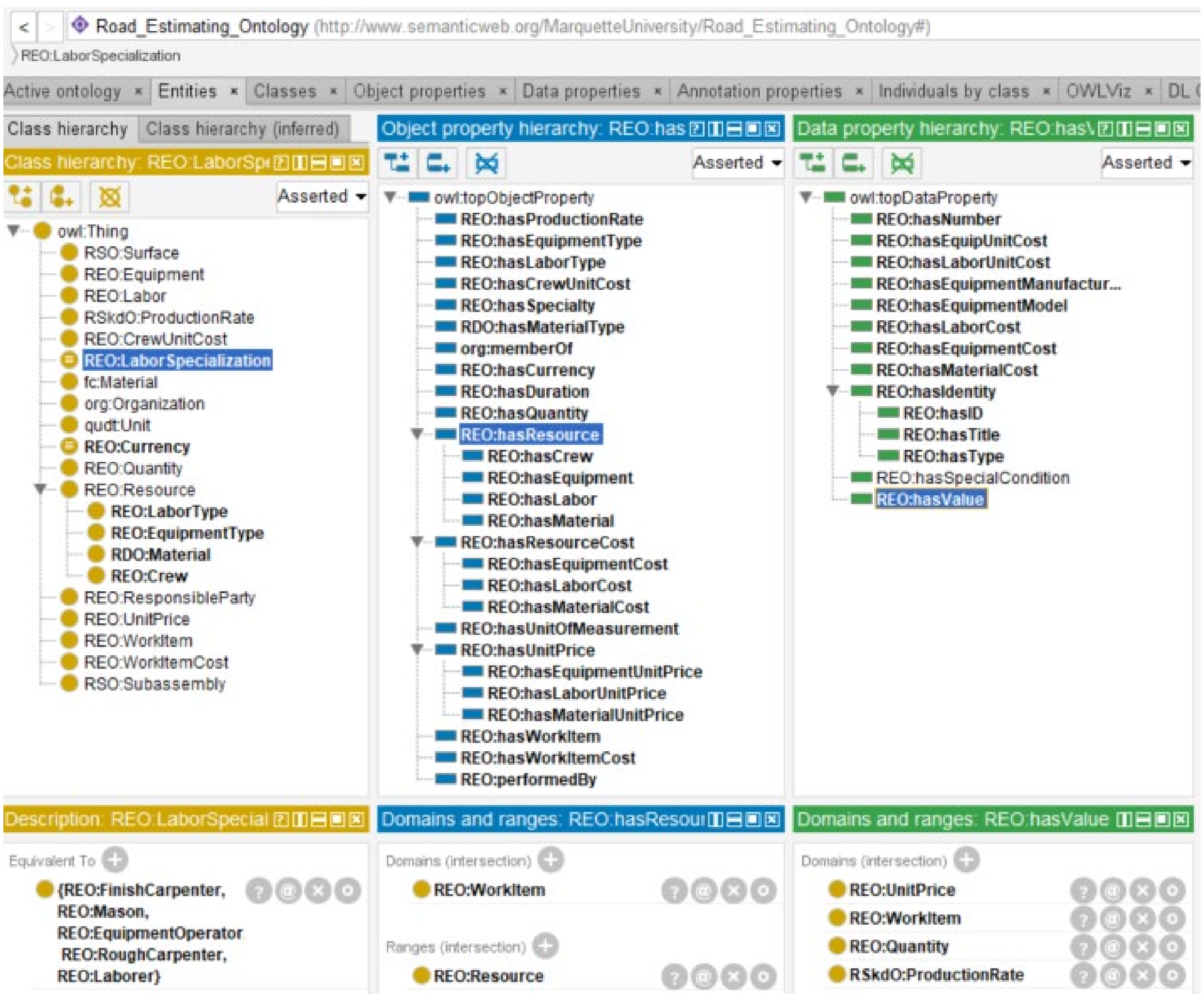


Figure 46 The Road Cost Estimating ontology implemented in Protégé

5.3 Road Scheduling Ontology

A schedule makes building a road infrastructure project much easier by providing a baseline for monitoring and controlling work (Kerzner 1998a, b). Also, it is the primary tool for facilitating the communication of construction planning among all stakeholders in a project (Karshenas and Sharma 2010). To enhance the communication and information integration with other domains involved in a project, some researchers have applied semantic web technologies to represent construction schedule information. For example, the Schedule ontology created by Niknam and Karshenas (2016) (URI:http://www.marquette.edu/Schedule_Ontology#) and the OZONE scheduling ontology created by Smith and Becker (1997). However, these are all created for building construction projects. A road infrastructure ontology is not available.

Several professional schedulers were consulted, and several non-ontological resources were referred to during the development of the Road Scheduling ontology proposed in this study. The non-ontological resources include Construction Planning and Scheduling book (AGC 1994) and the Construction Scheduling Manual (NJDOT 2013). The scheduling book provides a big picture of the critical activity properties of scheduling work. The elaborated Construction Scheduling Manual prepared by the New Jersey Department of Transportation serves as an excellent example of the various standard activity codes assigned by local DOT.

As discussed in Section 1.1.4, there are several scheduling methods available for road construction; however, the effectiveness of each method varies depending on the type of the project. For example, the critical path method (CPM), the most commonly used scheduling method, can be too complex and ineffective for linear projects at times

(Yamin 2001). As a result, the road schedule ontology developed specifically for road projects must consider the linear and repetitive nature of road projects. The Road Scheduling ontology developed in this study takes into account both CPM and the linear scheduling method (LSM) scheduling data. The following will discuss the CPM and LSM methods for the Road Scheduling Ontology, respectively.

The author defines ontology requirements for the Road Scheduling ontology as follows:

- **Purpose:** the Road Scheduling ontology is developed as a fundamental conceptual knowledge model of road scheduling information.
- **Scope:** the Road Scheduling ontology's scope is limited to a road's fundamental scheduling concepts.
- **Implementation language:** the Road Scheduling ontology is implemented in RDF/OWL language.
- **Intended end-users:** road project schedulers.
- **Intended use:** creating a Road Scheduling knowledge base

The QUDT ontology (Hodgson et al. 2011) will be used to represent units of measurement, and the time ontology (Ontology URI: http://www.w3.org/2006/time#) will be used to provide temporal information (W3C 2017).

The prefix RSkdO is assigned to the Road Scheduling ontology, which allows abbreviating the full URI of the ontology: http://www.semanticweb.org/MarquetteUniversity/Road_Scheduling_Ontology#. Figure 47 shows a general view of the architecture of the CPM-type schedules.

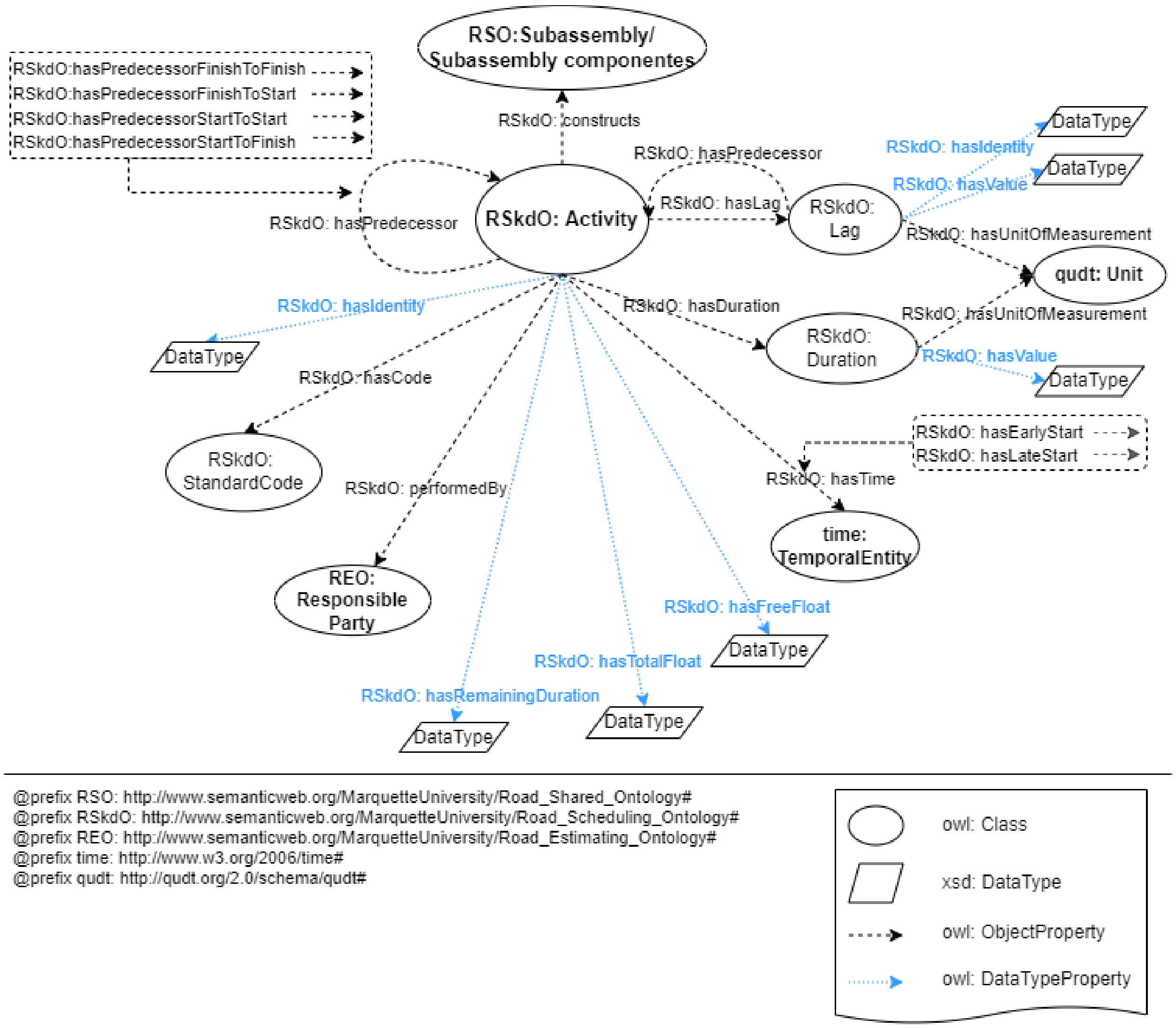


Figure 47 A general view of CPM ontology architecture

The CPM method comprehensively contains the predecessor, sequence relationships, lag time, ID, regulatory code, responsible party, various types of dates, and duration of each activity. These concepts are explained as follows:

- **Sequence relationships:** the four typical sequence relationships are semantically represented as object properties: hasPredecessorFinishToFinish, hasPredecessorFinishToStart, hasPredecessorStartToStart, and hasPredecessorStartToFinish.

- **ID**: each work item should be specified with its identity for management purposes.
- **Regulatory code**: codes of each activity usually comply with the standard codes set by the federal or local transportation department. Examples of codes include the Work Breakdown Structure (WBS) code, Project Area (AREP) code, construction stage code, and count code (NJDOT 2013). The common types of codes are shown in Figure 48.
- **Duration and remaining duration**: duration refers to the period of time between the start and end of an activity at a particular location. The remaining duration will also be necessary for schedule control when an activity is in progress.
- **Dates**: examples of activity dates used in scheduling work are early start date, late start date, early finish date, late finish date, free float, and total float. The finish dates can be easily calculated by adding duration to the start dates.
- **Lag:** refers to the amount of time that exists between the early finish of one activity and the early start of the next activity (Hinze 2004). Since activity can have several predecessors, the lag of activity should also specify the exact predecessor. The lag is defined with the value and time unit (like day and month). The unit of measurement is defined in the QUDT ontology (Hodgson et al. 2011).

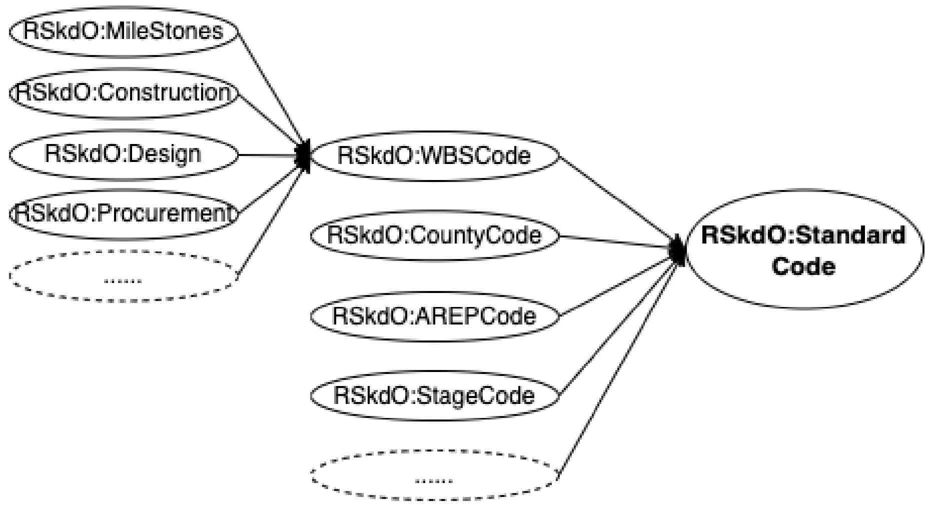


Figure 48 The architecture of Standard Codes

In comparison, the linear scheduling method (LSM) (Harmelink and Rowings 1998) is tailor-made for continuous linear projects, i.e., a road project. The LSM provides an intuitive representation of the activity sequence, making it simple to read and communicate between construction sectors. It employs an activity-velocity diagram to depict the time and location at which a specific crew handles a specific operation (AGC 1994). The diagram indicates many details, such as activity sequences, resources involved, activity durations, and activity buffers. As a result, the sequence relationships, ID, standard codes, activity start time, activity duration, and activity buffers are defined in the LSM part. The meaning of the sequence relationships, ID, standard codes, and activity duration are the same as those defined in the CPM ontology. The following are several new properties:

- **Start time:** here, the start time refers to the actual start of an activity. In the LSM, the activity without any predecessor is the beginning activity. The start time of the beginning activity equals the beginning time of the segment's construction. When the beginning activity has been finished, it becomes the predecessor of the next activity. The start time of the next activity is equal to the end time of the predecessor activity plus the time buffer in between.

- **Production rate**: the production rate of an activity can be calculated by dividing the quantity of work items performed by the activity duration (AGC 1994). In an activity-velocity diagram, the production rate is often represented as the slope of an activity line.
- **Buffer**: the time buffer is similar to the lag defined in the CPM ontology, while the space buffer refers to the distance separating the two activities. Both the time buffer and the space buffer are called buffers in the linear scheduling method. Figure 49 depicts an overview of the linear scheduling ontology architecture.

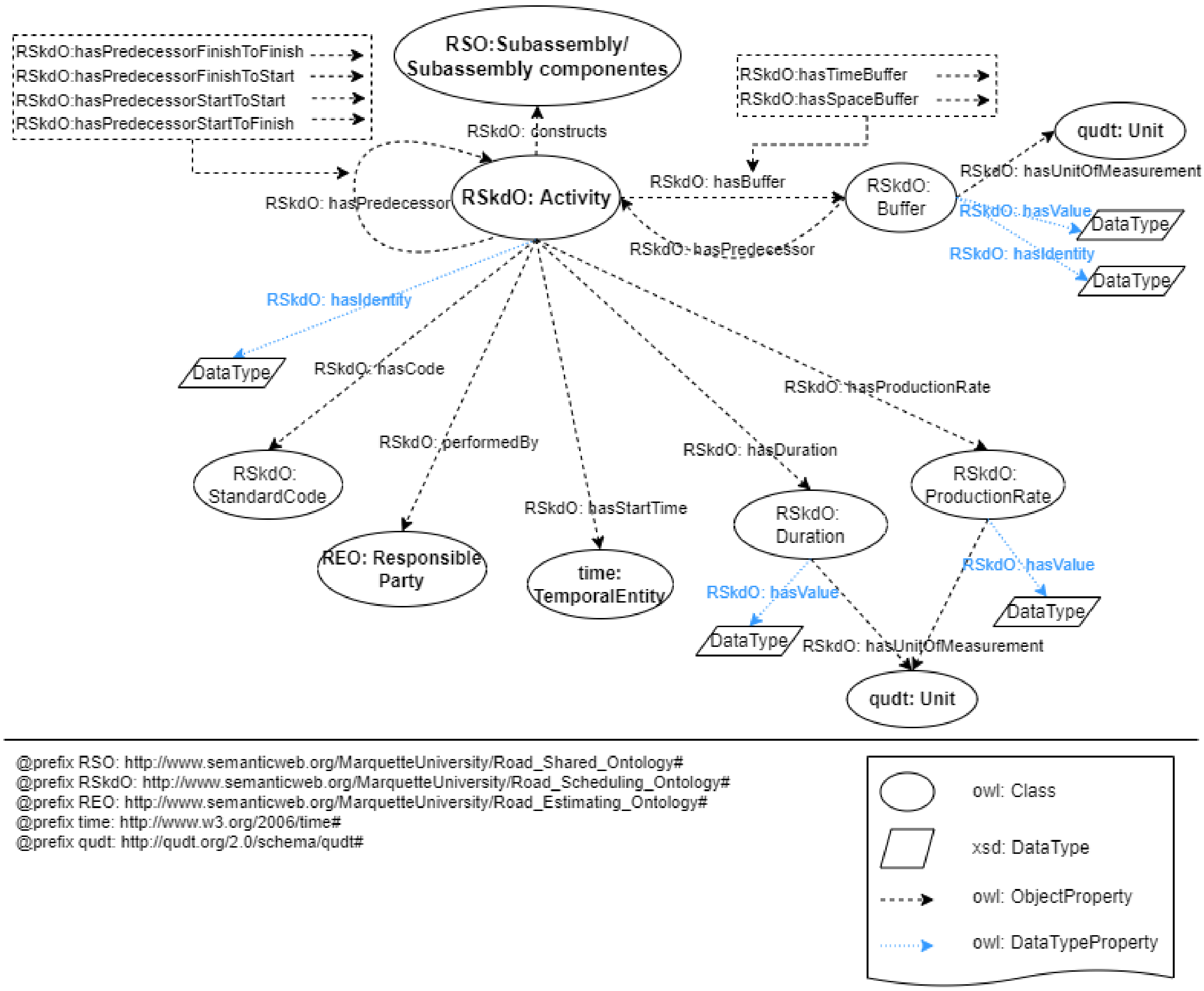


Figure 49 A general view of LSM ontology architecture

The following is an LSM example of the activity, the construction of the Surface_01, which is created in the Road Scheduling knowledge base. The Surface_01 is the same instance defined in the design domain knowledge base. Figure 50 shows that in the scheduling knowledge base, the work item Surface_01_Construction is defined with an ID (Activity_01) and a construction code (C.01). The activity starts at the same location 1 day after the BaseCourse_01 is built. The xyz_Company performs it from 2021-02-01. This activity will take one day to complete. The work production rate is

466.88 cubic feet per hour. The QUDT ontology (Hodgson et al. 2011) is reused to provide the unit of measurement.

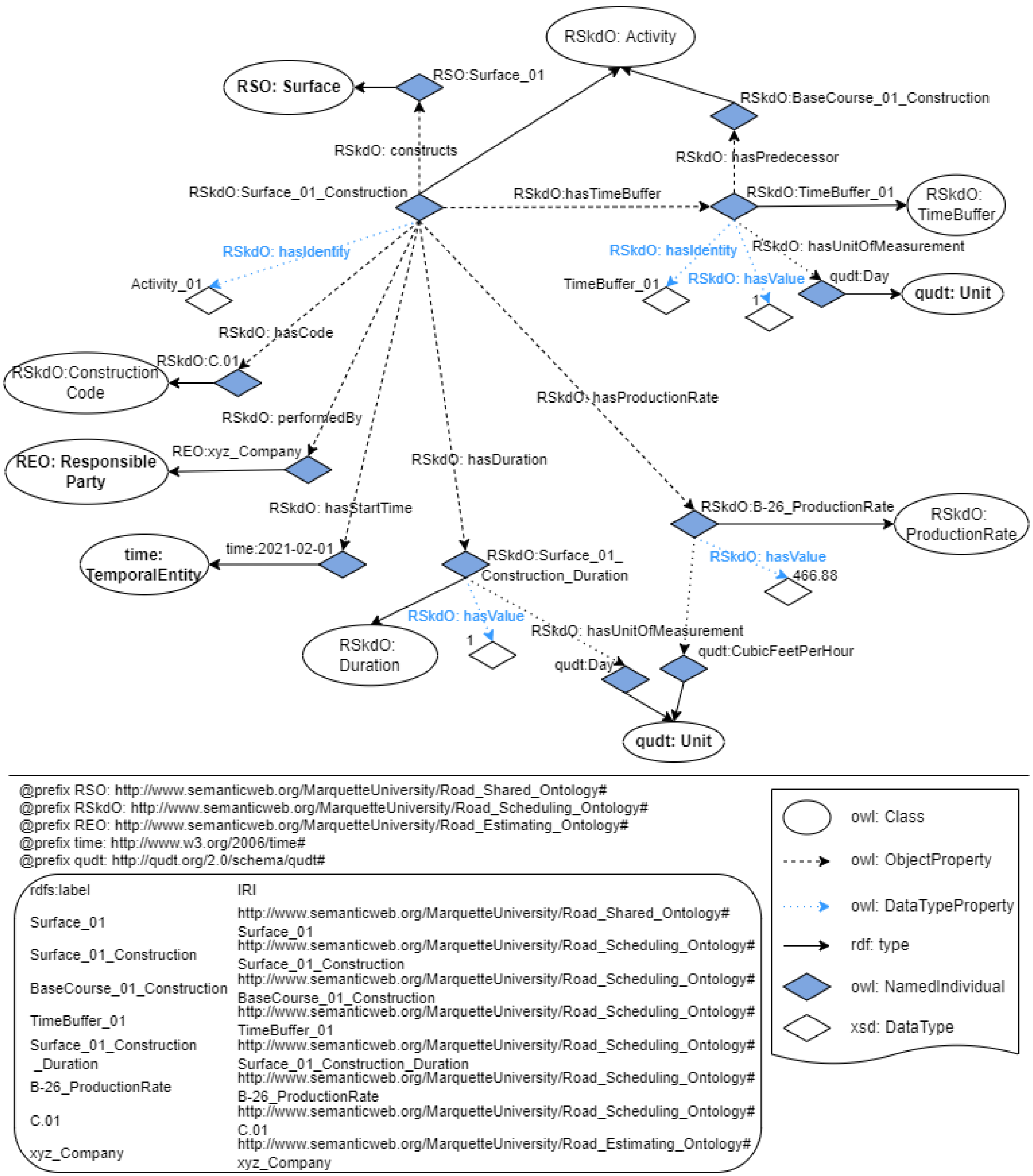


Figure 50 An activity example: Surface_01_Construction

In this study, the Road Scheduling ontology is coded in the Protégé software (Stanford University 2015). The Protégé user interface allows users to open web-based ontologies and reuse them when creating a new ontology. Figure 51 depicts the Protégé implementation of the Road Scheduling Ontology. It is a screenshot of the Road Scheduling Ontology's realization. In Figure 51, the left panel depicts the hierarchy of concepts (classes), and the middle panel depicts the hierarchy of object properties, and the right panel depicts the data type properties defined in the ontology.

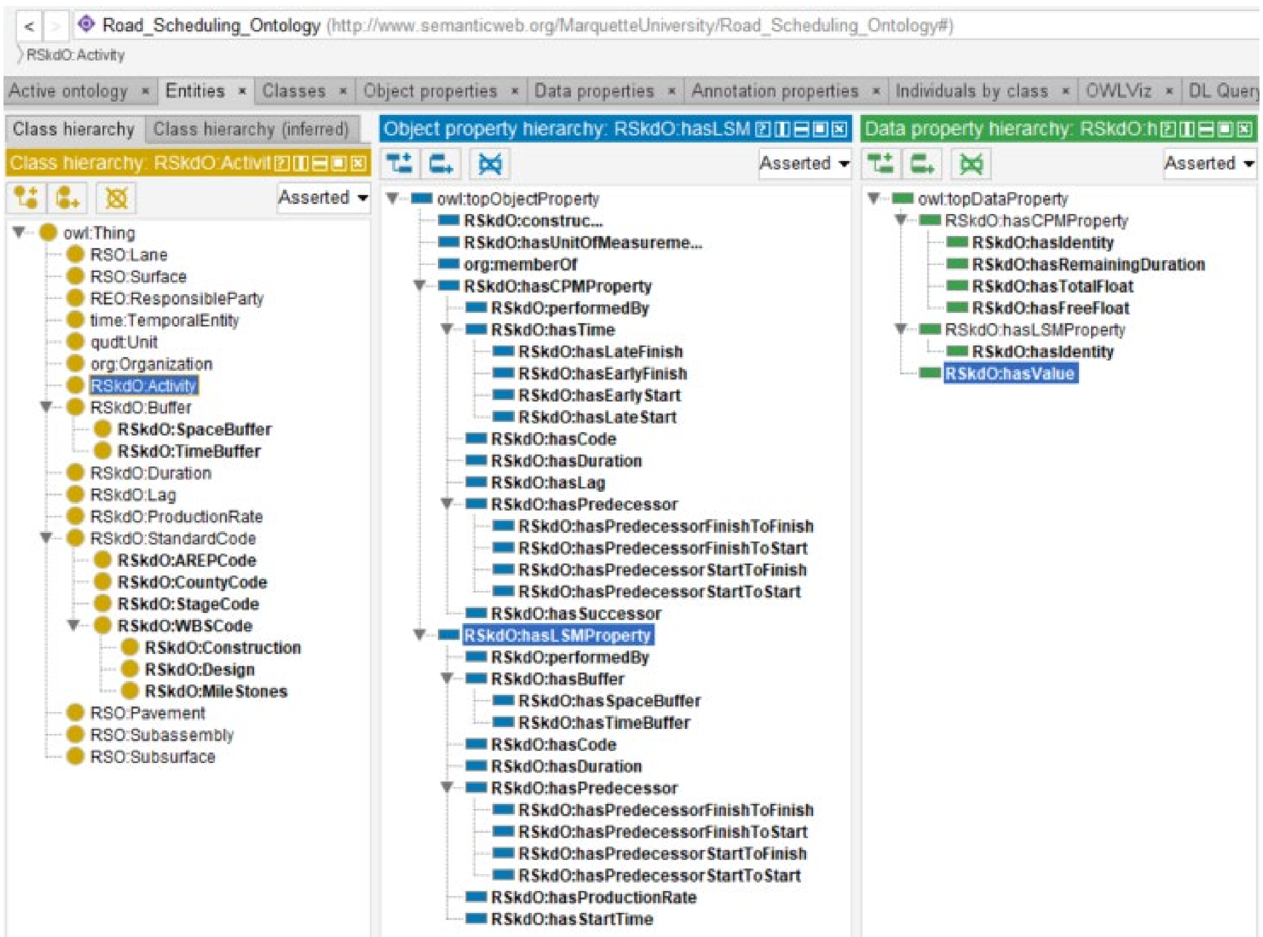


Figure 51 The Road Scheduling ontology implemented in Protégé

CHAPTER 6: KNOWLEDGE BASES AND MODEL VALIDATION

Instead of using software or a piece of programming code to transfer information among domains, the Semantic Web services serve as the general mediator for all the domains whose ontology is extended from the Road Shared ontology. As a result, the retrieval of information generated throughout the life cycle of a road infrastructure project becomes much faster and more accurate.

The information repository created using a domain ontology is known as the domain knowledge base. Each domain knowledge base is domain-specific. In this study, the design, estimating, and scheduling knowledge bases are created by applying the Road Design ontology, Road Cost Estimating ontology, and Road Scheduling ontology to a road project. The design knowledge base includes identities, properties, and the mutual relationships among design domain entities. Likewise, the cost estimating knowledge base contains identities, resources involved, responsible parties, unit prices, and the relationships of work items involved in a project. Similarly, the scheduling knowledge base contains construction activity identities, activity codes, dates, durations, production rates, activity properties, and the relationships among activities.

As discussed in Chapter 2, Section 2.5, the domain ontologies developed in this study are implemented in RDF/OWL format. To create a domain knowledge base, the data should be in the same format. If not, a converter module should be developed to convert the data into the RDF/OWL format. The methodology used to create a domain knowledge base is displayed in Figure 52, using the design domain as an example.

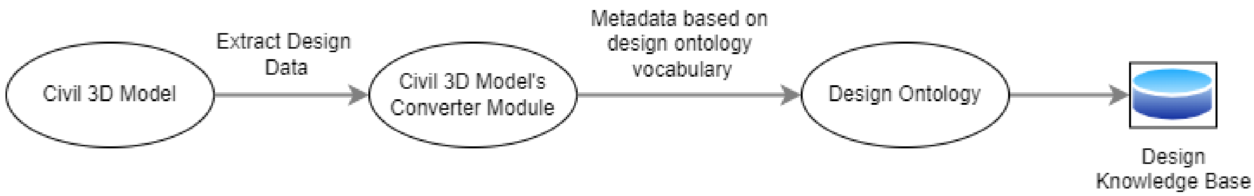


Figure 52 A general view of design knowledge base creating methodology

The converter module developed for Autodesk Civil 3D software uses Civil 3D API. According to the Civil 3D Developer's Guide published by Autodesk (Autodesk 2021), the three APIs available for retrieving properties of a Civil 3D model created in the Civil 3D database are as follows:

1. .NET API — allows writing extensions to Autodesk Civil 3D in any .NET language. In general, the Autodesk Civil 3D.NET API performs significantly faster than the COM API. Development requires Microsoft Visual Studio 2008 SP1 or better.
2. COM API — allows creating clients that access the COM API from managed (.NET) or unmanaged (C++) code. In addition, this API can be used in the Visual Basic for Applications (VBA) IDE, which is available as a separate download. VBA support is deprecated.
3. Custom Draw API (in C++) — is an extension of the AutoCAD ObjectARX API that allows customizing the way Autodesk Civil 3D renders objects. Development requires Microsoft Visual Studio. In this study, .NET API is used.

OpenRDF Sesame triplestore (Sesame 2015) is used to save a knowledge base, which provides a query endpoint that will allow local and remote access to its data over the Internet. In this study, the design, estimating, and scheduling knowledge bases are created as endpoints. This study employs the query language, Simple Protocol and Resource Description Framework Query Language (SPARQL) (Prud'Hommeaux and Seaborne 2008), which is tailored specifically for RDF (Antoniou et al. 2012). Thus, these knowledge bases can be directly queried with SPARQL (Niknam and Karshenas 2015).

The following provides several query examples, ranging from simple to complex, to display the methods of information retrieving and integrating from knowledge bases developed in this study.

6.1 Query examples from individual knowledge bases

Figure 53 provides an example of the information retrieval process from the road design domain knowledge base. To retrieve the data stored in a domain knowledge base, users send commands at the user end to a specific domain knowledge base. After the target domain knowledge base has processed the commands, the retrieved results will be sent back to the users.

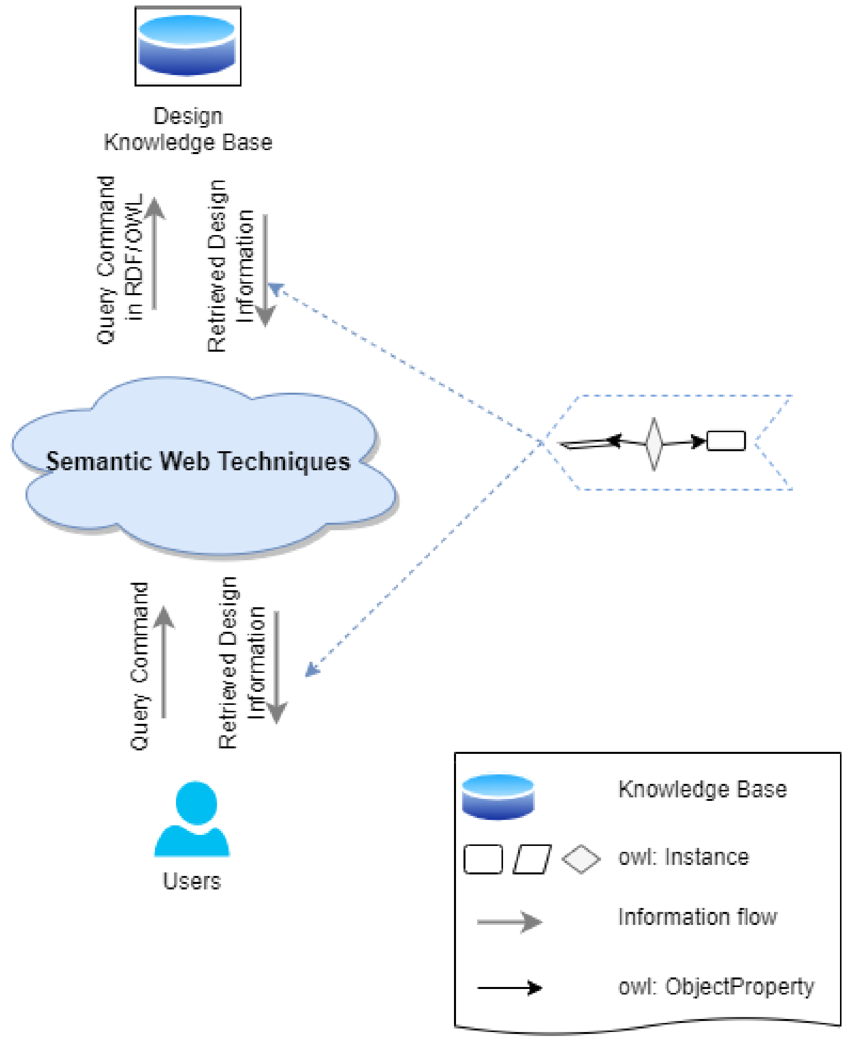


Figure 53 A Schematic View of the Information Maintained in Each Domain

6.1.1 Query 1

Query 1 is an example of retrieving information from the design knowledge base. It retrieves the assembly used for each segment and then asks for the results to be ordered by the ID of the segment. The query command is as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX RSO: <http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#>
SELECT DISTINCT ?Segment ?Assembly
        WHERE {?Segment RSO:hasAssembly ?Assembly}
        ORDER BY ASC(?Segment)
```

The query result is shown below:

Segment	Assembly
RSO:Seg_01	RSO:Assembly_01
RSO:Seg_02	RSO:Assembly_02

This query, with minor changes, can be applied to query cases such as (1) querying the names and IDs of the top-level objects such as road, phase, assembly, and subassembly of the road project, and (2) querying the start time and location of each segment and other related information.

6.1.2 Query 2

Query 2 is another example of retrieving information from the design knowledge base. It retrieves design properties such as materials used and the layer thickness for subassembly Surface_01. Additionally, each dimension property value should come with a corresponding unit of measurement. The unit of measurement is defined in the QUDT ontology (Hodgson et al. 2011). The query command is as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX RSO: <http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#>
PREFIX RDO: <http://www.semanticweb.org/MarquetteUniversity/Road_Design_Ontology#>
PREFIX fc: <http://www.freeclass.eu/freeclass_v1#>
PREFIX qudt: <http://qudt.org/2.0/schema/qudt#>

SELECT DISTINCT ?Surface ?Material ?Thickness ?ThicknessValue ?Unit
        WHERE {
        ?Surface RDO:hasMaterial ?Material.
        ?Surface RDO:hasThickness ?Thickness.
        ?Thickness RDO:hasValue ?ThicknessValue.
        ?Thickness RDO:hasUnitOfMeasurement ?Unit
        }
```

The query result is shown as follows:

Surface	Material	Thickness	ThicknessValue	Unit
RSO:Surface_01	RDO:Surface_01_Material	RDO:Surface_01_Thickness	"0.083"^^<http://www.w:	qudt:ft

This query, with minor changes, can be applied to query cases such as (1) query other design parameters of the Surface_01, such as the skid resistance, and (2) query the design information of other road elements, such as the length and width of the segment or the slope of the lane.

6.1.3 Query 3

Query 3 is an example of retrieving information from the cost estimating knowledge base. It retrieves the model of the equipment assigned to the work item Surface_01_Construction. The query command is as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX REO: <http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology#>

SELECT DISTINCT ?WorkItem ?Crew ?EquipmentType ?Equipment ?DataType
        WHERE {
        ?WorkItem REO:hasIdentity 'WorkItem_01'.
        ?WorkItem REO:hasCrew ?Crew.
        ?Crew REO:hasEquipmentType ?EquipmentType.
        ?EquipmentType REO:hasEquipment ?Equipment.
        ?Equipment REO:hasEquipmentModel ?DataType.
        }
```

The query result is shown as follows:

WorkItem	Crew	EquipmentType	Equipment	DataType
REO:Surface_01_Construction	REO:B-26	REO:EquipmentType_1	REO:TrussScreed_01	"ZPL-300y_ConcreteTrussScreed"

This query, with minor changes, can be applied to query cases such as (1) query the number of equipment assigned to the work item, (2) query the unit price of each equipment employed, (3) query the labor specialty, number, and unit price information for the work item, and (4) query the work item cost details.

6.1.4 Query 4

Query 4 is an example for retrieving information from the scheduling knowledge base. It retrieves schedule properties such as the time buffer of the activity, Activity_01. Since the time buffer should be specified between which two activities, the predecessor activity should be retrieved as well. The first query retrieves the time buffer identity of the Activity_01. The second query retrieves the buffer value, unit of measurement, and predecessor information. The query command is as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX RSkdO: <http://www.semanticweb.org/MarquetteUniversity/Road_Scheduling_Ontology#>
PREFIX qudt: <http://qudt.org/2.0/schema/qudt#>

SELECT DISTINCT ?Activity ?TimeBuffer
        WHERE {
        ?Activity RSkdO:hasIdentity 'Activity_01'.
        ?Activity RSkdO:hasTimeBuffer ?TimeBuffer.
        }

SELECT DISTINCT ?TimeBuffer ?Activity ?DataType ?Unit
        WHERE {
        ?TimeBuffer RSkdO:hasIdentity 'TimeBuffer_01'.
        ?TimeBuffer RSkdO:hasPredecessor ?Activity.
        ?TimeBuffer RSkdO:hasValue ?DataType.
        ?TimeBuffer RSkdO:hasUnitOfMeasurement ?Unit.
        }
```

The query results are shown as follows:

Activity	TimeBuffer
RSkdO:Surface_01_Construction	RSkdO:TimeBuffer_01

TimeBuffer	Activity	DataType	Unit
RSkdO:TimeBuffer_01	RSkdO:BaseCourse_01_Construction	"1"^^<http://www.w3.org/2001/XMLSchema#integer>	qudt:Day

With minor changes, these queries can be applied to retrieve information such as start time, standard code, responsible party, duration, and remaining duration of an activity.

6.2 Cross query from a combination of knowledge bases

Domain knowledge bases allow cross-domain information sharing. Users can query multiple properties, even if the properties are defined in different knowledge bases. Figure 54 provides an example of the cross-domain information retrieval process from a variety of domain knowledge bases.

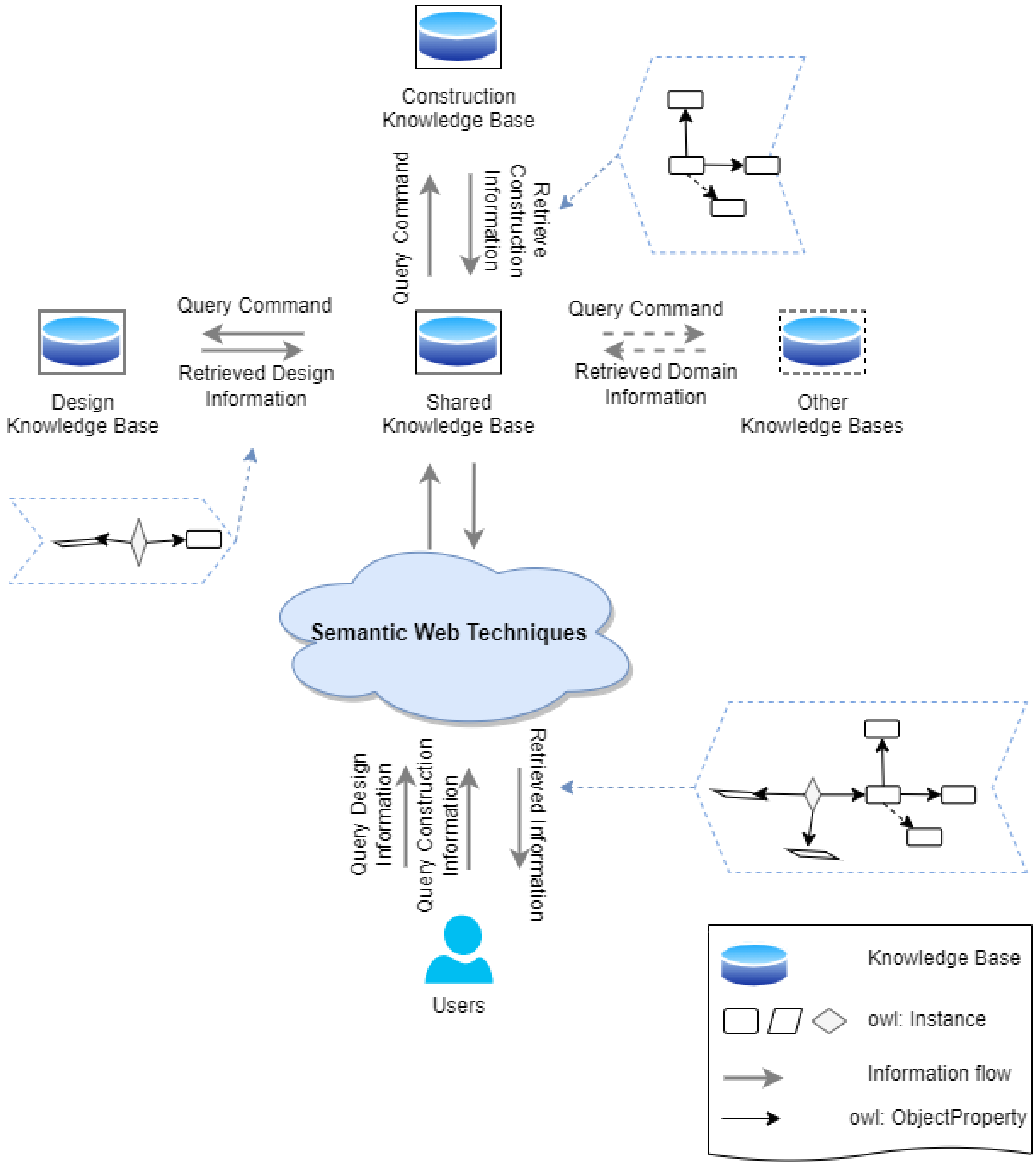


Figure 54 Information flow when combining design and construction information

6.2.1 Query 5

The following is an example for retrieving information about the construction time, region, and centerline name of a construction phase, Phase_01. The data is stored in the road design, time, and location knowledge bases, respectively. The query command is shown as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX RSO: <http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#>
PREFIX RLO: <http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#>
PREFIX time: <http://www.w3.org/2006/time#2016#>

SELECT DISTINCT ?Phase ?TemporalEntity ?AbsoluteLocation ?Alignment
        WHERE {
        ?Phase RSO:hasBeginningTime ?TemporalEntity.
        ?Phase RSO:hasBeginningPoint ?AbsoluteLocation.
        ?Phase RSO:hasAlignment ?Alignment
         }
```

The query result is shown as follows:

Phase	TemporalEntity	AbsoluteLocation	Alignment
RSO:Phase_01	time:2021-01-01	RLO:Point001	RSO:Align_01

6.2.2 Query 6

The following is a cross query example for retrieving information from the design and cost estimating knowledge bases. Assume an estimator needs the dimension information of the Surface_01 and the corresponding material unit price to prepare the cost estimation for the work item Surface_01_Construction. The dimension information is stored in the design knowledge base, and the material unit price is stored in the cost estimating knowledge base. The query command is as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX RSO: <http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#>
PREFIX RDO: <http://www.semanticweb.org/MarquetteUniversity/Road_Design_Ontology#>
PREFIX REO: <http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology#>
PREFIX fc: <http://www.freeclass.eu/freeclass_v1#>
PREFIX qudt: <http://qudt.org/2.0/schema/qudt#>

SELECT DISTINCT ?Segment ?Length ?Width ?Surface ?Thickness ?Material ?UnitPrice
        WHERE {
        ?Segment RSO:hasIdentity ?Seg_01.
        ?Segment RDO:hasLength ?Length.
        ?Segment RDO:hasWidth ?Width.
        ?Segment RSO:hasAssembly ?Assembly.
        ?Assembly RSO:hasSubassembly ?Subassembly.
        ?Subassembly RSO:hasLane ?Lane.
        ?Lane RSO:hasComponent ?Pavement.
        ?Pavement RSO:hasLayer ?Surface.
        ?Surface RDO:hasThickness ?Thickness.
        ?Surface REO:hasWorkItem ?WorkItem.
        ?WorkItem REO:hasMaterial ?Material.
        ?Material REO:hasUnitPrice ?UnitPrice.
        }
```

The query result is shown as follows:

Segment	Length	Width	Surface	Thickness	Material	UnitPrice
RSO:Seg_01	RDO:Seg_01_Length	RDO:Seg_01_Width	RSO:Surface_01	RDO:Surface_01_Thickness	RDO:Surface_01_Material	REO:Ready-mixedConcrete_UnitPrice

6.2.3 Query 7

The following is another cross-query example for retrieving information from the cost estimating and scheduling knowledge bases simultaneously. Assume a procurement staff needs the equipment model information of the work item Surface_01_Construction and the corresponding work start time to schedule the equipment procurement order. The equipment model information is stored in the cost estimating knowledge base, and the start time is stored in the scheduling knowledge base. The query command is as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX REO: <http://www.semanticweb.org/MarquetteUniversity/Road_Estimating_Ontology#>
PREFIX RSkdO: <http://www.semanticweb.org/MarquetteUniversity/Road_Scheduling_Ontology#>
PREFIX time: <http://www.w3.org/2006/time#2016#>

SELECT DISTINCT ?WorkItem ?TemporalEntity ?Equipment ?DataType
            WHERE {
            ?Surface RSO:hasIdentity 'Surface_01'.
            ?Surface RSkdO:constructedBy ?Activity.
            ?Activity RSkdO:hasStartTime ?TemporalEntity.
            ?Surface REO:hasWorkItem ?WorkItem.
            ?WorkItem REO:hasCrew ?Crew.
            ?Crew REO:hasEquipmentType ?EquipmentType.
            ?EquipmentType REO:hasEquipment ?Equipment.
            ?Equipment REO:hasEquipmentModel ?DataType.
            }
```

The query result is shown as follows:

WorkItem	TemporalEntity	Equipment	DataType
REO:Surface_01_Construction	time:2021-02-01	REO:TrussScreed_01	"ZPL-300y_ConcreteTrussScreed"

CHAPTER 7: SUMMARY, CONCLUSION, AND RECOMMENDATIONS FOR FUTURE WORK

7.1 Summary and Conclusion

An AEC project requires the efficient collaboration of multiple sectors to get work done. Throughout an infrastructure project's lifespan, data exchange is frequently and in a massive amount. The data sources include but are not limited to planning documents, bidding documents for procurement of goods, road 3D digital models, estimating assemblies, takeoff quantity files, work item list, traffic data, and maintenance history records. However, each domain has its own methods for storing and managing data which have evolved along with its development. As a result, domain data is usually stored in diverse data formats. The data could be in text document, pdf, XML, drawing, or database table format.

Additionally, in different domains, the taxonomy and concept definition could conflict. For example, an assembly in the cost estimating domain includes all the necessary material and labor to complete a unit of work, while an assembly refers to the combination of objects that make up a road such as lanes, shoulders, and median in the design domain. All of these hinder the sharing and exchange of information among domains.

To ensure the accuracy and the efficiency of the data exchange operations, there are three critical requirements such as (1) store project data once, in the place where it is generated, (2) store data semantics along with the data, and (3) provide specifically authorized internet access to users across the Internet.

This study adopts the Semantic Web technologies to represent a road infrastructure project's data in a machine-processable structural framework to facilitate the integration and exchange of information among domains. Data models created using semantic web technologies greatly fulfill the previous three requirements. The semantic web approach creates a specific vocabulary to define the road infrastructure data and multiple ontologies that represent contextual relationships behind the vocabulary. In this study, the shared ontology, called Road Shared ontology, provides the architecture of main concepts and relationships among them to be reused by multiple road infrastructure knowledge domains. Subsequently, the shared ontology is extended by each knowledge domain to create its own domain ontology by adding domain-specific properties to the concepts defined in the shared ontology. This collection of road infrastructure ontologies not only creates an extensible, machine-processable, and Internet-tailored data framework but also fills the gaps in road-related ontologies and promotes the information integration of road projects.

In this study, the design domain, location domain, estimating domain, and scheduling domain were studied. Accordingly, the Road Location ontology, Road Design ontology, Road Cost Estimating ontology, and Road Scheduling ontology are developed, respectively. The Road Design ontology includes the material and dimensional properties attached to a subassembly that designers define; the Road Location ontology completes the Road Shared ontology with location information, which includes the two major object positioning methods: absolute- and relative- type; the Road Cost Estimating ontology defines the resource, resource quantity, production rate, and cost for each work item; and the Road Scheduling ontology covers the concepts and relationships among them necessary for CPM and the linear scheduling method (LSM). Moreover, the criteria used, development process, and main architecture of these ontologies are discussed in detail.

A corresponding domain knowledge base is created based on each domain ontology and extracted domain data. The data of each domain is converted directly into the domain knowledge base with a converter module. Converter modules are usually developed with software APIs. The semantic web information integration approach eliminates the need for mediators and reduces human involvement.

Additionally, the semantic approach not only allows the integration of distributed data but also facilitates machine processing of the exchanged data. All the data is stored once, in the place it is generated. No extra data transfer efforts are required. Regarding information retrieval, the querying language SPARQL, which is tailored for RDF, is used. Several SPARQL query examples are presented in the following section.

7.2 Suggestions for Future Work

The development of an ontology is an iterative process through the entire lifecycle of the ontology (Noy and McGuinness 2001). The ontology must be continuously supplemented and refined, which requires a long time of joint efforts. So far, the ontologies proposed in this study are a start step for applying semantic technologies in road infrastructure projects. There are some recommendations for future work.

Firstly, the proposed knowledge bases are designed only for professional domain engineers. It could be more user-friendly by establishing querying and inferring end-user interface. The querying interface should be more visualized and facilitate relief for users from the programming codes.

In addition, the domain ontologies proposed in this dissertation only cover a few domains. In many road infrastructure-related fields, the investment in the development of the domain ontology is still lacking. An example of this is road traffic. Many existing ontologies developed in the transportation domain mainly aim at travel and public transit. As a result, road-related traffic ontology (such as traffic monitoring video, speed sensor data, and traffic flow data) is rare. Another example comes from the domain of pavement material. Most of the existing material ontologies are related to building materials or focus on some physical and chemical properties of the materials themselves. Pavement materials often use different mix ratios according to regions' local environments and road usage requirements. AASHTO provides some standards in this regard (AASHTO 2008). However, the development of a standard ontology in this road-related material domain is still lacking.

Another ontology development direction is about the complexity of the road infrastructure. In this study, the complex road network is not included. In a comprehensive road network, complex segments such as roundabouts and intersections should be considered.

Lastly, this study takes the road infrastructure as a typical representation of the transportation infrastructure. Future work can transfer the methodologies used in this study to other linear types, such as railway and watercourses.

BIBLIOGRAPHY

AASHTO. (2008). *Guide Specifications for Highway Construction (9th Edition).* American Association of State Highway and Transportation Officials (AASHTO). Retrieved from https://app.knovel.com/hotlink/toc/id:kpGSHCE001/guide-specifications/guide-specifications

AASHTO. (2011). *A Policy on Geometric Design of Highways and Streets, 2011*. 6th edition. AASHTO, Washington, DC.

Adams, T. M., N. A. Koncz, and A. P. Vonderohe. (2001). NCHRP Report 460: *Guidelines for the implementation of multimodal transportation location referencing systems* (No. Project SP20-27 (3) FY'96). TRB, National Research Council, Washington, D.C., 2001.

AGC (1994). *Construction Planning and Scheduling* (Publication No. 1107.1) Associated General Contractors of America, Washington, DC.

Akinyemi, A., Sun, M., & Gray, A. J. (2018). An ontology-based data integration framework for construction information management. *Proceedings of The Institution of Civil Engineers-Management, Procurement and Law*, *171*(3), 111-125.

Amirian, P., Winstanley, A. C., & Basiri, A. (2013). NoSQL storage and management of geospatial data with emphasis on serving geospatial data using standard geospatial web services.

Alexander. K. (2013). "The Difference Between a Triplestore and a Relational Database." http://krisalexander.com/innovation/2013/07/16/the-difference-between-a-triplestore-and- a-relational-database/ 2015).

Allemang, D., and Hendler, J. (2011). *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier.

Antoniou, G., Groth, P., Harmelen, F. V., and Hoekstra, R. (2012). *A semantic web primer*. MIT press., third edition, The MIT Press, 2012, 96-97.

Autodesk (2021). Civil 3D Developer's Guide https://knowledge.autodesk.com/support/civil-3d/learn-explore/caas/CloudHelp/cloudhelp/2020/ENU/Civil3D-DevGuide/files/GUID-275A6271-7758-4C14-9703-989B1B007E3E-htm.html

Bergman, M. (2009). "Advantages and Myths of RDF." *AI3, April.*

Berners-Lee, T. (1998). "Why RDF model is different from the XML model." http://www.w3.org/DesignIssues/RDF-XML (06/02, 2015).

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American 284 (5),* 34–43.

Breunig, M., & Zlatanova, S. (2011). 3D geo-database research: Retrospective and future directions. *Computers & Geosciences, 37(7),* 791-803.

Cadkin, J. (2002). Understanding dynamic segmentation–working with events in ArcGIS 8.2. ArcUser magazine, October-December issue, 2002, 40-43.

Cambridge. (2015). "RDF vs. XML." https://www.cambridgesemantics.com/semantic-university/rdf-vs-xml.

Chung, S. M., & Mah, P. S. (1995). Schema integration for multidatabases using the unified relational and object-oriented model. In *Proceedings of the 1995 ACM 23rd annual conference on Computer science* (pp. 208-215).

Collins, S. R., Navathe, S., & Mark, L. (2002). XML schema mappings for heterogeneous database access. *Information and Software Technology*, *44*(4), 251-257.

Costin, A. (2016). *A new methodology for interoperability of heterogeneous bridge information models* (Doctoral dissertation, Georgia Institute of Technology).

Cyganiak, R., and Jentzsch, A. (2010). Introduction to: Linked Data. (06/02, 2015). http://www.dataversity.net/linked-data-an-introduction/.

Doerr, M., Hunter, J., & Lagoze C. (2003). Towards a core ontology for information integration. *Journal of Digital information*, *4*(1).

Dueker, K.J., & Butler, J.A. (1997). GIS-T Enterprise Data Model with Suggested Implementation Choices. Journal of the Urban and Regional Information Systems Association, *URISA-WASHINGTON DC-*, *10*, 12-36.

Dueker, K. J., & Butler, J. A. (2000). A geographic information system framework for transportation data sharing. *Transportation Research Part C: Emerging Technologies*, *8*(1-6), 13-36.

Egenhofer, M. J. (1994). Spatial SQL: A query and presentation language. *IEEE Transactions on knowledge and data engineering*, *6*(1), 86-95.

El-Diraby, T. E. (2013). Domain ontology for construction knowledge. *Journal of Construction Engineering and Management*, *139*(7), 768-784.

El-Diraby, T. E., & Kashif, K. F. (2005). Distributed ontology architecture for knowledge management in highway construction. *Journal of Construction Engineering and Management*, *131*(5), 591-603.

El-Gohary, N. M., & El-Diraby, T. E. (2010a). Dynamic knowledge-based process integration portal for collaborative construction. *Journal of Construction Engineering and Management*, *136*(3), 316-328.

El-Gohary, N. M., & El-Diraby, T. E. (2010b). Domain ontology for processes in infrastructure and construction. *Journal of Construction Engineering and Management*, *136*(7), 730-744.

El-Gohary, N. M., & El-Diraby, T. E. (2011). Merging architectural, engineering, and construction ontologies. *Journal of Computing in Civil Engineering*, *25*(2), 109-128.

ESRI. (2006). About geographic coordinate systems http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=About_geographic_coordinate_systems

ESRI. (2007). An overview of linear referencing http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=An_overview_of_linear_referencing

ESRI. (2008). The Geodatabase: Modeling and Managing Spatial Data https://www.esri.com/news/arcnews/winter0809articles/the-geodatabase.html

FDM. (2019). FACILITIES DEVELOPMENT MANUAL, Chapter 11: Design, Wisconsin Department of Transportation, 2019. https://wisconsindot.gov/rdwy/fdm/fd-11-00toc.pdf

Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). Methontology: from ontological art towards ontological engineering.

Findley, D. J. (2016). Highway Engineering, Institute for Transportation Research and Education, North Carolina State University, 2016

Fisher, M., Blace, R., Hebeler, J., & Perez-Lopez, A. (2011). *Semantic web programming*. John Wiley & Sons.

freeClass. (2022). freeClass building materials. https://www.freeclass.eu/?freeclass_v1

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing?. *International journal of human-computer studies*, *43*(5-6), 907-928.

Gruber, T. (2001). What is an ontology, Stanford Knowledge Systems Laboratory. http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

Grüninger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies.

Guarino, N. (1997). "Understanding, building, and using ontologies—A commentary to Using explicit ontologies in KBS development by van Heijst, Schreiber, and Wielinga." *International journal of human-computer studies*, *46*(2-3), 293-310.

Harmelink, D. J., & Rowings, J. E. (1998). Linear scheduling model: Development of controlling activity path. *Journal of Construction Engineering and Management*, *124*(4), 263-268.

Hinze, J. (2004). *Construction planning and scheduling* (Vol. 228, p. 229). Upper Saddle River, NJ: Pearson/Prentice Hall.

Hodgson, R., Keller, P. J., Hodges, J., and Spivak, J. (2011). QUDT–Quantities, Units, Dimensions and Data Types Ontologies. 2013.
http://qudt.org.

Horridge, M., Jupp, S., Moulton, G., Rector, A., Stevens, R., & Wroe, C. (2009). A practical guide to building owl ontologies using protégé 4 and co-ode tools edition1. 2. *The university of Manchester*, *107*.

ISO 19107. (2001). Geographic Information—Spatial Schema. International Organization for Standardization, Geneva, Switzerland, 2001.

ISO 19111. (2002). Geographic Information—Spatial Referencing by Coordinates. International Organization for Standardization, Geneva, Switzerland, 2002.

ISO 19115. (2002). Geographic Information—Metadata. International Organization for Standardization, Geneva, Switzerland, 2002.

ISO 19133. (2003). Geographic Information—Location Based Services Tracking and Navigation. International Organization for Standardization, Geneva, Switzerland, 2003.

Karshenas, S., & Niknam, M. (2013). Ontology-based building information modeling. In *Computing in Civil Engineering (2013)* (pp. 476-483).

Karshenas, S., & Sharma, A. (2010). Visually scheduling construction projects. In *Construction Research Congress 2010: Innovation for Reshaping Construction Practice* (pp. 490-499).

Kavanagh, B.F. and Glenn, S.J. (1992). Surveying Principles and Applications (Third Edition), Prentice Hall, Englewood Cliffs, New Jersey, USA

Keet, C. (2018). *Introduction to ontology engineering.* College Publications.

Kerzner, H. (1998a). *Project management: A systems approach to planning, scheduling, and controlling*, 6th Ed., Wiley, New York.

Kerzner, H. (1998b). *In search of excellence in project management: Successful practices in high performance organizations*, Wiley, New York.

Koncz, N., & Adams, T. M. (2002). A data model for multi-dimensional transportation location referencing systems. *URISA journal*, *14*(2), 27-41.

Kuchibhotla, H. N., Dunn, D., & Brown, D. (2009). Data integration issues in IT organizations and a need to map different data formats to store them in relational databases. In *2009 41st Southeastern Symposium on System Theory* (pp. 1-6). IEEE.

LandXML. (2017). Available online: http://www.landxml.org/

Lefler, N., Council, F., Harkey, D., Carter, D., McGee, H., & Daul, M. (2010). *Model inventory of roadway elements-MIRE, version 1.0* (No. FHWA-SA-10-018).

Maier, F., Chummers, L. E., Pulikanti, S., Struthers, J. Q., Mallela, J., & Morgan, R. H. (2017). *Utilizing 3D Digital Design Data in Highway Construction-Case Studies* (No. FHWA-HIF-17-027). United States. Federal Highway Administration. Office of Infrastructure Research and Development.

Mamoulis, N. (2011). Spatial data management. *Synthesis Lectures on Data Management*, *3*(6), 1-149.

Mascardi, V., Locoro, A., & Rosso, P. (2009). Automatic ontology matching via upper ontologies: A systematic evaluation. *IEEE Transactions on knowledge and data engineering*, *22*(5), 609-623.

Megan Katsumi. (2016). Public Transit Ontology. Retrieved from: http://ontology.eil.utoronto.ca/icity/PublicTransit/1.2/

Miller, H. J., & Shaw, S. L. (2001). *Geographic information systems for transportation: principles and applications*. Oxford University Press on Demand.

Motik, B., Sattler, U., & Studer, R. (2005). Query answering for OWL-DL with rules. *Journal of Web Semantics*, *3*(1), 41-60.

Nandini, D., & Shahi, G. K. (2019). An ontology for transportation system. *Kalpa Publications in Computing*, *10*, 32-37.

NCHRP. (1974). NCHRP Synthesis of Highway Practice 21: Highway Location Reference Methods. *Transportation Research Board, National Research Council, Washington, DC*, 29.

Nesi, P., Badii, C., Bellini, P., Cenni, D., Martelli, G., & Paolucci, M. (2016). Km4City Smart City API: an integrated support for mobility services. In *2016 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 1-8). IEEE.

Niknam, M. (2015). *A semantics-based approach to construction cost estimating* (Doctoral dissertation, Marquette University).

Niknam, M., & Karshenas, S. (2015). Integrating distributed sources of information for construction cost estimating using Semantic Web and Semantic Web Service technologies. *Automation in construction*, *57*, 222-238.

Niknam, M., & Karshenas, S. (2016). Integrating BIM and project schedule information using semantic web technology. In *Construction Research Congress 2016* (pp. 689-697).

Niknam, M., & Karshenas, S. (2017). A shared ontology approach to semantic representation of BIM data. *Automation in Construction*, *80*, 22-36.

NJDOT. (2013). Construction Scheduling Manual, Department of Transportation, the state of New Jersey, 2013 https://www.state.nj.us/transportation/eng/documents/BDC/pdf/CSM20130430.pdf

Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology.

O'Leary, D. E. (1997). Impediments in the use of explicit ontologies for KBS development. *International Journal of Human-Computer Studies*, *46*(2-3), 327-337.

Ontology Engineering Group. (2015). FreeClassOWL Ontology http://smartcity.linkeddata.es/ontologies/www.freeclass.eufreeclass_v1.owl.html

Pauwels, P., Zhang, S., & Lee, Y. C. (2017). Semantic web technologies in AEC industry: A literature overview. *Automation in construction*, *73*, 145-165.

Pinto, H. S., Staab, S., & Tempich, C. (2004). DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies. In *ECAI* (Vol. 16, p. 393).

Prud'Hommeaux, E., and Seaborne, A. (2008). "SPARQL query language for RDF." W3C Recommendation, 15.

Reed, D. (2006). Take a good look. Data Strategy, 2(4), 24- 29. Retrieved November 17, 2008, from Business Source Complete database.

Ross, D., Scarponcini, P., Tucker, J., and Bolt. M. (2002). Mn/DOT Location Data Management Project P075—Implementing the NCHRP 20-27(2) LRS Conceptual Model, Geospatial Information Systems for Transportation Symposium (GIS-T). Atlanta, Ga., 2002.

RSMeans. (2015). RSMeans Construction Cost Database. https://www.rsmeans.com/

Scarponcini, P. (2005). ISO 19133 tracking and navigation standard: 6.6 Linear reference system standard. *Transportation research record*, *1935*(1), 77-84.

Schreiber, G., Wielinga, B., & Jansweijer, W. (1995). The KACTUS view on the 'O'word. In IJCAI workshop on basic ontological issues in knowledge sharing (Vol. 20, p. 21).

Segaran, T., Evans, C., and Taylor, J. (2009). Programming the semantic web. O'Reilly Media, Inc.

Sesame bhttp://rdf4j.org/N, 2015 (2015).

Sequeda, J. (2012). "Introduction to: RDF vs XML." http://www.dataversity.net/introduction-to-rdf-vs-xml/ (06/02, 2015).

Smith, S. F., & Becker, M. A. (1997, March). An ontology for constructing scheduling systems. In *Working Notes of 1997 AAAI Symposium on Ontological Engineering* (pp. 120-127). Stanford, CA: AAAI Press.

Stanford University. (2015). "Protege." http://protege.stanford.edu/ (06/02, 2015).

Suárez-Figueroa, M. C., García-Castro, R., Villazón-Terrazas, B., & Gómez-Pérez, A. (2011). Essentials in ontology engineering: methodologies, languages, and tools.

Suárez-Figueroa, M. C., Gómez-Pérez, A., Motta, E., & Gangemi, A. (2012). Introduction: Ontology engineering in a networked world. In *Ontology engineering in a networked world* (pp. 1-6). Springer, Berlin, Heidelberg.

Sure, Y. (2003). *Methodology, tools & case studies for ontology based knowledge management* (Doctoral dissertation, Karlsruhe, Univ., Diss., 2003).

Sure, Y., Staab, S., & Studer, R. (2004). On-to-knowledge methodology (OTKM). In *Handbook on ontologies* (pp. 117-132). Springer, Berlin, Heidelberg.

Törmä, S. (2013). Semantic linking of building information models. In *2013 IEEE Seventh International Conference on Semantic Computing* (pp. 412-419). IEEE.

Uschold, M., & Jasper, R. (1999). A framework for understanding and classifying ontology applications. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden* (Vol. 2).

Uschold, M., & King, M. (1995). *Towards a methodology for building ontologies* (pp. 1-13). Edinburgh: Artificial Intelligence Applications Institute, University of Edinburgh.

van Nederveen, S., Luiten, B., Böhms, M. (2015). *Linked Data for Road Authorities*. Proc. of the 32nd CIB W78 Conference 2015, 27th-29th 2015, Eindhoven, The Netherlands, 577-588.

Vittori, C. M., Dorneles, C. F., & Heuser, C. A. (2001). Creating XML documents from relational data sources. In *International Conference on Electronic Commerce and Web Technologies* (pp. 60-70). Springer, Berlin, Heidelberg.

Vonderohe, A., Chou, C. L., Sun, F., & Adams, T. (1997). A generic data model for linear referencing systems. In *Research Results Digest 218. National Cooperative Highway Research Program. Transportation Research Board*.

W3C. (2008). SPARQL Query Language for RDF (01/15, 2008) https://www.w3.org/TR/rdf-sparql-query/

W3C. (2014). "The organization ontology." http://www.w3.org/TR/vocab-org/

W3C Standard. (2015). "Ontology." http://www.w3.org/standards/semanticweb/ontology

W3C (2016). Extensible Markup Language (XML) https://www.w3.org/XML/

W3C (2017). Time Ontology in OWL Homepage: https://www.w3.org/TR/owl-time/

Yamın, R. A., & Harmelink, D. J. (2001). Comparison of linear scheduling model (LSM) and critical path method (CPM). *Journal of construction engineering and management*, *112*(4), 476-491.

Yim, H. J., Seo, D., Jung, H., Back, M. K., Kim, I., & Lee, K. C. (2017). Description and classification for facilitating interoperability of heterogeneous data/events/services in the Internet of Things. *Neurocomputing*, *256*, 13-22.

Zeiler, M. (1999). *Modeling our world: the ESRI guide to geodatabase design* (Vol. 40). ESRI, Inc..

Ziering, E. A. (2007). *TransXML: XML schemas for exchange of transportation data* (Vol. 576). Transportation Research Board.

APPENDIX 1

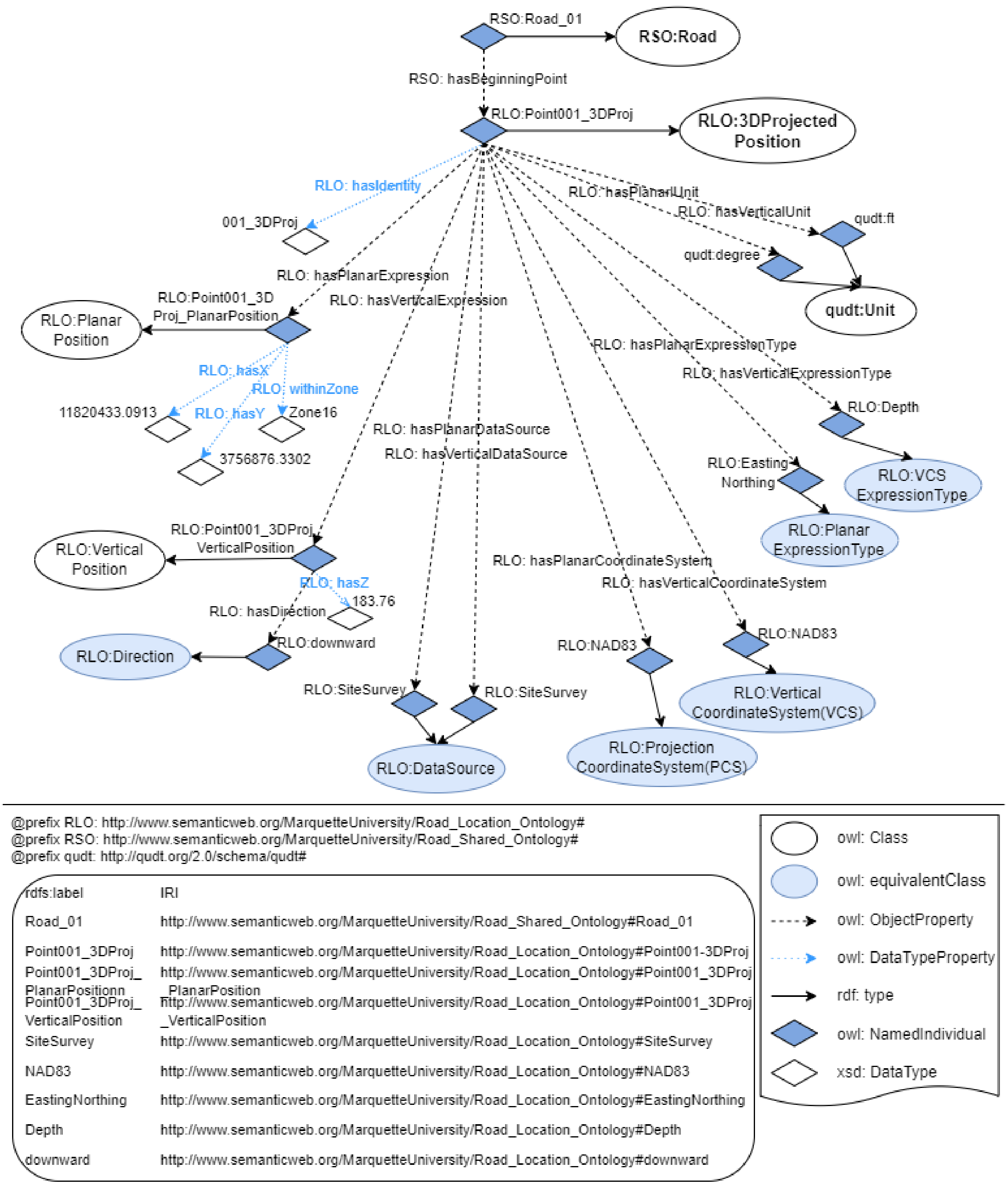


Figure 55 A point location example of 3D projected position

APPENDIX 2

RDF/XML REPRESENTATION OF THE ROAD EXAMPLE GIVEN IN CHAPTER 4

```
<?xml version="1.0"?>
<rdf:RDF
xmlns="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#"

xml:base="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology"
     xmlns:dc="http://purl.org/dc/elements/1.1/"

xmlns:RLO="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#"

xmlns:RSO="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:prov="http://www.w3.org/ns/prov#"
     xmlns:qudt="http://qudt.org/2.1/schema/qudt#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:skos="http://www.w3.org/2004/02/skos/core#"
     xmlns:time="http://www.w3.org/2006/time#2016#"
     xmlns:vaem="http://www.linkedmodel.org/2.0/schema/vaem#"
     xmlns:dtype="http://www.linkedmodel.org/1.1/schema/dtype#"
     xmlns:qudt2="http://qudt.org/schema/qudt/"
     xmlns:terms="http://purl.org/dc/terms/"
     xmlns:vaem1="http://www.linkedmodel.org/schema/vaem#"
     xmlns:dtype3="http://www.linkedmodel.org/schema/dtype#">
    <owl:Ontology
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#"/
>



    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Object Properties
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->
```

```
    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasAlignmen
t -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asAlignment">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Phase"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Alignment"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasAssembly
-->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asAssembly">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Segment"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Assembly"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasBeginnin
gPoint -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asBeginningPoint">
```

```
        <rdfs:subPropertyOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#hasLocation"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasBeginnin
gTime -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asBeginningTime">
        <rdfs:subPropertyOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#hasTime"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasCompone
nt -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asComponent">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Lane"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Pavement"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Subsurface"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasDevice -
->
```

```
   <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asDevice">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Road"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Device"/>
   </owl:ObjectProperty>



   <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasEndPoint
-->

   <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asEndPoint">
        <rdfs:subPropertyOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#hasLocation"/>
   </owl:ObjectProperty>



   <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasEndTime
-->

   <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asEndTime">
        <rdfs:subPropertyOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#hasTime"/>
   </owl:ObjectProperty>



   <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasEvent -->
```

```
    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asEvent">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Road"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Event"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasLane -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asLane">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#TraveledWay"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Lane"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasLayer -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asLayer">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Pavement"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Subsurface"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Layer"/>
    </owl:ObjectProperty>
```

```
    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasLocation
-->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asLocation">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Phase"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Road"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Segment"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#AbsoluteLocation"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasPhase -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asPhase">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Road"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Phase"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasSegment
-->
```

```
    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asSegment">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Phase"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Segment"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasSubassem
bly -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asSubassembly">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Assembly"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Subassembly"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasTime -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asTime">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Phase"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Segment"/>
        <rdfs:range rdf:resource="http://www.w3.org/2006/time#TemporalEntity"/>
    </owl:ObjectProperty>
```

```
    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasTraffic -
->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asTraffic">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Road"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Traffic"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#within -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
within">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Road"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Jurisdiction"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#withinSite -
->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
withinSite">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Road"/>
```

```
        <rdfs:range
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Site"/>
    </owl:ObjectProperty>



    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Data properties
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->




    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#hasIdentity -
->

    <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#h
asIdentity">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Assembly"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Phase"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Road"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Segment"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Subassembly"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:DatatypeProperty>



    <!--
```

```
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Classes
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->




    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#AbsoluteL
ocation -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology
#AbsoluteLocation"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Alignment -
->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
Alignment"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Assembly -
->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
Assembly"/>



    <!-- http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Base
-->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#B
ase">
```

```
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Layer"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Binder -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#B
inder">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Layer"/>
    </owl:Class>



    <!-- http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Curb
-->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#C
urb">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Subassembly"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#CurveDesign
Profile -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#C
urveDesignProfile">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#DesignProfile"/>
    </owl:Class>
```

```
    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#CurveGroun
dProfile -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#C
urveGroundProfile">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#GroundProfile"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#CurveHorizo
ntalAlignment -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#C
urveHorizontalAlignment">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#HorizontalAlignment"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#DesignProfil
e -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
DesignProfile">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#VerticalAlignment"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Device -->
```

```
    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
Device"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Drainage -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
Drainage">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Subassembly"/>
    </owl:Class>



    <!-- http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Event
-->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#E
vent"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#GroundProfil
e -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
GroundProfile">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#VerticalAlignment"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#HorizontalAl
ignment -->
```

```
    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
HorizontalAlignment">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Alignment"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Jurisdiction -
->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#J
urisdiction"/>



    <!-- http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Lane
-->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#L
ane"/>



    <!-- http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Layer
-->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#L
ayer"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Median -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
Median">
```

```
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Subassembly"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Pavement -
->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#P
avement"/>



    <!-- http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Phase
-->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#P
hase"/>



    <!-- http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Road
-->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#R
oad"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#SealCoat -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
ealCoat">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Layer"/>
    </owl:Class>
```

```
    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Segment -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
egment"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Shoulder -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
houlder">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Subassembly"/>
    </owl:Class>



    <!-- http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Site -
->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
ite"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#StraightDesi
gnProfile -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
traightDesignProfile">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#DesignProfile"/>
    </owl:Class>
```

```
    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#StraightGrou
ndProfile -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
traightGroundProfile">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#GroundProfile"/>
    &lt;/owl:Class&gt;



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#StraightHori
zontalAlignment -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
traightHorizontalAlignment">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#HorizontalAlignment"/>
    &lt;/owl:Class&gt;



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Subassembly
-->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
ubassembly"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Subbase -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
ubbase">
```

```
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Layer"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Subgrade -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
ubgrade">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Layer"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Subsurface -
->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
ubsurface"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Surface -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
urface">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Layer"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#TackCoat -
->
```

```
    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#T
ackCoat">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Layer"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Traffic -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#T
raffic"/>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#TraveledWa
y -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#T
raveledWay">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Subassembly"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#VerticalAlig
nment -->

    <owl:Class
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
VerticalAlignment">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Alignment"/>
    </owl:Class>
```

```
    <!-- http://www.w3.org/2006/time#TemporalEntity -->

    <owl:Class rdf:about="http://www.w3.org/2006/time#TemporalEntity"/>



    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Individuals
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->




    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#Point001 -
->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology
#Point001">
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#AbsoluteLocation"/>
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#Point100 -
->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology
#Point100">
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#AbsoluteLocation"/>
    </owl:NamedIndividual>
```

```
    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology#Point200 -
->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontology
#Point200">
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#AbsoluteLocation"/>
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Align_01 -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
Align_01">
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Alignment"/>
        <hasIdentity>Align_01</hasIdentity>
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Assembly_0
1 -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
Assembly_01">
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Assembly"/>
        <hasSubassembly
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Shoulder_01"/>
        <hasSubassembly
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Traveled-Way_01"/>
        <hasIdentity>Assembly_01</hasIdentity>
```

```
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Assembly_0
2 -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
Assembly_02">
        <hasIdentity>Assembly_01</hasIdentity>
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Pavement_0
1 -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#P
avement_01">
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Pavement"/>
        <hasLayer
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Surface_01"/>
        <hasIdentity>Pavement_01</hasIdentity>
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Phase_01 -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#P
hase_01">
        <hasAlignment
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Align_01"/>
        <hasBeginningPoint
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#Point001"/>
```

```
        <hasBeginningTime rdf:resource="http://www.w3.org/2006/time#2021-01-01"/>
        <hasEndPoint
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#Point100"/>
        <hasEndTime rdf:resource="http://www.w3.org/2006/time#2021-03-01"/>
        <hasSegment
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Seg_01"/>
        <hasSegment
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Seg_02"/>
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Road_01 -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#R
oad_01">
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Road"/>
        <hasBeginningPoint
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#Point001"/>
        <hasEndPoint
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#Point200"/>
        <hasPhase
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Phase_01"/>
        <within
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#WisDOT"/>
        <hasIdentity>Road_01</hasIdentity>
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Seg_01 -->
```

<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
eg_01">
<rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Segment"/>
<hasAssembly
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Assembly_01"/>
<hasBeginningPoint
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#Point001"/>
<hasBeginningTime rdf:resource="http://www.w3.org/2006/time#2021-01-01"/>
<hasEndPoint
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Location_Ontolo
gy#Point100"/>
<hasEndTime rdf:resource="http://www.w3.org/2006/time#2021-02-01"/>
<hasIdentity>Seg_01</hasIdentity>
</owl:NamedIndividual>

<!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Seg_02 -->

<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
eg_02">
<rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Segment"/>
<hasAssembly
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Assembly_02"/>
<hasIdentity>Seg_02</hasIdentity>
</owl:NamedIndividual>

<!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Shoulder_01
-->

<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
houlder_01">

```
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Shoulder"/>
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Surface_01 -
->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#S
urface_01">
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Surface"/>
        <hasIdentity>Surface_01</hasIdentity>
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#Traveled-
Way_01 -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#T
raveled-Way_01">
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#TraveledWay"/>
        <hasComponent
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Pavement_01"/>
        <hasIdentity>Traveled-Way_01</hasIdentity>
    </owl:NamedIndividual>



    <!--
http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#WisDOT -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology#
WisDOT">
```

```
        <rdf:type
rdf:resource="http://www.semanticweb.org/MarquetteUniversity/Road_Shared_Ontology
#Jurisdiction"/>
    </owl:NamedIndividual>



    <!-- http://www.w3.org/2006/time#2021-01-01 -->

    <owl:NamedIndividual rdf:about="http://www.w3.org/2006/time#2021-01-01">
        <rdf:type rdf:resource="http://www.w3.org/2006/time#TemporalEntity"/>
    </owl:NamedIndividual>



    <!-- http://www.w3.org/2006/time#2021-02-01 -->

    <owl:NamedIndividual rdf:about="http://www.w3.org/2006/time#2021-02-01">
        <rdf:type rdf:resource="http://www.w3.org/2006/time#TemporalEntity"/>
    </owl:NamedIndividual>



    <!-- http://www.w3.org/2006/time#2021-03-01 -->

    <owl:NamedIndividual rdf:about="http://www.w3.org/2006/time#2021-03-01">
        <rdf:type rdf:resource="http://www.w3.org/2006/time#TemporalEntity"/>
    </owl:NamedIndividual>
</rdf:RDF>



<!-- Generated by the OWL API (version 4.5.9.2019-02-01T07:24:44Z)
https://github.com/owlcs/owlapi -->
```