Development of Earthwork Ontology and its Application

Alhusain Mohamed Taher

A Thesis
In the Department

of

Concordia Institute for Information Systems Engineering (CIISE)

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Information Systems Engineering) at

Concordia University

Montreal, Quebec, Canada

March 2021

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Alhusain Mohamed Taher

Entitled: **Development of Earthwork Ontology and its Application**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information Systems Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Liangzhu Wang	Chair
Dr. Jeff Rankin	External Examiner
Dr. Sang Hyeok Han	External to Program
Dr. Anjali Awasthi	Examiner
Dr. Jamal Bentahar	Examiner
Dr. Amin Hammad	Thesis Supervisors
Dr. Faridaddin Vahdatikhaki	

Approved by

Dr. Mohammad Mannan, Graduate Program Director

April, 30 2021
Date of Defence

Dr. Amir Asif, Dean
Gina Cody School of Engineering and Computer Science

ABSTRACT

Development of Earthwork Ontology and its Application

Alhusain Mohamed Taher, Ph.D.
Concordia University, 2021

In a typical construction project, a significant amount of information is communicated to various stakeholders at different phases of the project lifecycle. The communication of this information tends to be informal and ad-hoc in the majority of the cases, which makes it more susceptible to loss of information or misinterpretation. Earthwork operations, which are one of the main operations of construction projects, also struggle with the challenge of effective information communication. There is an apparent shortcoming regarding the unified structure for data and information exchange in this domain. The existing models and ontologies do not address the explicit semantic representation of earthwork operations. Accordingly, there is a need for a knowledge model to formalize the efficient communication of information. An ontological model can be used to organize the domain knowledge so that it can be utilized and reused by the stakeholders.

The primary purpose of this study is to develop an ontology for the earthwork domain that can be used to create the semantics-based integration method to support the communications between the different disciplines and stakeholders in the earthwork domain. Accordingly, the objectives of this study are: (1) To extract the explicit and tacit knowledge required for the earthwork domain; (2) To formalize the extracted knowledge by developing the Earthwork Ontology (EW-Onto); (3) To develop methods for linking and coupling EW-onto with other existing relevant ontologies in the construction domain to extend its application for safety and productivity; and (4) To evaluate the integrated ontology (IEW-Onto) and apply the ontological model in supporting application development, which is a Multi-Agent System (MAS) in the earthwork domain.

In the proposed framework, the ontology integrates the different components in the domain. The extended earthwork ontology (called Integrated Earthwork Ontology or IEW-Onto) is composed of the concepts, relationships, and axioms in this domain and can represent the semantic values of the entities and the relationships. Each entity is linked with other entities with different types of relationships, such as *is-a*, *part-of*, *operates*, and *coordinates*. IEW-Onto benefits from the available ontologies in the construction domain, and links with other ontologies, such as sensor and soil ontologies. IEW-Onto is used to build the earthwork operation model as a pattern to represent the operations and processes sequences, which provide a reusable pattern for several applications such as MAS. The developed MAS can cope with the complexity of earthwork operations' communication at the fleet level and addresses safety issues. In the MAS, every piece of equipment is represented by a dedicated computer agent. This Ontology-based MAS is expected to improve the safety of earthwork operations. Different evaluation methods were used to evaluate EW-Onto and IEW-Onto, including checking consistency, survey, data-driven and application-based validations. The evaluation results show that both ontologies have consistency and provide a high level of clarity, richness, comprehensiveness, interpretability, and effectiveness of the presented knowledge in the earthwork domain.

To my parents' souls

To my wife

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisors, Dr. Amin Hammad and Dr. Faridaddin Vahdatikhaki, for their guidance, insightful discussion, encouragement, and support throughout this thesis's development.

I would, also, like to thank the member of my committee, Dr. Jeff Rankin, Dr. Sang Hyeok Han, Dr. Anjali Awasthi, and Dr. Jamal Bentahar for their valuable inputs and precious time.

I want to express my deepest gratitude for the constant support, understanding and love that I received from my wife, my daughters, and my big family during the past years. This accomplishment would not have been possible without their support.

I want to express my gratitude to my colleagues in the lab. I spent countless hours with them in the past few years.

I would like to dedicate this thesis to my parents' souls.

Finally, I would like to thank all the people who assisted me in completing this work. Thank you all.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AEC	Architecture, Engineering, and Construction
AASHTO	American Association of State Highway and Transportation Officials
AI	Artificial Intelligent
AL	Attributive Language
AMG/C	Automated Machine Guidance and Control
BIM	Building Information Modeling
BLE	Bluetooth Low Energy
BPM	Business Process Management
BrIM	Bridge Information Modeling
CAD	Computer-Aided Design
CIM	Civil Information Modeling
DBA	Database Agent
DDA	Design Document Agent
DL	Description Logics
DTM	Digital Terrain Model
EW-Onto	Earthwork Ontology
FIPA	Foundation for Intelligent Physical Agents
FM	Facilities Management
FO	Foundation Ontology
FOL	First-Order Logic
GCA	General Coordinator Agent
GPS	Global Positioning System
IAI	International Alliance for Interoperability
IC-PRO-Onto	Infrastructure and Construction PROcess Ontology
ICIS	International Construction Information Society
IDEF	Integrated DEFinition
IEW-Onto	Integrated Earthwork Ontology
IFC	Industrial Foundation Classes
ifcXML	IFC Extensible Markup Language
IoT	Internet of Things
IR	Infrared
ISO	International Organization for Standardization
JHA	Job Hazard Analysis
KM	Knowledge Management
LCH	Leacock and Chodorow
LoD	Level of Detail
LOS	Line-of-Sight
MAS	Multi-Agent System

MSDF	Multi-Sensor Data Fusion
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OA	Operator Agent
OntA	Ontology Agent
ORA	Occupational Risk Assessment
OSHA	Occupational Safety and Health Administration
OSP	Ontology of Soil Properties and Processes
OWL	Ontology Web Language
PDA	Project Document Agent
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
ResA	Resource agent
RF	Radio Frequency
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator
SensorML	Sensor Model Language
SSA	Site State Agent
SSN	Semantic Sensor Network
RTLS	Real-Time Location System
SWRL	Semantic Web Rule Language
TCA	Team Coordinator Agent
TOA	Time of Arrival
TOVE	TOronto Virtual Enterprise
TSA	Teams Setup Agent
UML	Unified Modeling Language
Uniclass	UNIfied CLASSification
US	Ultrasound Signals
USCS	Unified Soil Classification System
USDA	U.S. Department of Agriculture
UWB	Ultra-Wide Band
W3C	World Wide Web Consortium
WLAN	Wireless local area network
XML	Extensible Markup Language

CHAPTER 1 INTRODUCTION

1.1 BACKGROUND

Earthwork operations can be represented as a collection of complex and dynamic tasks, which are affected by the workflow, information flow, and other random factors of construction activities (Cheng et al., 2011). Formalizing the information exchange between the different stakeholders is one of the challenges of construction projects. Information is produced from a large amount of data collected from different sources related to the project, such as earthwork equipment, target operations, surrounding environment, and planning and execution details. These data are used after being processed to enhance the communication between different operators, increase productivity and safety, and improve the decision-making process. In recent years, the advent of technologies such as Building Information Modeling (BIM) and Automated Machine Guidance and Control (AMG/C), in which various sensors and location systems, such as the Global Positioning System (GPS), are used to support equipment operators, has significantly increased the amount and scope of information and data flow in earthwork operations (Hammad et al. 2013). On the other hand, the ever-increasing complexity of modern projects means that the number of stakeholders that are involved in a project is growing. In turn, this translates to an increased volume of information generated throughout a project. To ensure the success of the project, it is indispensable to seamlessly integrate and manage this information.

The equipment can be instrumented with sensors (e.g., GPS receivers) to collect the data needed to guide or control the equipment (Hammad et al. 2013). This information can be formalized and modelled as ontologies to support the different earthwork operations' stakeholders. Ontologies have been used under the Artificial Intelligent (AI) umbrella to capture the knowledge in a domain (Russell et al. 2010).

In recent years, ontologies have been developed in different domains to share and reuse the knowledge and to improve the communication and logical reasoning between the various entities in these domains. Ontology has different definitions, but the most used definition is "a formal and explicit specification of a shared conceptualization" (Gruber 1993). In AI, the term "ontology" has one of two meanings: "a representation vocabulary, often specialized to some domain or subject matter; a body of knowledge describing some particular domain, using the representation vocabulary." (Chandrasekaran et al. 1999). Gruber (1993) claimed that the ontology captures and converts the knowledge into machine-readable, interpretable, and explicit presentations. Ideally, ontologies are used to formalize the understanding of the domain and provide the machine-human interaction. Ontologies are central for different domains, such as commerce, medicine, and food sciences (Cantais et al. 2005; Hu et al. 2003; Leukel et al. 2006).

In construction, ontologies have been developed to share and reuse knowledge and to improve the communication and logical reasoning between various entities (Katranuschkov et al. 2009). Examples of such ontologies include an ontology-based framework for identifying job hazards in construction (Zhang et al. 2015), transaction ontology in the domain of infrastructure management (Zeb and Froese 2012) and domain ontology for processes in infrastructure and construction (El-Gohary and El-Diraby 2010).

Nevertheless, while existing ontologies cover a vast area within the construction sector, they fail to fully support earthwork operations. Most of the software tools used for the management of earthwork operations are based on the properties of entities in the domain and do not support the integration based on an ontology or semantic representations (Liu et al. 2016). The existing information models, such as BIM, which is widely used in construction projects, provide a repository of the digital and shared information models. BIM models are information-centric with less focus on knowledge modelling (Ho et al., 2013; Wu, 2013). In the meanwhile, with the increasing demand to improve productivity and efficiency, managing and sharing the knowledge play an important role in the project lifecycle. BIM does not fully support semantic representation at various levels (e.g., operations, processes, etc.). Given the magnitude and criticality of earthwork operations, this is a major limitation.

The advantages of using an ontology within the earthwork domain include: (1) The ontology links and identifies the relationships between the concepts and classifies the knowledge in a hierarchical way accepted by the experts and the end-users in the domain. Moreover, the ontology can facilitate the management of earthwork operations and simplify information exchange and interoperability. (2) The information, which is structured in the context of a robust knowledge, can be used to increase the stakeholders' knowledge of earthwork operations. This knowledge can improve the communication to increase productivity, safety and enhance the decision-making process. (3) Ontologies are the cornerstone of the linked data with the ability to be implemented in different languages (e.g., OWL) accepted by the World Wide Web Consortium (W3C) (Radulovic et al.

2015). Linked data does not only overcome the interoperability issue by enabling the linking of different heterogeneous data sets among the same domain, but also it facilitates extending the data to be linked to other data from other sources.

1.2 PROBLEM STATEMENT

According to a study by Thomas et al. (2018), which included 599 construction industry leaders from the United States, Australia, New Zealand, United Kingdom, and Canada, on average, 52% of all reworks in construction are caused by poor data and miscommunication. These reworks cost the construction industry in 2018 in the US alone $65.2 Billion. About 48% ($31.3 Billion) of this cost is due to poor data integration and miscommunication between stakeholders in the projects. Moreover, 35% of professional time is spent on non-optimal activities. These non-optimal activates cost more than 14 hours/person each week, including looking for project data/information, conflict resolution, and dealing with mistakes and rework. Therefore, and due to the growth of the construction projects' complexity, coupled with the growth of the number of stakeholders with different interests, there is a need for consistent and formalized collaboration to share the knowledge and improve the communications among the project stakeholders.

In construction projects (e.g., roads, bridges, highways, and dams), earthwork operations are one of the main portions of a project. More than 20% of the total cost of road projects is dedicated to earthwork operations (Vahdatikhaki 2015). Consequently, capturing and representing the knowledge about the earthwork domain (e.g., classifications, properties, relationships, etc.) and sharing it among the stakeholders play an important role in the project's success at different levels. One of the main challenges is the lack of a unified and consistent knowledge representation of the earthwork domain among project stakeholders. The existing tools are based on textual documents and the graphical representation models (e.g., BIM) rather than an integrated knowledge representation (Liu et al. 2016). The semantic representation and the taxonomies at different levels (e.g., operations, processes, etc.) and between the different disciplines are still implicit, and thus, limit sharing the integrated knowledge within the earthwork domain.

Based on the review of the existing studies (e.g., El-Diraby and Osman, 2011; El-Gohary and El-Diraby, 2010; Labban et al., 2013; Viljamaa and Peltomaa, 2014; Wang et al., 2010; Wang and Boukamp, 2011, Zhang et al., 2015), the following research gaps can be identified: (1) A formal knowledge representation and explicit classification of the earthwork resources (e.g., equipment) is missing; (2) The knowledge that supports the decision-making to improve the productivity and safety in the earthwork domain is fragmented; (3) There is a demand for smart construction support; and (4) The integration of semantically rich data into earthwork planning tools is missing.

1.3 RESEARCH HYPOTHESIS

It is important to formalize the related knowledge and represent it in a way that can be shared among different disciplines. These needs are not limited to formalizing and sharing the earthwork domain knowledge, but also to coupling with the knowledge that is interrelated to the domain. As such, an integrated knowledge representation, including the concepts, relationships, axioms, and taxonomies of the earthwork domain is needed.

It is hypothesized that the use of an ontology within the earthwork domain can help: (1) link and identify the relationships between concepts, define earthwork semantics, and classify knowledge in a hierarchical way accepted by experts and end-users. This would help establish a common ground for streamlined communication within the domain, which can eventually reduce the chance of miscommunication and misinterpretation of information during the design and construction phase; (2) facilitate the management of earthwork operations and simplify information exchange and interoperability between currently fragmented systems. This will allow easy development of integrative systems that build on the current specialized software to further automate and optimize the planning of earthwork operations; and (3) increase the stakeholders' knowledge of earthwork operations through the provision of the information, which is structured in the context of robust knowledge (Park et al., 2013). This knowledge can improve communication to increase productivity, safety and enhance the decision-making process. On a more practical note, an earthwork ontology can help develop platforms for easy integration of various types of data towards different goals. One example is a safety rule checker that integrates a BIM model with the project schedule and safety regulations to identify potential safety risks during the design phase. Another example is a platform that can link the inventory list of suppliers (e.g., equipment rental companies) with the planning of a project to help automate, streamline, and optimize the procurement of appropriate resources at the right time/price.

1.4 RESEARCH OBJECTIVES

The primary purpose of this study is to develop an ontology for the earthwork domain that can be used to create the semantics-based integration method to support the communications between the different disciplines and stakeholders in the earthwork domain. Accordingly, the objectives of this study are: (1) To extract the explicit and tacit knowledge required for the earthwork domain; (2) To formalize the extracted knowledge by developing the Earthwork Ontology (EW-Onto); (3) To develop methods for linking and coupling EW-onto with other existing relevant ontologies in the construction domain to extend its application for safety and productivity; and (4) To evaluate the integrated ontology (IEW-Onto) and apply the ontological model in supporting the development an application, which is a Multi-Agent System (MAS) in the earthwork domain.

1.5 RESEARCH SCOPE

The goal of the ontology is to provide a formal representation of the domain of interest. The ontology should provide the conceptual representation from different perspectives (e.g., technical, and managerial standpoints). The proposed research scope is defined by the following aspects:

- The study focuses on the earthwork operations, processes, and tasks, including classifications of the entities, the relationships between them, and developing the framework to contain all these components as a domain ontology.

- The developed ontology models the earthwork operation domain. This domain connects to other domains, such as the different project management knowledge areas. The concepts and relationships will be captured from the earthwork project perspective. Moreover, the ontology covers the planning and execution phases of the earthwork project.

- The intended users of the developed ontology are the stakeholders in the earthwork domain. The developed ontology may be utilized as support and foundation for other applications, such as developing MAS or simulation.

1.6 THESIS LAYOUT

The remaining chapters of the thesis are organized as the following:

- Chapter 2 presents the literature review of the research. This review covers the main areas of the research: ontologies, earthwork operations, using ontologies in construction, and MAS.

- Chapter 3 introduces an overview of the research methodology, including the steps for the development of EW-Onto, IEW-Onto, and their scope.

- Chapter 4 presents the initial development of EW-Onto in detail. It starts with the proposed method for the development of EW-Onto, including defining the concepts and the taxonomies, the implementation of the proposed method, and EW-Onto evaluation.

- Chapter 5 presents the development methods for integrating EW-onto with other existing relevant ontologies in the construction domain. It starts with the proposed framework, including the elements of IEW-Onto, the development process, and the integration process. Then, the implementation of IEW-Onto is presented. Finally, two evaluation approaches are applied to evaluate IEW-Onto.

- Chapter 6 discusses the summary, conclusions, contributions, limitations, and provides recommendations for future work.

CHAPTER 2 LITERATURE REVIEW

2.1 INTRODUCTION

This chapter presents a review of the literature in different areas that are related to the research. These areas include the earthwork operations and the major types of earthwork projects (e.g., dams and roads), the equipment's parts and attachments, earthwork operation levels, and the different types of earthwork operations. The earthwork domain includes various information and variables that affect safety, productivity, and the complexity of the interaction in earthwork operations. Therefore, simulation techniques are used to capture these complexities and create different plains to execute these complex operations. Thus, such computer models could benefit EW-Onto, which provides a unified and consistent representation of the complexity of the earthwork domain. Earthwork operations are reviewed in Section 2.2.

Soil classifications play significant roles in selecting the suitable equipment and the attachments that should be used to perform the operation. Moreover, these classifications are indispensable for safety and affect the productivity of earthwork operations. Therefore, the unified and consistent representation of these classifications will play an essential role in improving the safety of the earthwork domain. Soil classification is reviewed in section 2.3.

Data models in construction such as IFC and classification systems such as OmniClass are reviewed in Section 2.4 to provide an overview of the standards available in the construction industry and how important it is to provide a unified representation that can integrate with these different data models. Furthermore, data collection technologies are introduced. These different technologies applied in the construction industry are studied to cover the possibilities of integration with EW-Onto. These technologies are introduced and discussed in Section 2.5.

The new earthwork support technologies, including MAS and ontologies, are reviewed in Section 2.6. MAS and the data collection technology are studied and represented in different researches, such as (Skobelev et al. 2020; Vahdatikhaki et al. 2017; Dibley et al. 2012). These technologies are applied in recent years to improve safety and productivity in the earthwork domain and provide the evidence about the usability of ontologies to be integrated and linked with MAS in construction.

The ontology development principles, including the components of ontology, ontology languages, development methodologies, reusing the ontologies, and the ontology evaluation approaches, are reviewed in Sections 2.7 and 2.8.

Semantic technologies are increasingly used in the Architecture, Engineering, and Construction (AEC) industry and complement existing approaches (Hamdan and Scherer 2020). Different ontologies have been developed for different purposes in construction and could be reutilized and linked with different data resources using linked data. In linked data, ontologies represent the knowledge in flexible models (e.g., RDF), and thus, they are considered as the cornerstone of linked data systems. Linked data and their coupling with ontologies in construction are reviewed in Section 2.9.

2.2 EARTHWORK OPERATIONS

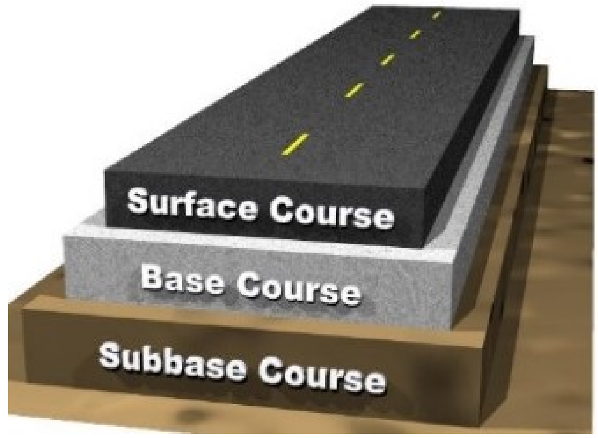
2.2.1 Major Types of Earthwork Projects

Earthwork is one of the most significant operations in roads and highways, earth dams, railroads, and airfields projects, as well as the foundations of buildings. In highway and road projects, earthwork is composed of different operations, such as cleaning, excavation, and embankment construction. As shown in Figure 2-1(a), an embankment is constructed of various types of materials, such as soil and rocks. These materials are structured as layers (e.g., subbase and base layers), which are called base course materials.

Compaction and grading are applied on each layer to create coherent and consistent multilayers as the basic structure for pavement, as shown in Figure 2-1(b). Constructing an embankment consists of different processes to make the soil more stable. In earth dam projects, a dam is structured by well-compacted earth, which may be mixed with watertight concrete, as shown in Figure 2-1(c). Earthwork is also an important task for constructing the foundation of buildings, as shown in Figure 2-1(d) through different operations, such as removing unwanted materials from the site and digging and dumping the earth into dumping areas.

The diversity of earthwork operations, which are coupled with different stakeholders, different disciplines, and a diversity of variables, such as disparate equipment, several attachments for each piece of equipment, and different technologies involved in the same project, have major impacts on the safety and the productivity of the earthwork project. However, it is important to handle and represent these components and variables in a consistence way that can be shared between the different stakeholders in the same project.

(a) Road Embankment Construction.



(b) Basic Structure for Pavement.

(c) Earth Dam.

(d) Building Foundation.

Figure 2-1 Types of earthwork for different projects (Delaware Department of Transportation 2020; Pavement Interactive 2010)

Given that earthwork operations are heavily equipment-driven, and given that different types of equipment can be used for different tasks (i.e., by using different attachments), the first step toward harmonization of knowledge in this domain is to properly classify different pieces of earthwork equipment (and their attachments) with respect to different tasks for which they can be used.

2.2.2 Earthwork Equipment and their Attachments

Earthmoving is performed by a variety of equipment individually or combined as a fleet. Different pieces of equipment have several types, sizes, and functionalities, which affect the selection and usage of the equipment. Most types of earthwork equipment consist of common parts, such as the engine, cab, cylinders, etc., that are usually similar, and even the experts may not be able to identify them in a unified way.

Hoes are the main type of *excavators* and have several types, sizes, and functionalities, and are used for different earthwork applications. Hoes are mainly used for excavation operations standalone or teamed with trucks to perform hauling operations. Front shovels, also called power shovels, are used for heavy excavation operations above the grade. Dozers can be used as excavation machines and to haul the soil or other materials by pushing over the earth's surface for a short distance less than 500 ft (about 152 m) (Gransberg et al. 2006). Dozers are used for excavating below the grade similar to other equipment, such as hoes and scrapers, with different work specifications (e.g., the speed and the dimensions of the workspace). Scrapers are used for rough cutting and filling of the topsoil for a distance in the range of 500 ft to 2 miles (about 152 to 3,219 m). Trucks are combined with other equipment (e.g., a front shovel, hoe, dozer, or loader) for hauling the materials for distances over 2 miles (about 3,219 m). Given that there is some overlap between the functionalities of various equipment, it is important to classify earthwork equipment into a well-organized taxonomy. Although textbooks define the scope of the equipment (e.g., Peurifoy et al. 2010; Gransberg et al. 2006), to the best of the author's knowledge, a comprehensive taxonomy of earthwork equipment is missing.

Equipment attachments are separate parts that are attached to the equipment to perform different types of tasks without changing the whole equipment; thus, increasing the equipment’s versatility and usability and reducing costs. Figure 2-2(a) illustrates examples of the attachments for a backhoe. As shown in Figure 2-2(b), a compaction wheel is attached to a hoe and used for the compaction operation in narrow spaces. In this case, the hoe, which is mainly used to excavate the earth below grade, is used as a compaction equipment by replacing the bucket with a compaction wheel or a vibratory plate. The change in the task assigned to the excavator will alter the classification of the equipment, from the functionality point of view, from the original classification as excavation equipment to another classification as compaction equipment. On the other hand, it is important for the project coordinators to know if the attachments of the equipment (e.g., grapple, hammer, and compact plate) are available or not. Therefore, it is important to formalize, represent and share the classifications for these equipment and their attachments, including the concepts, the relationships and the related regulations and rules in a consistent way to enhance the safety and improve productivity in the earthwork domain.

(a) for Backhoe

(b) Compaction wheel attached to Hoe

Figure 2-2 Equipment attachment (Debbie 2016; JCB 2021)

2.2.3 Levels of Earthwork Operations

Earthwork operations account for a considerable portion of the total cost of a project (Vahdatikhaki et al. 2017). Therefore, any improvements in these operations can result in significant savings both in time and cost of the overall project (Rezazadeh Azar and McCabe 2011).

Depending on the type of the project, earthwork can comprise different operations. For instance, in highway and road projects, earthwork is composed of cleaning, excavation, embankment construction, compaction and grading (Delaware Department of Transportation 2020). Each one of these operations, in turn, can be classified further into more detailed functional elements. Commensurate with the concept of Level of Detail (LoD) in scheduling and in design (Stephenson 2007), the breakdown of a project into more granular functional elements can be achieved at several hierarchical layers. Each of these layers is scoped to address certain needs (e.g., planning, scheduling, resource levelling, task assignment, safety management, etc.) and certain target groups (e.g., managers, sub-contractors, planners, site superintendents, workers, and operators). Halpin et al. (1992) have presented such a taxonomy in the form of project, operation, process, and task. This hierarchy is illustrated in Figure 2-3.

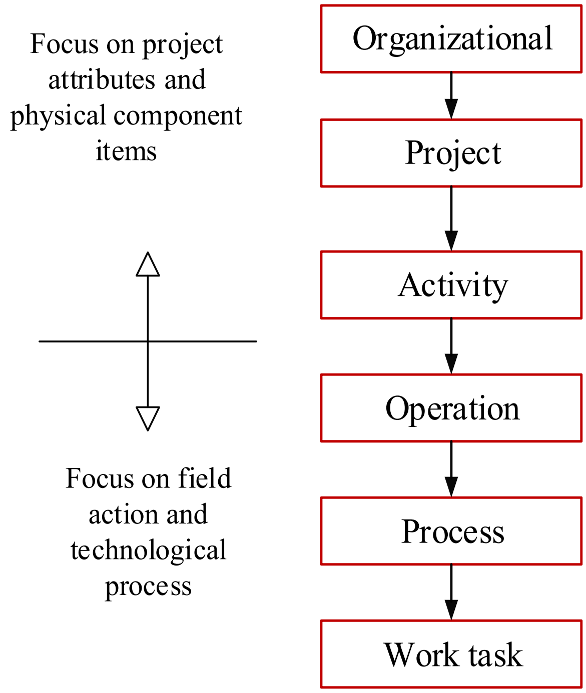


Figure 2-3 Hierarchy levels of construction management (Halpin et al. 1992)

Others have presented similar classifications (El-Gohary and El-Diraby 2010; Zhang et al. 2015). As such, there are different representations of the project taxonomy in the domain, which can cause miscommunication and ambiguity in the exchanged information between the stakeholders in the same project. Understanding the accurate definition and scope of each of these layers is very important for the consistent classification and organization of earthwork information. Therefore, representing a consistent and clear taxonomy for the earthwork project, including the operations, processes, tasks, and microtasks, is one of the objectives of this research. In the next section, the different operations in the earthwork domain are presented.

2.2.4 Types of Earthwork Operations

Earthwork contains different operations, including cleaning, excavating, compacting, and finishing or grading operations. These operations are executed in a specific order. Different textbooks (Gransberg et al. 2006; Nunnally 2004; Peurifoy et al. 2010) describe the relationships between earthwork operations and the equipment performing those operations. Knowing the details of earthwork operations gives a better understating of the properties of these operations and the pieces of equipment to perform them. As shown in Table 2-1, there is a variety of equipment used in various operations. Although the naming of the equipment may suggest a specific type of operations (e.g., loader is used for loading the soil to the truck), this equipment can be used to perform other operations (as will be explained in Section 2.2.2). The information and terms are collected from different resources (e.g., Delaware Department of Transportation, 2020; Gransberg et al., 2006; Peurifoy et al., 2010).

Table 2-1 Definition of different operations and the equipment to perform them

Operation	Definition	Main Equipment	Alternative Equipment
Cleaning and grubbing	Cleaning the site, Removing and disposal the Trees, stumps, rubbish, undergrowth, buildings, and any other materials or objects not needed.	Prep Equipment, Compact Loaders, Hoes, Dozers	Drilling Equipment
Excavation and Embankment Construction	Digging up and hauling earth *Cut*, forming the embankment *Fill,* or disturbing the compacted earth *Rip*.	Front Shovels, Hoes	Scrapers, Dozers
Hauling	Removing unsuitable martial.	Trucks	Scrapers, Dozers
Compaction	Compacting the materials to the required density to improve the properties.	Compactors	Dozers
Grading	Shaping the materials to the required grade.	Graders	Trimmers, Gradalls

(a) Cleaning and Grubbing

This operation comprises removing, grubbing, and disposing of all unwanted objects (e.g., trees, debris, and old building). This is the first operation to prepare the site for other operations.

(b) Excavation and Embankment Construction

Excavation and embankment construction are the operations that follow the cleaning and grubbing operation. The excavation consists of moving soil or loose rocks. The embankment construction consists of shaping the roadbed, slopes, channel, ditches, and road shoulders.

(c) Hauling

In earthwork operations, hauling can be represented as an operation performed mainly by trucks to remove or move the materials between places. Other pieces of equipment, such as scrapers and dozers, can be used to perform this operation for a short distance as explained in Section 2.2.2

(d) Compaction

Compaction is the operation performed to change the loose soil properties into a particular density specification to meet the requirements. For example, each layer of the embankment construction of the roads should be compacted to the required density, which affects the stability of roads.

(e) Grading

This operation follows the excavation and compaction operations. There are two types of grading operations: rough grading and finish grading. This operation shapes the soil and grades it into the required level in the design documents.

These different types of earthwork operations consist of processes, tasks, and microtasks, which are performed by many equipment and comprise different stakeholders. These operations create a very complex work environment that needs to be seamlessly coordinated. The drawback of the methods used for managing and coordinating the earthwork project is that they are mostly ad-hoc and did not contain the knowledge that can be shared and reused among the project stakeholders. Moreover, having deeper understating of the operation's complexity and properly modeling these operations with their hierarchy and properties play a critical role in safety and productivity improvement in the earthwork domain. Therefore, computer models (e.g., simulation models) are used to capture the complexity of these operations and create the virtual environment, that is logically similar to the real context of the earthwork operations before the actual implementation.

2.2.5 Simulation Models for Earthwork Operation

Earthwork operations are performed in an environment that contains a variety of variables, such as the type of materials to be excavated, the distance between loading and hauling areas, and the operator's experience. Moreover, these variables are affected by uncertainty factors, such as weather conditions and accidents, which consequentially affect the time and cost of the operation. Simulation is used to represent the real-world system by modelling the components and functions of this system and integrating them within the simulation engines. Simulation tools, such as

STROBOSCOPE, are used to model complex construction operations (Martínez 1998). Figure 2-4 illustrates a simulation model developed using Stroboscope for earthmoving operation, which is a combination of excavation and hauling (Vahdatikhaki 2015). The model represents an excavator and a truck, and the behaviors of these pieces of equipment (e.g., relocation, hauling, dumping, and loading). Furthermore, detailed *micro*-behaviors are represented in this model (e.g., the excavator swing to the truck). This model describes the operation pattern of performing the earthmoving, including the tasks performed by each piece of equipment. However, building the simulation model requires extensive training, which may not be available for the staff responsible for planning (Martínez 1998). Ontology can be used as a consistent foundation of the knowledge about the resources, and the operations in the earthwork domain, which can be used in simulation model development. On the other hand, good simulation models can be used to add more concepts and improve the understanding of the complex parts throughout the development of the ontology.

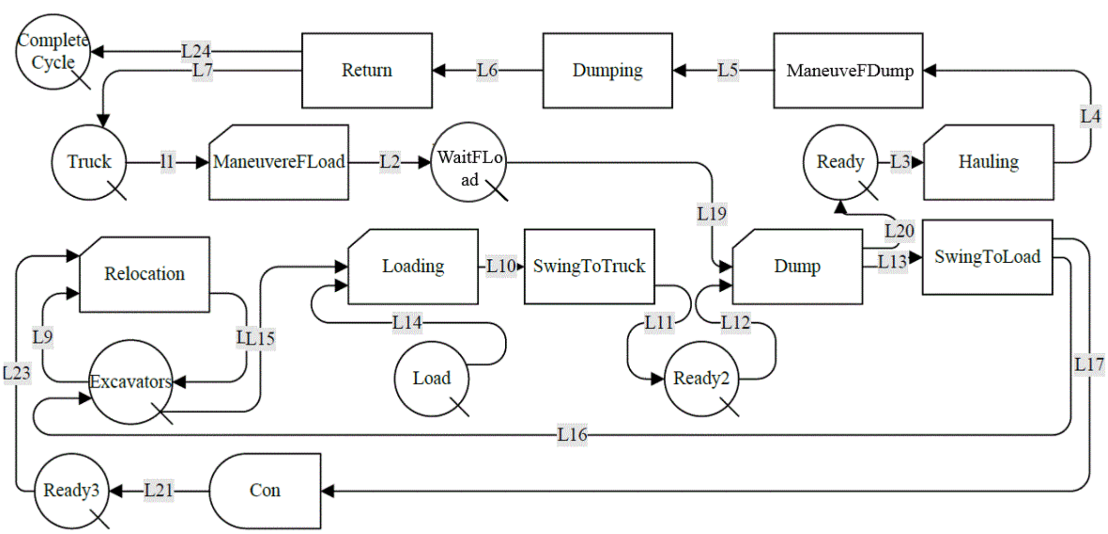


Figure 2-4 Simulation model for earthmoving operation (Vahdatikhaki 2015)

Although there are different techniques that can be used to build the simulation models, still there are other considerations related to the safety rules and regulations that need to be linked to the elements and the activities in these models. For example, the effects of the soil types on the operations' safety and productivity. There is a variety of soil classifications that are used by the stakeholders in the earthwork project. Thus, defining a unified classification of the soil in EW-Onto will provide a robust knowledgebase that can be used to construct the simulation models.

2.3 SOIL CLASSIFICATION

Soil and rocks are the materials that make up the shell of the earth and play an important role in construction (Nunnally 2004). Different soils with similar properties can be classified according to their behavior and properties in terms of simple indices (Atkinson 2000). In earthwork operations, it is necessary to know the classification of the soils to identify how to deal with them by choosing the suitable equipment and method to increase the level of productivity and quality and decrease the operation cost (Peurifoy et al. 2010). In road projects, where the soil is an essential material, the soil affects road stability, supports the structure, and distributes the forces on the road. Different factors influence the stability of the embankment construction. The negative effects of some factors could be limited by the design of soil structure based on the behavior of the soil.

There are basic characteristics of soil that used to classify the soil, such as the size range of grain, the shape of the grain. Several classification systems exist, such as Unified Soil Classification System (USCS) (Gadouri et al. 2018); classification of American Association of State Highway and Transportation Officials (AASHTO) (Pratt et al. 2000); classification of U.S. Department of Agriculture (USDA) (García-Gaines and Frankenstein 2015); and Massachusetts Institute of Technology (MIT) classification system (Kulhawy and Chen 2009).

The types of soil and rock play an important role in the selection of equipment in different earthwork operations. For example, in a compaction operation, knowing the type of materials is important for choosing the appropriate compaction method as shown in Table 2-2.

Table 2-2 Appropriate compaction methods based on soil type (Peurifoy et al. 2010)

Material	Impact	Pressure	Vibration	Kneading
Gravel	Poor	No	Good	Very Good
Sand	Poor	No	Excellent	Good
Silt	Good	Good	Poor	Excellent
Clay	Excellent with confinement	Very Good	No	Good

Choosing the right equipment and method will increase productivity, improving quality and decreasing operation costs. For example, and as shown in Table 2-2, using a compactor with vibration to compact the silt will lead to poor compaction results. Moreover, the type of soil in the workzones plays a critical role in selecting the required resources and how the processes and tasks will be performed according to the related safety regulations. Therefore, providing formal and consistent shareable presentations of all the components related to safety and productivity in the earthwork domain through the ontology and linked with the related regulations and rules will pave the way to improve safety and increase productivity.

2.4 DATA MODELS IN THE CONSTRUCTION INDUSTRY

In the construction industry, where several groups collaborate intensively and work on one project, it is vital to have compatible tools and models (Laakso and Kiviniemi 2012). Using different software applications and platforms requires standards to enable collaboration and communications between various stakeholders in the same project.

BIM is used to capture, store, analyze, and visualize building lifecycle information in a systematic and structured way and is increasingly implemented in the construction industry (Liu et al. 2016). BuildingSMART developed a data standard for BIM called Industry Foundation Classes (IFC) (Laakso and Kiviniemi 2012). IFC includes the physical elements as well as the processes and activities (Behrman 2002), and it is used to improve the quality throughout the lifecycle of the building design, construction, and maintenance (Isikdag et al. 2007). OmniClass is a construction classification system developed by the International Organization for Standardization (ISO) and the International Construction Information Society (ICIS) for construction information (OmniClass 2020). Another classification system in the construction industry is UNIfied CLASSification (Uniclass), which provides the classification of the asset through a number of tables, such as table *Pr*, which provides the classification of products, and table *Ss*, which provides the classification of asset systems (Heaton et al. 2019; NBS 2020).

While the data modelling and standardization for the building industry is relatively well-established, other civil infrastructures do not enjoy the same level of maturity in data modelling and standardization. An example of data modelling that can be used beyond the building scope is LandXML. This is a data exchange format based on Extensible Markup Language (XML) that contains such data as the terrain, maps, pipelines, roads, railways, and other infrastructure objects (Rebolj et al. 2008). Furthermore, buildingSMART's ongoing projects on infrastructure modelling are committed to enhancing the data communication, collaboration, and management of infrastructure by extending the concepts of BIM to infrastructure projects (BuildingSMART 2020). Figure 2-5 illustrates the components included in the scope of these projects. The upper-level models are rail, road, bridge, and tunnel. The earthwork model is the interface between the upper models and the alignment and terrain models.

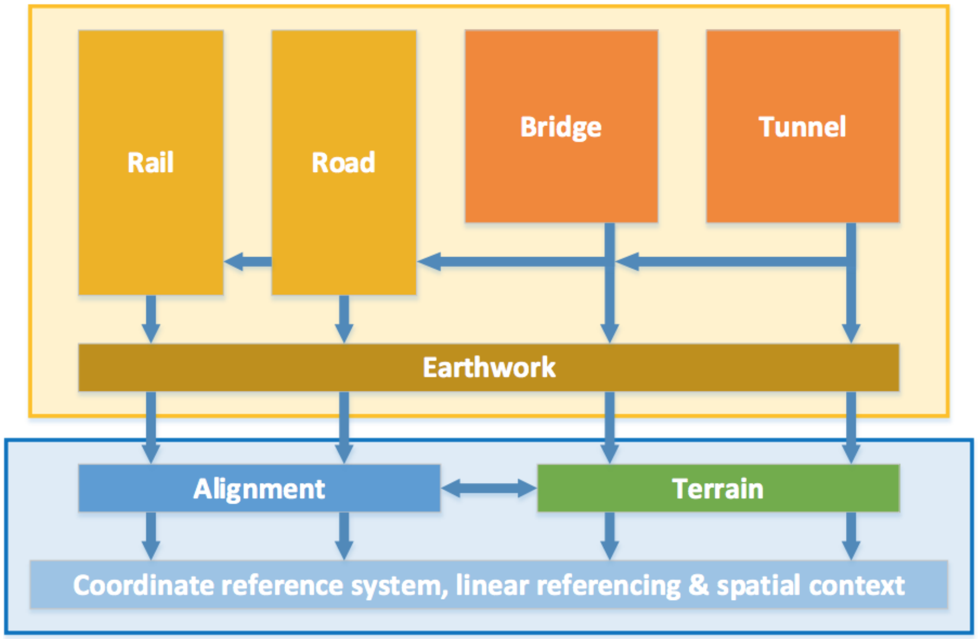


Figure 2-5 BuildingSMART for infrastructure models (BuildingSMART 2020)

As an example of these projects, IFC4.2 can be mentioned, which has incorporated IFC-Bridge extension (Borrmann et al., 2019). Civil Information Modeling (CIM) refers to the application of BIM in civil infrastructure facilities, especially horizontal projects (e.g., bridges and tunnels) (Cheng et al., 2016). Nevertheless, although data models such as LandXML can be of use in the earthwork domain, a comprehensive model for data exchange in this domain is evidently missing.

The formalization efforts in the construction industry allow the stakeholders to exchange valuable information, not only during the construction project but also after the project completion and through the operation stage of the project (i.e., during facilities management). Ontologies can be integrated with different software applications which represent data differently to provide data interoperability (e.g., BIM and Geographic Information Systems (GIS)) (Le et al. 2019). Quinn et al. (2020), proposed an approach to integrate the data from the sensors with BIM for facility management using the ontology and the linked data architecture. This approach increases the data queries flexibility and provides a more complex analysis of the data. Therefore, the earthwork ontology will play an important role to provide knowledge not only to the earthwork domain but also to the other models, as illustrated in Figure 2-5. To build a reliable ontology for the earthwork domain, this ontology should have the ability and flexibility to integrate with the technologies that have been applied in the earthwork domain.

2.5 DATA COLLECTION TECHNOLOGIES

There are a variety of technologies that are used in construction to collect data. These technologies vary in the form of data delivered and the environment where they can be used. In the next paragraphs, data collection technologies are briefly explained.

Using Real-Time Location System (RTLS) technologies in the construction industry has attracted interest in the past decade (Li et al. 2016). RTLSs are used in construction sites to track and determine the coordinates of objects indoor and outdoor. In recent years, different RTLSs have been developed with different levels of quality, cost, and limitations. RTLS data is not just for real-time uses; it can be also used for post-processing analysis. RTLS hardware consists of tags, which communicate with the receivers and use different algorithms to calculate the locations, such as the Time of Arrival (TOA) and Received Signal Strength Indicator (RSSI) (Li et al. 2016). RTLS has different technologies, which are applied in different environments, such as office buildings, hospitals, and roads for safety and security purposes.

Ultra-wideband (UWB) is a Radio Frequency (RF) technology, which is used in indoor and outdoor environments. The usage of UWB has been investigated by different researchers to verify the accuracy when it is used in different environments (Cho et al. 2010; Maalek and Sadeghpour 2013; Siddiqui 2014).

Vision-based positioning systems are used in indoor and outdoor environments with up to 88% of accuracy (Li et al. 2016; Zhang et al. 2020). In construction, vision-based positioning systems are used to track workers and equipment (e.g., wheel loaders, dozers, and tower cranes). Moreover, vision-based systems are used to identify the dangerous behavior of workers (Han and Lee, 2013; Park et al., 2011).

GPS is used to estimate the location in outdoor environments and cannot be used indoor because it needs Line-of-Sight (LOS) from the satellites. GPS is used in construction sites mostly to track and register the equipment and materials locations continuously (Hildreth et al. 2005). The accuracy of using GPS is investigated in different researches, such as the study of (Lu et al., 2007).

Radio Frequency Identification (RFID) is another RTLS technology, which solves the issues of GPS in indoor environments (Chon et al. 2004). RFID is used to track equipment, workers, and materials (Wu et al., 2010). Moreover, RFID is integrated with other technologies, such as GPS to cover large areas (Razavi and Haas 2010), laser scanning in the indoor environment (Valero et al. 2015), and total stations for accurately positioning objects (Sakamoto et al. 2012).

Existing Wireless Local Area Network (WLAN) can be used as positioning systems based on the strength of the signals. Woo et al. (2011) investigated the feasibility of using WIFI-based WLAN positioning systems in the indoor environment (i.e., shield tunnel) to locate the workers. Their experiments show that the accuracy of the system is within 5 m.

The ultrasound positioning system is another positioning system that is used for tracking objects using Ultrasound Signals (US). In construction, Skibniewski and Jang (2009) proposed a framework to combine the US and RF to increase the accuracy of using RF only. They investigated and compared the accuracy using simulation results and found that the accuracy is less than 0.2 m in the LOS environment because the US cannot penetrate objects without enough signal strength.

Infrared (IR) is a technology that is used in LOS environments. IR is initially used in construction to track the resources (e.g., equipment, objects, and workers) using the 3D range camera (Teizer et al. 2007) and capture the 3D images for objects such as wallboards, pipes and humans using a high-frame-rate sensor camera (Chi et al. 2009).

Bluetooth technology has wide uses in construction. Bluetooth is a technology that can be utilized indoor and underground to tackle the absence of GPS and RFID. Bluetooth Low Energy (BLE) can track and monitor workers and assets with low power and cost. The other advantage of using Bluetooth technology is that most of the workers have smartphones that have already build-in Bluetooth sensors. Park et al. (2015) evaluated the performance of Bluetooth technology in a construction site for tracking equipment and workers to prevent collisions in work zones.

Siddiqui (2014) proposed a Multi-Sensor Data Fusion (MSDF) approach to overcome the limitation of UWB RTLS. The framework is intended to cope with the challenges of the dynamic environment at construction sites by combining two sensory data sources, which are UWB RTLS and video. Table 2-3 lists the main properties and limitations of using UWB and image processing in construction projects.

Data collection technologies are the backbone of the future smart construction, and each technology has a specific operational performance, which is affected by the environment and weather conditions (Apanaviciene et al. 2020; Edirisinghe 2019). Moreover, the limitations of these technologies (e.g., the limited battery life of sensors, the sensitivity and accuracy of sensors, the limited field of view of cameras, the resolution of images and videos) are further affected by the nature of construction projects, which in turn affect the safety, productivity, and quality of construction projects as further explained in the next section.

Table 2-3 Comparison of UWB & image processing technologies for construction projects (Siddiqui 2014)

Required Features	**UWB**	**Image Processing**
Localization	3D	Mostly 2D
Identification of specific equipment	Yes	No
Real-time processing	Yes	No
Update rate	Limited	High
Missing data	High	Low
Coordinate system	Global	Pixels
Multipath and radio noise effect	Yes	No
Weather and light conditions effect	No	Yes
Line of sight and occlusion issues	Provides a location with error	Provides a location with more training
Training required	No	Yes
Cost of deployment	High	Low
Configuration at site	Difficult	Easy
Tagging issues (e.g., battery replacement)	Yes	No

2.5.1 Safety Monitoring and Control in Construction

A major factor for success in the construction industry is developing and ensuring safety procedures to identify potential hazards before they occur. Job Hazard Analysis (JHA) is used to define the relationships between jobs, tools, workers, and the surrounding environment, which can result in hazards, and to provide a list of procedures and resources for preventing or mitigating these hazards (OSHA 2020a). Occupational Risk Assessment (ORA) (Pinto et al., 2011) is a process that is performed on construction sites to gather information from different sources of hazards (Lu et al. 2015). The check-list technique (Mattila et al. 1994) is used in ORA to define the safety issues at the early stages of the work. Zhang et al. (2013) outlined a framework for early hazard identification by integrating a 4D BIM and safety regulations to identify the hazards and apply and visualize prevention procedures automatically (Kiviniemi et al. 2011). Heterogeneous data resources at the construction site provide sensory data with different levels of accuracy and efficiency. Zhang et al. (2017) compared sensor technologies from three main viewpoints: the complexity of the applied algorithm, the complexity of the layout, and the limitations to apply them on construction sites. As illustrated in Table 2-4, they targeted three leading technologies: location sensor-based technology (e.g., GPS, UWB, and RFID), vision-based sensing technology, and wireless sensor network technology. RTLSs have been combined with other technologies such as MAS to enhance the coordination and safety issues related to earthwork equipment (Vahdatikhaki et al. 2017).

Table 2-4 Comparison of the three sensor-based technologies' adaptability (Zhang et al. 2017)

Sensor-Based Technology		**Algorithm Complexity**	**Installation Complexity**	**Construction Environment Limitation**
Locating sensor-based technology	GPS	Low	Low	Suitable for outdoor environment
	UWB	Low	Moderate	Accuracy affected by the arrangement of signal transmitters and receivers. Signals blocked or interfered by obstacles. Signals interfered by metal objects.
	Zigbee	Low	Moderate	
	RDIF	Low	Moderate	
	WLAN	Low	Moderate	Signals blocked or interfered by obstacles.
	Ultrasond	Low	Moderate	Signals blocked or interfered by obstacles. Signals interfered by metal objects.
Vision-based sensing technology		High	Moderate	Vulnerable to the impact of surrounding environment, such as lighting condition and background color.
Wireless sensor network		Moderate	High	Signals blocked or interfered by obstacles or other electronic signals in network communication. Difficult to solve the energy supply problems.

As such, it is important to select the technologies that are compatible with the characteristics of the construction site (e.g., size of the site and the available locations to install the sensors), which in turn are coupled with the different properties, limitations, advantages, and disadvantages of each technology. Therefore, the main key to implementing and managing these technologies consistently and unambiguously is to formalize their concepts and the relationships and integrate them with earthwork domain ontology. The next sections discuss the new technologies that are used in the earthwork domain. These technologies are used for different proposes, such as enhancing safety and improving productivity in the earthwork domain.

2.6 NEW EARTHWORK-SUPPORT TECHNOLOGIES

Given the complexity of earthwork operations and the amount of data that need to be processed for efficient planning and execution of the operations, a variety of systems and technologies are developed for earthwork operations in recent years (e.g., Caterpillar 2020; Vahdatikhaki et al. 2017; Kim et al. 2012). During the execution of projects, several new technologies can enhance earthwork operations. Most notably, AMG/C is a technology that integrates 3D design models with the real-time sensor data (e.g., GPS) to provide different levels of assistance to the operators of the earthwork equipment (Kaufmann and Anderegg 2008). Vahdatikhaki et al. (2017) proposed the application of a Multi-Agent System (MAS) as a means to support larger fleet-level coordination for the earthwork operations. Intelligent compaction is another technology used to support and improve the efficiency of compaction jobs (Anderegg et al. 2006). Although these technologies are able to enhance the earthwork operations at different levels, there is very little interoperability and data exchange between these systems, resulting in a great degree of redundancy in their application (Vahdatikhaki et al. 2017). Thus, it is essential to create a consistent taxonomy of the autonomy levels and link them to earthwork equipment classification. Earthwork equipment autonomy classification could benefit from more mature autonomy classifications in other domains, such as the self-driving car in the car industry. While there are different standards for characterizing the Level of Autonomy (LoA) the self-driving car, the Society of Automotive Engineering (SAE) (SAE 2018) metric is the most widely recognized one. This metric as shown in Figure 2-6 employs a scale of 0 (fully non-autonomous) to 5 (fully autonomous) (Melenbrink et al. 2020; Sifakis 2019). Semantics modeling of earthwork information in the form of an ontology would pave the way for the adoption of new technologies in the domain such as autonomy classification.

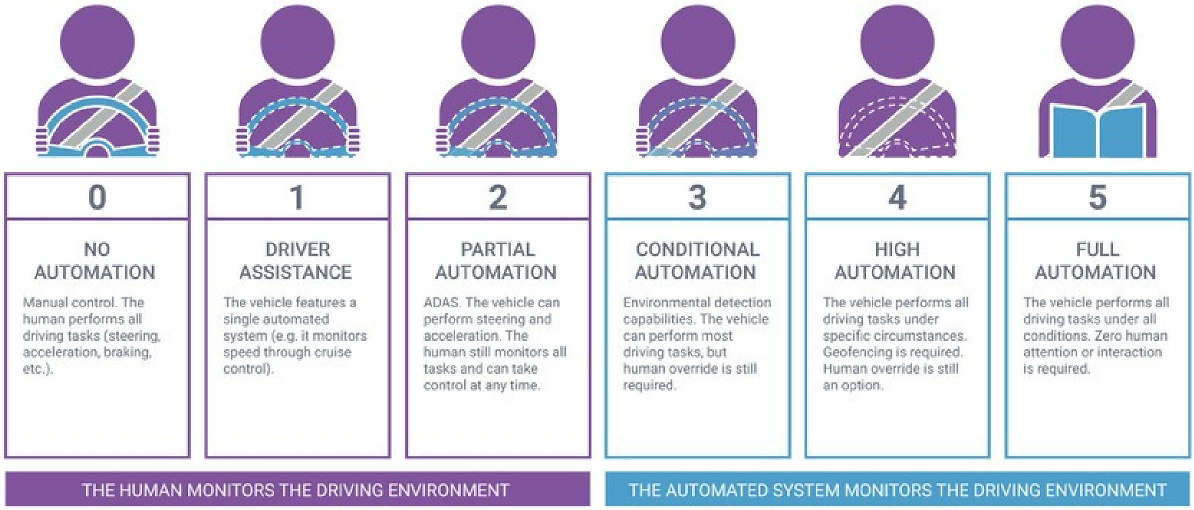


Figure 2-6 Levels of autonomy, adapted from (Synopsys 2020)

Ontology has the ability to link the concepts related to the autonomy and the classification in earthwork domain. Developing the ontologies is the core of semantic systems, which provides the understandable semantics not only for the human but also between the machines. Moreover, the new technologies such as MAS is associated with the ontologies to move from the centralized approaches of management to distributed and flexible solutions (Skobelev et al. 2020).

2.6.1 MAS for Earthwork Operations

An intelligent agent is an agent capable of perceiving its environment and making decisions about how to react to the received information (Russell et al. 2010). The term *Precepts* is used to describe the inputs of an agent, and the output of the agent is called *Actions*. A MAS supports communication in a distributed environment. Zhang and Hammad (2011) discussed MAS approaches for path-planning problems of construction equipment to avoid collisions and create new paths for the cranes, as well as the negotiation between the agents to accomplish their goals. In our previous work, and as shown in Figure 2-7, an updated version of the MAS is proposed to facilitate the earthwork operations (Vahdatikhaki et al. 2017). According to the level of responsibility, the agents are grouped into four categories: *Operator Agents* (OA): These agents support the operators of each equipment to achieve their tasks. Since the machines are equipped with GPS and other types of sensors, the agents can use this accurate sensory data to determine the precise location and the state of the equipment. *Coordinator Agents:* There are two types of coordinator agents, *Team Coordinator Agent* (TCA) and *General Coordinator agent* (GCA). The

TCA is responsible for coordinating and supporting the teams and sub-teams, depending on the characteristics of the project. *Information Agents:* These agents are responsible for providing and updating the information to the operational and coordination levels. The *Site State Agent* (SSA) delivers the *Digital Terrain Model* (DTM) of the site. The *Design Document Agent* (DDA) provides and updates the 3D model depending on the changes that are made in the project. *Project Document Agent* (PDA) holds the documents about the project, such as the schedule, the resources, safety regulations, and construction methods. However, this MAS development approach for earthwork is not based on a specific ontology; and therefore, it may suffer from semantic inconsistencies. Examples of these inconsistencies are the non-unified representations of hierarchies of equipment and the relationships between *project, operations*, *processes,* and *tasks*.

Ontologies are used with MAS for a specific domain in construction to overcome the issues of inconsistency, interoperability and ununified representation of the knowledge. On the other hand, MAS uses this knowledge to configure its agents.

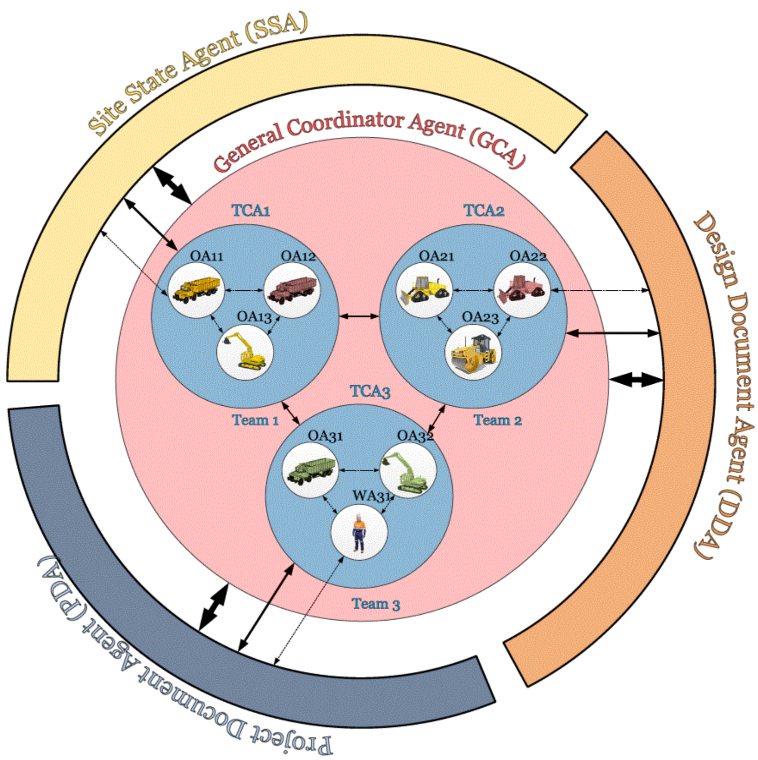


Figure 2-7 Multi-agent System architecture (Vahdatikhaki et al. 2017)

2.6.2 Ontology-Based MAS in Construction

Several research works tried to develop MAS based on an ontology of a specific domain related to construction. Zeb and Froese (2012) developed a transaction ontology in the infrastructure domain. The aim of the transaction ontology is to formalize the communication processes and define the message templates between the municipal and provincial governments. Skobelev et al. (2020) proposed a method using ontology and MAS for resource planning management, where the ontology is used to formalize the concepts and the relationships that are related to resources and planning management. Dibley et al. (2012) combined the ontology with a MAS and proposed an ontology framework for sensor-based building monitoring. The ontology that is used in their research combines three sub-ontologies, which are: building ontology, sensors ontology, and the general-purpose ontology to support the real-time building monitoring.

These works show the advantages of using the ontologies and coupling them with MAS, where the ontologies are used to provide the formalized description of the concepts and the relationships of the domain of interest. Therefore, in this research, a MAS will be used to demonstrate the capability of integrated earthwork ontology to provide the required knowledge to the MAS in the earthwork domain. The next section provides a review of using the ontology with other information models in domains related to construction.

2.7 ONTOLOGICAL MODELING IN CONSTRUCTION

In the construction industry, ontologies have been developed to improve workflow and to share knowledge about the various stakeholders' process planning. The collaboration based on knowledge sharing and integration in construction is not limited to the construction enterprise but can be extended to the integration at the level of the construction supply chain (El-Gohary and El-Diraby 2010). Ontologies are one of the advanced technologies that have been used in construction to facilitate not only human-to-human but also machine-to-machine communications by formalizing the information exchange scheme (Taher et al. 2017). Previous studies have used ontologies and combined them with different modelling techniques (e.g., BIM) (Lee et al. 2014; Zhang et al. 2013). Dhakal et al. (2020) proposed ontology-based semantic modelling to support knowledgebase document classification related to disaster-resilient construction practices. Lee et al. (2014) proposed an ontological approach for quantity take-off using BIM as a data source. The developed ontology can be used to infer suitable items based on the estimated cost. Zhong et al. (2012) developed an approach to integrate construction processes with regulations related to quality compliance. Zhang et al. (2015) established an approach for the storage and application of safety management knowledge in construction. This ontology is formulated to include a product model, a process model, and a safety model. Ding et al. (2016) proposed an approach for combining BIM with an ontology that organizes construction risk knowledge semantically. Zhang et al. (2015) presented a framework using an ontology to formalize the company's JHA of activities. Wang and Boukamp (2011) created a framework to improve access to the company's JHA. The framework uses the ontology to organize knowledge about activities, jobs' steps, and the related hazards. El-Gohary and El-Diraby (2010) presented an ontology for infrastructure and construction processes. El-Diraby and Kashif (2005) presented a distributed ontology architecture for Knowledge Management (KM) in highway construction. The architecture was developed as an extension of the e-COGNOS ontology. Viljamaa and Peltomaa (2014) developed a method to intensify construction process control and to enhance process management and the accessibility of information for subcontractors. There are general key concepts that were previously presented in research on the AEC ontologies, such as Domain Ontology for Construction Knowledge (El-Diraby 2012), transaction ontology in the domain of infrastructure management (Zeb and Froese 2012), and domain ontology for processes in infrastructure and construction (El-Gohary and El-Diraby 2010).

Most of the abovementioned ontologies were developed from scratch, and there is a need for more research in the area of ontology integration, especially for construction safety applications. Moreover, these ontologies were developed for different branches in construction and road projects. However, these ontologies did not cover the earthwork domain.

2.8 PRINCIPLES OF ONTOLOGY DEVELOPMENT

One of the main motives to use the ontology is to represent the knowledge in a domain in a way that can be processed by machines (Gómez-Pérez and Benjamins 1999). The ontology, in simple words, is a set of relations between the concepts as shown in Equation (2.1) (Thomopoulos et al. 2013).

$$\boldsymbol{\Omega} = \{\boldsymbol{\mathcal{C}}\ ,\ \boldsymbol{\mathcal{R}}\} \tag{2.1}$$

where Ω is the ontology, $\mathcal{C}$ is the concepts of this ontology, and $\mathcal{R}$ is the relationships between these concepts.

One of the most important points that ontology developers need to consider is the integration at the enterprise level, where different sub-domain ontologies need unified linguistics, high level of generality, and rigorous vocabulary (Guarino 1998). Moreover, ontologies are the foundation of content-based information access, which provides semantic interoperability over the web (Flotyński 2020; Marković and Gostojić 2020).

Axioms are the "statements that are true in the domain" (Boris et al. 2020). Class expression axioms establish the relationships between the classes including SubClassOf, EquivalentClasses, DisjointClasses, and DisjointUnion. These axioms organize the relationships amongst the concepts and are used by the reasoners to check the consistency of the ontology.

It is important here to mention the difference between ontology and database from the knowledge representation perspective. Both of these data models have some analogous features. However, there are differences between the ontologies and the database schema (Benevolenskiy et al. 2012): (1) The main difference between the database schema and the ontology is the purposes of each of them. Databases are used to structure data in a way that makes it efficient to be retrieved through the queries. Whereas ontologies are focusing not only on the data but also on their semantics; (2) The ontological representation can be built without the instances. Whereas in the database schema, the instances are essential; and (3) Ontologies provide the taxonomy and the class hierarchy, while databases have tables structure.

Closed World Assumption (CWA) and Open World Assumption (OWA) are essential in the logic of knowledge representation. CWA on the database expresses that if a fact is not known (to be true), it must be false. CWA is commonly used in database applications, where the system is assumed to be complete. In OWA, if a fact is not known (to be true), it will be just unknown. Therefore, the missing information is not considered to be false. OWA is useful when the information is integrated from different resources (Bergman 2009). Ontology represents the knowledge from different resources and builds the formalization based on OWA. This knowledge representation links the components and makes it explicitly available through an ontology language (e.g., OWL). As mentioned above, concepts and relationships are the main components of ontology. Hence, in the next paragraphs, these components are explained in detail.

2.8.1 Components of Ontology

As mentioned above, concepts are one of the main components of the ontology and should be presented unambiguously. The main concepts in an ontology can be represented through the following component types: (1) Entities: Entities cover the different abstract concepts in an ontology, such as process, task, actor, and product; (2) Attributes: Each individual of an entity has some attributes (i.e., data properties) that make it different from the others. For example, the equipment attributes, such as weight, equipment capacity, type, and brand; (3) Relationships: The relationships among the ontology concepts should be defined. El-Gohary and El-Diraby (2010) defined the major types of relationships: subsumption relationship and partonomy relationship. The subsumption relationship reflects *is-a* relationship between the concepts. Partonomy relationship is a combination of the *part-of* relationship between the concept and its sub-concepts, which are built as partonomic hierarchies. In addition, object properties can be used to create other links between the concepts (e.g., Hoe *type-of* Excavator); (4) Modalities: Describe the entity from different points of view at a particular time (e.g., situations); (5) Strategies: Strategies refer to the mechanisms that are used to accomplish the operations, processes, and tasks in the project; (6) Rules and Regulations: Describe the related safety, productivity, and quality rules and regulations in the domain. Developing an ontology and linking these components is different from one domain to another. Regardless of the domain and the components included in the developed ontology, there are various methodologies and languages that could be used to build and describe the development steps. These languages and methodologies are explained in the next section.

2.8.2 Ontology Languages and Development Methodologies

Ontologies aim to represent the implicit knowledge in a domain in an explicit way by establishing an organized structure of related concepts and relationships. Different languages can be used to represent the ontology. Description Logics (DL) is a language to formalize the knowledge representation that provides a high-level description of the world to be used in intelligent systems (Baader et al. 2003). DL delivers syntax to describe the knowledge using expressions built as atomic concepts, atomic roles, and role constructors. DL has three formalism components: Terminological Component (TBox), Assertion Component (ABox), and Role Component (RBox). TBox axioms describe the general properties of concepts and contain the essential knowledge in the form of taxonomy or terminology such as concept inclusion. ABox axioms contain the assertional knowledge for specific individuals in the domain, whereas RBox refers to roles' properties, such as role equivalence axioms and role inclusion (Krötzsch et al. 2012). Semantic Web Rule Language (SWRL) is used to express rules as sequences of axioms and facts. The rules can be saved as a part of the developed ontology. SWRL can work with reasoner systems such as Pellet (Sirin et al. 2007) and HermiT (Glimm et al. 2014) to infer the implicit knowledge included in the ontology (Bassiliades 2020).

The Foundation Ontology (FO), also known as the top-level or upper ontology, is an ontology that describes the most general terms across different domains. To develop an ontology, the developer has the option of using one of the available methodologies. However, out of 151 research papers reviewed by Zhou et al. (2016), most of the studies use their own methodologies to develop the ontology. Zhou et al. (2016) explained that certain studies might include some steps from the previous methodologies for developing their ontologies. The ontology development methodology depends to a great extent on the specific domain, the level of detail, and the starting point of the development (i.e., an extension of an existing ontology or development of a new ontology). In the 1990's, a number of methodologies for developing ontologies were developed (Corcho et al. 2003). Cyc Project (Lenat et al. 1990), TOronto Virtual Enterprise (TOVE) (Grüninger and Fox 1995), KACTUS Project (Schreiber et al. 1995), Skeletal methodology (Uschold and Gruninger 1996), and METHONTOLOGY (Fernández-López et al. 1997) are general methodologies used to build ontologies. IDEF5 is an ontology capture method and one of the Integrated DEFinition (IDEF) family languages that support the analysis and design of models (Noran 2004). Table 2-5 shows some ontology development methodologies and the steps of each methodology. Other methodologies or approaches are also used to build ontologies by re-using existing ontologies or integrating two or more ontologies. Examples of these methodologies are Ontolingua (Farquhar et al. 1997) and SENSUS (Swartout et al. 1996). Reusing existing ontologies to build the new ontology provides the conjunction between the concepts and the relationships from these ontologies instead of constructing the whole ontology from scratch each time (Leung et al. 2014).

Table 2-5 Examples of ontology development methodologies

	METHODOLOGIES			
	TOVE	METHONTOLOGY	SKELETAL	IDEF5
Main Steps	- Motivating scenarios - Informal competency questions - Terminology Specification - Formal competency questions - Axiom specification - Completeness theorems	- Specification - Knowledge acquisition - Conceptualization - Formalization - Integration - Implementation - Maintenance - Evaluation - Documentation	- Identifying purpose and scope - Building ontology: - Ontology capture - Coding - Integrating existing ontologies - Evaluating - Documentation - Initial guidelines for each step	- Organizing and scoping - Data collection - Data analysis - Initial ontology Development - Ontology refinement and validation

2.8.3 Reusing Ontologies to Build New Domain Ontology

One of the main purposes of building ontologies is to extend and reuse them for knowledge integration in multiple related domains because building a robust knowledge representation that covers these domains needs to combine heterogeneous information. On the other hand, developing different ontologies for the same domain leads to overlapping efforts and potential misunderstanding of the concepts represented in these ontologies (Choi et al. 2006). Thus, mapping ontologies facilitates reusing them for a specific domain, instead of creating them from scratch, and makes the integrated ontology more inclusive and comprehensive with respect to the concepts and the relationships in the domain. There are three methods to map ontologies:

(1) Ontologies merging: is the process of combining two or more ontologies presenting the information in similar or overlapping domains to create another ontology in the same domain with minor changes (Pinto et al. 1999).

(2) Ontologies alignment: is the process of creating links between two ontologies, which usually have related and complementary domains (Choi et al. 2006; Noy et al. 2008).

(3) Ontologies integration: The integration process combines ontologies that are built for different domains to reuse some of their components (Pinto et al. 1999). Thus, this method saves the effort to redevelop these components that are needed in the integrated ontology. The integration process has two main steps: performing the integration process and adding more knowledge to the integrated ontology (Pinto and Martins, 2001).

The resulting ontology contains the concepts, relationships, axioms, and rules, form the reused ontologies. This ontology should be evaluated to get the reliability and make it available to other ontology developers. The next section explains the ontology evaluation approaches.

2.8.4 Ontology Evaluation Approaches

Ontology evaluation is one of the essential steps in ontology development (Haghighi et al. 2013). The selection of the suitable evaluation approach depends on the purpose of the evaluation and the aspects of the developed ontology. Different ontology components are selected for the evaluation, such as vocabulary, taxonomy, and semantic and syntactic relationships. There are different approaches and criteria to evaluate each of these components. Experts can evaluate most of these components, whereas the data-driven approach can evaluate vocabulary, taxonomy, and semantic relationships (Brank et al. 2005). The next paragraphs provide a review of four different approaches for ontology evaluation:

(1) The gold standard: This approach aims to compare the developed ontology with a high-level "golden" standard or another ontology considered a benchmark in the domain by measuring the similarity between them. Velardi (2006) proposed a method to evaluate an ontology by comparing the extracted text (e.g., terms) with WordNet entities, which is considered as a lexical ontology that includes broad coverage of cognitive synonyms (*synsets*) (Singh and Sharan 2014).

(2) Data-driven evaluation: Data-driven evaluation is a quantitative method used where the developed ontology is compared with the source of knowledge, such as a corpus (Brewster et al. 2004). An automated extracting process is applied to the corpus (i.e., WordNet) to extract the terms; then, the overlapping terms between the corpus and the developed ontology are counted. If the terms used in the developed ontology do not appear in the corpus or vice versa, the ontology is penalized (Brewster et al. 2004). Haghighi et al. (2013) claimed that this method is more suitable for measuring an ontology coverage. Brewster et al. (2004) suggested using the data-driven method to evaluate which level the ontology fits with the corpus. In their method, each class in the ontology representing a term is compared with WordNet's *synsets*. The number of terms used in the ontology and appearing in WordNet reflects the level of ontology's richness, comprehensiveness, interpretability and clarity (Brewster et al. 2004). The WU and Palmer (WUP) index (Wu and Palmer, 1994) is a taxonomy-based similarity measure which represents the depths of the *synsets* in WordNet taxonomies along with the depth of Least Common Subsumer (LCS). LCS is the last common node in the path of two words. The index is given by:

$$WUP(C_1, C_2) = \frac{2*depth(LCS)}{(depth(s1)+depth(s2))} \quad (2.2)$$

where C_1 and C_2 are the concepts with the two corresponding *synsets* ($s1$ and $s2$) in the WordNet taxonomies.

Leacock and Chodorow (LCH) index (Leacock and Chodorow 1998) is an enhanced taxonomy-based similarity measure based on the shortest path between two concepts using node counting. The calculation is given by:

$$LCH(C_1, C_2) = -\log\left(\frac{\text{len}}{2*D}\right) \quad (2.3)$$

where len is the shortest path between the two *synsets*, and D is the deepest level in the WordNet taxonomy. The drawback of LCH is that there is no maximum value of range in the formula.

(3) Application-based evaluation: Application-based evaluation is the evaluation of a developed ontology using an application. This approach judges whether the ontology is suitable to perform the task and measure the ontology's performance. This approach is useful for measuring the capabilities of a developed ontology to meet the objectives, and it is not used to evaluate the design or the contents of the ontology (Haghighi et al. 2013).

(4) Criteria-based evaluation: Yu et al. (2007) proposed a qualitative method for evaluating ontologies using a list of criteria, including completeness, consistency, conciseness, expandability, and sensitivity. Except for the consistency criteria, which can be performed successfully by the ontology tools, this method is performed manually. criteria-based evaluation is more suitable for evaluating ontologies in early stage of development (Xing et al., 2019).

It should be noted that more than one approach can be used to evaluate an ontology. The selection of the approach depends on the nature of the developed ontology and whether a qualitative or quantitative evaluation is needed.

Semantic modelling of earthwork information in the form of an ontology would pave the way for the adoption of the linked data approach towards earthwork-support technologies (Curry et al., 2013; Lee et al., 2016).

2.9 LINKED DATA AND ONTOLOGY

The concept of linked data is derived from the idea of using the semantic web to connect the data and transfer the web into a universal knowledgebase, where the data from different datasets are linked (Lee et al., 2016). Berners-Lee (2009) identified the basic principles for developing the Linked data: (i) using Uniform Resource Identifier (URI) for each entity to be represented; (ii) each entity is provided with the Hypertext Transfer Protocol (HTTP) URI; (iii) using the web standards, such as Resource Description Framework (RDF) to describe the data and SPARQL Protocol and RDF Query Language (SPARQL); and (iv) including links to other resources URLs already available in the web.

Semantic modelling and linked data are the approaches that can be used to create links to share the information between the different stakeholders (Lee et al. 2016; Curry et al. 2013). It is evident that these approaches are successfully used in other disciplines. However, such approaches have never been applied to the earthwork domain. The next sections explain the linked data elements where the ontologies are the main components to create the unified representation of the knowledge from different resources.

2.9.1 Resource Description Framework

Resource Description Framework (RDF) is a standard model for interchanging the data and providing a description model using the triple form that contains three elements: subject, predicate, and object (Ian et al. 2004). In RDF, the user can define his terminologies in schema language called RDF schema. In Resource Description Framework Schema (RDFS), the user can define the vocabularies, the relationships, the range, and the domain of these relationships (Martinez-Rodriguez et al. 2020). A resource in RDF has a URI. URI gives a unique name schema to each part of RDF. Each subject (e.g., books, authors, places) has its own URI. Moreover, the predicates have their own unique URI that links the subjects with objects in the statements. SPARQL queries are used to retrieve meaningful information from RDF files. The values given to the subject through the predicates are either other resources or literals (e.g., string or integer). RDF syntax contains *rdf: RDF element*, which includes different descriptions:

(a) The *rdf: resource attribute*: *rdf: resource attribute* is used to link the different resources in RDF. For example, if there is a piece of equipment that works in an excavation operation (e.g., hoe), and at the same time, there is another equipment of the same type working in the same operation, there is a need to use a formal specification of the fact that these two equipment are not the same.

(b) The *rdf: type*: To introduce the resources in RDF, *rdf: type* property is used to state that this resource is a type of another resource. For example, hoe *rdf: type* excavator, *rdf: type* earthwork equipment and earthwork equipment *rdf: type* resource.

(c) Nested descriptions: The description of one resource can be used in another resource description. RDFS define the vocabularies and the relationships in the RDF document. In RDFS, RDF elements are also used. RDFS can express its ingredients, such as SubClassOf, and SubPropertyOf (Martinez-Rodriguez et al. 2020).

2.9.2 Query Language for RDF

SPARQL is a query language used to retrieve and manipulate the data stored in RDF format (Ali and Qayyum 2019). SPARQL is W3C candidate recommendation for querying and provides the abilities to make the queries over the RDF graph (triple) and return the subjects, predicates, or the objects in the statements. The quires in SPARQL match the RDF graphs and return the related results. The query returns the classes in the RDF and stored in the variables. The prefixes are used to replace the long URI in the quire's statements. SPARQL can create queries from different URIs and gives one final result. This feature provides the ability to gather information from different resources using URIs.

2.9.3 Linked Data in Construction

In construction, linked data has been used to share the data between heterogeneous data sources. Lee et al. (2016) proposed a framework to utilize BIM and linked data to share the data about defects to overcome the limitations of the traditional ways to manage this data. Defect ontology is one of the main components of the framework. BIM is used to provide the information about the elements, which is transferred to RDF as well as the collected data about the defects. Linked data is used to overcome the interoperability challenges to enable data from different domains to be merged in broad scenarios and presented to different stakeholders. Curry et al. (2012) presented an approach to build a holistic building performance analysis using linked data, which enables the building stakeholders to share data from multiple domains. Quinn et al. (2020) presented an integration technique for mapping the sensor networks, which are involved in monitoring building conditions and building control points with Facility Management-enabled BIM. Radulovic et al. (2015) proposed guidelines for developing linked data related to energy consumption in buildings.

The guidelines include the processes to transfer the data to linked data. Pedro et al. (2017) presented an approach to share and integrate construction safety information from different sources using linked data and semantic web technologies. Their approach includes developing a safety ontology, the information about accidents, JHA, and safety rules.

2.9.4 Linked Data for Safety in Construction

The construction domain remains the most accident-prone industry with high number of serious injuries or deaths (Le et al. 2014). Safety management is necessary to check the safety documents and related safety regulations and rules for each operation and task. Different resources provide information about the expected hazards and how we can avoid them. Organizations provide safety regulations and rules and other documents that can be used to mitigate and avoid accidents in the earthwork domain. However, these safety-related rules, regulations, and documents are often unstructured and fragmented. Thus, finding the related contents that are required for safety issues in a timely manner is a challenging and inefficient task.

To address this issue and enhance the retrieving process of these different contents, linked data and semantics technology can be used to integrate and share safety information including the hazards, the collected safety information and safety regulations. This approach will help to find the related safety rules and regulations related to the hazards in the workzones and improve safety management in the earthwork domain.

Ontologies are the cornerstone of the linked data with the ability to be implemented in different languages (e.g., OWL) that are accepted by W3C (Radulovic et al. 2015). Linked data overcome the interoperability issue, link the different data sets with different formats and facilitate the organization data expansion by linking their own data to other data from other sources. Therefore, developing EW-Onto and integrating it with other related ontologies to the earthwork domain is expected to provide the robust knowledgebase that can be further extended to be linked with other sub-domain of knowledge (e.g., legal, government, and environment).

2.10 SUMMARY AND CONCLUSIONS

This chapter was dedicated to the review of the literature on the earthwork domain and several areas that pertain to the topic of the present research. This chapter includes reviewing the types and the levels of earthwork operations. Different ontologies that have been developed in construction for different purposes are reviewed. Topics related to ontology development methodologies and the evaluation approaches are explained.

Based on the above-presented literature review, it can be concluded that while there is a need for a formal representation of knowledge in the earthwork domain, there is a palpable absence of an earthwork ontology. Also, while different earthwork-support tools/systems are complementary and inter-dependent at the functional level, there is very little interoperability between these tools/systems. It is shown that similar problems have been already addressed successfully in other domains by adopting a semantic approach to develop relevant ontologies.

CHAPTER 3 OVERVIEW OF RESEARCH METHODOLOGY

3.1 INTRODUCTION

The literature review indicated that the current earthwork domain lacks the formal and consistent representation of the concepts, relationships, and semantic modeling required to cope with smart construction advancements. Therefore, an overview of the research methodology is presented in this chapter for the development of EW-Onto to enhance data exchange in the earthwork domain, and extend EW-Onto to include the safety regulations and integrate it with soil and sensor ontologies. This overview of includes the research scope and the research phases and components.

3.2 RESEARCH SCOPE

The scope of the research is illustrated in Figure 3-1, including the scopes of EW-Onto and IEW-Onto. The bottom of the triangle covers available *process models* and the top of the triangle addresses the concept of *smart construction*. EW-Onto focuses on the main components in earthwork domain including: (1) the different resources related to earthwork operations (e.g., excavators); and (2) the operations, which represent the logic of performing earthwork operations (e.g., compaction), processes, tasks, and micro-tasks under the operations. The concepts and the relationships in these components are captured in EW-Onto.

IEW-Onto benefits from available ontologies, such as Ontology of Soil Properties and Processes (OSP) (Du et al. 2016) and Semantic Sensor Network (SSN) (Compton et al. 2012). This integration aims to support the concept of smart construction, which ensures the improvement of productivity in safer workzones with a high level of quality. SSN applies Sensor Model Language (SensorML) in its classification and taxonomies presentation. SensorML represents the sensory data in XML format to enable interoperability (OGC 2020). Also, IEW-Onto includes the performance guidelines, which are related to performing the operations and processes in ways that are conforming with safety rules and guidelines. An example of these guidelines is Occupational Safety and Health Administration OSHA Regulations (OSHA 2020a).

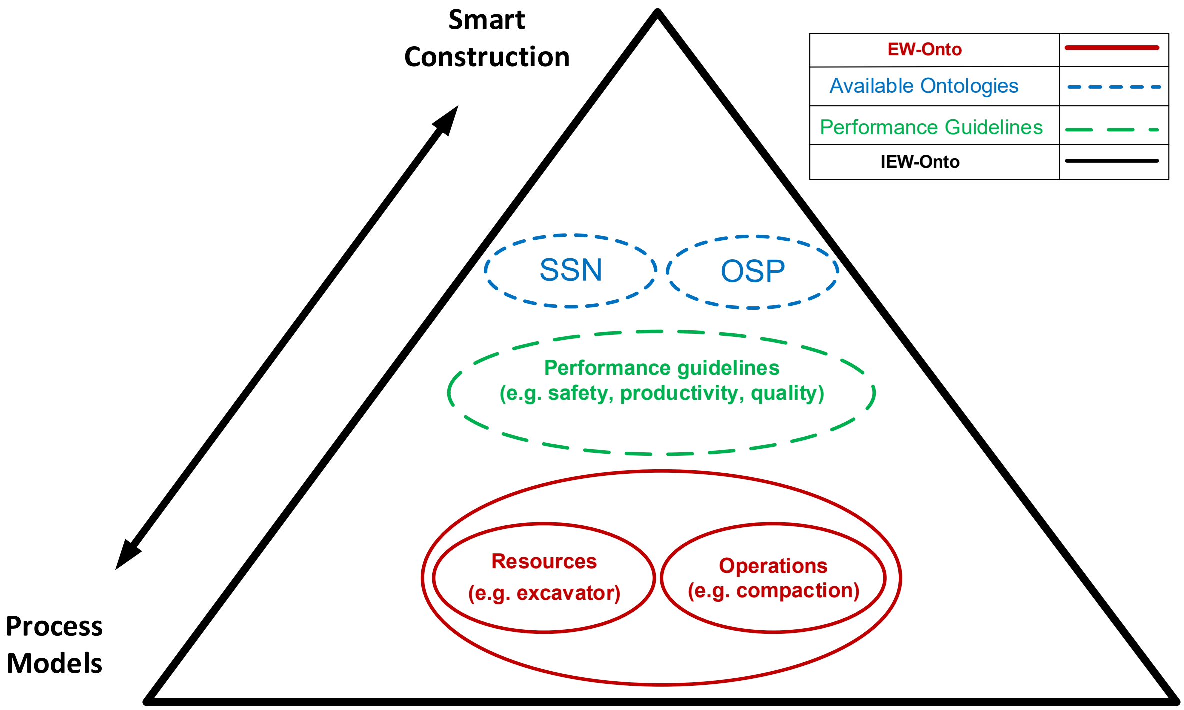


Figure 3-1 Scope of research

3.3 RESEARCH PHASES AND COMPONENTS

Figure 3-2 shows the research phases and the components in each phase.

3.3.1 EARTHWORK ONTOLOGY

Chapter 4 addresses the first and second objectives of this research (i.e., to extract the explicit and tacit knowledge required for the earthwork domain and to formalize the extracted knowledge by developing EW-Onto). The development of EW-Onto starts with defining the concepts and building taxonomies for earthwork operations and equipment. As shown in Figure 3-2, creating EW-Onto has the following steps: (1) defining the scope of EW-Onto; (2) defining the concepts and the taxonomies in the domain. The taxonomies include the equipment taxonomy and the project taxonomy; (3) EW-Onto coding using ontology editor (i.e., Protégé); (4) verifying EW-Onto using the consistency checker; (5) improving EW-Onto by adding more relationships; (6) validating EW-Onto using a survey; and (7) documenting EW-Onto.

3.3.2 EXTENDING AND INTEGRATING EARTHWORK ONTOLOGY

In order to use EW-Onto as the knowledgebase for developing the next generation of decision-support systems for safety management of earthwork operations, it is necessary to extend it to cover safety-related regulations, sensing technologies, and soil properties. Chapter 5 addresses the third and fourth objectives in this research (i.e., to extend EW-Onto to enhance operation safety by adding rules based on safety regulations and integrating with related concepts from sensor and soil ontologies).

The steps to developing IEW-Onto can be summarized as following: (1) defining the scope of IEW-Onto including defining the candidate ontologies for the integration processes; (2) formalizing the unstructured safety knowledge from different resources (e.g., OSHA regulations and best practices) in IEW-Onto; (3) classifying and structuring the unstructured sensor data; (4) defining the concepts and the taxonomies from the related ontologies. The integration process, which is part of the IEW-Onto development phase, includes three main steps: (a) defining the candidate ontologies that are related to the earthwork domain and facilitate the extended knowledge representation; (b) analyzing the candidate ontologies; and (c) implementing the integration process; (5) IEW-Onto coding, which include the implementation of the integration processes; (6) verifying IEW-Onto using the consistency checker and verifying the safety rules; (7) improving IEW-Onto by adding more concepts and relationships; (8) performing the evaluation process using data-driven and application-based approaches; and (9) documenting IEW-Onto. The output of this phase addresses the objectives of developing IEW-Onto.

Figure 3-3 illustrates the main components of Chapter 4 and Chapter 5.

3.4 SUMMARY

This chapter provided an overview of the research methodology including the scope of the research and a brief description about developing EW-Onto as well as the steps to extend it to IEW-Onto by integrating the related ontologies and adding the safety rules.

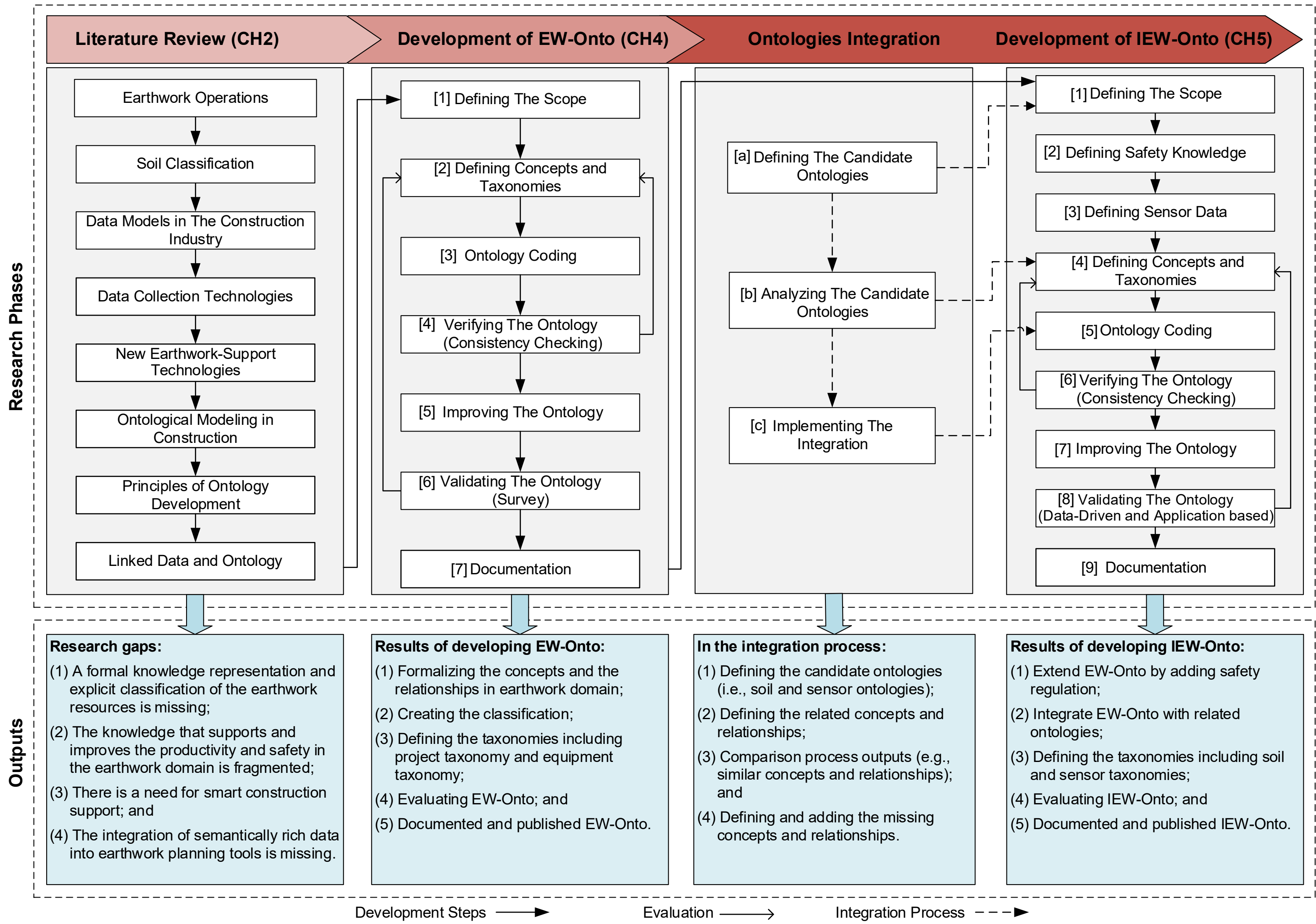


Figure 3-2 Overview of the research methodology

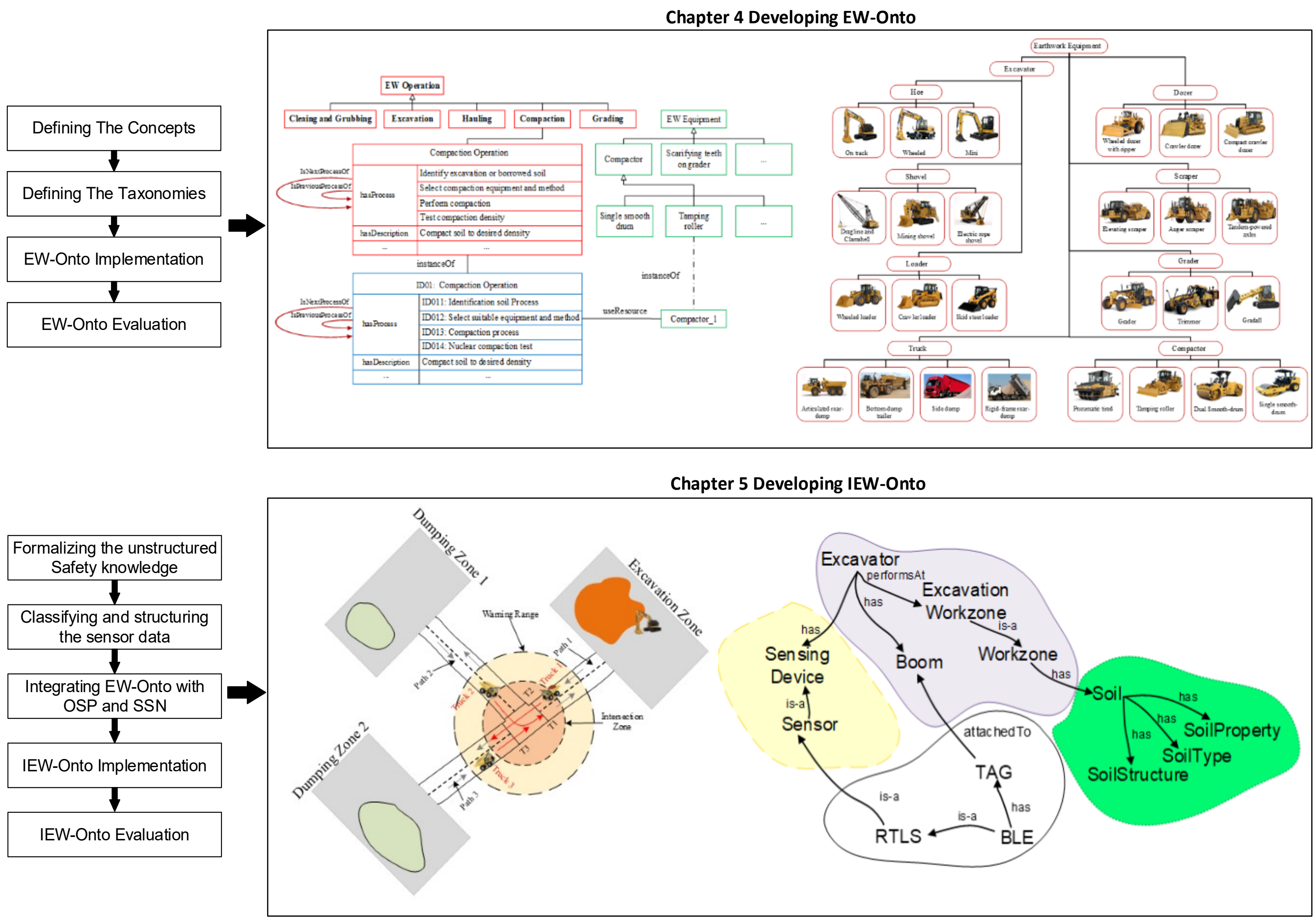


Figure 3-3 The main research components

CHAPTER 4 FORMALIZING KNOWLEDGE REPRESENTAION IN EARTHWORK OPERATIONS THROUGH DEVLOPMENT OF DOMAIN ONTOLOGY

4.1 INTRODUCTION

Earthwork operations are an integral part of many medium and large construction projects. For instance, more than 20% of the total cost of road projects is dedicated to earthwork operations (Artun et al. 2019; Parente et al. 2015; Hare et al. 2011; Smith et al. 1996). These operations are complex and dynamic in nature, and there are many different factors that need to be monitored closely to ensure the success of a project (Cheng et al., 2011). The successful planning and execution of these operations rely on the generation, processing, transfer, and analysis of a large volume of information deriving from various sources such as earthwork equipment, target operations, planning details, execution details, and surrounding environment. These data are processed and used to enhance communication between different operators, increase productivity and safety, and improve the decision-making process. However, given the diversity and heterogeneity of the information generated in earthwork operations, the smooth management of the information is a challenging task. In recent years, ontological modeling and development have been used to address similar challenges in different domains (Hou et al. 2020; Pfaff et al. 2018; Meenachi and Baba 2012; Guizzardi 2005). Ontologies enable semantic interoperability, which can pave the way for managing the information sources in complex environments (Viljamaa and Peltomaa 2014).

The purpose of this chapter is to formalize knowledge representation in earthwork operations through the development of domain ontology that is called Earthwork Ontology (EW-Onto). This ontology should be able to link to the regulations and the rules related to the domain and include them in the planning and execution procedures. Also, this ontology needs to link with the other data models that are already used in the domain such as IFC (Weber et al. 2019) and road/terrain models (Lee and Kim, 2011).

The developed ontology can be used to create a semantic-based integration method to support communication between the different disciplines and stakeholders in earthwork domain. Semantic technology supports information exchanges among different systems and between the stakeholders including agents, and applications. The semantic interoperability is one of Industry 4.0

interoperability architecture levels besides operational, systematical and technical interoperability (Da Rocha et al., 2020; Lu, 2017; Mrugalska and Wyrwicka, 2017). The new digitalized construction industry, which is defined as a pure and simple instantiation of Industry 4.0, is called Construction 4.0. Construction 4.0 promises to improve productivity, quality, and resource efficiency (Boton et al., 2020; Craveiroa et al., 2019). EW-Onto provides the conceptual model to facilitate the advancement in the earthwork domain by connecting and representing the domain ontological primitives including the concepts (e.g., operations and resources), relationships, and axioms. EW-Onto has the extensibility to accommodate a wide spectrum of semantic contexts to satisfy Construction 4.0 requirements.

The structure of the chapter is as follows: Section 4.2 introduces the proposed method including the research methodology, the development workflow, defining the concepts, and building the taxonomies. Section 4.3 covers the initial implementation of EW-Onto. Section 4.4 presents the EW-Onto evaluation. Finally, Section 4.5 discusses the summary and conclusions.

4.2 PROPOSED METHOD

This section introduces the proposed method to develop EW-Onto including the scope of EW-Onto to show the boundaries of the work, EW-Onto development workflow, and the definition of the concepts and the taxonomies.

4.2.1 Overview of EW-Onto Development Workflow

In this section, the main steps for developing EW-Onto will be explained. METHONTOLOGY is adapted to develop EW-Onto because this methodology is: (1) application-independent and mature (Corcho et al. 2003); (2) well- documented and clear for the development activities; (3) based on experience acquired from developing the ontology for other domains (Fernández-López et al. 1997); and (4) recommended by Foundation for Intelligent Physical Agents (FIPA), which is the standard for the interaction in MAS (Fernández-López 1999).

The best practices and knowledge in the earthwork domain are used to develop EW-Onto. The development workflow is illustrated in Figure 4-1 EW-Onto development lifecycle based on METHONTOLOGY includes initial, development, and final stages. The initial stage is composed of the steps to define the scope, main concepts and the taxonomies of EW-Onto. The development stage is dedicated to building and verifying the initial structure of EW-Onto. The final stage is to add new relationships or adjust existing ones and to perform validation of EW-Onto through interviews with experts and end-users. Knowledge acquisition, evaluation, and documentation are carried out during the whole lifecycle of the ontology development. The processes in IDEF add more details about the METHONTOLOGY states and activities including the input, the output, the control, and the mechanism for each process. The development stages are explained in detail in the following paragraphs.

Initial stage: The initial stage of the ontology development comprises two steps. Process ID1*:* Defining the scope of the EW-Onto based on the requirements. The requirements (e.g., terms, data properties, and object properties) are collected from various sources, such as textbooks and online resources. The developed ontology aims to address the Competency Questions (CQ), such as: why do we need to develop the ontology? What are the domains and the scope of the ontology? Who are the users of the ontology? Answering these questions addresses parts of the requirements in the earthwork domain, which are related to the concepts and relationships. The answers to these questions are linked to each other. For example, the question about the need to develop EW-Onto gives an idea about how EW-Onto will be used by the users and for which purposes. Moreover, the scope of EW-Onto addresses the target users, which in turn will limit or extend the concepts and the relationships included in EW-Onto. Also, it gives an idea of the size of the development and the level of detail that should be covered in EW-Onto. Process ID2: Defining concepts and taxonomies for EW-Onto. In this step, the data related to the ontology are gathered to construct the ontology. Both previous steps require communication with and feedback from experts and end-users in the domain. Furthermore, the list of requirements obtained from reviewing construction books and literature helps develop the ontology (e.g., Nunnally, 2004; Peurifoy et al., 2010; Vahdatikhaki, 2015). These two steps correspond to the Specification, Conceptualization, formalization and Knowledge acquisitions, in METHONTOLOGY.

Integration process aims to reuse the available ontologies, such as soil ontology and sensor ontology that extend EW-Onto to include other concepts and relationships. The integration process will be considered in Chapter 5.

Development stage*:* The development stage includes two main steps. Process ID3*:* The first step is developing the domain ontology. Depending on the maturity level and availability of ontologies in a domain, a new ontology can be developed as either an extension (or continuation) of other pertinent ontologies or as a new development. In this research, since there are no earthwork domain ontologies, the ontology needs to be developed from scratch. Process ID3 corresponds to the Implementation in METHONTOLOGY. Process ID4: The second step is to verify the developed ontology. The verification of ontology aims to judge the ontology content from a technical point of view (e.g., concepts, relations, scope, and taxonomy). The verification is partially performed using the consistency rules and competency questions. The result of this step is the semi-final ontology, which is capable of representing the knowledge in the domain.

***Final stage*:** Process ID5: Maintenance is covered in this stage. The developer extends the ontology by adding rules and the regulations, such as earthwork-related regulations from OSHA (OSHA 2020a). These regulations are used as constraints when formalizing the relationships between the concepts. Semantic Web Rule Language (SWRL) can be used to translate these regulations into a machine-readable format (Ian et al. 2004). Process ID6: The experts and end-users may either recommend new relationships or adjust the existing ones to improve the ontology. Process ID7: The ontology validation aims to prove that the developed ontology complies with the real world. Thus, this validation is carried out by domain experts and the end-users of EW-Onto.

Verification and validation are associated with the evaluation activity in METHONTOLOGY, and are covered in Process ID4 and Process ID7. A criteria-based evaluation method is used to evaluate the EW-Onto for different reasons: (1) EW-Onto is not extended or built based on a previous ontology; (2) There are no similar ontologies in earthwork domain that can be used to perform other evaluation methods such as the golden standard; and (3) Because there is currently no application that can use the developed EW-Onto, the use of an application-based evaluation method is not feasible. Process ID8: Documentation, attempts to document each step to deliver the ontology. This activity is carried out during the development of EW-Onto.

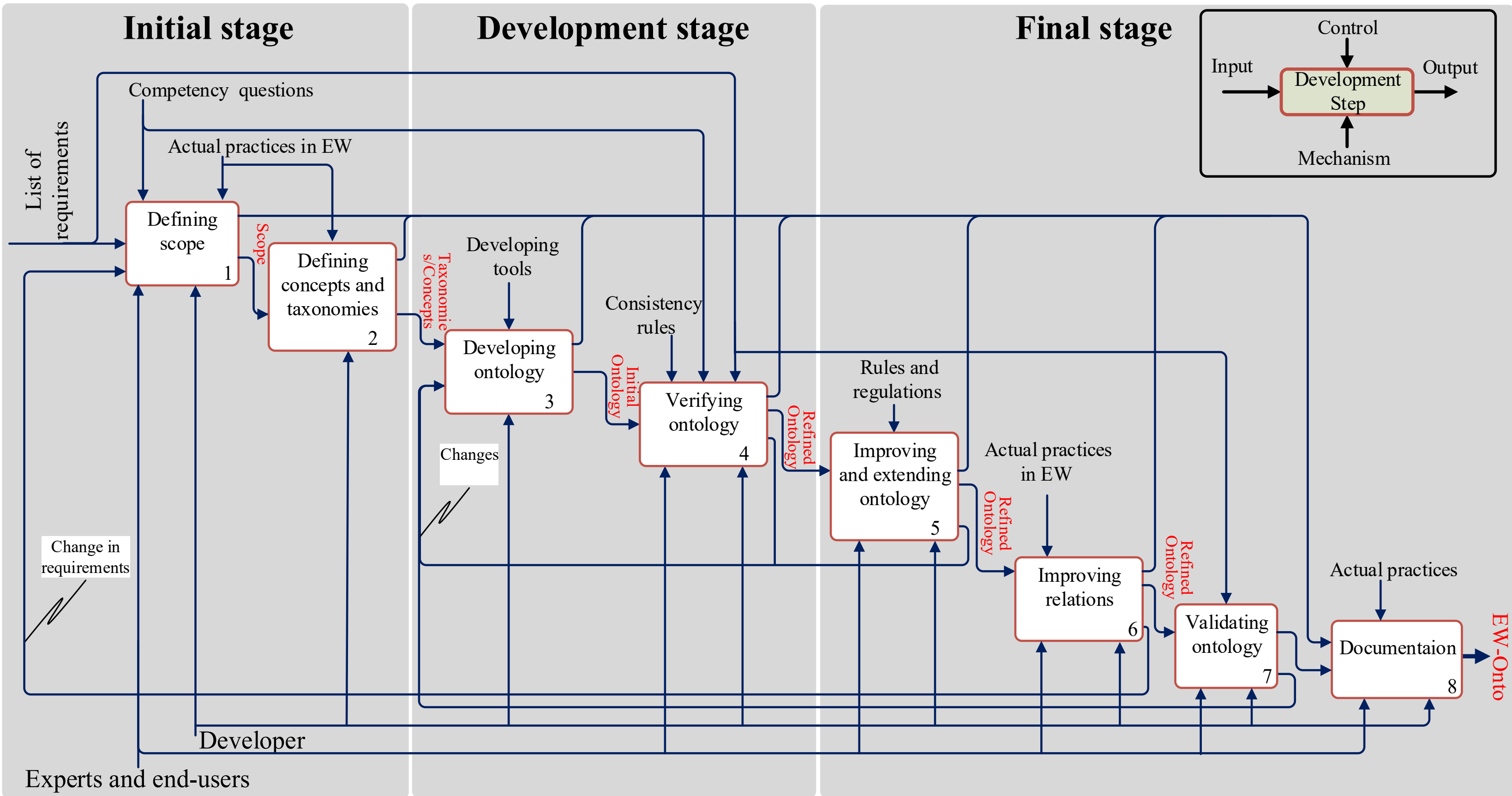


Figure 4-1 Workflow steps to develop EW-Onto

4.2.2 Defining the Concepts and Building Taxonomies for EW-Onto

The basic concepts and taxonomies in earthwork domain are discussed in this section.

4.2.2.1 Basic Concepts in Earthwork Domain

The concepts in the construction domain are composed of different abstract groups or sets. Unified Modeling Language (UML) (Bennett et al. 2001) is used to create the class diagram. Figure 4-2 represents an overview of the main entities and their relationships in EW-Onto. As shown in this class diagram, a *team* consisting of several pieces of equipment performs an *operation*. Each *operation* consists of a number of *processes*, and each *process*, in turn, consists of a number of tasks. Each *task* can be further decomposed into a number of *micro tasks.* A *project* is a collection of *operations* performed by a firm or a group of firms to deliver a product or a set of products using resources for a known period. An *operation* is a group of associated *processes* that have a start and end time to deliver a part of the *project* based on a specific method. A composite operation is a combination of two or more *operations*. An example of composite operations is earthmoving, which is a combination of an excavation operation and a hauling operation. A *process* is a group of related *tasks* that are performed to deliver detailed elements of a *product*. A *task* is the essential actions that are performed by equipment (e.g., for the hoe: digging, dumping, and relocating) or labor. Each task consists of micro-tasks, which describe each action performed by equipment in detail (e.g., for the hoe: swing to digging, digging, swing to dumping, and dumping). A *micro-task* duration is usually in the range of minutes or seconds. Each entity in the class diagram has relationships with other entities, and each individual of these entities has data properties. A *general coordinator*, is the main supervisor on the construction site and supervises the *team coordinators.* A *team coordinator* is responsible for a *team* and deals with the issues related to his team. The relationship *coordinates* link the *team coordinator* and the *team*. The *team* is composed of two or more pieces of equipment that perform a specific operation or process. For example, the excavation team is composed of one or more hoe and one or more truck to move the soil from one place to another. There are different factors that govern the number and specifications of each piece of equipment in the team, such as the schedule, cost, and quality of products. The *team performs operations*, which *lead to* the *products*. The *performance guidelines* contain the rules and regulations that relate to safety, productivity, and quality. *operation*s, *processes*, *tasks*, *micro-tasks*, *products*, *site*, and *resources* should follow these guidelines. Moreover, products, resources, and site *is equipped with sensors*. The information, which are collected about different entities, can be used to build knowledge about these entities in EW-Onto, which in turn can be used to improve the safety and increase the productivity. EW-Onto can be represented through the following main concepts:

(1) *Entities*: Entities in EW-Onto consist of *projects*, *operations*, *processes*, *tasks*, *micro-tasks products*, *resources*, and *actors*. *projects*, *operations*, *processes*, *tasks*, and *micro-tasks* are concepts linked to each other through various relationships, such as a *project has operations*, an *operation produces products* and *uses resources*, and *processes* are *part-of operations*. A *Product* is the outcome of *projects*, *operations*, *processes*, *tasks*, and *micro-tasks*. The *product* may consist of sub-products, and each of them may belong to a different category. There are two categories of products: physical products and non-physical products. Physical products are also divided into (1) simple products, which are usually the outcome at the level of task or micro-task, such as trenches, holes, and documents; and (2) complex products, which are the outcome at the level of project, such as roads or bridges. On the other hand, non-physical products are abstract components, such as experience and knowledge that workers and engineers acquire at different levels. A *resource* can be expendable or reusable. Expendable resources are any materials that are consumed at any level of a project, such as sand, cement, or water; whereas reusable resources are any resources that can be reused, such as equipment, tools, or human resources. An *actor* is a type of entity that can affect the state of other entities. For instances, an *operator operates* the equipment, and a hoe is used for digging and changing the surface of the earth. There could be more than one actor involved at different levels.

(2) *Attributes*: Attributes in EW-Onto can be divided into three main types, namely basic, temporal, and impact attributes. Basic attributes describe the main characteristics of the individuals of an entity. For example, the capacity of equipment or an operator ID. These attributes do not change during *operations*, *processes*, *tasks*, or *micro-tasks*. Temporal attributes describe the entity from a temporal point of view, and they can be changed. An example of these attributes is the volumetric changes of the bulk materials (e.g., soil, rock, and clay). In other words, the volume of a certain load may take different values during operations between natural conditions (i.e., *in place*), after digging, and after compaction. Impact attributes are those that are influenced by the effects of another entity, for example, the changes in the stockpile areas as a result of volume swept by excavation.

(3) *Relationships*: The relationships in EW-Onto are categorized into three main types: internal, external, and transitive. The internal relationships are the relationships between different entities and concepts. For example, the relationships between two or more pieces of equipment that are grouped to perform a specific process and the concept *team*. These pieces of equipment are *part-of* this team. External relationships reflect the external relationships between the concept (e.g., *team*), and another entity. For example, as shown in Figure 4-2, the relationships between the *team* and the upper level of coordination, (i.e., *general coordinator*). Transitive relationships describe the relationships between entities that are not linked directly (e.g., *operator* and *team*).

(4) *Modalities*: Modalities can be one of four main categories: management, situation, temporal and engineering. Management modality is the description of an operation when it belongs to one of the processes of project lifecycle (initiating, planning, execution, monitoring and control, and closing). Situation modality is the description of operations, processes, and tasks. There are two types of situation modality: planned and unplanned. Operations, Processes, and Tasks which should be performed based on a specific schedule and should have a start time, duration, and finish time are called planned operations, processes, or tasks. On the contrary, unplanned operations, processes or tasks, which are not scheduled before their start time. An example of unplanned processes is the process that is performed to cope with an emergency or a mistake in the performance. Temporal modality is the description of an operation based on the state that the operation belongs to during a certain duration. Engineering modality is the description of operations, processes, and tasks. This description identifies to which discipline they belong (e.g., earthwork operation belongs to civil engineering).

(5) *Strategies*: Strategies refer to mechanisms that are used to accomplish operations and tasks. Strategies can be obtained from previous best practices for similar operations (Razuri et al. 2007). In the earthwork domain, a strategy is composed of two types, *work performing guide* and *work performing method*. The *work performing guide* covers the techniques that are used to carry out the operation at the construction site. For example, splitting equipment between two teams containing equal number of different trucks and excavators depends on some constraints (e.g., time, the size of work, equipment availability). The *work performing method* is a more detailed description of how to perform an operation.

(6) *Rules and Regulations Axioms*: Axioms provide explicit logical texture between *concepts*, *relationships*, and *constraint.* There are three types of axioms: *permissive axioms*, *transitive axioms*, and *constraints and regulation axioms*. A *permissive axiom* is confined to the axiom that defines the simple meaning of concepts in the ontology. For example, the meaning of unplanned operation, which is an operation that is performed when accidents occur and lead to tasks that are originally not scheduled. A *transitive axiom* is the axiom that transfers from one level to another. For example, an operation is complete if and only if all tasks that belong to this operation are complete. A *constraints and regulation axioms* are the axioms that can be expressed in an explicit way using the rules. These rules should be followed by the different stakeholders of the project. The references to these types of axioms are mainly from organizations, such as OSHA (OSHA 2020a) and the best practice records.

4.2.2.2 Developing the Earthwork Taxonomies

Based on the concepts defined in Section 4.2.2.1, the next step is to build the main taxonomies in EW-Onto. These are (1) the classification of earthwork equipment; and (2) the classification of the *projects*, *operations*, *processes*, *tasks*, and *micro-tasks*. These classifications are presented in Figure 4-2 as a high-level structure and will be explained in the next sections.

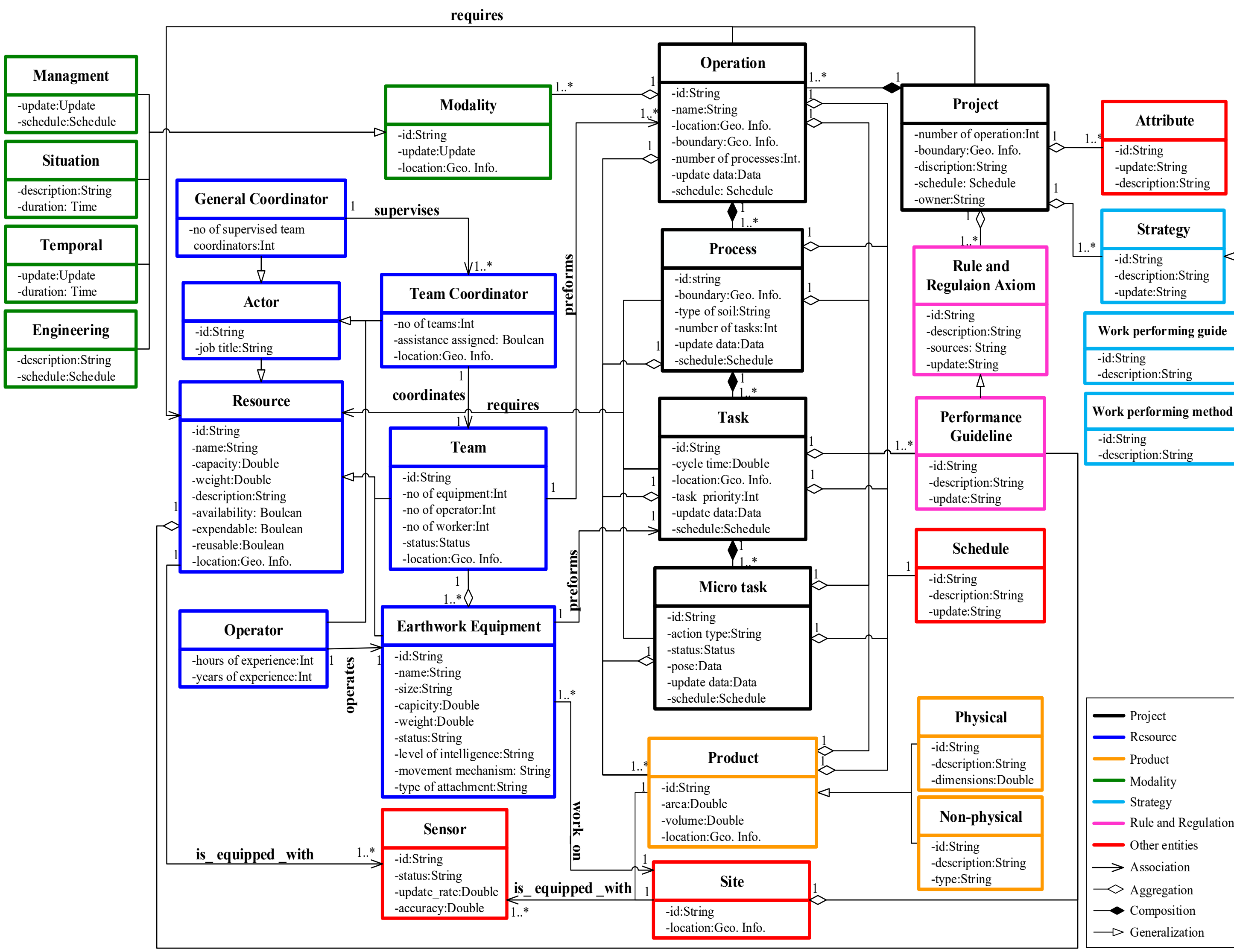


Figure 4-2 High-level structure of EW-Onto

Earthwork equipment taxonomy: The equipment classification will facilitate the development of EW-Onto. This classification is used to build the taxonomy related to the equipment in EW-onto, which is Process ID2 in the methodology.

Figure 4-3 provides some details about the equipment and its related properties. In the class diagram, extended details are added to the excavator as an example of the classification levels. The properties listed under *earthwork equipment* class are related to all earthwork equipment. Under the *excavator* class, the properties are more specific for the excavator, such as *bucket* capacity and *counterweight* clearance.

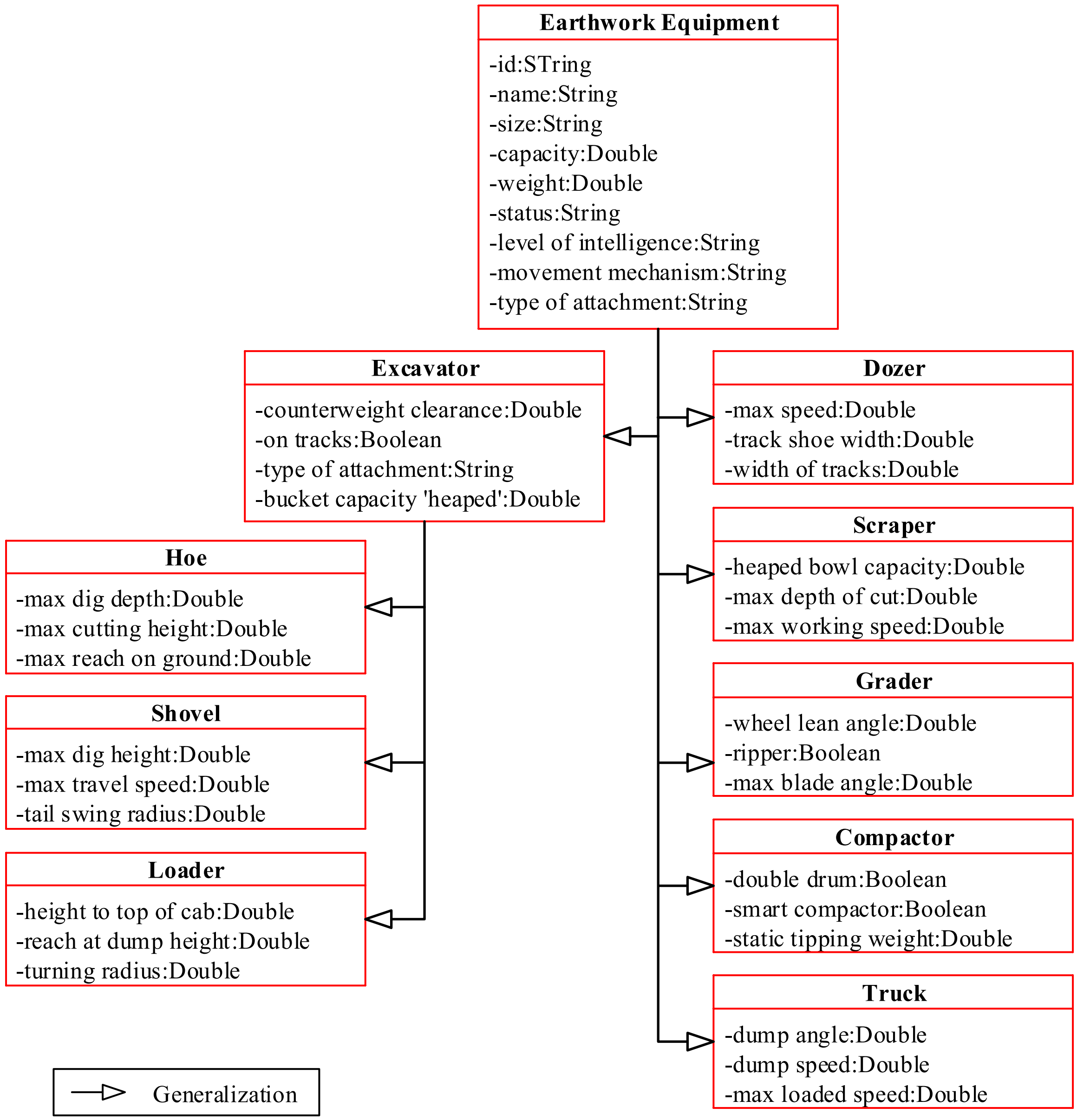


Figure 4-3 Earthwork equipment class diagram

Figure 4-4 illustrates the hoe parts as an example of the parts names and the details of the equipment. Each part has its own data properties, which affect the performance and limit or extend the functionality of the equipment. More detailed data properties are listed under each sub-class of *excavator*, such as maximum dig depth and maximum reach on the ground.

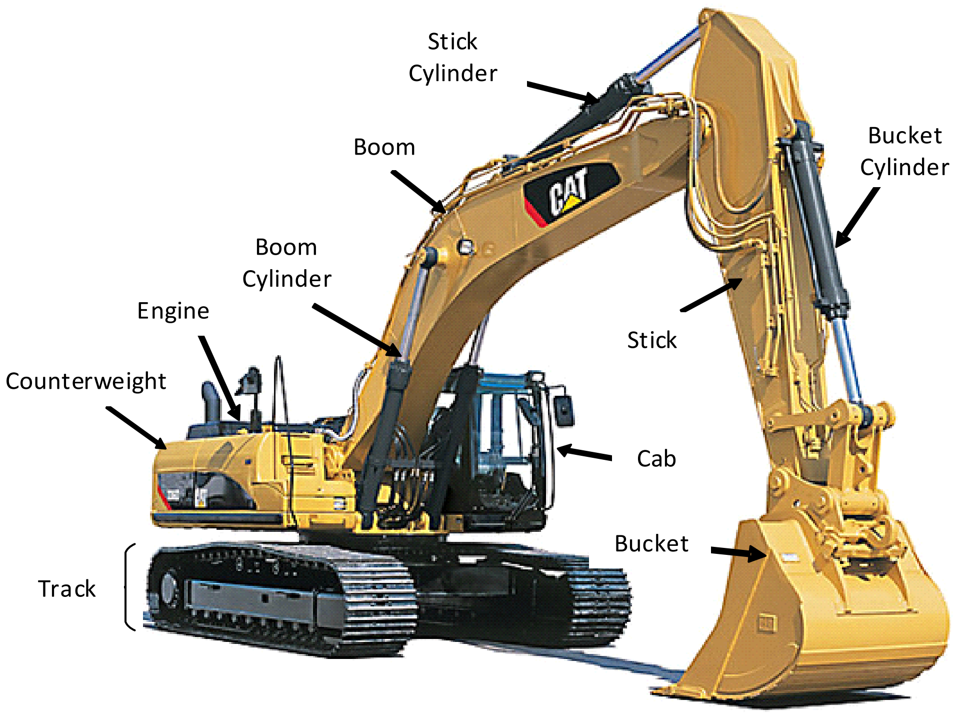


Figure 4-4 Hoe's main parts

Figure 4-5 shows a visual example of earthwork equipment classification as a tree.

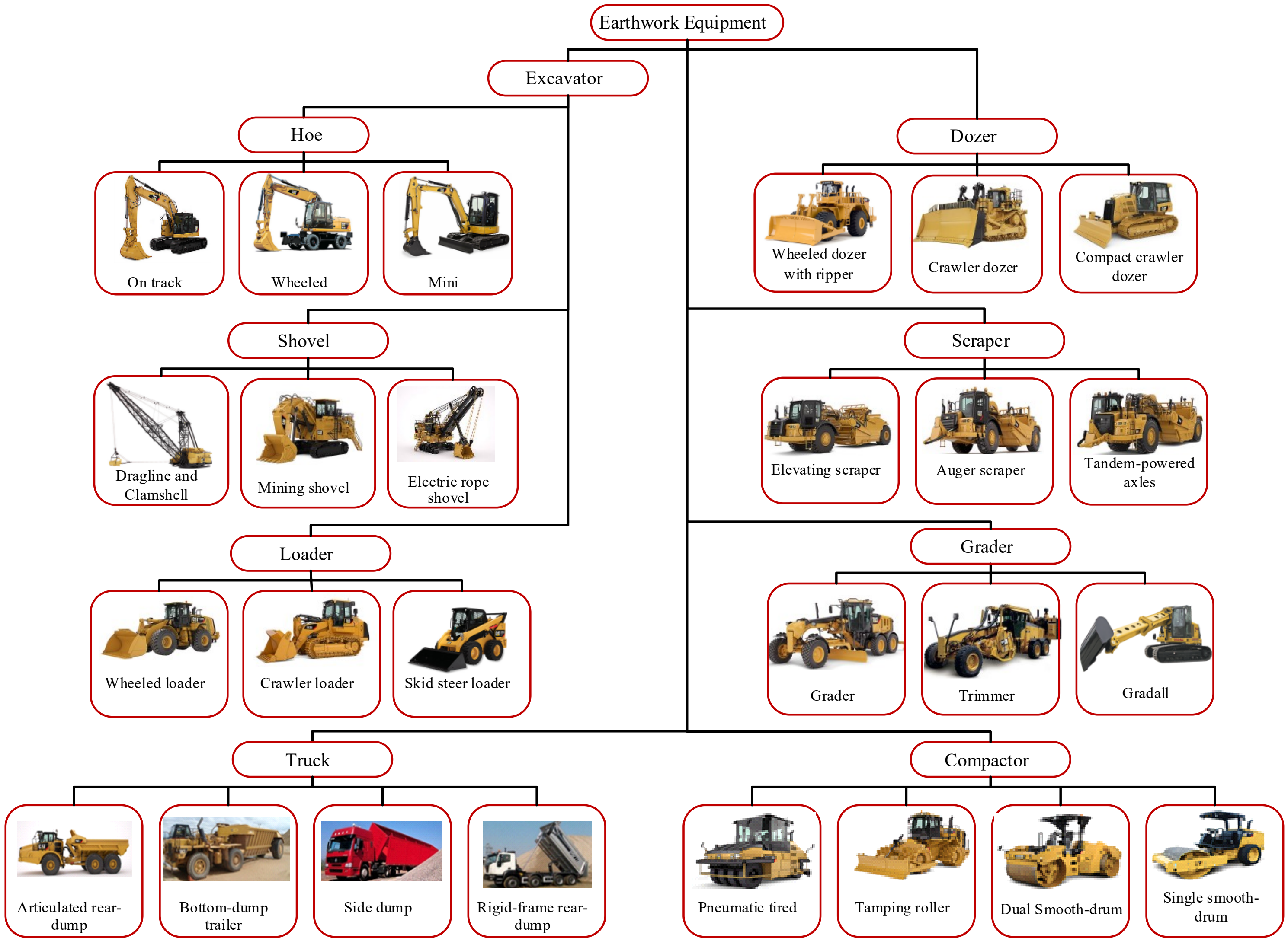
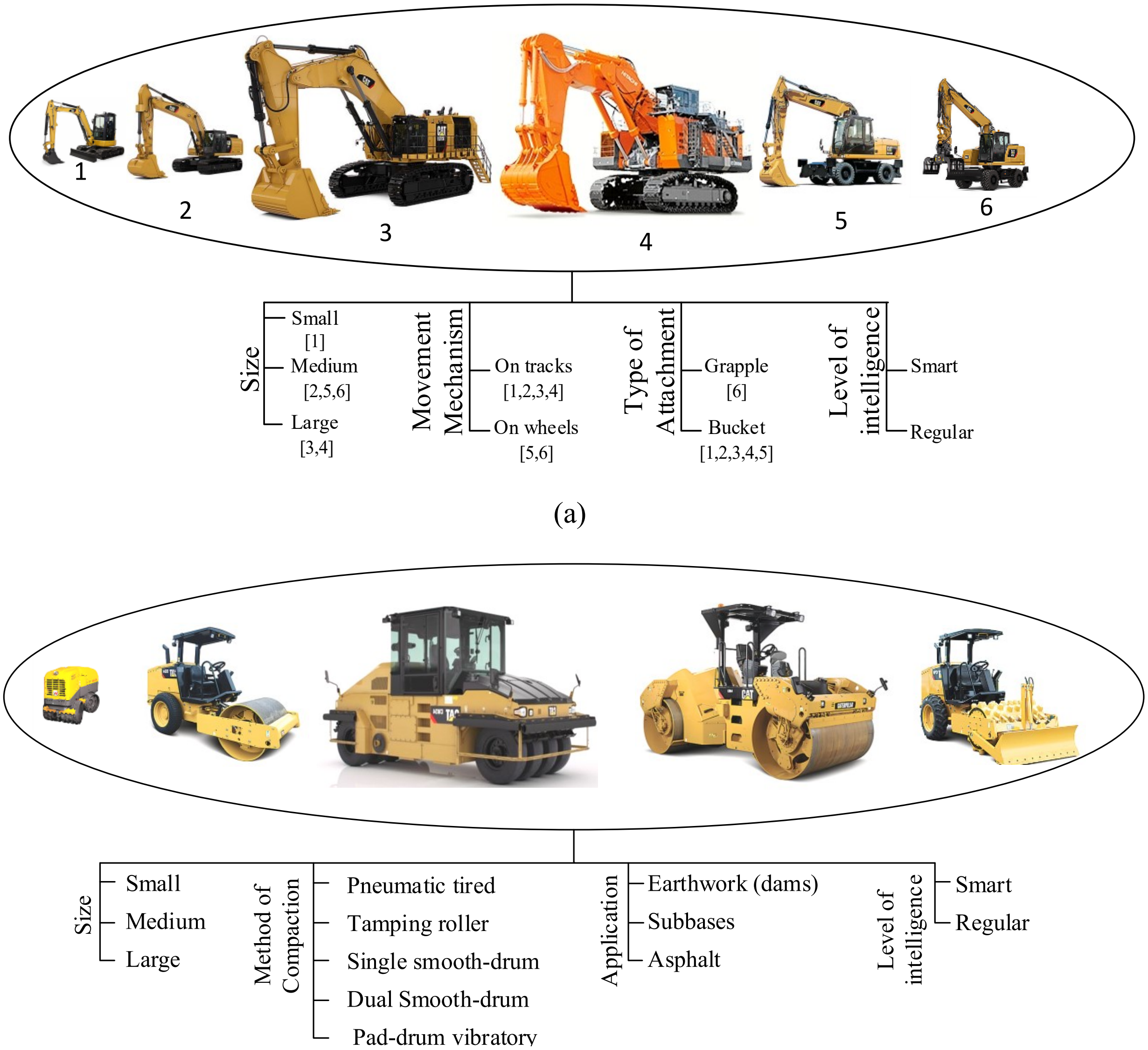


Figure 4-5 Example of classification for equipment

Taivalsaari, (1996) emphasized that there are no general rules that can be used as the basis for objects classification. In the earthwork domain, different equipment can be classified using different criteria. For example, Figure 4-6(a) depicts the classifications that can be utilized for hoes based on different points of view. As shown in Figure 4-6(a), if the size is used for the classification, then equipment 3 and 4 are classified as large. On the other hand, if the movement mechanism is used for the classification, then equipment 1, 2, 3 and 4 are under the same class because they are all mounted on tracks; and equipment 5 and 6 are mounted on wheels but have different attachments (i.e., *grapple and bucket*). Figure 4-6(b) shows another example of the classifications that can be utilized for compaction equipment. Different compaction methods can be used in different applications (e.g., asphalt or subbase), and different equipment with different sizes, either smart or regular, use these methods. These various classifications lead to the need for creating an unambiguous way of classifying the equipment in the earthwork domain.

EW-Onto contains the data properties of each type of equipment. This information, in turn, can be used to classify the equipment according to different perspectives. For example, each piece of equipment is classified based on the size using *hasSize* relationship that can take the values small, medium, and large. Using the SWRL rule (*Earthwork_Equipment (?EWE) ^hasSize (?EWE,"small") ->Small (?EWE)),* all the pieces of equipment that are small will be inserted into the class *Small.* Moreover, using the Description Logics (DL) query (Tudose et al. 2013) Earthwork_Equipment and *hasSize* value *"small"* gives a list of small equipment regardless of other properties.



(a)

(b)

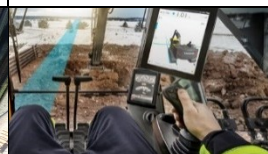
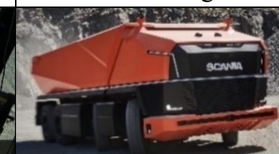
Figure 4-6 Examples of faceted classifications for hoes (a) and compaction equipment (b)

The above-mentioned classification of equipment can be extended to consider the level of autonomy of the earthwork equipment. As mentioned in the literature review, the 0 to 5 autonomy scale of SAE (SAE 2018) can be adapted for earthwork equipment classification. The level of autonomy is linked with the Degree of Freedom (DoF) that the system can control. As shown in Table 4-1 the autonomy in earthwork equipment can be classified as follows:

- Level 0: There is no automation at this level and the equipment is fully controlled by the operator.

- Level 1: The system focuses on a single DoF and helps the operator to perform the task (e.g., by showing the operator of the excavator the required level of the blade on the screen based on the GPS position and design data). This level is the starting point of AMG and sensor involvement in operating the equipment.
- Level 2: At this level, the equipment is partially automated. In other words, the system is controlling multiple DoF simultaneously (e.g., controlling the level and angle of the blade of a grader). The system still requires the operator involvement in driving the equipment and recovering from a potential failure. At this level, the equipment type is shifted from AMG to AMC category and more involvement of the sensors in operating the equipment.
- Level 3: The equipment is conditionally automated. At this level, the system is capable of controlling multiple DoF simultaneously under some conditions (e.g., the system controls the Boom and the Bucket, while the operator controls the Stick). The operator is required.
- Level 4: the equipment is highly automated. The system executes the tasks under a certain condition (e.g., a specific speed) and the operator remotely monitors the equipment. At this level, the system is able to cope with the unexpected disorder and does not require the operator's assistance except in case of a potential failure.
- Level 5: the equipment is fully automated. The system can complete the tasks under any conditions without an operator.

Table 4-1 Proposed autonomy levels for earthwork equipment

Level	**0**	**1**	**2**	**3**	**4**	**5**
Description	No Automation	Driver Assistance	Partial Automation	Conditional Automation	High Automation	Full Automation
Automated Functions	Regular Equipment	AMG With Operator	AMC With Operator	AMC With Partial Operator Control	AMC with Remote Monitoring	Robotic Equipment
Example	 The Cat 308 CR excavator. (Caterpillar, 2021).	 Dig assist from Volvo (McLean, 2019).	 The 14M3 Grader from Caterpillar, automatically controlling the blade (Holling and Johanna, 2016).	 Trimble earthworks grade control platform Version 2.0 (Fox, 2020).	 Scania AXL, a new cab-less concept truck (Scania, 2019).	 The NASA "Glenn Digger" excavator (Bauman et al., 2016).

Defining the autonomy level of the equipment provides a consistent classification. Thus, the equipment could be classified based on these levels beside the other classification mentioned above. The equipment with different levels of autonomy could be used at the construction projects' different operations from cleaning, earthmoving, embankment construction, compaction, and grading. These operations are explained in the next paragraphs.

Project taxonomy: In earthwork domain, there are different types of *projects*, *operations*, *processes*, *tasks*, and *micro-tasks*. This hierarchy was briefly described in Section 4.2.2.1.

There are several operations in earthwork domain. The next paragraphs explain these operations based on the sequence of performing them in the site (Delaware Department of Transportation 2020). These operations are: (1) cleaning and grubbing; (2) earthmoving, which covers the excavation and hauling operations; (3) compaction; and (4) grading.

Cleaning and grubbing: Figure 4-7 shows the processes related to the cleaning and grubbing, including: (1) checking the project plan and design documents that contain the details of this operation (e.g., the limits and boundaries of workspaces); (2) checking the obstructions and the materials in the boundaries of the site; (3) checking and determining the location of underground facilities (e.g., electric cables, sewage pipes, and gas pipelines) to avoid the risk of accidents; (4) selecting the suitable equipment and method based on the materials, boundaries and planning documents; (5) performing the cleaning operation. A hoe could be used as a cleaning machine if the bucket is replaced with a hammer or grapple attachments; (6) A hoe can be combined with trucks as a team for cleaning and hauling earth from the site; and (7) validating the cleaned site using design documents until the results are satisfactory.

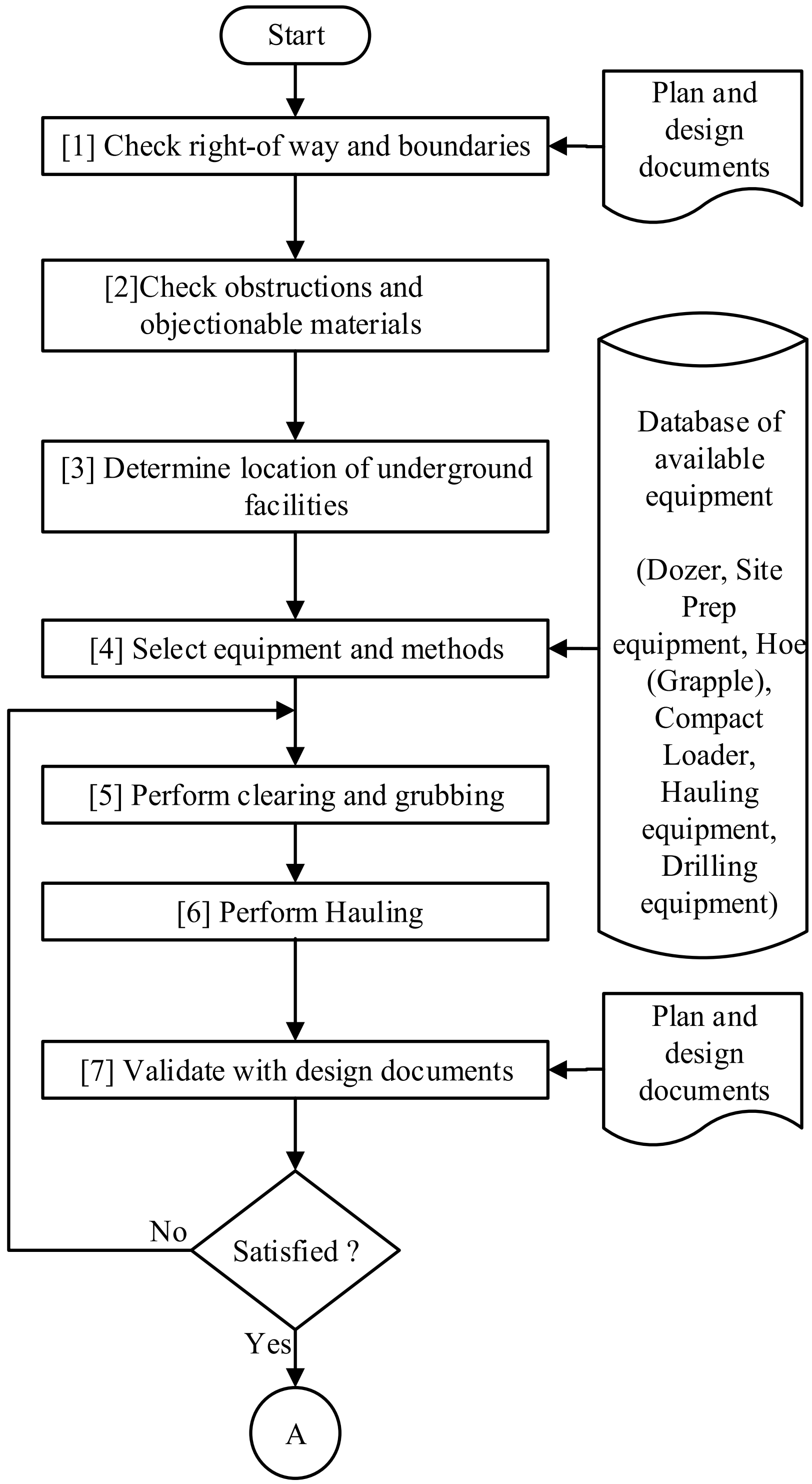


Figure 4-7 Cleaning and grubbing operation

Earthmoving: Earthmoving is a composite operation combining the excavation and hauling operations and is categorized into two types based on the performed work. When the operation is performed to dig up and haul the soil, it is called *cut;* whereas placing a portion of the excavated soil to form an embankment is called *fill*. Figure 4-8 indicates the processes to perform the excavation and embankment construction: (1) Identifying the excavation and embankment construction boundaries, which is one of the main factors for selecting the suitable equipment. For example, the distance between the cut and fill locations is needed in order to choose the equipment to perform the cut and fill operations (i.e., scraper, dozer, or a combination of trucks and excavators); (2, 3) Identifying the types and amounts of the soil using the design and planning documents, and selecting the specific type and number of equipment; (4) The operation is performed according to the rules and regulations related to the earthwork operations (e.g., OSHA regulations (OSHA 2020a)); (5) If the operation is *cut*, the excavated soil is disposed and hauled to the dumping area; (6) In case the operation is *fill,* the soil is hauled to the desired locations; (7) The soil is tested to check the required specification; (8) In the case of roadway excavation, where the soil is usually obtained within the boundaries of the earthwork site, if the soil is not enough for the embankment construction portion, extra soil must be imported from other places; (9) In some cases, the soil needs to be treated to cope with the uncontrollable effects (e.g., the weather) by applying the stabilization processes.

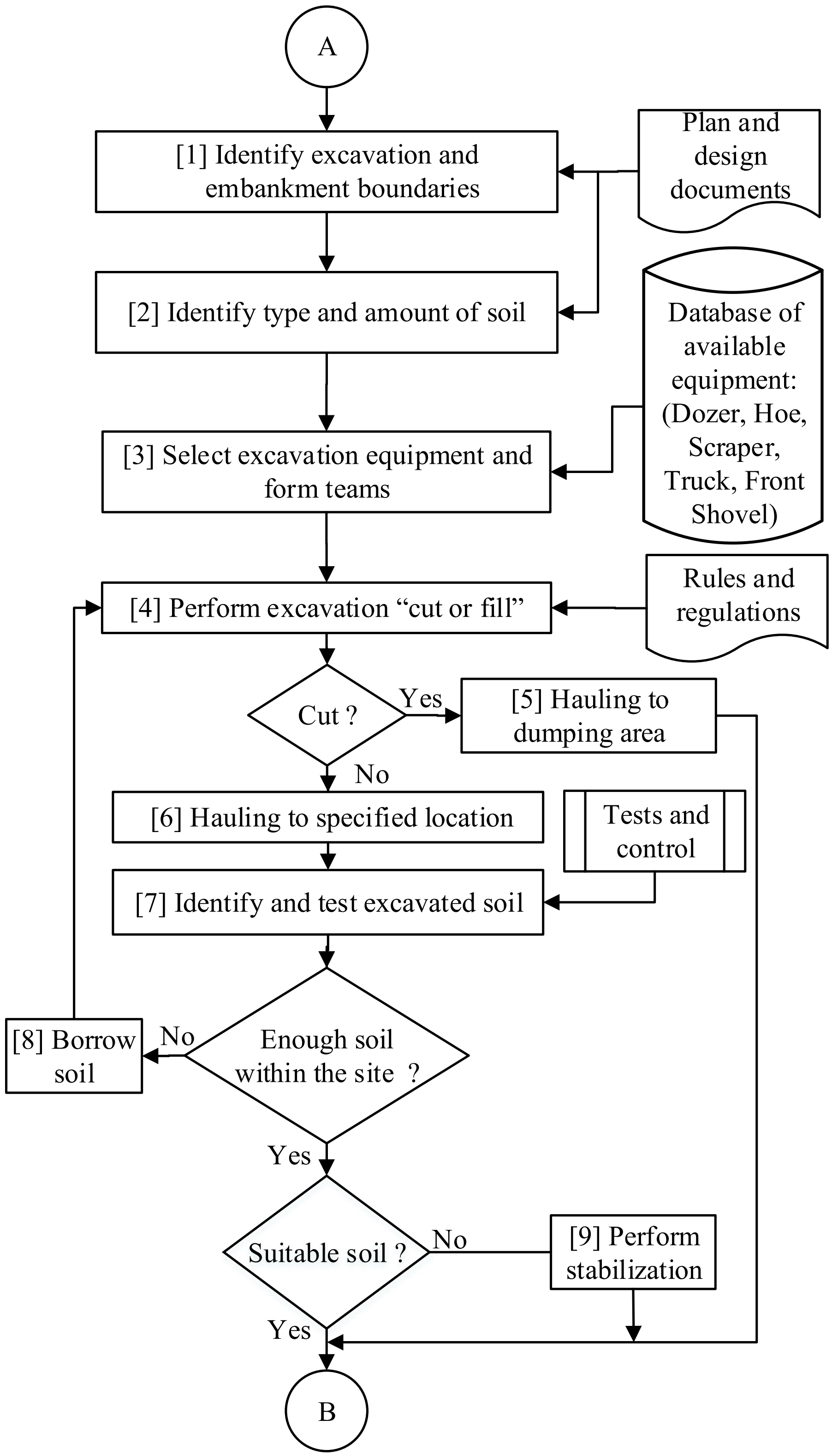


Figure 4-8 Earthmoving operation

Compaction: Figure 4-9 indicates the steps for performing the compaction operation. The steps start with (1) identify the soil; (2) based on the type of the soil, the suitable compactor is selected; (3) the compaction processes is performed with respect to the rules and regulations related to compaction (i.e., a specific compaction method for a specific type of soil); (4) the compaction density is tested and the compaction process proceeds till the required density is achieved. It is necessary to know the classification of the soils to identify the suitable equipment and method to increase the level of productivity and quality and decrease the operation cost (Peurifoy et al. 2010). In road projects, the soil affects the road stability, supports the structure and distributes the forces on the road.

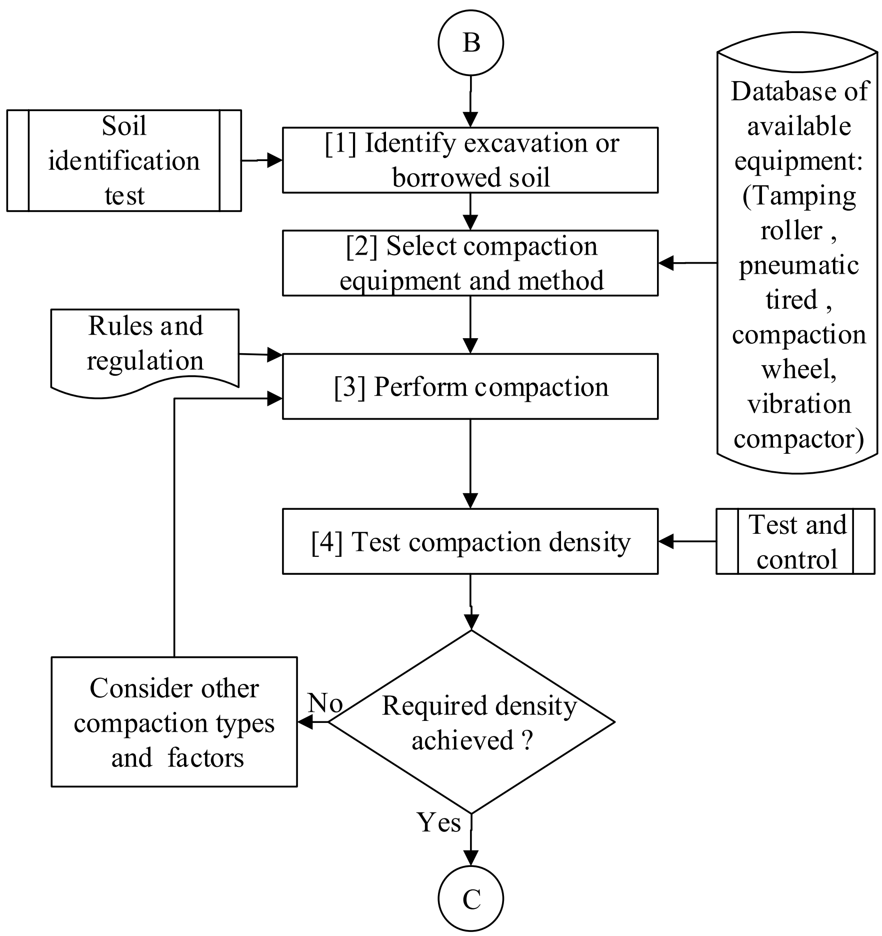


Figure 4-9 Compaction operation

Grading: Graders are used for moving, leveling, spreading, and mixing the soil, which is not considered as a heavy excavation operation. In some operations, other equipment, such as a trimmer, is used to perform the grading operation (e.g., grading the canal slope). Figure 4-10 illustrates the steps to perform the grading operation of layers in road projects, which include: (1) identifying the course soil and the boundaries of the workspace by checking the project plan and design documents; (2) selecting the suitable grading equipment; (3) performing the grading taking into account the rules and regulations that are related to grading operations; (4) performing the compaction process alongside the grading process specially for the course soil layers; (5) testing the moisture level of the soil using different methods, if the level of moisture is not satisfactory, it can be controlled by adding water or leaving the soil to dry; and (6) after the suitable level of moisture is achieved, other specifications are tested (e.g., slop degree, and soil stiffness).

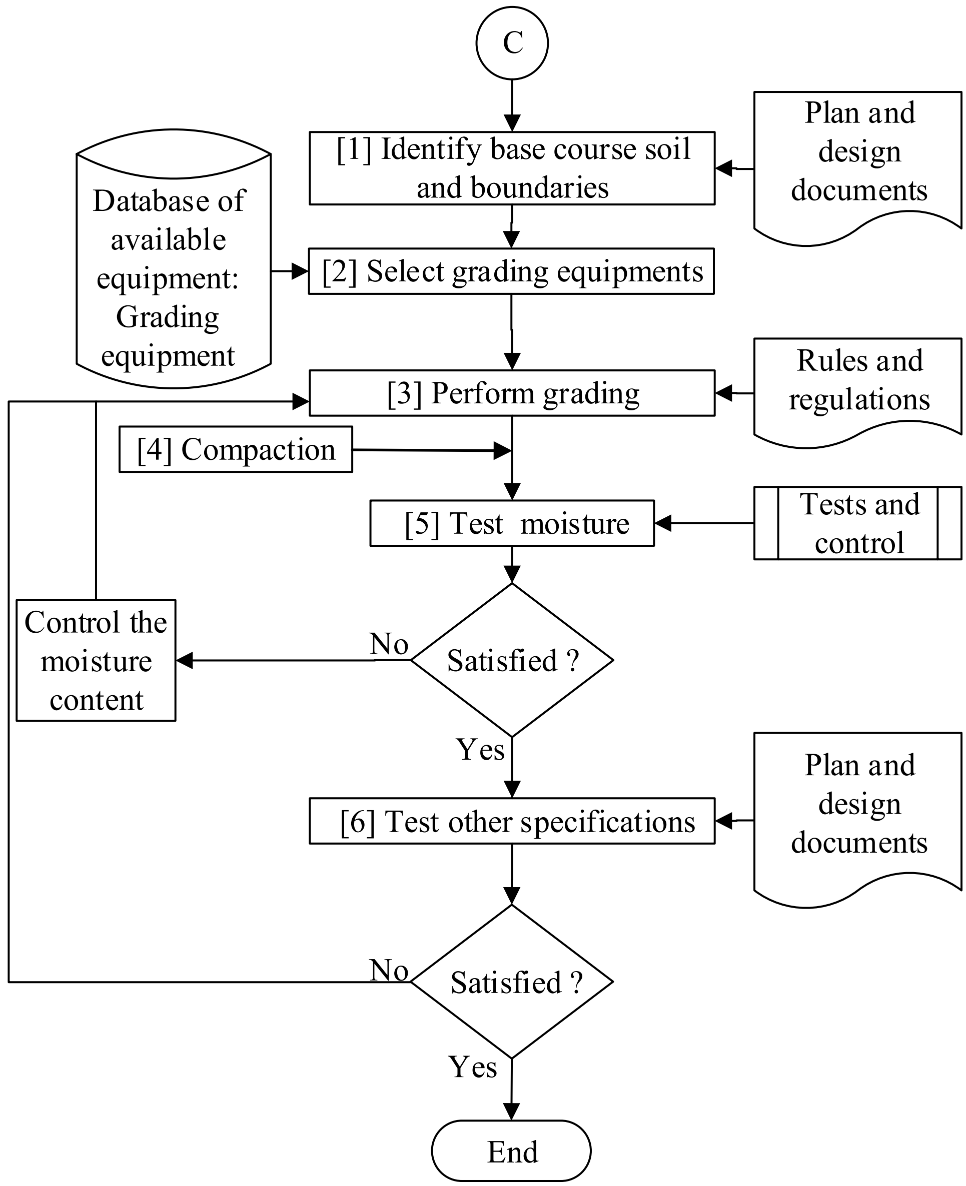


Figure 4-10 Grading operation

4.2.3 Example of Operation Representation in EW-Onto

As an example of representing earthwork operations, related processes, and the resources used to accomplish these processes, Figure 4-11 shows a partial representation of earthwork operations. The figure describes the compaction operation at two levels: the upper level, which is structured as a template, includes the general description of the processes under this operation using the *isNextProcessOf,* and *isPreviousProcessOf* relationships, which link a certain process with the previous and next processes, respectively. It also describes the operation using *hasDescription* relationship. The lower level represents the instances of this class. As an example, *identification soil* process is an instance of *identify excavation or borrowed soil* process at the template level. Moreover, at the instance level, the resources that are used to perform a process are linked through the *useResource* relationship. For example, *Compactor_1*, which is an *instanceOf Tamping roller,* is used to perform the compaction process.

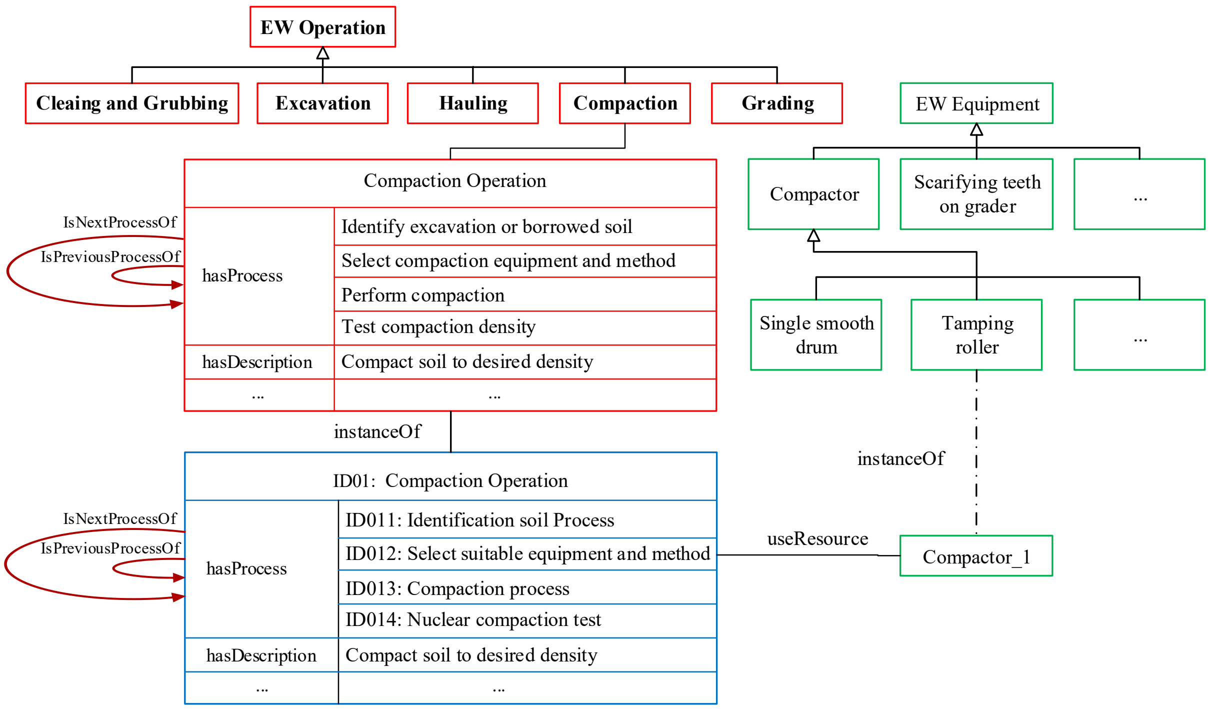


Figure 4-11 Partial presentation of compaction operation

Figure 4-12 shows the extended class diagram of the levels in EW-Onto. This figure illustrates the *project*, *operation*, *process*, *task*, and *micro-task* classes. The upper level presents the operations in earthwork. The next level includes the processes and their instances. Excavation process with id ID014, is an example of an instance of an excavation process. The task level presents the list of tasks, which are under one process (i.e., ID014 Excavation process). Each task has id, which extends the id of the related process (e.g., Digging id is ID0141, and dumping id is ID0142). Micro-tasks level includes all the micro-tasks that are related a task. Each task extends to the list of micro-tasks. For example, the task ID0141 Dig is extended to the micro-tasks: ID01411 swing to digging, and ID01412 dig. At each level, the different processes, tasks, and micro-tasks are linked through the *IsNext* and *IsPrevious* relationships.

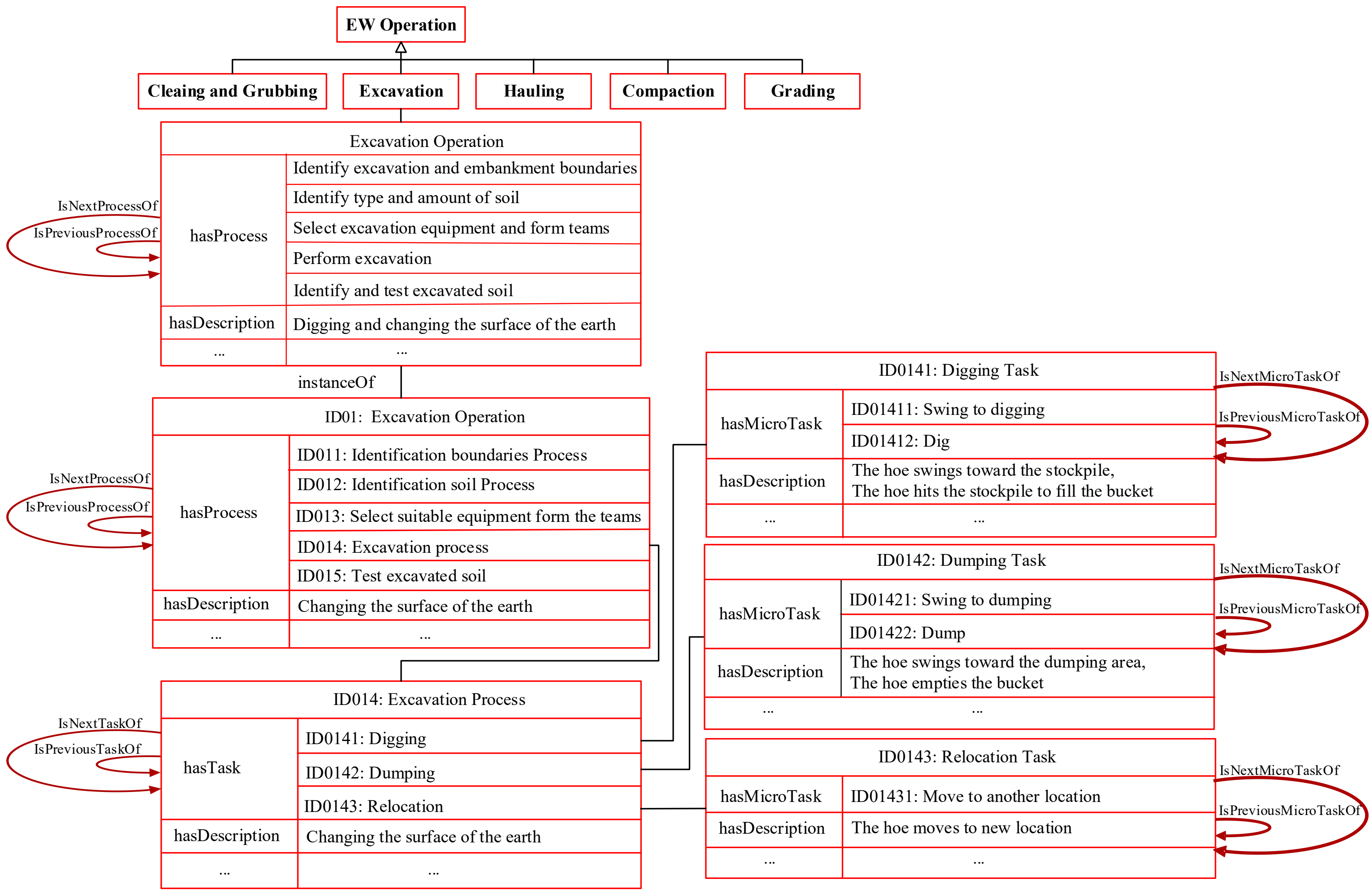


Figure 4-12 Extended diagram of project decomposition levels in EW-Onto

To clarify the need for representing micro-tasks, a hoe is used to explain the various movements that the hoe has, which can generate various potential risks. Table 4-2 shows the tasks, the micro-tasks, and the related risks for the hoe. The table also contains the definition of each micro-task, and the potential risk that may occur when the equipment performs the micro-tasks. For example, a hoe working in the limited workspace (e.g., next to a congested road) needs to avoid the traffic it is moving. These risks need to be considered by different participants in the task (e.g., coordinators, operators, and planners). Providing these details in an explicit formalized way can help avoiding possible accidents and delays.

Table 4-2 Micro tasks and related risk for hoe

Task	Micro-task	Definition	Potential Risks
Digging	Swing to digging	The hoe swings toward the stockpile	Hitting other equipment, moving vehicles, objects (e.g., barrier) or workers-on-foot
	Dig	The hoe hits the stockpile to fill the bucket	Hitting underground utilities (e.g., power lines)
Dumping	Swing to dumping	The hoe swings toward the dumping area	Hitting other equipment, moving vehicles, objects (e.g., barrier) or workers-on-foot
	Dump	The hoe empties the bucket	Hitting other equipment (e.g., truck), objects or workers-on-foot
Relocation	Move to another location	The hoe moves to a new location	Hitting other equipment, objects, workers-on-foot, or utilities

Based on the understanding of the nature of the operations and tasks, there is a set of factors and limitations that affect the selection and the usage of the equipment in earthwork operation, such as availability, the equipment's ability to perform the work, maximizing profit, possibility of using the equipment in the future, and availability of parts and services.

4.3 IMPLEMENTATION OF PROPOSED ONTOLOGY

This section covers the development of EW-Onto to clarify how the ontology components, including the unified equipment classifications and the hierarchies of the Project, Operation, Process, Task, and Micro-Task are implemented in EW-Onto.

Various tools are available for building an ontology. These tools can help define new concepts and relations or extend an existing ontology. *Protégé* and OntoEdit are examples of these development tools. DL is the formalism for the knowledge representation that provides high-level descriptions of the world for intelligent systems (Baader et al. 2003). Web Ontology Language (OWL) is a language designed to give the ability to the applications to process the information in a way that is also readable by humans (McGuinness and Van Harmelen 2004). RDF is a standard model for interchanging the data. RDF provides a description model using the triple form that contains three elements: subject, predicate, and object (Horridge et al. 2009). EW-Onto is available at **https://www.ew-onto.info/**. The RDF file are shown in Appendix A.

In this research, Protégé (Musen 2015) was chosen as the development environment. From a practical point of view, Protégé is a stable, well supported, and updated platform for developing ontologies. Protégé is a free development environment that is used to build the knowledge model for a particular domain. Protégé (version 5.2.0) is used to create and edit EW-Onto including classes, object properties, data properties, instances, and to add the rules. Protégé has a number of free plug-ins that can be added to the environment (e.g., SWRL rule engine, Pellet, and HermiT reasoner) (Ian et al. 2004). Figure 4-13 shows a partial class hierarchy of EW-Onto presented in Protégé.

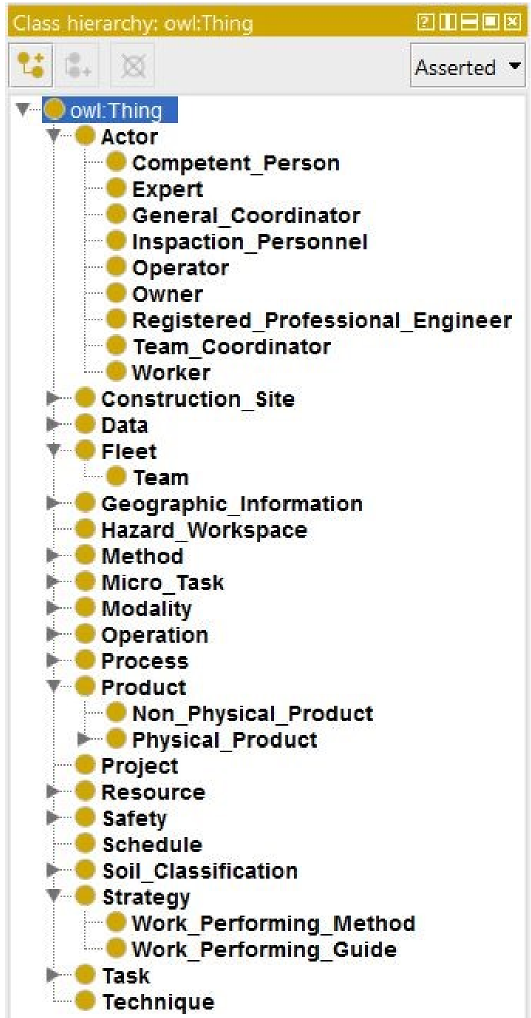


Figure 4-13 Example of EW-Onto class hierarchy in Protégé

Figure 4-14 shows a partial hierarchy of EW-Onto that illustrates the classifications of earthwork equipment.

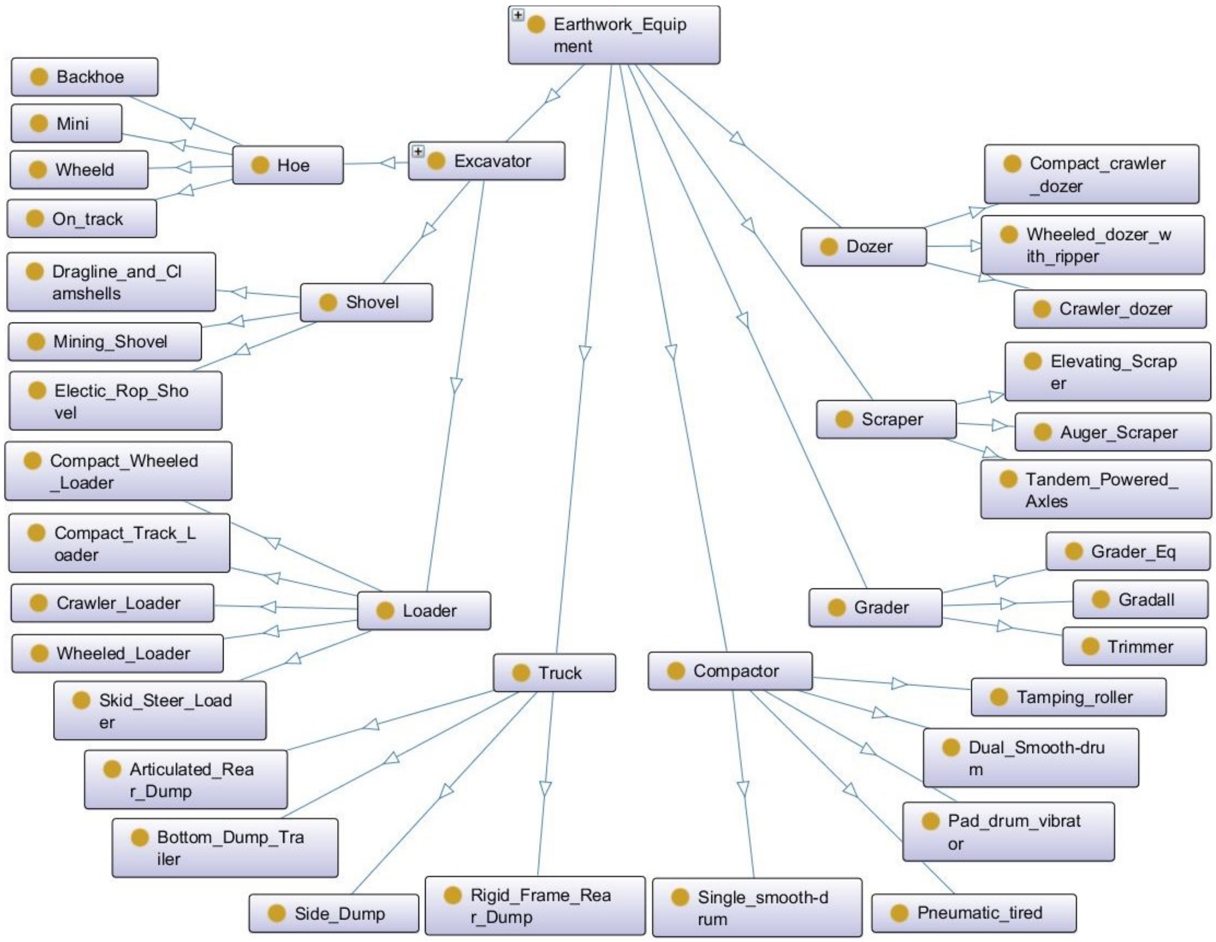
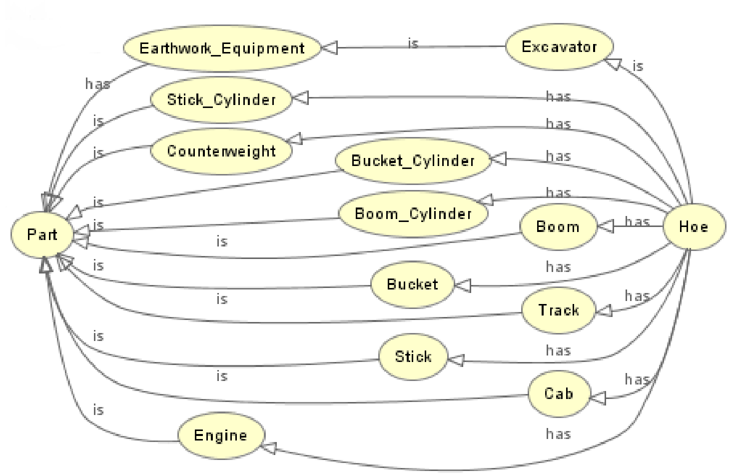
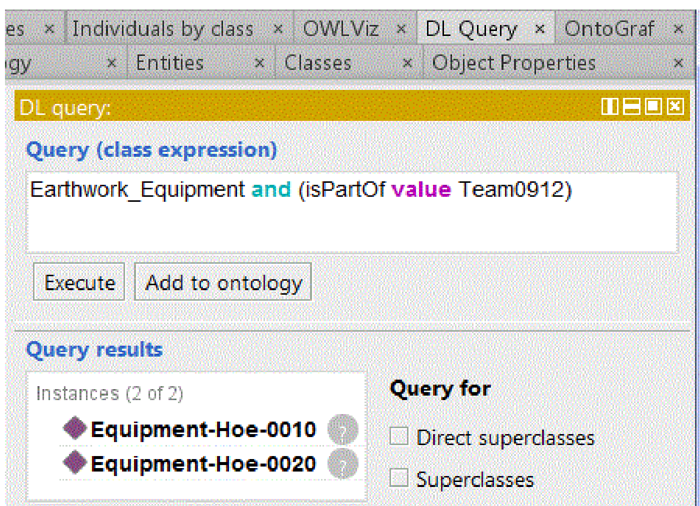


Figure 4-14 Partial hierarchy of EW-Onto for equipment

Figure 4-15(a) illustrates a portion of OWL visualization of the developed EW-Onto related to the equipment. Each node represents a superclass or subclass, and their relationships through the subsumption *is* relationships (e.g., Hoe *is* Excavator). The other relationship is *has* a relationship (e.g., Hoe *has* Boom, and *has* Bucket). In EW-Onto, each subclass may contain more subclasses based on the level of detail and the scope of the ontology. An individual can be created with specific properties (e.g., name, ID, and serial number). Pieces of equipment are related as a team in EW-Onto, the relationships between the equipment and its team can be expressed by *isPartOf* relationship to show the pieces of equipment that are part of a team. Figure 4-15(b) shows the result of the query over the ontology about the equipment that *isPartOf* a specific team, which is Team0912. The result of this query is a list of pieces of equipment, including two hoes (i.e., *Equipment-Hoe-0010* and *Equipment-Hoe-0020*).



(a) Partial view of OWL visualization of EW-Onto



(b) Example of query of equipment involved in a team.

Figure 4-15 Examples of visualization and queries in EW-Onto

SWRL provides a formal human-readable syntax format that can be used to translate the regulations to a machine-readable format. Thus, the regulations that are related to the earthwork domain are written in the SWRL language to represent some definitions of inferred classes or classifications. The following SWRL rule indicates that a scraper is the suitable equipment for earthmoving operation with a distance in the range of 500 ft - 2 miles (152 – 3,219 m).

EarthMoving_Operation(?EMO)^Operation03(?Ope03)^isTypeOf(?Ope03, ?EMO) isTypeOf (?eq,Equipment)^hasDistance(?Ope03, ?dis)^swrlb: lessThan(?dis, "3219"^^xsd:int) ^swrlb: greaterThan(?dis, "152"^^xsd:int)->Scraper(?eq)^Suitable_equipment (?eq,?Ope03)

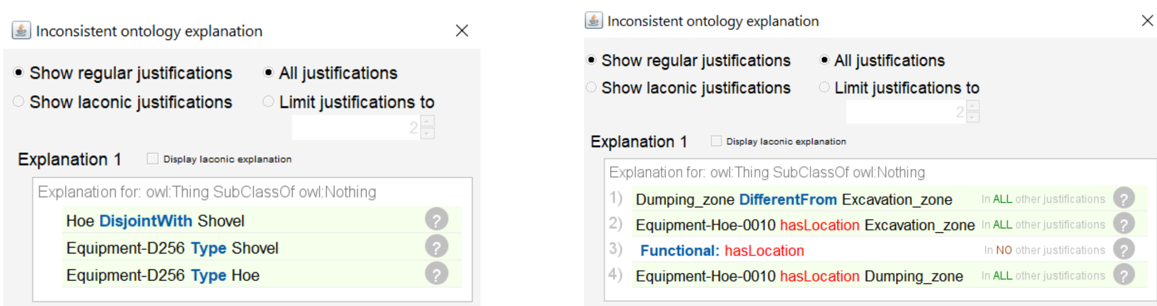
4.4 EW-ONTO EVALUATION

Evaluating the developed ontology is one of the main steps in the ontology development stage. The ontology can be evaluated by the related experts and end-users. The evaluators should check if EW-Onto is able to answer the Competency Questions (CQ) listed in the initial step in "Overview of Ew-Onto Development Workflow" section. Answering these questions in the early stages is an appropriate means of evaluating the developed ontology.

4.4.1 Consistency Checking Using Protégé

Protégé editor contains different description logic reasoners, such as Pellet, FACT++, and HermiT. In protégé, the reasoner is used to perform the verification process and check the consistency for EW-Onto. Pellet checks the relationships amongst the classes, finds the implicit relationships between them, and provides rule support. An example of the class axiom that can be checked by the reasoner is the disjoint classes' axiom, which states that no individual can be at the same time an instance of two classes. Pellet reasoner was used during EW-Onto development stage and gave the explanation of some inconsistencies in the ontology. Figure 4-16(a) shows the explanation results of the inconsistency assertion in EW-Onto. As illustrated in this figure, *Hoe* and *Shovel* are stated as *Disjoin* Classes. Therefore, the reasoner shows an inconsistency when *Equipment-D256* belongs to both classes at the same time. Another example for the relationship inconsistency is when the *hasLocation* relationship is stated as *Function* axiom, which means an individual can be linked with another individual only once using this relationship. Therefore, and as illustrated in Figure 4-16(b), the *Equipment-Hoe-0010 hasLocation Dumping_zone*. The reasoner shows an inconsistency when the same equipment is related with *hasLocation* relationship to another value *Excavation_zone*.

These results were used as feedback and input to the process ID3 to fix them in the development stage before moving to the final stage. Pellet reasoner checks the relationships, axioms, classes and find the implicit subclasses in EW-Onto.



(a) Class inconsistency (b) Relationship inconsistency

Figure 4-16 Examples of the inconsistency results using Pellet reasoner in Protégé

4.4.2 The Survey

The survey aims to evaluate the adequacy of the semantic representation of the concepts, taxonomies, and relationships, focusing on the following criteria: clarity, accuracy, and comprehensiveness. A survey is sent to experts, selected based on their knowledge in construction and familiarity of using information technologies in construction. The survey includes thirteen questions, which are related to the different components of EW-Onto. These questions reflect the concepts' coverage, taxonomies, faceted classifications, and semantic relationships between the classes. They also aim to measure the clarity, comprehensiveness, completeness, usefulness, and accuracy of EW-Onto.

Table 4-3 lists the survey questions and examples of the provided comments. Some of the figures and tables included in this research were provided in the survey to present the ontology to the respondents. A five-point Likert scale is used (except for Q3 and Q5) to get quantitative values of the answers (1 being strongly agree, very clear, or utmost comprehensive and 5 being strongly disagree, not at all clear, or missing a lot of concepts).

Table 4-3 The evaluation questions and examples of the comments

Q.No	The questions
Q1	Name, organization, area of expertise, and years of experience.
Q2	Figure 4-5 shows an example of an earthwork equipment classification. Does this figure capture the concepts that are related to the equipment in an extensive taxonomy? ○ 1 Strongly agree ○ 2 Agree ○ 3 Neither agree nor disagree ○ 4 Disagree ○ 5 Strongly disagree **Comments:** "For earthwork, sheepsfoot and padfoot rollers are often used".
Q3	Figure 4-6(a) and (b) show examples of faceted classifications that can be utilized for hoes and compactors based on different criteria. What other criteria do you think could be used to add more classification, and to which level of detail? **Comments:** - "The fuel type and the weight of the equipment". - "Speed, fuel consumption, and price can be considered"; "The manufacturing company, operation cost"; "Performance aspects". - "For compactors, the level of intelligence could be more specific, i.e., location only, density (stiffness) or both".
Q4	Figure 4-4 and Figure 4-15(a), which are focusing on hoes, show a hoe's main parts and a partial view of the semantic relationships among these parts in EW-Onto as presented in Protégé. Does grouping the different parts as a system (e.g., hydraulic system) facilitate the semantic presentations in EW-Onto? ○ 1 Strongly agree ○ 2 Agree ○ 3 Neither agree nor disagree ○ 4 Disagree ○ 5 Strongly disagree **Comments:** - "I think grouping the parts can add more understanding of the equipment".
Q5	Figure 4-12 shows the extended diagram of project decomposition levels in EW-Onto. To which level of detail should EW-Onto be designed? ○ Operation level ○ Process level ○ Task level ○ Micro-Task level **Comments:** - "The level of detail is depending on the purpose of the ontology". - "I think all levels are necessary, but applicable to different stakeholders". - "Desired LOD depends on the use case".

Table 4-3 The evaluation questions and examples of the comments (Continued)

Q.No.	The questions
Q6	Table 4-2 shows the tasks, Micro-Tasks, and the source and potential risks for a hoe. Do you agree that it is appropriate to allocate the risks at Task and Micro-Task levels to better manage the safety of construction sites? o 1 Strongly agree o 2 Agree o 3 Neither agree nor disagree o 4 Disagree o 5 Strongly disagree **Comments:** - "At these levels, the risks are more likely because of the human intervention". - "This detailed risk allocation to tasks and micro-tasks should not affect the simplicity of use".
Q7	Figure 4-2 shows the high-level structure of EW-Onto using UML Class Diagram. The Figure captures the main classes that represent the key concepts in earthwork domain based on the literature review. Regarding the concepts and the relationships between these concepts, do you find this representation clear? o 1 Very clear o 2 Clear o 3 Somewhat clear o 4 Not so clear o 5 Not at all clear **Comments:** No comments
Q8	For the same previous Figure (Figure 4-2), regarding the concepts and the relationships between them, do you find this representation comprehensive? o 1 Utmost comprehensive o 2 Comprehensive o 3 Somehow comprehensive o 4 Not comprehensive o 5 Missing a lot of concepts **Comments:** - "For the main concepts and relationships, I think that UML is comprehensive".
Q9	After you check the concepts and the relationships presented by EW-Onto. Does the ontology include the relevant concepts and their lexical representations? o 1 Strongly agree o 2 Agree o 3 Neither agree nor disagree o 4 Disagree o 5 Strongly disagree **Comments:** - "In the earthwork domain, yes. For the construction domain, ontology need more information to add". - "Ontology needs to link with other ontologies related to the domain. No one ontology can cover everything in the domain".

Table 4-3 The evaluation questions and examples of the comments (Continued)

Q.No.	The questions
Q10	Do you agree that EW-Onto captures and accurately represents the knowledge in the domain? ○ 1 Strongly agree ○ 2 Agree ○ 3 Neither agree nor disagree ○ 4 Disagree ○ 5 Strongly disagree **Comments:** - "It is easy to locate the concepts in the hierarchy".
Q11	Does EW-Onto represent the concepts and relationships in a way that can be used in applications in the earthwork domain? ○ 1 Strongly agree ○ 2 Agree ○ 3 Neither agree nor disagree ○ 4 Disagree ○ 5 Strongly disagree **Comments:** - "I think the ontology can be integrated with databases to develop smart applications".
Q12	Do you agree that integrating EW-Onto with other computerized systems, such as Multi-Agent Systems (MAS), will enhance the communications between the different disciplines to improve safety and productivity in earthwork operations? ○ 1 Strongly agree ○ 2 Agree ○ 3 Neither agree nor disagree ○ 4 Disagree ○ 5 Strongly disagree **Comments:** - "Also, to support distributed decision support systems". - "Ontology can be used for more applications to improve resource allocation, quality, and supply chain".
Q13	Some examples of OSHA safety regulations are translated to axioms in EW-Onto. Do you agree that adding these regulations will improve the usage of EW-Onto in safety applications? ○ 1 Strongly agree ○ 2 Agree ○ 3 Neither agree nor disagree ○ 4 Disagree ○ 5 Strongly disagree **Comments:** No comments

Table 4-4 shows the respondents' profiles. The 40 respondents have a total of 292 years of experience in construction and a total of 160 years of experience in information technology and ontology research.

Table 4-4 The respondents' profiles

Number of respondents	Areas of expertise	Years of experience (total)
3	Civil engineering and construction management	49
2	Facility management	32
6	Construction	66
4	Architecture	65
6	Project management	68
15	Information technology applications in earthwork	145
1	Asphalt process control	12
3	Ontology researcher	15

Table 4-5 lists the results of the survey answers. Questions Q2 to Q9 are about the classifications of the equipment, the concepts representation in EW-Onto, and the relationships between them. For Q2, 94.7% of the respondents strongly agree or agree with the classification of the equipment in EW-Onto. Q3, which is related to the faceted classification, received different comments. The comments emphasized adding other classification criteria, such as weight, fuel consumption, and cost. For Q4, 89.4% of the respondents strongly agree or agree that EW-Onto is providing the semantic representation through the relationships between the concepts. For Q5, which assesses the respondents' different perspectives on the project composition levels, most answers selected the task level (32.4%), followed by the micro-task level (29.4%), then the process level (20.6%) and the operation level (17.6%). Q6, which is related to safety risk allocation at different levels (i.e., tasks and micro-tasks), got 84.2% strongly agree or agree responses. The answers emphasize that the risks should get appropriate allocation at task and micro-task levels. Q7 about the clarity of concepts and the relationships between them in EW-Onto got 63.1% very clear or clear responses. Q8 about the comprehensive coverage of the concepts and the relationships in EW-Onto got 73.7% utmost comprehensive or comprehensive responses. The responses show that there are some significant values of standard deviation for Q7 and Q8, which could be explained by the lack of familiarity with ontology aspects. Q9 is asking if EW-Onto presents the lexical values of the concepts in the domain. The question got 81.9% strongly agree or agree responses. The comments highlight that linking EW-Onto with other related ontologies could add more information.

Questions Q10 to Q13 aim to evaluate the accuracy and usefulness of EW-Onto. Q10 about the accuracy of the representation of concepts and the relationships in EW-Onto received 95.4% strongly agree or agree responses. Q11 about the usefulness of EW-Onto in developing applications got 91% strongly agree or agree responses. In Q12, the experts were asked if they agree that integrating EW-Onto with other computerized systems, such as Multi-Agent Systems (MAS), will enhance the communications between the different disciplines to improve safety and productivity in earthwork operations. Q12 got 95.5% strongly agree or agree responses. The respondents highlighted the different applications where EW-Onto can be used, such as, distributed *Decision Support System* (DSS), supply chain, resource allocation and the applications related to quality monitoring. Q13, which is also about the usefulness of the EW-Onto in safety applications, got 100% strongly agree or agree responses. The graphical representation of the results is shown in Appendix B.

Table 4-5 Distribution of the responses

Q. No.	Ave.	SD	Results				
Q2	1.84	0.49	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly Disagree
			21%	73.7%	5.3%	0%	0%
Q4	1.74	0.64	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly Disagree
			36.8%	52.6%	10.6%	0%	0%
Q5	NA	NA	Operation Level	Process Level	Task Level	Micro Task Level	
			17.6%	20.6%	32.4%	29.4%	
Q6	1.58	0.75	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly Disagree
			57.9%	26.3%	15.8 %	0%	0%
Q7	2.05	0.94	Very clear	Clear	Somewhat clear	Not so clear	Not clear at all
			36.8%	26.3%	31.6%	5.3%	0%
Q8	2.16	0.74	Comprehensive	Somehow Comprehensive	Utmost Comprehensive	Not Comprehensive	Missing lots of concepts
			57.9%	21%	15.8%	5.3%	0%
Q9	2.14	0.46	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly Disagree
			4.6%	77.3%	18.1%	0%	0%
Q10	1.77	0.52	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly Disagree
			27.3%	68.1%	4.6%	0%	0%
Q11	1.68	0.63	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly Disagree
			41%	50.0%	9 %	0%	0%
Q12	1.64	0.57	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly Disagree
			40.9%	54.6%	4.5%	0%	0%
Q13	1.50	0.5	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly Disagree
			50%	50%	0%	0%	0%

4.5 DISCUSSION ABOUT THE RESULTS OF THE SURVEY

The hypotheses mentioned in the introduction have been verified through the questionnaire. Table 4-6 shows each hypothesis and the related questions. The answers to these questions have high positive values. As illustrated in the table, the first hypothesis is about the concepts and the relationships in EW-Onto and the level of acceptance by the experts and the end-users. The answers show that EW-Onto is representative of the concepts and the relationships of the earthwork domain, with some room for further extensions for EW-Onto coverage by adding more concepts and relationships. The second, third, and fourth hypotheses are about the practical implications and the benefits of EW-Onto. The answers show high acceptance of the practical implications and benefits that EW-Onto provides. These answers emphasize that EW-Onto constitutes a core earthwork ontology that can be used as a basis for developing further applications for the management of earthwork operations.

Table 4-6 The hypotheses and the related questions' results

Expected benefits of EW-Onto	Related questions in the survey (% of positive evaluation)
Link and identify the relationships between concepts, define earthwork semantics, and classify knowledge in a hierarchical way accepted by experts and end-users	Q2: Classification of the equipment (94.7%) Q4: Semantic representation (89.4%) Q7: Clarity of concepts and the relationships (63.1%) Q8: Comprehensive coverage (73.7%) Q9: Lexical values of the concepts (81.9%) Q10: Accuracy of the representation (95.4%)
Facilitate the management of earthwork operations and simplify information exchange and interoperability between currently fragmented systems	Q12: Integrating EW-Onto with other computerized systems, such as MAS, will enhance the communications between the different disciplines to improve safety and productivity in earthwork operations. (95.5%)
Increase the stakeholders' knowledge of earthwork operations through the provision of the information, which is structured in the context of robust knowledge	Q5: Project composition levels (N.A.) Q6: Safety risk allocation (84.2%)
Help developing platforms for easy integration of various types of data towards different goals.	Q11: Usefulness in developing applications (91%) Q13: Usefulness in developing safety applications (100%)

4.6 SUMMARY AND CONCLUSION

The communication issues and the relationships between the entities in the project play a significant role in earthwork projects. The developed EW-Onto defines the concepts and relationships in the earthwork domain. The conceptual ontology elements and the different classifications of equipment in this domain are presented. The hierarchies in EW-Onto, which are related to the resources (e.g., equipment) and the different project levels (i.e., operations, processes, tasks, and micro-tasks) are built. The scope of EW-Onto is explained to illustrate the boundaries of related technologies that will benefit EW-Onto.

Based on the literature review, it was found that there is no ontology focusing on the earthwork domain. The development of EW-Onto started from defining the concepts and building taxonomies for earthwork operations and equipment following the METHONTOLOGY approach. In addition, several rules have been extracted from safety codes and implemented as SWRL rules. The ontology has been implemented using Protégé. The consistency of EW-Onto has been checked and it has been evaluated using a survey.

The following conclusions can be stated: (1) The METHONTOLOGY approach was effective in the development of EW-Onto; (2) The results of the evaluation show that EW-Onto was able to give a clear, accurate, and comprehensive understanding of the concepts, constraints, axioms, and relationships in the domain; and (3) The respondents provided favorable evaluation of EW-Onto in developing practical applications by integrating various types of knowledge.

CHAPTER 5 EXTENDING EARTHWORK ONTOLOGY TO ENHANCE OPERATION SAFETY

5.1 INTRODUCTION

Ensuring workers and equipment safety is a vital concern in the construction domain. According to the Occupational Safety and Health Administration (OSHA) (OSHA 2020a), about 21.1% of work fatalities in 2016 occurred in construction. Ignoring safety regulations, weather, and reckless equipment operators are the main factors that lead to accidents on construction sites (Williams et al. 2018). In addition to the loss of lives, these accidents affect all aspects of the construction work, including schedules, productivity, costs, and the reputation of construction firms. In earthwork operations, which account for 20% of the total cost of road-building projects (Smith et al. 1996), the most hazardous operation is excavation, especially trenching, where accidents include cave-ins, toxic atmospheres, and falls (OSHA 2020b).

To reduce the occurrence of construction accidents, OSHA (OSHA 2020c) has developed a technique called Job Hazard Analysis (JHA) to identify, evaluate, and control these types of hazards (Zhang et al. 2015). It is one of the various methods used to check if the different variables related to workers, tools, equipment, and the environment are according to regulations and rules. The data collected from construction sites using different technologies can enhance construction site safety. Combining human experience and best practices is another way of avoiding accidents. It is necessary to link the hazards at different levels of the project with other information about the construction site to improve decision-making related to safety, including the products, equipment, and surrounding environment.

In recent years, ontologies have been applied to give a formal structure to knowledge and to integrate a variety of domain knowledge to improve cross-functional developments. As mentioned in Chapter 4, the author has previously developed a comprehensive Earthwork Ontology (EW-Onto) to support and enhance the communication and provide knowledge about the resources (e.g., excavators, trucks, compactors) and operations (e.g., excavation, hauling, compaction) in the earthwork domain (Taher et al. 2017). In order to use EW-Onto as the knowledgebase for developing the next generation of decision-support systems for earthwork safety management, it is necessary to extend it to cover safety-related regulations, sensing technologies, and soil properties (Hammad et al. 2012). Therefore, this research aims to: (1) Extend EW-Onto to enhance operation safety by adding rules based on safety regulations and integrating related concepts from sensor and soil ontologies, and (2) Evaluate the integrated ontology using data-driven and application-based approaches. The new ontology, called Integrated EW-Onto (IEW-Onto), should describe the concepts related to earthwork safety and their relationships explicitly and unambiguously so that different stakeholders can reuse the captured knowledge in a formal language. For example, an operator-support system can be developed, which enhances safety by applying safety rules and regulations, taking into consideration the variables of the site (e.g., equipment locations and speeds). The rest of this chapter is organized as follows: The proposed framework is presented in Section 5.2. Section 5.3 covers the IEW-Onto implementation. Section 5.4 discusses the evaluation of IEW-Onto. Finally, Section 5.5 discusses the summary, and conclusions.

5.2 PROPOSED FRAMEWORK

5.2.1 Elements of IEW-Onto

As shown in Figure 5-1, IEW-Onto extends EW-Onto by adding the knowledge and rules related to safety (e.g., OSHA Regulations (OSHA 2020a)), and integrates concepts and relationships from available soil ontology (Du et al. 2016) and sensor ontology (Compton et al. 2012). In addition, some new concepts and relationships are defined to enhance the knowledge representation in the earthwork safety domain based on the available literature. For example, specific types of sensors used for tracking equipment are added, as will be explained in Section 5.2.3. Each component of IEW-Onto is briefly explained in the following.

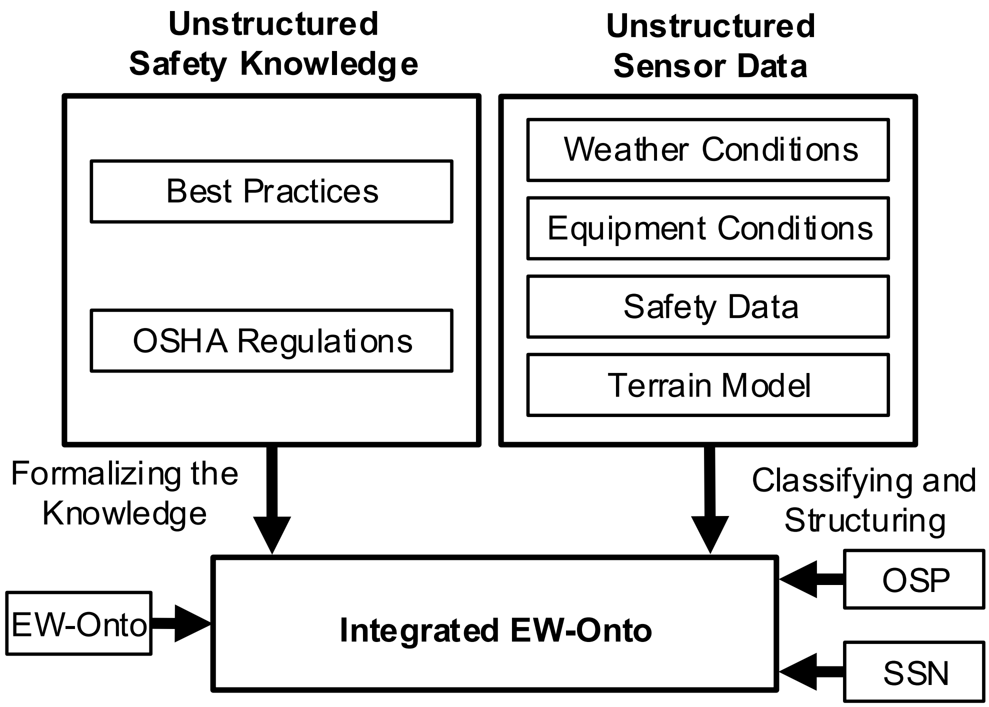


Figure 5-1 Integrating ontologies and knowledge in the development process

(a) Unstructured safety knowledge: Unstructured knowledge related to safety regulations is extracted from OSHA documents (OSHA 2020a) and best practices in the earthwork domain such as (CCGA 2020). SWRL is used to express these regulations as formal rules. The following rules show some examples. The pseudocode and SWRL implementation are listed in Section 5.3.2.

Rule 1. Soil classification: OSHA standard number 1926 (OSHA 2020d) provides the guidelines for classifying the soil based on various properties. This rule is an example of classifying the soil based on these quantitative properties, obtained from sensors or lab tests. Soil classification is used to link to other rules related to hazards in the workzones. The rule checks if the structure of the soil is cohesive, the silt and clay percentage are higher than 15%, and the Unconfined Compressive Strength (UCS) value is higher than or equal to 1.5 Ton per Square Foot (TSF), then the soil is classified as *Type A*.

Rule 2. Cave-in hazard: This rule is derived from OSHA regulations (OSHA 2020a). This rule checks if there is an indication of potential cave-in hazards in the workzone and a need for protection systems. The rule checks if the depth (*d*) of workzone (*wz*) is greater than 153 cm to classify the workzone as a hazard workzone and link it to the type of the hazard (i.e., cave-in). Moreover, the workzone is categorized as a workzone that needs a protection system. Also, the excavation operation is classified as has a hazard.

Rule 3. Workzone with multiple layers of different soil types: OSHA regulations for hazard recognition in trenching and shoring (OSHA 2020c) provides the guidelines to apply slops in trenches. The rule checks if the depth (*d*) of workzone (*wz*) is greater than 367 cm and less than 609 cm and has multiple layers of different soil types (e.g., soil type B over soil type A). The rule specifies the slop degree for each layer of soil. Moreover, the workzone is categorized as a workzone with the hazard (i.e., cave-in) and needs a safety procedure, which is sloping excavation. Also, the excavation operation is classified as has a hazard.

The following rules are related to hauling operation safety.

Figure 5-2 illustrates a hazardous situation associated with three trucks performs their tasks (i.e., hauling and return from the dumping zones) and the uncontrolled intersection in their paths. The safety rule is about who gets the priority and right-of-way and who needs to stop or slow down at the intersection. The assumption here is that the trucks are equipped with sensors (e.g., GPS and RTLS), which indicate the locations of the trucks. In this scenario, the following two rules apply when the trucks enter the warning range (Zhao et al. 2017).

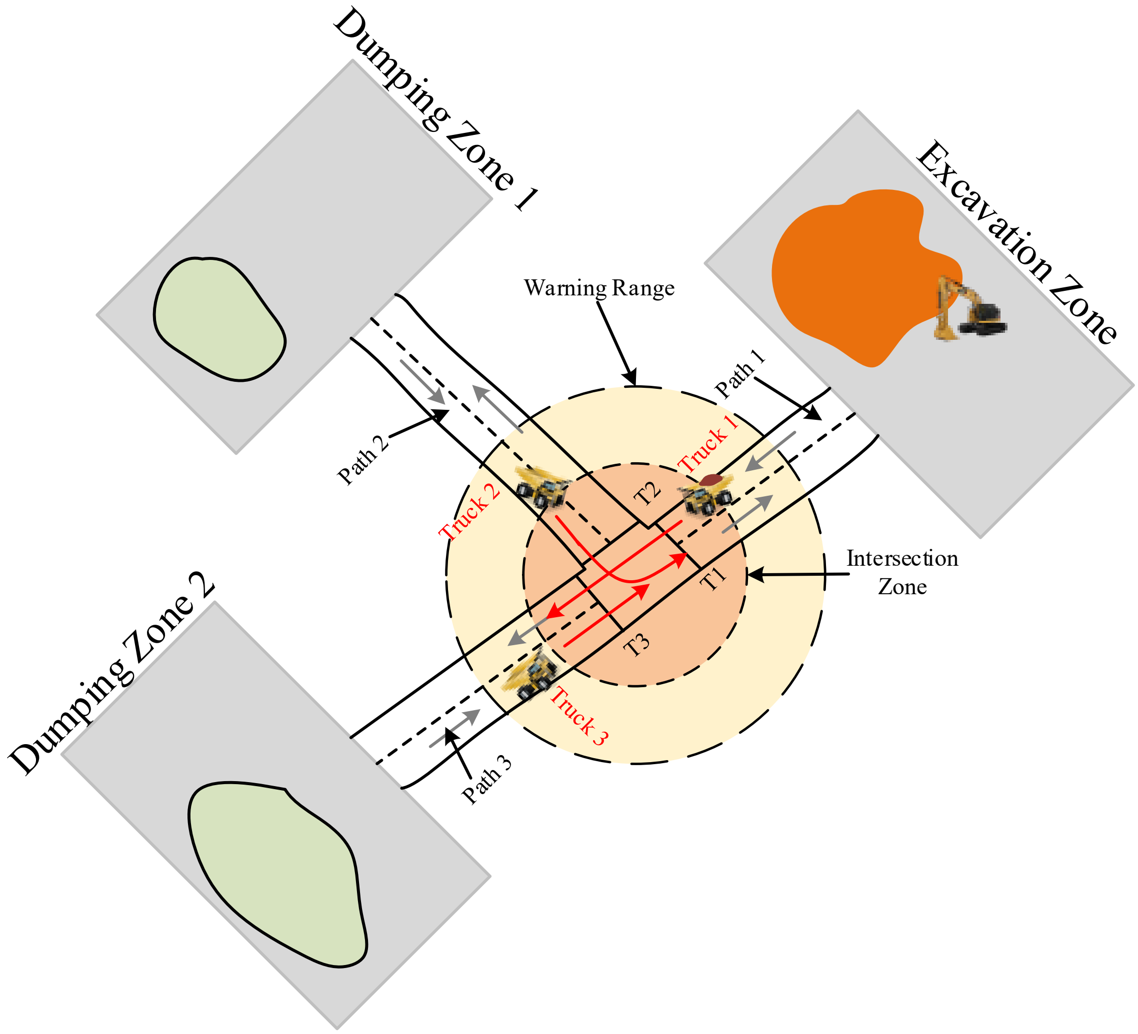


Figure 5-2. Example of an uncontrolled intersection

Rule 4. Truck priority: This rule eliminates the confusion about who gets the priority at the uncontrolled intersection. The rule states that the truck which carries a load and is already at the intersection has the priority over the other trucks to proceed.

Rule 5. Truck collision avoidance: This rule illustrates an example of the orders that trucks receive based on the situation of a truck with the priority. In this rule, Truck 1 is loaded and its direction is straight. It is located at the intersection and has the priority based on Rule 4. Thus, Truck 2, which is located at the intersection point on Path 2 (i.e., T2) and it is turning left, receives the warning about the possibility of collision and is ordered to stop. In the case that there is another truck on Path 3, as illustrated in Figure 5-2, and it is in the warning range, this truck receives an order to stop or slow down based on its direction. For example, Truck 3, which is located on Path 3, receives the warning about the possibility of collision and is ordered to stop or slow down.

(b) Unstructured sensor data: A verity of sensory data can be collected from an earthwork site. This data includes: (1) Weather conditions, such as wind speed and direction, humidity level, temperature, snow, and rain, affect the schedule and the performance of the equipment. Thus, the safety and productivity of the project are affected. Weather conditions can be obtained from the meteorological forecast provider (UBIMET 2020); (2) Equipment conditions, which show the internal status of the equipment, such as fuel level, temperature, and the hydraulic system performance. These data can be obtained from sensors attached to the equipment; (3) Safety data, such as the equipment speed and proximity between equipment and workers, which can be obtained using RTLS or CV technologies; (4) Terrain model, including the changes in the terrain and update from the site. This data can be obtained from different resources in different formats, such as LandXml file form total stations, point cloud from LiDAR, and video and images from cameras and drones; and (5) Soil conditions including the properties, which can be obtained from sensors or lab tests and affect the classification of soil (e.g., moisture level and density). The data should be structured and saved in a database to be available for queries from the stakeholders of the project. This data is usually represented using specialized software, and it may not be understandable for all stakeholders in the project. Providing a structured representation of this data facilitates data sharing among stakeholders.

(c) Structured Knowledge: Taxonomies, concepts, and relationships representing the knowledge about soil, and sensors are borrowed from the soil ontology and sensor ontology and integrated with EW-Onto. The integration process is explained in Section 5.2.3.

5.2.2 IEW-Onto Development Processes

Figure 5-3 shows the processes of the development methodology of IEW-Onto. The methodology is adapted from Chapter 4. The methodology has three stages: the initial, development, and final stages. It should be noted that IDEF has also been used in the development of IEW-Onto because of its simplicity (Taher et al. 2017). The following paragraphs summarize these processes.

(a) Initial stage: The initial stage of the ontology development comprises two processes: **(1)** ***Process ID1: Defining the scope of IEW-Onto based on the requirements.*** As mentioned in Chapter 3, the scope of IEW-Onto addresses the target users (e.g., safety system developers, safety managers), which will limit or extend the concepts and the relationships included in the final ontology. It also gives an idea of the size of the development and the level of detail that should be covered in IEW-Onto. **(2)** ***Process ID2: Defining concepts and taxonomies for IEW-Onto.*** In this step, the related knowledge (i.e., concepts, relationships, and taxonomies) is gathered to construct IEW-Onto. The mapping method controls which components of the candidate ontologies should be selected and included in IEW-Onto. The structured and unstructured knowledge mentioned in Section 5.2.1 are used as input to this process.

(b) Development stage: The development stage includes two main processes: **(1)** ***Process ID3: Developing IEW-Onto.*** This process begins with the defined components from the preceding stage to create and formalize the conceptual model of IEW-Onto. Mapping the ontologies using the integration process is performed to create the initial IEW-Onto. Section 5.2.3 explains the details of the integration process. **(2)** ***Process ID4: The verification of IEW-Onto aims to evaluate the ontology content from a technical perspective (e.g., concepts, relationships, taxonomy, and scope).*** This process starts with performing the consistency checking using the Reasoning Engine (RE) to check for any conflicts and validate the relationships.

(c) Final stage: The final stage comprises four processes: ***(1) Process ID5: Improving and extending IEW-Onto by adding the earthwork safety rules from OSHA and other sources.*** OSHA rules are translated to SWRL, as explained in Section 5.2.1. These regulations are used as constraints when formalizing the relationships between the concepts. **(2)** ***Process ID6: Improving relationships.*** The experts and end-users may either recommend new relationships or adjust the existing ones to improve IEW-Onto. **(3)** ***Process ID7: Validating the ontology.*** IEW-Onto validation aims to prove that it complies with the requirements using a data-driven approach and application-based approach mentioned in Chapter 2. The validation process is covered in detail in Section 5.4 **(4)** ***Process ID8: Documentation.*** This process aims to document all the previous steps to deliver the ontology.

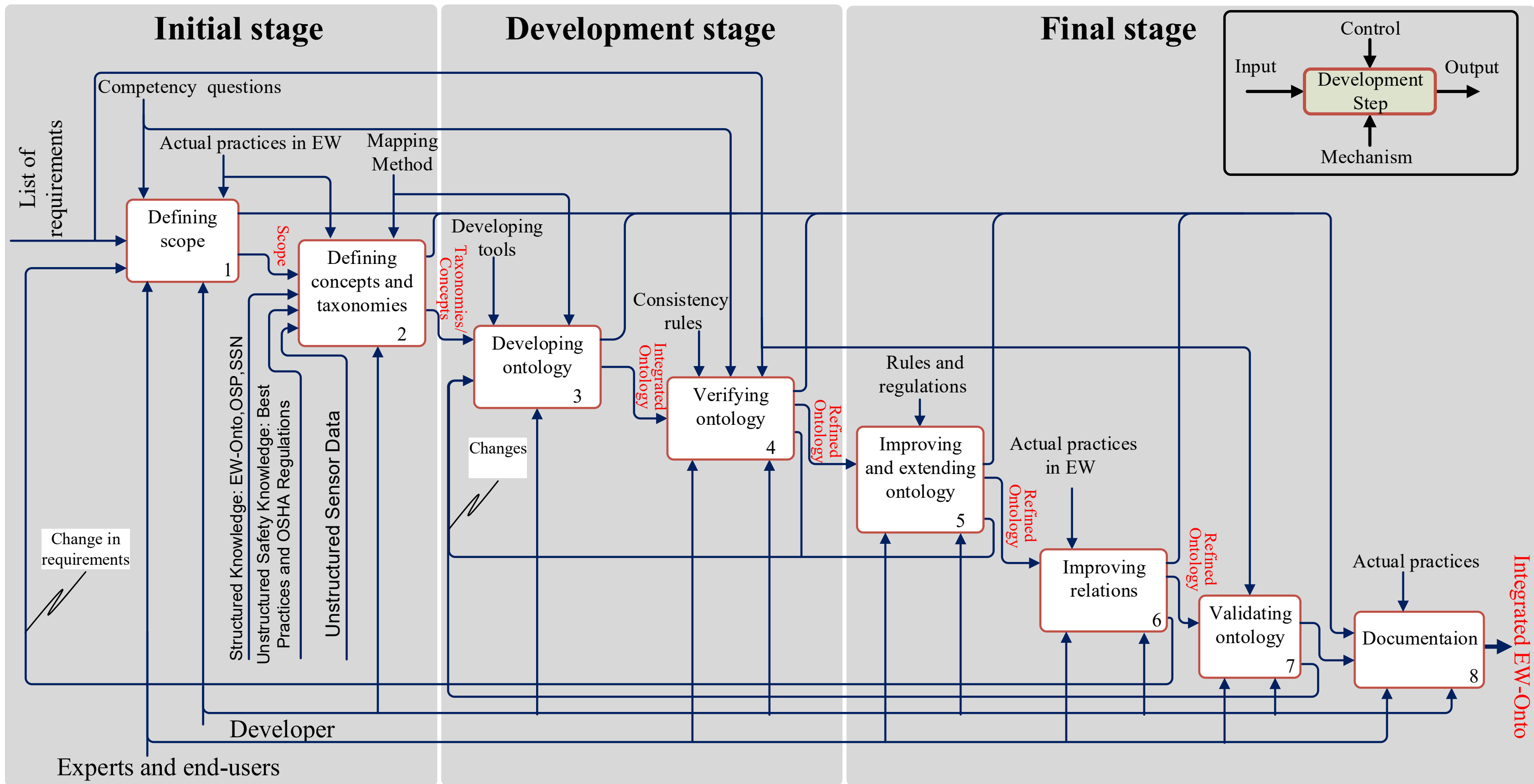


Figure 5-3 IEW-Onto development methodology

5.2.3 Integration Process of IEW-Onto

The scope of the IEW-Onto falls in formalizing and representing the safety knowledge in the earthwork domain, including the integration with the related available ontologies. The development of IEW-Onto requires combining and reusing the knowledge from these related ontologies. Though the IEW-Onto aims to be general and extensible, in order to control the scope, the hazards in earthwork (i.e., in workzones and at uncontrolled intersection) were selected to demonstrate the effectiveness of IEW-Onto. This section focuses on the integration process of IEW-Onto, which aims to link the concepts related to earthwork, soil, sensors, and safety regulations. Figure 5-4 shows an example of the mapping between the elements in IEW-Onto: A Hoe *has* a device, which is a sensor. The sensor is RTLS, which is BLE, and *has* a Tag. This tag is *attachedTo* to the boom of the hoe. The hoe *performsAt* ExcavationZone that *has* a Workzone. This workzone *hasSoilType* SoilType-A, which *has* SoilStructure. Furthermore, monitoring soil properties can be done using sensors to improve safety.

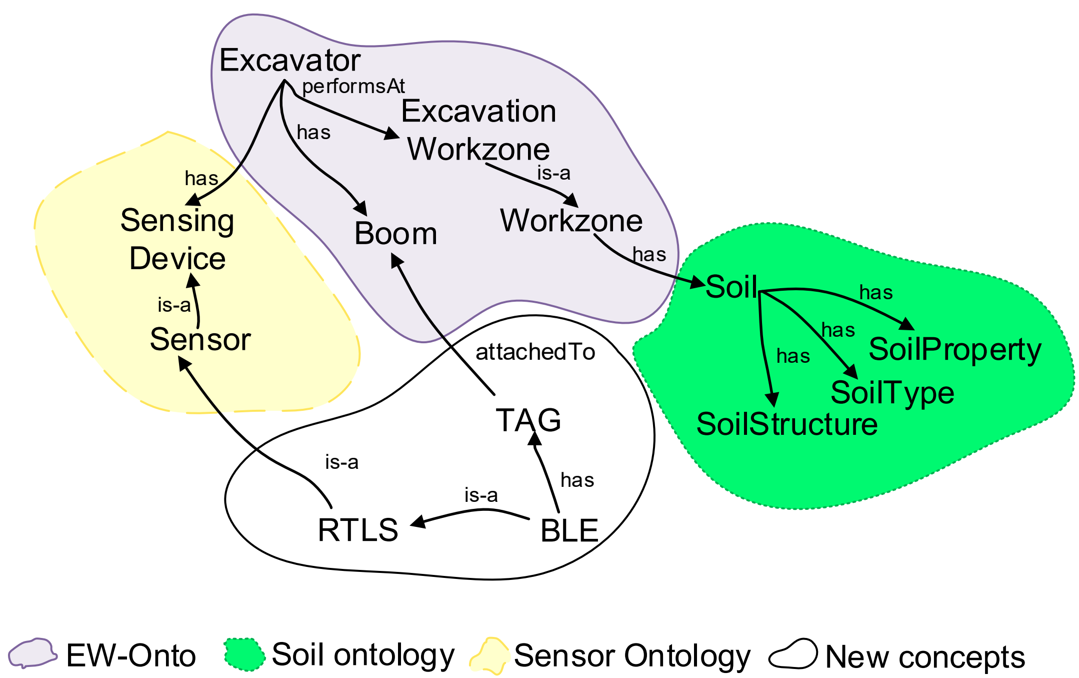
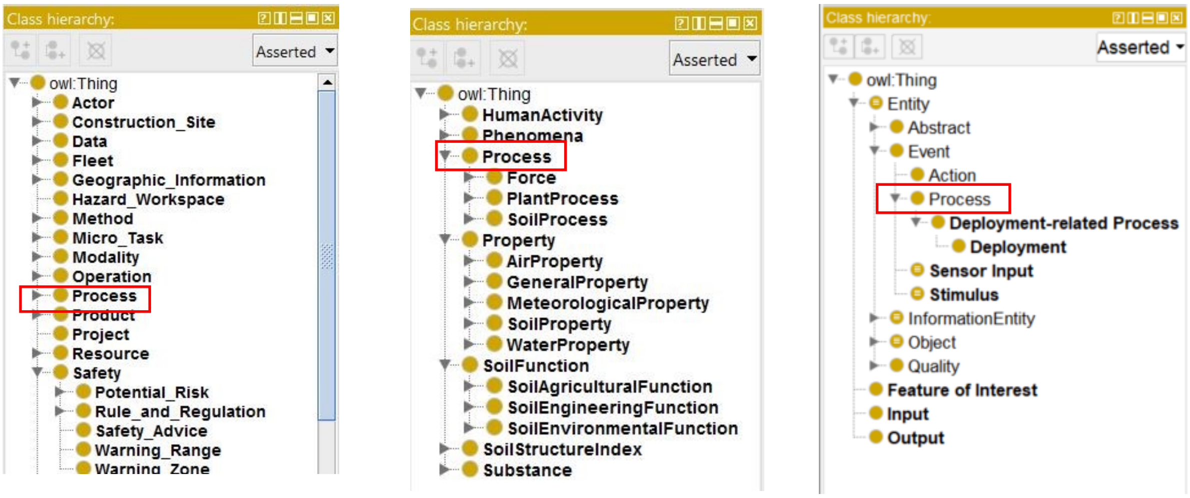


Figure 5-4 An example of concept mapping in IEW-Onto

As mentioned in Section 2.8.3, there are different methods to reuse ontologies. The ontology integration method is selected in this research because it gives more flexibility to map the existing concepts in each candidate ontology to fit into IEW-Onto. The next paragraphs explain the integration steps.

(a) Identifying candidate ontologies: The following candidate ontologies for the integration are identified based on the concepts and relationships that are needed in the final IEW-Onto: (1) Ontology of Soil Properties and Processes (OSP) (Du et al. 2016), which covers the processes (e.g., soil compaction) that lead to changes in the soil properties, and (2) Semantic Sensor Network (SSN) (Compton et al. 2012), which includes knowledge about the physical properties of sensing devices, observations, and management processes. SSN includes the taxonomy, definitions of concepts, and properties adapted from the available standards (i.e., SensorML (Botts and Robin 2007) and Observation and Measurements (O&M) (Cox 2007)). The main classes of SNN are Event, Input, Output, and Object. SSN is more expressive than other sensor ontologies, such as OntoSensor (Russomanno et al. 2005) and CSIRO (Neuhaus and Compton 2009). The abovementioned two candidate ontologies are available in Web Ontology Language (OWL) format, which facilities the integration process.

(b) Analyzing the candidate ontologies: In this step, the candidate ontologies are evaluated from different perspectives, such as the overall structure, concepts, relationships, and quality and clarity of definitions. For example, similar concepts with identical terms should be identified and distinguished from each other during the integration process. Figure 5-5 shows examples of the taxonomies of EW-Onto, OSP, and SSN. As illustrated in the figure, the same term *process* appears in the three ontologies. These replications will lead to inconsistent representation in IEW-Onto because they refer to specific concepts in three contexts. As shown in Figure 5-6(a), the *process* concept in EW-Onto is related to the earthwork context as an intermediate activity between *operation* and *task* in the hierarchy of the ontology.



(a) EW-Onto (b) OSP (c) SSN

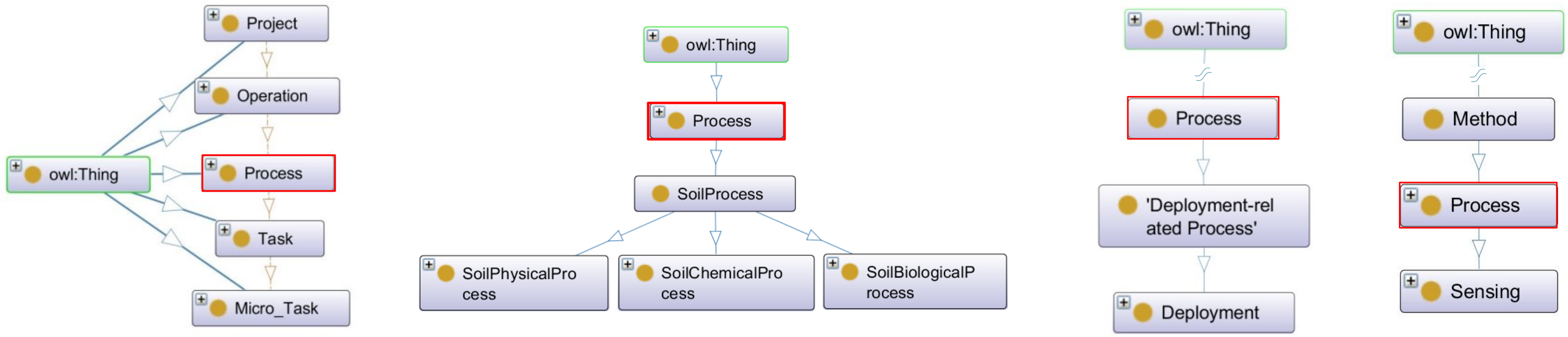
Figure 5-5 Example of the taxonomies of EW-Onto, OSP, and SSN

As shown in Figure 5-6(b), the *process* in OSP is the superclass of the *SoilProcess class,* which is linked to different soil processes (i.e., *soil physical process, soil chemical process, and soil biological process*). These processes affect the structure and stability of the soil. In earthwork operations, these changes need to be taken into account to avoid accidents. As shown in Figure 5-6(c), in SSN, the class *process* is presented in two places in the taxonomy: (i) *Process* concept groups the processes related to sensor deployment, such as installation, maintenance, and removal. (ii) *Process* concept related to the sensing context.

In order to maintain the consistency of the IEW-Onto, the identical terms referring to different concepts are modified. For instance, in the above example, the concept *process* in EW-Onto and OSP are renamed as *earthworkProcess* and *soilProcess*, respectively. The concept *process* in SSN under the *method* concept is renamed as *sensingProcess,* whereas the concept *process,* which is a superclass of *deploymentRelatedProcess* is renamed as *deploymentProcess*.

(c) Implementing the integration: As mentioned above, the candidate ontologies are available in OWL format, which facilitates the integration process. These ontologies are studied using the available documents and the comparison tool (explained in Section 5.3.1). Table 5-1 illustrates examples of the conceptual components (mentioned in Section 2.8.2) represented as terminology, assertion, and rule axioms, which define the concepts, individuals, and relationships in IEW-Onto, respectively. Protégé (Musen 2015) is used to integrate the concepts from the candidate ontologies with EW-Onto. The mapped classes and relationships in IEW-Onto retain their original Uniform Resource Identifier (URI) from the source ontologies.

Figure 5-7 shows the main concepts and relationships in the IEW-Onto. Several new relationships are added to link the concepts from the integrated ontologies. For example, the *System* and *Platform* concepts from SSN are linked through *onPlatform; w*hereas*, Resource, Site,* and *Actor* from EW-Onto are linked with concepts from SSN by representing them as a type of platform where the sensor systems can be installed. In addition, new concepts are linked to SSN concepts as new classes of sensing methods and technologies. Table 5-2 shows the relationships between concepts in IEW-Onto. The table describes how the relationships link these concepts.



(a) In EW-Onto (b) In OSP (c) In SSN

Figure 5-6 Process concepts in three ontologies

Table 5-1 Examples of TBox, ABox, and RBox components in the IEW-Onto

Axiom	Explanation	Examples		
		EW-Onto	**OSP**	**SSN**
TBox	Describe terminological knowledge	HazardWorkzone, CaveIn, Resource, Location	SoilPhysicalProperty, SoilDensity, SoilMoistureContent, Weather	Device, Sensor, Stimulus, Platform, ObservationValue
ABox	Describe knowledge about the individuals	Path1, IntersectionPointT1, ExcavationZone1	Soil1, SoilType-A	RTLS, GPS
RBox	Describe the properties of the roles	[Transitive] e.g., hasHazard: The property is transited from one individual to another over the chain of two individuals.	[Functional] e.g., hasSoilType: The property can have at most one value.	[Reflexive] e.g., hasPart: The individual is related to itself via this property.

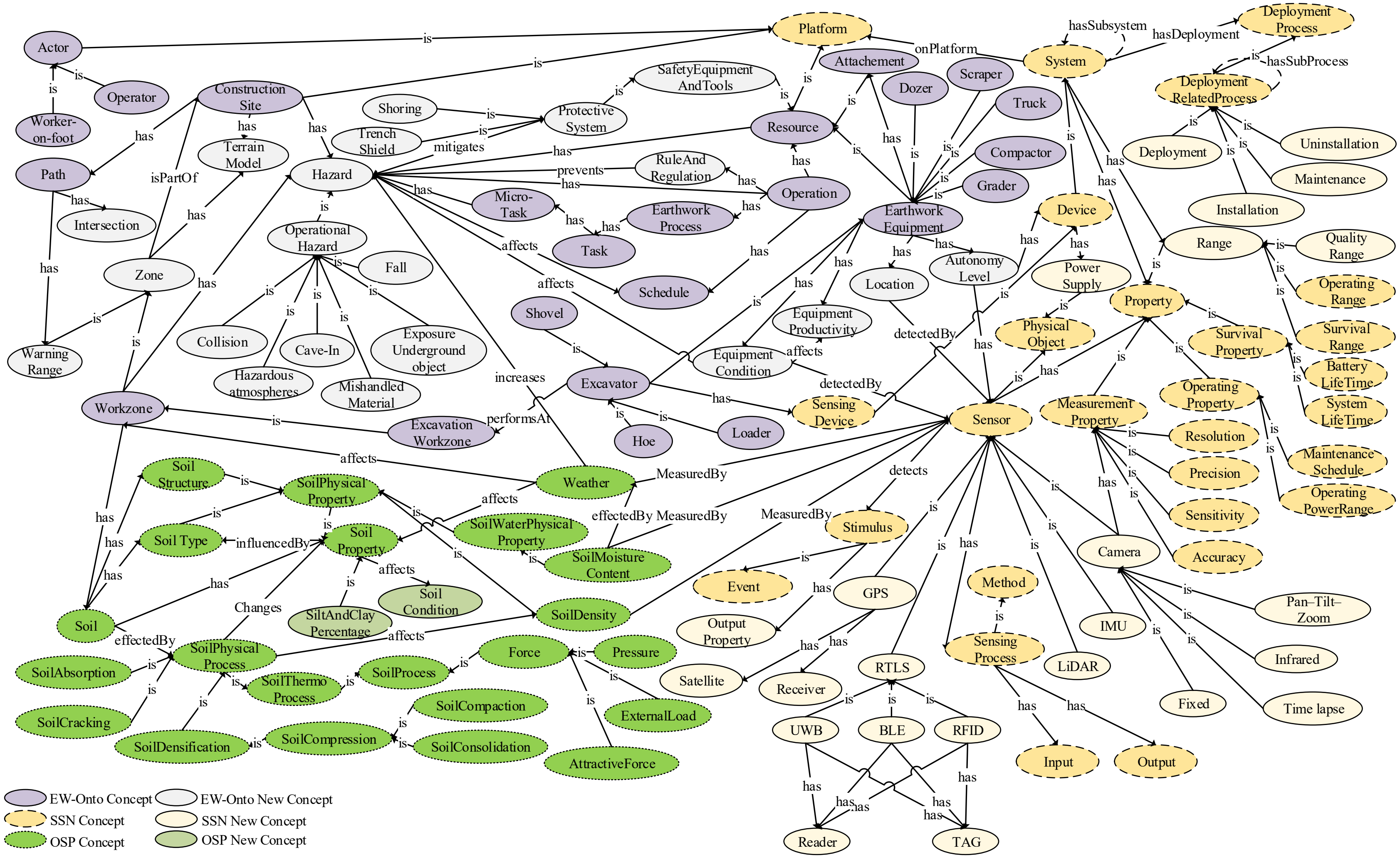


Figure 5-7 Examples of concepts and relationships in IEW-Onto

Table 5-2 The relationships between concepts

Concept	Relationship	Concept	Description
Actor ConstructionSite Resource	is-a	Platform	Sensors are attached to the platform. (e.g., GPS) are attached to equipment or installed on the construction site. Tags are attached to the workers
Stimulus Location EquipmentCondition	detectedBy	Sensor	Stimulus are detected by sensors. GPS and RTLS detect the equipment's location and condition.
Weather SoilMoistureContent SoilDensity	measuredBy	Sensor	Weather conditions (e.g., wind, temperature, humidity) are measured by sensors.

The number of components from each ontology and the total number of these components in IEW-Onto are illustrated in Table 5-3. There are 240 concepts in EW-Onto, 592 concepts in OSP, and 52 concepts in SSN. The 240 concepts from EW-Onto are extended with other 38 concepts, which are illustrated in Table 5-4. These concepts are added to EW-Onto to facilitate the safety knowledge representation by linking them with the EW-Onto concepts through the taxonomy and the relationships. Moreover, 14 object properties are added to the 37 original EW-onto object properties to cover the safety knowledge and link between the concepts (e.g., *EquipmentCondition detectedBy Sensor, Equipment hasLocation Workzone, Weather affects Workzone).* These object properties are shown in Table 5-3. A total of 16 object properties from OSP and 55 object properties from SSN are added to the object properties in IEW-Onto. Data properties are assigned and added to evaluate IEW-Onto using SPARQL queries and description logics queries. There are 284 concepts selected from OSP to be added to IEW-Onto. The selected concepts are more relevant to the earthwork domain. Other concepts are about the chemical, biological, and agricultural processes and properties. Table 5-6 shows the new concepts which are added to OSP. More concepts can be included in IEW-Onto in the future to facilitate other usages. There are 52 concepts from SSN selected to be in IEW-Onto. These concepts are needed to represent the knowledge about the different sensors that are used in the domain. Moreover, and to extend the coverage of SSN, 24 new concepts are added to SSN to cover the equipment and technologies that are used in the earthwork domain. These concepts are shown in Table 5-7. Some related concepts from EW-Onto appear in OSP and SSN ontologies and have the same contexts. These concepts were created during the development of EW-Onto and appeared in the other ontologies during the integrating process. These concepts from EW-Onto are aligned as unified concepts in IEW-Onto. Table 5-8 shows these concepts and their original ontology.

Table 5-3 Summary of the main components in IEW-Onto

Ontology	Original Concepts	Related Concepts	Added Concepts	Aligned Concepts	Total Number of Concepts (IEW-Onto)
EW-Onto	240	240	38	7	633
OSP	592	284	3	4	
SSN	52	52	24	3	
	Original Object Properties	Related Object Properties	Added Object Properties	Aligned Object Properties	Total Number of Object Properties (IEW-Onto)
EW-Onto	37	37	14	-	122
OSP	16	16	-	-	
SSN	55	55	-	-	
	Original Data Properties	Related Data Properties	Added Data Properties	Aligned Data Properties	Total Number of Data Properties (IEW-Onto)
EW-Onto	91	91	-	-	91
OSP	-	-	-	-	
SSN	-	-	-	-	
	Original Equivalent Classes	Related Equivalent Classes	Added Equivalent Classes	Aligned Equivalent Classes	Total Number of Equivalent Classes (IEW-Onto)
EW-Onto	1	1	4	-	38
OSP	74	32	-	-	
SSN	1	1	-	-	

Table 5-4 The added concepts to EW-Onto in IEW-Onto

Owl: Thing ⋮ Hazard *OperationHazard* *CaughtInBetween* *CaveIn* *Collision* *Electrocution* *UndergroundObjectExposure* *Fall* *HazardousAtmosphere* *MishandledMaterial* *StruckbyObject* *ProductHazard* *ResourceHazard* *ChemicalMaterial* *FlammableMaterial* *ToxicMaterial* *HazardWorkzone*	**Owl: Thing** ⋮ Resource ⋮ *SafetyEquipmentAndTool* *ProtectiveSystem* *ShoringSystem* *TrenchShield* *GuardrailSystem* *Ladder* *PersonalFallArrestSystem* *SafetyNet* *Scaffold* *Walkway* **Owl: Thing** ⋮ Resource ⋮ Tool ⋮ *InspectionTool* *AugersEarthDrill* *SoilPenetrometer*	**Owl: Thing** ⋮ ConstructionSite ⋮ Zone ⋮ *WarningRange* *WarningArea* ⋮ Workzone ⋮ *ExcavationEdge* **Owl: Thing** ⋮ AutonomyLevel RulesAndRegulations ⋮ *ProtectionProcedure* *ShieldingSystem* *ShoringSystem* *SlopingExcavation* *SafetyInstruction*

Table 5-5 The added object properties to EW-Onto in IEW-Onto

affects, detectedBy, goingTo, hasHazard, hasLocation, hasSoilType, hasSpeedLimit, isEquippedWith, mitigatets, movingFrom, needSafetyProcedure, needSafetyResource, prevents, requires

Table 5-6 The added concepts to OSP in IEW-Onto

Owl: Thing ⋮ Property GeneralProperty *SoilCondition*	**Owl: Thing** ⋮ Property ⋮ SoilProperty *SiltAndClayPercentage*	**Owl: Thing** ⋮ Property ⋮ SoilProperty SoilPhysicalProperty SoilCompressibility *UnconfinedCompressiveStrength*

Table 5-7 The added concepts to SSN in IEW-Onto

Owl: Thing	**Owl: Thing**	**Owl: Thing**
⋮	⋮	⋮
PhysicalObject	PhysicalObject	Quality
PowerSupply	System	Property
Sensor	Device	*OutputProperty*
Camera	*Reader*	*Range*
FixedCamera	*Receiver*	*QualityRange*
InfraredCamera	*Satellite*	
Pan-tilt-ZomeCamera	*Tag*	
TimeLapseCamera	DeplomentProcess	
GPS	Deployment-relatedProcess	
IMU	*Deployment*	
LiDAR	*Installation*	
RTLS	*Mantenance*	
BLE	*Uninstallation*	
RFID		
UWB		

Table 5-8 Examples of the related concepts alignment in IEW-Onto

EW-Onto	OSP	SSN	IEW-Onto
WeatherCondition	Weather	-	Weather
SoilClayLevel	SoilClayContent	-	SoilClayContent
SoilWaterLevel	SoilWaterContent	-	SoilWaterContent
SoilMoisture	SoilMoistureContent	-	SoilMoistureContent
ElectronicSystem	-	System	System
ElectronicDevice	-	Device	Device
SensorDevice	-	Sensor	Sensor

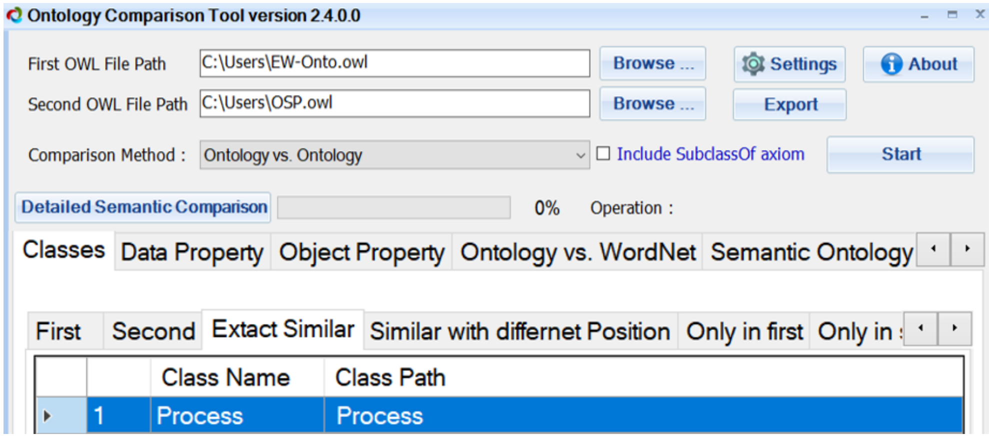
5.3 IEW-ONTO IMPLEMENTATION

5.3.1 Comparing Ontologies

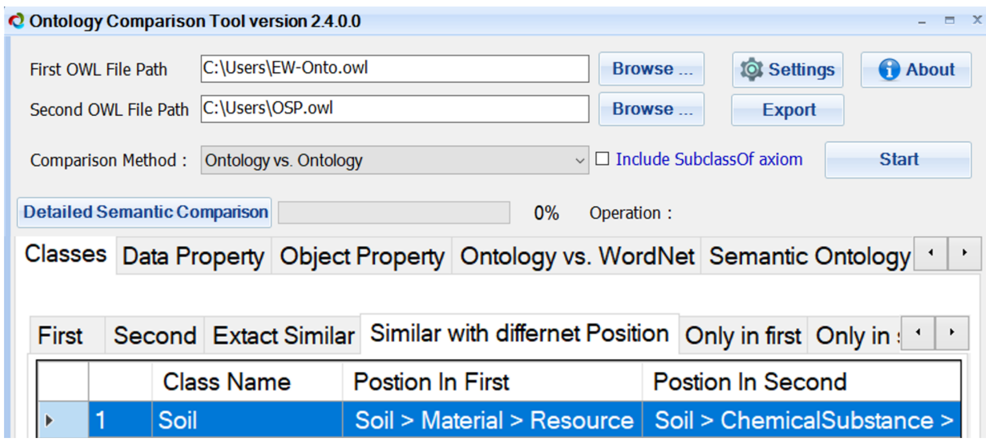
As mentioned in Section 2.8.3, when reusing ontologies, it is necessary to compare similar concepts. An ontology comparison tool is developed using C# to compare and find the similarities between the component ontologies (i.e., EW-Ont, OSP, and SSN). For example, Figure 5-8 shows the results of the comparisons between EW-Ont and OSP. The results of EW-Onto and OSP classes' comparison is shown in Figure 5-8(a), where similar classes are located in the hierarchy at the same taxonomy (the *process* is superclass in both ontologies). Whereas Figure 5-8(b) shows that similar classes are located in the hierarchy at different taxonomies (*soil* class appears in both ontologies but with different taxonomies). Also, the tool shows the hierarchies of these similar classes. Figure 5-8(c) shows the results for the object properties comparison. The results show that there are similar objects properties in both ontologies with close terminologies. For example, *partOf* and *isPartOf* are object properties in both ontologies. The ontology comparison tool gives ideas about the components in both ontologies that can be aligned to support the consistency of the representation. Furthermore, it gives the lists of classes, data properties, and object properties that are only included in either ontology. The core code for the comparison tool is shown in Appendix C.

5.3.2 Verification of Developed Rules

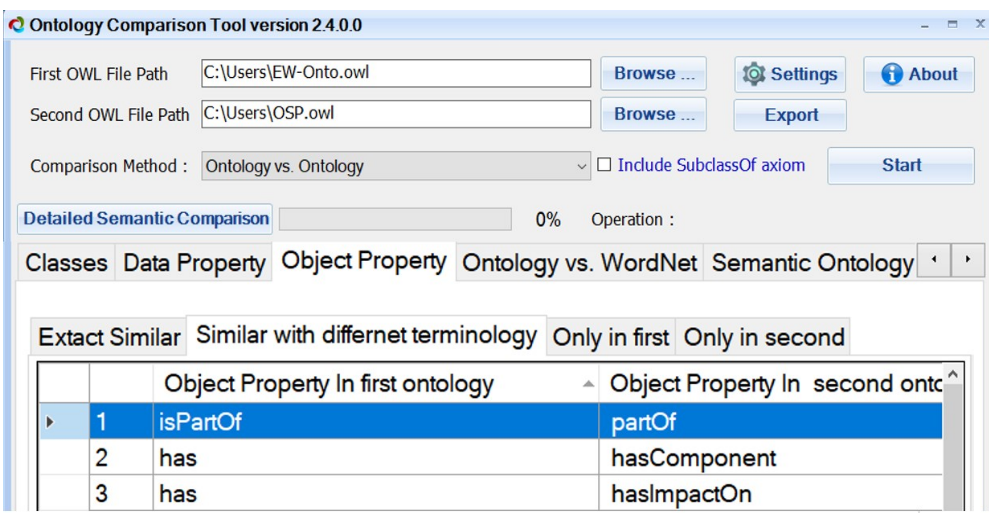
SWRL rules are used to add and edit the rules and the regulations using Protégé. Furthermore, the consistency of IEW-Onto is checked using the Pellet reasoner. Table 5-9 shows examples of rules with the pseudocode and SWRL implementation of each rule in IEW-Onto. IEW-Onto is available at **https://www.ew-onto.info/.**



(a) Similar classes in EW-Onto and OSP



(b) Similar classes in different positions in the taxonomies



(c) Similar object properties with different terminology

Figure 5-8 Examples of ontologies' similarity results between EW-Onto and OSP

Table 5-9 Examples of rules

Rule 1: Soil classification	

Pseudocode	Start
	Input: soil *structure,* Silt and Clay percentage *(scp),* the value of Unconfined Compressive Strength *(ucs)*
	Output: soil classification
	For each Soil sample do
	If soil *has structure == "Cohesive" and* scp > 15%
	and ucs > =1.5 TPF, and it is not *fissured*
	Set soil has *Type A*
	end if
	End

SWRL implementation	*Soil(?so) ^ hasStructure(?so,"Cohesive")* *^hasSiltAndClayPersentage(?so,?scp)* *^swrlb:greaterThanOrEqual(?scp,0.15)* *^hasUnconfinedCompressiveStrengthValue(?so,?ucs)* *^swrlb:greaterThan(?ucs,1.5)* *^isFissured(?so,false)* *-> hasType(?so,"A")*
Rule 2: Cave-in hazard	

Pseudocode	Start
	Input: The workzone depth (*d*), type of the earthwork operation (*exc*)
	Output: potential hazard, safety resource needed, workzone classification
	For each excavation operation do
	For each workzone in operation do
	If *d* > 153 cm
	Set workzone *has* hazard (CaveIn)
	Set workzone *needs* safety resource (ProtectionSystem)
	Set workzone *is* hazard workzone
	Set excavation operation *has* hazard
	end if
	end
	end
	End

Table 5-9 Examples of rules (Continued)

SWRL implementation	*ExcavationOperation(?exco)^Workzone(?wz)* *^CaveIn(?ca)* *^ProtectionSystem(?prosys)* *^hasWorkzone(?exco,?wz)* *^hasDepth(?wz,?d)* *^swrlb:greaterThan (?d,153)* *->HazardWorkzone (?wz)^hashazard(?wz,?ca)* *^ needSafetyResource(?wz,? prosys)^ has (?exco, Hazard)*
Rule 3: Workzone with multiple layers of different soil types	

Pseudocode	Start
	Input: The workzone depth (*d*), type of the earthwork operation (*exc*), type of soil at each layer in the workzone
	Output: potential hazard, safety procedure needed, workzone classification
	For each excavation operation do
	For each workzone in operation do
	If *d* > 153 and *d* < 609 cm
	If workzone *hasMultiSoilLayers* SoilLayer
	If SoilLayer *hasSoilType* == *Type B*
	Set *SlopAngle* == 45 degrees
	If SoilLayer *hasSoilType* == *Type A*
	Set *SlopAngle* == 53 degrees
	end if
	end if
	end if
	end if
	Set workzone *is* hazard workzone
	Set excavation operation *has* a hazard
	Set workzone *has* hazard (CaveIn)
	Set workzone *needs safety procedure* (SlopingExcavation)
	end
	End

Table 5-9 Examples of rules (Continued)

SWRL implementation	*ExcavationOperation(?exco)^Workzone(?wz)* *^Slop(?slo1)^Slop(?slo2)^Soil(?s1)^MultiSoilLayer(?sl)* *^Soil(?s2)^SlopingExcavation(?se)^isTypeOf(?slo1,?se)* *^SoilLayer(?sl1)^SoilLayer(?sl2)^CaveIn(?ca)* *^hasMultiSoilLayers(?wz,?sl)^hasLayerOfSoil(?wz,?sl1)* *^hasLayerOfSoil(?wz,?sl2)^hasType(?s1,"B")* *^ hasType(?s2,"A")^isTypeOf(?slo2,?se)* *^hasWorkzone(?exco,?wz)^hasSoilType(?sl1,?s1)* *^hasDepth(?wz,?d)^hasSoilType(?sl2,?s2)* *^swrlb:greaterThan (?d,153)^swrlb:lessThan(?d,609)* *-> hashazard(?wz,?ca)^has(?exco,Hazard)* *^hasSlopAngle(?sl1,45)* *^ hasSlopAngle(?sl2,53)* *^needSafetyProcedure(?exco,?se)*
Rule 4: Truck Priority	

Pseudocode	Start
	Input: The location, the status of the truck (loaded or not loaded)
	Output: Give the priority
	For each Truck do
	If *Truck* is *loaded and isLocatedAt* (Intersection)
	Set *Truck hasPriority* (*True*)
	end if
	End

SWRL implementation	Intersection(?int)^Truck(?t)^isLoaded(?t,true)^isLocatedAt(?t,?int) -> hasPriority(?t,true)

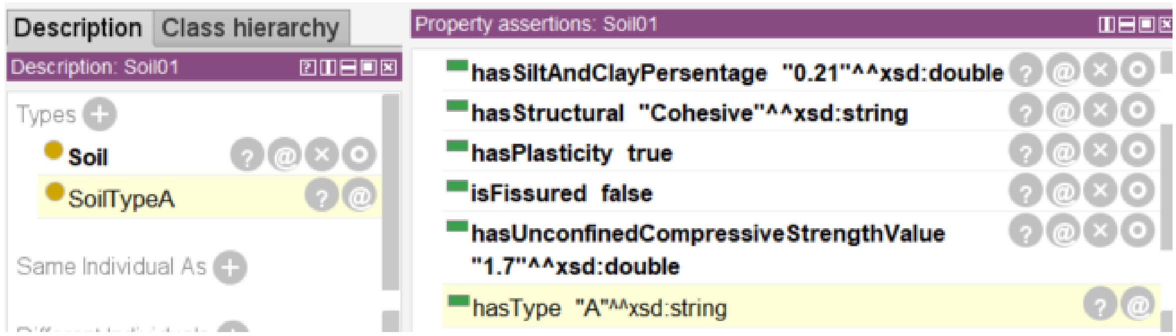
Table 5-9 Examples of rules (Continued)

Rule 5: Truck Collision avoidance	

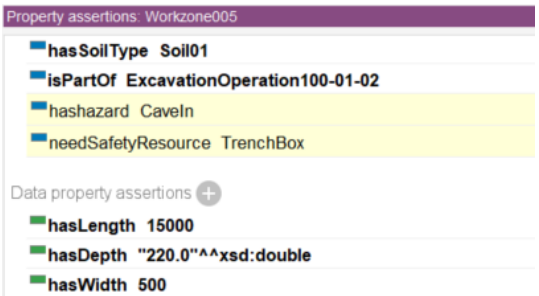
Pseudocode	Start
	Input: The location, the direction, truck label
	Output: The order to stop or slow down
	For each Truck do
	If *label* == "*Truck i*" and *location* == *isLocatedAt* (Intersection *i*) and direction= "GoingStraight"
	If *label* == "*Truck j*" and *direction* == "*GoingLeft*" and *location* == *isLocatedAt* (Intersection *j*) and *isUnder* (*WarningRange*)
	Set *Truck j hascollisionWarning*(*True*)
	Set *Truck j hasOrder* (*Stop or slow down*)
	end if
	end if
	end
	End

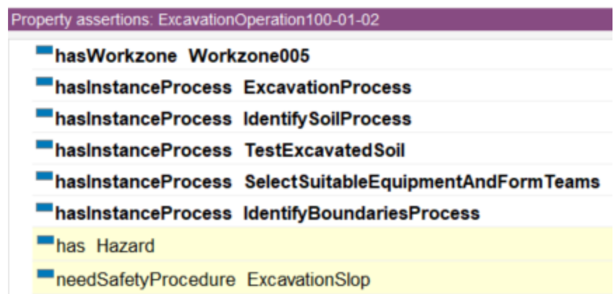
SWRL implementation	*Truck(?tr1)^hasLabel(?tr1,"Truck1")^Truck(?tr2)* *^hasLabel(?tr2,"Truck2")^WarningRange(?wr)* *^isLocatedAt(?tr1,Intersection1)* *^ isLocatedAt(?tr2,Intersection2)* *^hasDirection(?tr1,"GoingStraight")* *^hasDirection(?tr2,"GoingLeft")^isUnder(?tr2,?wr)* *-> hascollisionWarning(?tr2, true)^hasToStop(?tr2, true)*

Figure 5-9(a) illustrates an example of applying the reasoner engine on Rule 1 to classify the soil (see Section 5.2.1). Thus, *Soil01* is soil type A based on the values of this sample. Moreover, Figure 5-9(b) provides an example of linking the hazard with the soil type and the depth of the workzone. The instance of workzone (i.e., *Workzone005*) in Figure 5-9(b) has the soil instance *Soil1*. Based on Rules 2 (see Section 5.2.1) this workzone instantiates *hashazard* CaveIn and *needSafetyResource,* which is TrenchBox. As shown in Figure 5-9(c), the operation instance *ExcavationOperation100-01-02* is linked to *Workzone005* through the relationship *hasWorkzone*. Consequently, this operation has inferred the hazard and will assign *needSafetyProcedure* to TrenchSlop. Figure 5-9(c) shows how the operation is linked with the process using a *hasProcess* relationship. The figure illustrates that *ExcavationOperation100-01-02* has a list of processes.



(a) Soil classification

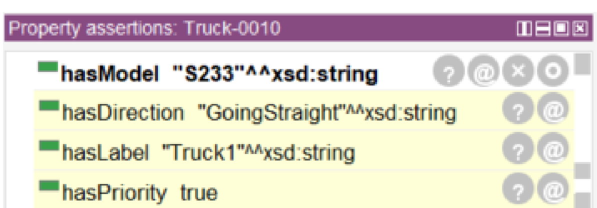


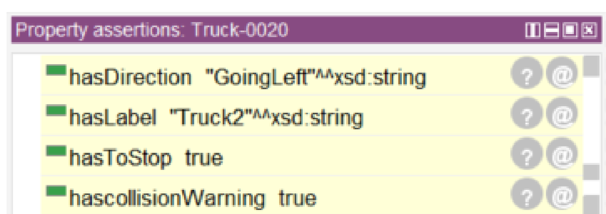


(b) Hazard in workzone (c) Hazard in excavation operation

Figure 5-9 Reasoning engine results

Figure 5-10 shows the results of the reasoning over IEW-Onto for the trucks at the uncontrolled intersection (see Figure 5-2). As shown in Figure 5-10(a), Truck 1 *hasDirection GoingStraight* and has the priority (*hasPriority: true*) (based on rule 4). Thus, in this case, and based on rule 5, Truck 2 receives a warning (*hascollisionWarning = true*) and the order to stop (*hasToStop=true*), as in Figure 5-10(b).





(a) (b)

Figure 5-10 Reasoning results in IEW-Onto for trucks at an uncontrolled intersection

5.4 IEW-ONTO EVALUATION

The build-in tools in Protégé (e.g., Pellet and DL queries) are used for checking the consistency of IEW-Onto taxonomy from the beginning of the development phase as part of the validation process. Protégé reasoner, and DL query plugins are used to make queries over the IEW-Onto. DL provides the human-understandable syntax to create the queries. The autonomy level of equipment is linked and depended on the type and the capability of the sensors and devices installed on the equipment. For example, GPS could help the operator in real-time improve accuracy and productivity. Moreover, sensors and devices' properties could shift the equipment from one level to another. For example, GPS with high accuracy could be used to control the equipment's location while performing the tasks. In contrast, another GPS could be used to show the location to the operator. Figure 5-11 shows an example of DL query result about the equipment equipped with GPS. The query result lists *Hoe-0030* and *Truck-0010* as equipment with GPS. Thus, with a variety of concepts and relationships that can be used to build the queries' expressions, different queries' can be executed over IEW-Onto to get the desired knowledge. The results show that IEW-Onto is able to provide the required knowledge based on the queries.

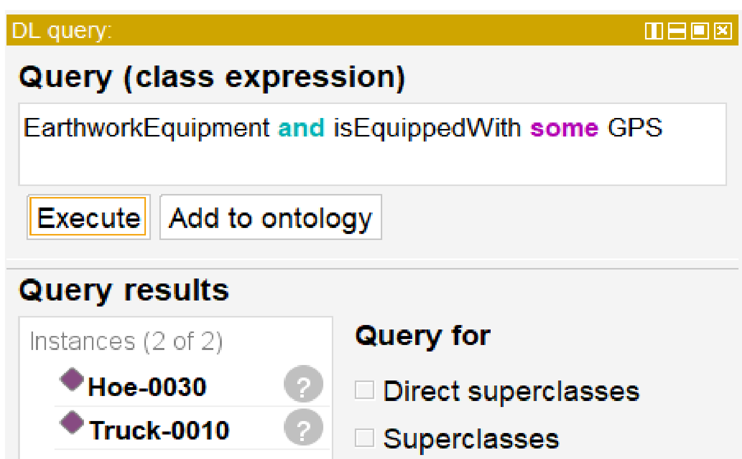


Figure 5-11 The results of the query in Protégé for equipment equipped with GPS

Other approaches can be used to evaluate IEW-Onto. These approaches are explained in Section 2.8.4. The drawback of the "gold standard" is that the evaluation is based on comparing the IEW-Onto with an existing benchmark ontology in the domain, which is not available at this time. A data-driven approach and application-based approach are used to evaluate IEW-Onto. These approaches are explained in the next sections.

5.4.1 Data-Driven Evaluation

5.4.1.1 Ontology-Corpus Measure

It is argued that the corpus of texts might be the most effective source of information that can be used for ontology evaluation (Brewster et al. 2004). Comparing the developed ontology with the corpus is mentioned in Section 2.8.4 as a data-driven evaluation approach. Therefore, the data-driven approach is used to evaluate the IEW-Onto. In the ontology-corpus evaluation, IEW-Onto terms are automatically extracted to find the similarity with the corpus. Since there is no specific corpus for the earthwork domain, WordNet is used as a corpus. Using the corpus against the ontology gives the measures of the lexical terms and reflects the coverage of the IEW-Onto. Python and C# are used to perform similarity measurement. The Natural Language Toolkit (NLTK) in Python provides the required statistical Natural Language Processing (NLP) tools. The result of the ontology-corpus evaluation gives 86.96%, reflecting the similarity between the terms in IEW-Onto and WordNet. These results indicate that IEW-Onto provides a high level of ontology's richness and clarity. Some terms are precisely matching with the synsets from WordNet (e.g., collision), whereas others show up also under a similar tab, which means that WordNet also has other similar terms to the term from IEW-Onto. For example, the team "Dozer" from IEW-Onto is matching the term "dozer" from WordNet, and there is another similar term in WordNet, which is "bulldozer". In this case, the similar terms may be used to improve the terminology in IEW-Onto.

5.4.1.2 Taxonomy-Based Measures

As mentioned in Section 2.8.4, WUP and LCH algorithms are used as semantic measures between the ontology and WordNet. These algorithms provide a quantitative measure of the ontology. In this evaluation, IEW-Onto is evaluated against WordNet based on the depths of its terms and the depths of synsets from WordNet. Equations 2.2 and 2.3 in Section 2.8.4 provide the calculations for the similarity. Figure 5-12 shows the results of the similarity measures. As shown in the figure, WUP is 90.48%, and LCH is 85.71%. The results reflect the number of terms used in IEW-Onto that are semantically identical and appear in WordNet. These results indicate that IEW-Onto provides a high level of ontology's comprehensiveness and interpretability. The core code for the semantic comparison tool is shown in Appendix D.

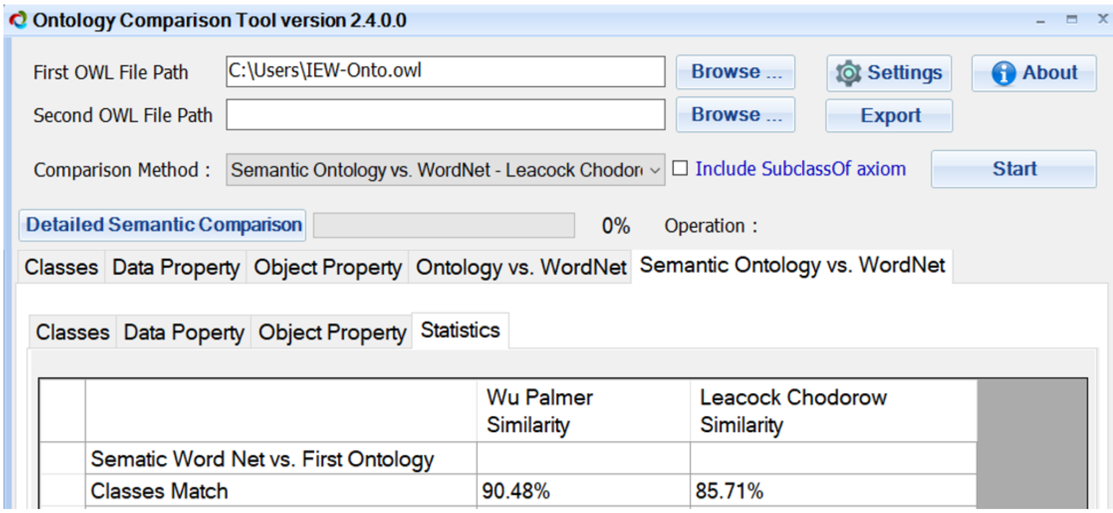


Figure 5-12 The results of WUP and LCH measures against IEW-Onto

5.4.2 Application-Based Evaluation

This evaluation typically evaluates how effective IEW-Onto is in the context of an application. MAS is practically used in dynamic and distributed environments, where two or more agents work and interact to achieve their goals. A dedicated agent supports each piece of equipment and other entities in the earthwork project. In our previous work (Vahdatikhaki et al. 2017), the developed MAS supports the equipment operators to improve safety, which is done without a formal representation of the related knowledge. However, in this work, we used IEW-Onto, which provides this missing knowledge representation. Moreover, MAS benefits from the knowledge and the safety rules defined in IEW-Onto instead of an ad-hoc approach. IEW-Onto is used to create MAS teams for earthwork operations and then to monitor these teams for safety issues during these operations.

The MAS comprises four types of agents with different functions: (1) Operator Agents (OA) represent the agents in the construction site. Each equipment operator is supported by designated agents and other layers of coordinator agents. These agents are formed as teams to reflect the real situation at the site. Thus, each team includes several operator agents and a Team Coordinator Agent (TCA), who communicates with the General Coordinator Agent (GCA). Another layer of agents which support these agents are the information agents (Vahdatikhaki et al. 2017); (2) Ontology Agent (OntoA) is responsible for accessing and making the queries to IEW-Onto; (3) Resource agent (ResA) provides information about the resources (e.g., equipment); (4) Safety agent (SA), which is responsible for responding to the safety issue; and (5) Database Agent (DBA) is responsible for updating the availability of the resources. As illustrated in Figure 5-13, the communications start when the GCA sends a request to the Teams Setup Agent (TSA) to create the teams for an operation. Upon the requested delivery, TSA verifies the request with the IEW-Onto through OntoA to get the number of teams required for this operation. OntoA sends the results back to TSA, who forwards it to GCA. GCA sends a message to ResA to determine the availability of the resources. At this point, ResA has to perform two main requests: (a) ResA sends a REQUEST to OntoA to create the quires to the IEW-Onto about the types of resources needed for this operation. After checking the rules related to the required resources. The query retrieves the list of equipment with their properties and sends back the results to OntoA. OntoA forwards the results to ResA. (b) ResA sends REQUEST to DBA to check if the required equipment is available or not. ResA sends back the list of equipment, which is combined from DBA and OntoA to GCA. GCA forwards the list to TSA to create the teams based on this list.

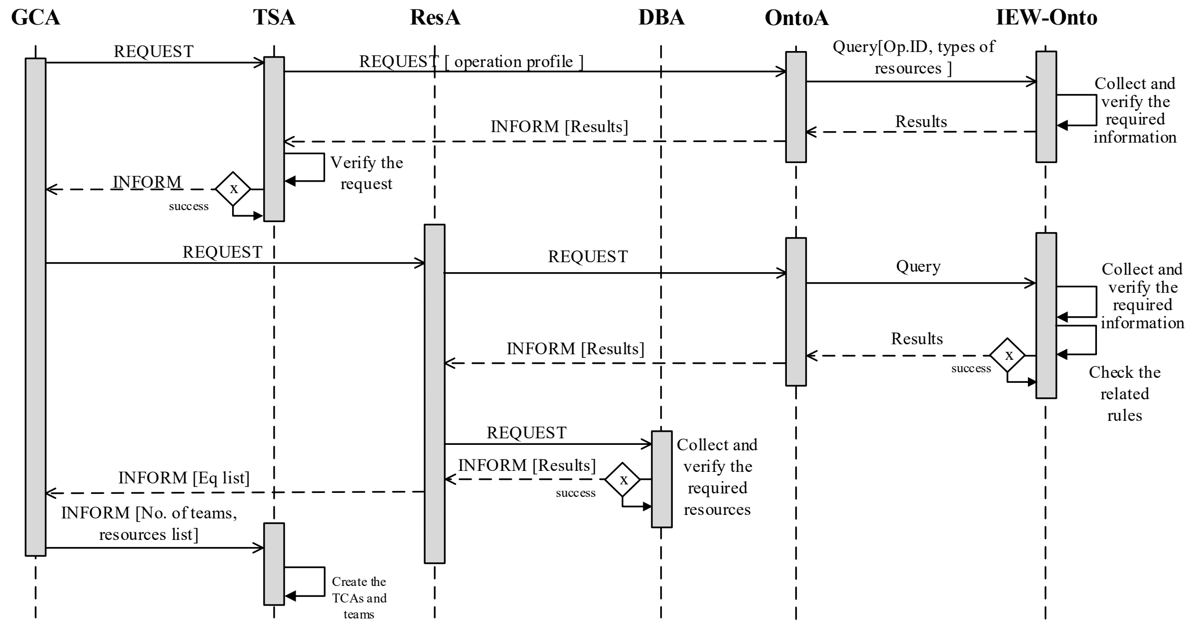


Figure 5-13 Sequence diagram of ontology-based MAS communication

In this example, the operation has two teams and requires six pieces of equipment (i.e., two hoes and four trucks). After TSA receives the number of teams in this operation, it starts creating the teams and assigning the equipment.

Java Agent Development Framework (JADE) is used in the development of MAS. JADE uses the Foundation for Intelligent Physical Agents (FIPA) specifications. These specifications provide the communication between the agents (Bellifemine et al. 1999). Figure 5-14 illustrates the created agents for the teams in JADE.

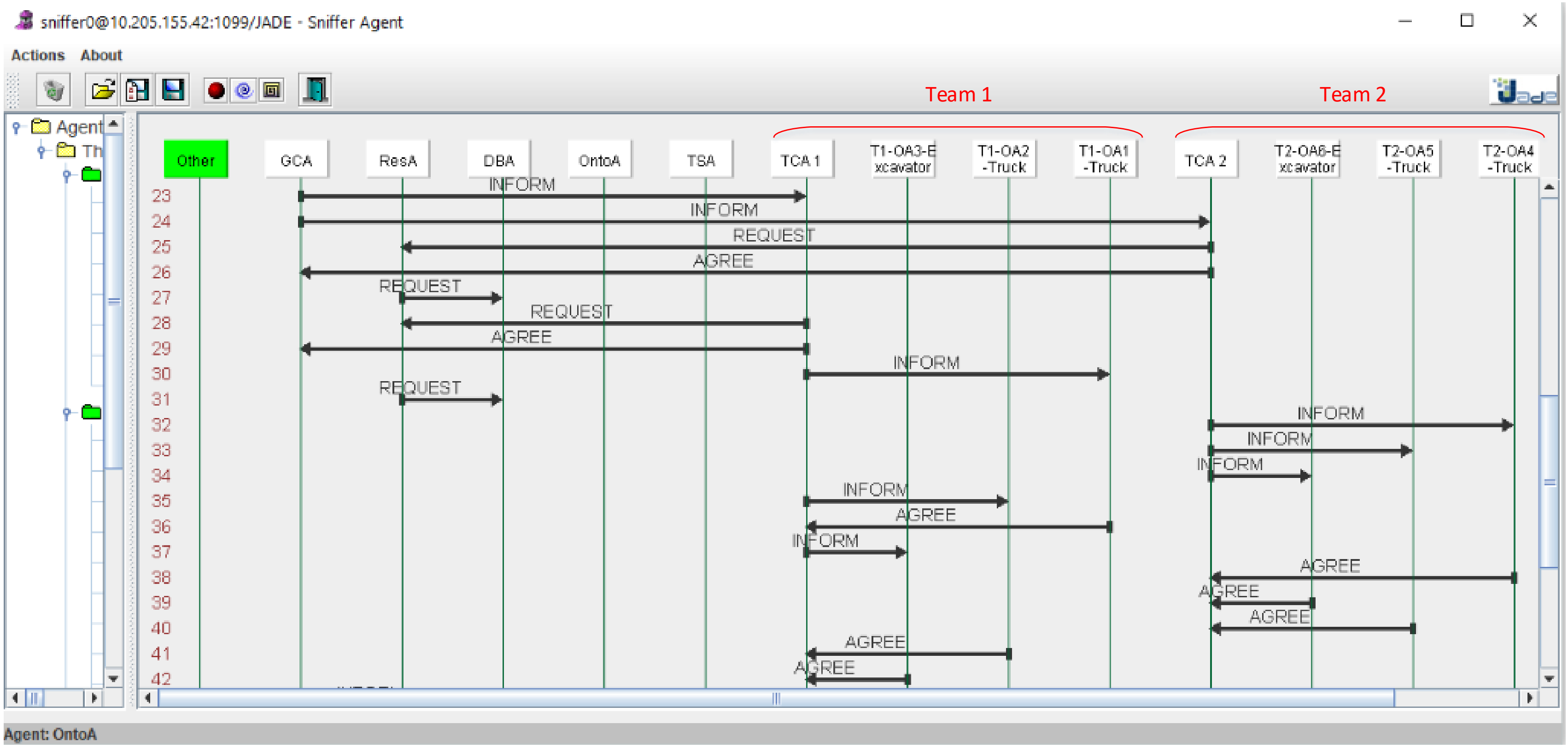


Figure 5-14 Creating the agents' teams based on the retrieved information from IEW-Onto

The IEW-Onto browser tool is used to show the inferred safety information from IEW-Onto. This tool is built to help SA retrieve the safety information form IEW-Onto. Figure 5-15 shows the information about the excavation operation, which has two workzones (i.e., *workzone001 and workzone005*). The tool shows the inferred information at the operation level (i.e., *Hazard is True*, *SafetyProcedure: ExcavationSlop* and *SlopAngle: 53*) (based on Rule 3). Moreover, it shows the inferred information at the workzone level in this operation. As shown in the figure, *workzone001* has a hazard (i.e., CaveIn). The reason of this hazard is the depth of the workzone (based on rule 2). Moreover, the resource (i.e., Trench Box) is needed in this workzone to eliminate or mitigate this type of hazard (based on Rule 2). Based on this safety inferred information, SA delivers this information to GCA to forward it to TCA in each team. TCAs forward the information to OAs in its team. Each truck in the team sends its location to TCA; then, TCA forwards it to SA, who checks if there are any safety issues related to the locations of the trucks. Based on the sensor data received from the trucks, the truck with priority will proceed, and the other trucks receive messages to stop (based on Rules 4 and 5). The core code for the ontology browser tool is shown in Appendix E.

As demonstrated above, IEW-Onto can provide knowledge about the operation and related hazards. Thus, the procedures and resources to mitigate these hazards can be planned and performed. The knowledge provided by IEW-Onto is not merely from the concepts and relationships that are listed; rather, it provides the inferred knowledge based on the facts and the rules. The application-based evaluation shows the applicability and usefulness of IEW-Onto in supporting the earthwork projects and creating the dedicated agents to support the teams for the operation.

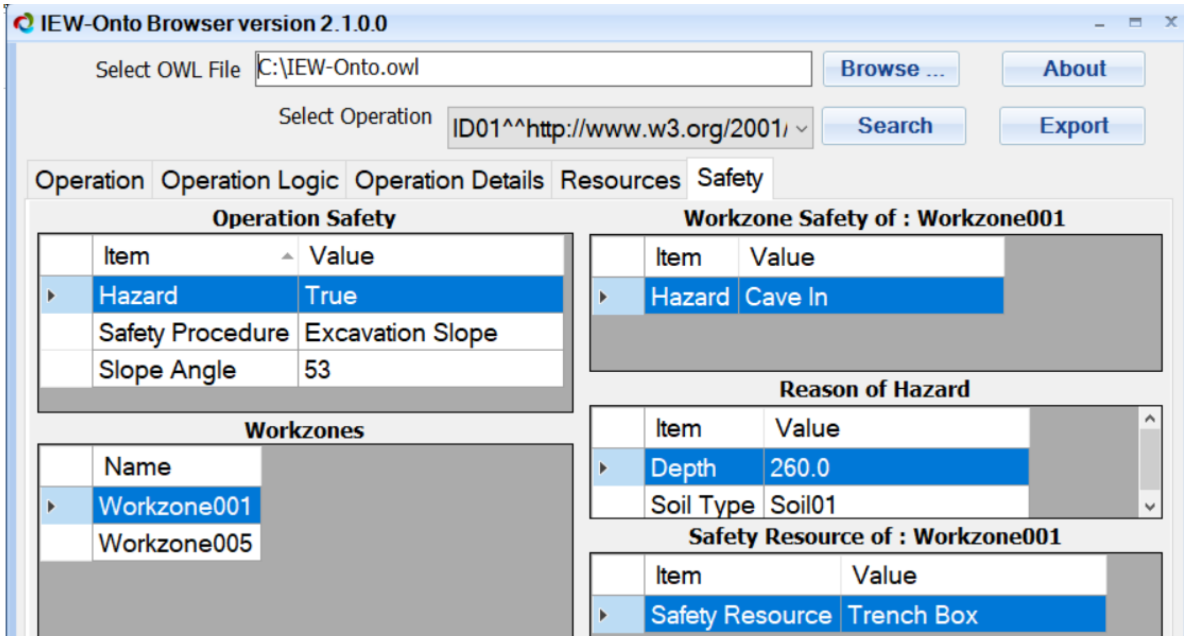


Figure 5-15 The inferred safety information in operation and workzones

5.5 SUMMARY AND CONCLUSIONS

In this chapter, EW-Ono is augmented with an additional knowledgebase and presented as IEW-Onto. IEW-Onto presents the integrated knowledge of the three main components, which are EW-Onto, OSP, and SSN ontologies. IEW-Onto is not merely a collection of concepts and relationships; rather, it defines a conceptualization of the earthwork domain, including the definition and the integration of concepts and relationships. IEW-Onto includes other relationships to link with the safety regulations. The development of IEW-Onto started with defining the concepts and the relationships, which are related to the earthwork domain among the unstructured data (e.g., safety data) and unstructured safety knowledge (e.g., OSHA regulations). The knowledge related to the soil from OSP and to sensors from SSN has been integrated with EW-Onto. A comparison tool is developed to analyze and compare these ontologies to find the similarities and the differences between them to provide consistency representation through IEW-Onto. Different rules were implemented as SWRL rules and included in IEW-Onto using Protégé.

Different evaluation methods were used to evaluate IEW-Onto, including checking consistency, data-driven and application-based validations. The evaluation results show that IEW-Onto has consistency and provides a high level of clarity, richness, comprehensiveness, interpretability, and effectiveness of the presented knowledge. The conclusions for this chapter can be stated: (1)

integrating the related ontologies to earthwork domain and represented as one integrated ontology provides a robust and consistence knowledge that can be used as a knowledgebase in the domain; (2) the integration processes prove that one single ontology can benefit from other pre-defined ontologies related to the domain; (3) The IEW-Onto provides a robust knowledgebase to enhance the safety in earthwork domain and (4) The IEW-Onto has several potential benefits, most notably, the scalability nature to include more concepts and relationships to support other related domains.

CHAPTER 6 SUMMARY, CONCLUSIONS, CONTRIBUTIONS, LIMITATIONS, AND FUTURE WORK

6.1 SUMMARY

Based on the literature review in Chapter 2, it was found that there is no ontology focusing on the earthwork domain. The communication issues and the relationships between the project entities play a significant role in the earthwork project. EW-Onto's development started from defining the concepts and building taxonomies for earthwork operations and equipment following the METHONTOLOGY approach. The developed EW-Onto defines conceptualization, which includes the definition of concepts and relationships in the earthwork domain. The conceptual ontology elements and the different classifications of equipment in this domain are presented. The hierarchies in EW-Onto, which are related to the resources (e.g., equipment) and the different project levels (i.e., *operations*, *processes*, *tasks*, and *micro-tasks*), are built. The ontology has been implemented using Protégé. The consistency of EW-Onto has been checked, and it has been evaluated using a survey.

EW-Onto is augmented with an additional knowledgebase and presented as IEW-Onto. IEW-Onto presents the integrated knowledge of the three main components, which are EW-Onto, OSP, and SSN ontologies. IEW-Onto is not merely a collection of concepts and relationships; instead, it defines a conceptualization of the earthwork domain, including the definition and the integration of concepts and relationships. IEW-Onto includes other relationships to link with the safety regulations. The development of IEW-Onto started with defining the concepts and the relationships, which are related to the earthwork domain among the unstructured data (e.g., safety data) and unstructured safety knowledge (e.g., OSHA regulations). The knowledge related to the soil from OSP and sensors from SSN has been integrated with EW-Onto. A comparison tool was developed to analyze and compare these ontologies to find the similarities and the differences between them to provide consistent representation through IEW-Onto. Different rules were implemented as SWRL rules and included in IEW-Onto using Protégé.

Different evaluation methods were used to evaluate IEW-Onto, including checking consistency, data-driven and application-based validations. The evaluation results show that IEW-Onto has consistency and provides a high level of clarity, richness, comprehensiveness, interpretability, and effectiveness of the presented knowledge.

6.2 CONCLUSIONS

Based on the results of this research, the following conclusions can be made: (1) The METHONTOLOGY approach was effective in the development of EW-Onto; (2) The results of the evaluation show that the developed EW-Onto was able to give a clear, accurate, and comprehensive understanding of the concepts, constraints, axioms, and relationships in the domain; (3) The respondents provided favorable evaluation of EW-Onto in developing practical applications by integrating various types of knowledge; (4) Integrating the related ontologies to the earthwork domain and represented as one integrated ontology provides a robust and consistent knowledge that can be used as a knowledgebase in the domain; (5) The integration processes prove that one single ontology can benefit from other pre-defined ontologies related to the domain; (6) IEW-Onto provides a robust knowledgebase to enhance the safety in earthwork domain; and (7) IEW-Onto has several potential benefits, most notably, the scalability nature to include more concepts and relationships to support other related domains.

6.3 CONTRIBUTIONS

The following points summarize the main contributions of this research. The contributions are presented with respect to the research objectives:

(1) Creating earthwork domain ontology

- Developing an ontology to formalize and represent the earthwork domain knowledge. EW-Onto provides the conceptualization, which offers a shared understanding among the different stakeholders and provides reusable knowledge. EW-Onto was developed from scratch. Most of the previous research about developing ontologies for construction is not available or is only theoretical studies.

- Developing different classifications of equipment and taxonomies in the earthwork domain to provide formal and consistence representations. The concepts and the relationships from the earthwork domain point of view were considered and described.

(2) Integrating the developed ontology

- Integrating EW-Onto with safety knowledge and other related ontologies improves safety in earthwork operations by considering the safety issues at different levels. The integrated ontology (IEW-Onto) presents the knowledge of the three main components, which are EW-Onto, OSP, and SSN ontologies.

- Linking the unstructured safety knowledge (e.g., OSHA regulations) and the unstructured data (e.g., safety data) in IEW-Onto.
- Developing the tools to browse, compare and evaluate the ontologies (not only EW-Onto or IEW-Onto). The ontology browser provides efficient knowledge about operations, processes, tasks, microtasks, resources, workzones, and potential hazards.

- The IEW-Onto has several potential benefits, most notably, the scalability, which allows to include more concepts and relationships to support other related domains. The developed ontology is available for developing further extensions and ontology-based applications.

6.4 LIMITATIONS AND FUTURE WORK

While this research has successfully achieved its objectives, the following limitations still remain to be considered in future work:

(1) Although many studies claim that they developed an ontology in the construction domain, there is a lack of published ontologies to be reused in the IEW-Onto. On the other hand, extending IEW-Onto to include the related data models to the earthwork domain, such as IFC-Road, and LandXml can be investigated in the future. Moreover, considering the developed ontology's scalability, an approach using linked data can be studied in the future to link IEW-Onto with sensory safety data.

(2) Adding the rules related to safety, productivity, quality, or resource allocation and translating these rules from text to axioms requires much more effort to be fully developed. Safety was the main application of IEW-Onto discussed in this study. Other applications can benefit from IEW-Onto to facilitate other needs in the earthwork domain, such as integrating IEW-Onto with simulation models, finding the optimum locations for sensors, and hazards preparedness. These applications can be investigated in future work.

(3) The difficulty to find an adequate number of participants to evaluate the developed ontologies who know about the ontologies and the construction at the same time. Therefore, there is a need to evaluate the developed ontologies using a large sample to assure statistical significance and include more potential ontologies users.

REFERENCES

Ali, A., and Qayyum, O. (2019). “Inference New Knowledge Using Sparql Construct Query.” *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, IEEE, Sukkur, Pakistan., 1–4.

Anderegg, R., Von Felten, D. A., and Kaufmann, K. (2006). “Compaction monitoring using intelligent soil compactors.” *GeoCongress 2006: Geotechnical Engineering in the Information Technology Age*, 1–6.

Apanaviciene, R., Vanagas, A., and Fokaides, P. A. (2020). “Smart Building Integration into a Smart City (SBISC): Development of a New Evaluation Framework.” *Energies*, Multidisciplinary Digital Publishing Institute, 13(9), 2190.

Artun, A., Radaev, A. E., and Badenko, V. L. (2019). “Minimizing Earthwork Volumes by Optimizing Vertical Alignment with Linear Programming Algorithm.” *Construction of Unique Buildings and Structures*, 82(7), 7–13.

Atkinson, J. (2000). “Soil classification.” <https://environment.uwe.ac.uk/geocal/SoilMech/classification/default.htm> (Oct. 29, 2020).

Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., and Nardi, D. (2003). *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge university press.

Bassiliades, N. (2020). “A Tool for Transforming Semantic Web Rule Language to SPARQL Infererecing Notation.” *International Journal on Semantic Web and Information Systems (IJSWIS)*, IGI Global, 16(1), 87–115.

Bauman, Steve, Paul Newman, and Alain Izadnegahdar. (2016). “A Basic Robotic Excavator (the ‘Glenn Digger’): Description, Design, and Initial Operation.” NASA Technical Reports Server (NTRS): <https://ntrs.nasa.gov/citations/20160004045> (Jan. 26, 2021).

Behrman, W. (2002). “Best practices for the development and use of XML data interchange standards.” *Center for Integrated Facility Engineering Technical Report*, 131, 27.

Bellifemine, F., Poggi, A., and Rimassa, G. (1999). “JADE–A FIPA-compliant agent framework.” *Proceedings of Practical Application of Intelligent Agents and Multi-Agents (PAAM)*. Vol. 99. London, UK.

Benevolenskiy, A., Roos, K., Katranuschkov, P., and Scherer, R. J. (2012). “Construction processes configuration using process patterns.” *Advanced Engineering Informatics*, Elsevier, 26(4), 727–736.

Bennett, S., Skelton, J., and Lunn, K. (2001). *Schaum’s outline of UML*. Schaum’s outline series, McGraw-Hill, New York.

Bergman, M. K. (2009). “The Open World Assumption: Elephant in the Room.” *AI3: Adaptive Information*, <https://www.mkbergman.com/852/the-open-world-assumption-elephant-in-the-room/> (Nov. 12, 2020).

Berners-Lee, T. (2009). “Linked Data - Design Issues.” <https://www.w3.org/DesignIssues/LinkedData.html> (Sep. 27, 2020).

Boris, M., Peter F., P.-S., and Bijan, P. (2020). “OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition).” <www.w3.org/TR/owl2-syntax/#Axioms> (May 28, 2020).

Borrmann, A., Muhic, S., Hyvärinen, J., Chipman, T., Jaud, S., Castaing, C., Dumoulin, C., Liebich, T. and Mol, L., (2019). “The IFC-Bridge project–Extending the IFC standard to enable high-quality exchange of bridge information models”. In Proceedings of European Conference for Computing in Construction, 377–386.

Boton, Conrad, Louis Rivest, Oussama Ghnaya, and Moataz Chouchen. 2020. “What Is at the Root of Construction 4.0: A Systematic Review of the Recent Research Effort.” *Archives of Computational Methods in Engineering* 1–20.

Botts, M., and Robin, A. (2007). “OpenGIS sensor model language (SensorML) implementation specification.” *OpenGIS Implementation Specification OGC*, 7.

Brank, J., Grobelnik, M., and Mladenic, D. (2005). “A survey of ontology evaluation techniques” *In Proceedings of Conference on Data Mining and Data Warehouses (SiKDD 2005).* Ljubljana, Slovenia.

Brewster, C., Alani, H., Dasmahapatra, S., and Wilks, Y. (2004). “Data driven ontology evaluation.” *International Conference on Language Resources and Evaluation*, Portugal.

BuildingSMART. (2020). “BuildingSMART Specifications.” <www.buildingsmart-tech.org/specifications> (May 21, 2020).

Cantais, J., Dominguez, D., Gigante, V., Laera, L., and Tamma, V. (2005). “An example of food ontology for diabetes control.” *Proceedings of the International Semantic Web Conference 2005 workshop on Ontology Patterns for the Semantic Web*, 1–9.

Caterpillar Inc. (2020). “Equipment Attachments.” <www.cat.com/en_US/products/new/attachments.html> (Jun. 20, 2020).

Caterpillar Inc. (2021). “308 CR Mini Excavator.” <https://www.cat.com/en_US/products/new/equipment/excavators/mini-excavators/108939670331.html> (Jan. 27, 2021).

CCGA. (2020). “Canadian Common Ground Alliance - Best Practices.” <https://www.canadiancga.com/best-practices> (Sep. 19, 2020).

Chandrasekaran, B., Josephson, J. R., and Benjamins, V. R. (1999). "What are ontologies, and why do we need them?" *IEEE Intelligent Systems and their applications*, IEEE, 14(1), 20–26.

Cheng, F. F., Wang, Y. W., Ling, X., and Bai, Y. (2011). "A Petri net simulation model for virtual construction of earthmoving operations." *Automation in Construction*, Elsevier, 20(2), 181–188.

Cheng, J., Lu, Q., and Deng, Y. (2016). "Analytical review and evaluation of civil information modeling." *Automation in Construction*, Elsevier, 67, 31–47.

Chi, S., Caldas, C. H., and Kim, D. Y. (2009). "A methodology for object identification and tracking in construction based on spatial modeling and image matching techniques." *Computer-Aided Civil and Infrastructure Engineering*, Wiley Online Library, 24(3), 199–211.

Cho, Y. K., Youn, J. H., and Martinez, D. (2010). "Error modeling for an untethered ultra-wideband system for construction indoor asset tracking." *Automation in Construction*, Elsevier, 19(1), 43–54.

Choi, N., Song, I.-Y., and Han, H. (2006). "A survey on ontology mapping." *ACM Sigmod Record*, ACM New York, NY, USA, 35(3), 34–41.

Chon, H. D., Jun, S., Jung, H., and An, S. W. (2004). "Using RFID for accurate positioning." *Journal of Global Positioning Systems*, Scientific Research Publishing, 3(1–2), 32–39.

Compton, M., Barnaghi, P., Bermudez, L., GarcíA-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., and Herzog, A. (2012). "The SSN ontology of the W3C semantic sensor network incubator group." *Web semantics: Science, Services and Agents on The World Wide Web*, Elsevier, 17, 25–32.

Corcho, O., Fernández-López, M., and Gómez-Pérez, A. (2003). "Methodologies, tools and languages for building ontologies. Where is their meeting point?" *Data & Knowledge Engineering*, Elsevier, 46(1), 41–64.

Cox, S. (2007). "OGC Implementation Specification 07-022r1: Observations and Measurements-Part 1-Observation schema." *Open Geospatial Consortium*.

Craveiroa, F., Duartec, J. P., Bartoloa, H., & Bartolod, P. J. (2019). "Additive manufacturing as an enabling technology for digital construction: A perspective on Construction 4.0." *Automation in Construction,* 103, 251–267.

Curry, E., O'Donnell, J., and Corry, E. (2012). "Building optimisation using scenario modeling and linked data." In *First International Workshop on Linked Data in Architecture and Construction (LDAC 2012),* 6–8.

Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M., and O'Riain, S. (2013). "Linking building data in the cloud: Integrating cross-domain building data using linked data." *Advanced Engineering Informatics*, Elsevier, 27(2), 206–219.

Da Rocha, H., Espirito-Santo, A., and Abrishambaf, R. (2020). “Semantic Interoperability in the Industry 4.0 Using the IEEE 1451 Standard.” *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 5243–5248.

Debbie, M. (2016). “Heavy Equipment Attachments Expand Versatility, Reduce Operation Costs.” <www.constructionequipmentguide.com/heavy-equipment-attachments-expand-versatility-reduce-operation-costs/28220> (Jul. 15, 2020).

Delaware Department of Transportation. (2020). “Standard Construction Procedures, General earthwork information, Delaware Department of Transportation.” <www.deldot.gov/information/pubs_forms/manuals/construction_manual/index.shtml> (Jun. 20, 2020).

Dhakal, S., Zhang, L., and Lv, X. (2020). “Ontology-based semantic modelling to support knowledge-based document classification on disaster-resilient construction practices.” *International Journal of Construction Management*, Taylor & Francis, 1–20.

Dibley, M., Li, H., Rezgui, Y., and Miles, J. (2012). “An ontology framework for intelligent sensor-based building monitoring.” *Automation in Construction*, Elsevier, 28, 1–14.

Ding, L., Zhong, B., Wu, S., and Luo, H. (2016). “Construction risk knowledge management in BIM using ontology and semantic web technology.” *Safety Science*, Elsevier, 87, 202–213.

Du, H., Dimitrova, V., Magee, D., Stirling, R., Curioni, G., Reeves, H., Clarke, B., and Cohn, A. (2016). “An ontology of soil properties and processes.” *In International Semantic Web Conference,* Springer, 30–37.

Edirisinghe, R. (2019). “Digital skin of the construction site.” *Engineering, Construction and Architectural Management*, Emerald Publishing Limited.

El-Diraby, T. (2012). “Domain ontology for construction knowledge.” *Journal of Construction Engineering and Management*, American Society of Civil Engineers, 139(7), 768–784.

El-Diraby, T., and Kashif, K. (2005). “Distributed ontology architecture for knowledge management in highway construction.” *Journal of Construction Engineering and Management*, American Society of Civil Engineers, 131(5), 591–603.

El-Diraby, T., and Osman, H. (2011). “A domain ontology for construction concepts in urban infrastructure products.” *Automation in Construction*, 20(8), 1120–1132.

El-Gohary, N. M., and El-Diraby, T. E. (2010). “Domain ontology for processes in infrastructure and construction.” *Journal of Construction Engineering and Management*, American Society of Civil Engineers, 136(7), 730–744.

Farquhar, A., Fikes, R., and Rice, J. (1997). "The ontolingua server: A tool for collaborative ontology construction." *International Journal of Human-computer Studies*, Elsevier, 46(6), 707–727.

Fernández-López, M. (1999). "Overview of methodologies for building ontologies." CEUR Publications. In *IJCAI99 Workshop on Ontologies and Problem-solving Methods: Lessons Learned and Future Trends*, 430.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). "Methontology: from ontological art towards ontological engineering." *AAAI Technical Report SS-97-06*, American Association for Artificial Intelligence, 33–40.

Flotyński, J. (2020). "Creating explorable extended reality environments with semantic annotations." *Multimedia Tools and Applications*, Springer, 1–31.

Fox, S. (2020). "Trimble Earthworks 2.0 grade control platform boasts new features for faster ROI." <https://www.heavyequipmentguide.ca/article/33774/trimble-earthworks-20-grade-control-platform-boasts-new-features-for-faster-roi> (Jan. 30, 2021).

Gadouri, H., Harichane, K., and Ghrici, M. (2018). "Assessment of sulphates effect on the classification of soil–lime–natural pozzolana mixtures based on the Unified Soil Classification System (USCS)." *International Journal of Geotechnical Engineering*, Taylor & Francis, 12(3), 293–301.

García-Gaines, R. A., and Frankenstein, S. (2015). "USCS and the USDA soil classification system: Development of a mapping scheme." Cold Regions Research and Engineering Laboratory (US).

Glimm, B., Horrocks, I., Motik, B., Stoilos, G., and Wang, Z. (2014). "HermiT: an OWL 2 reasoner." *Journal of Automated Reasoning*, Springer, 53(3), 245–269.

Gómez-Pérez, A., and Benjamins, R. (1999). "Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods." *Proceedings of IJCAI and the Scandinavian AI Societies. CEUR Workshop*. Stockholm, Sweden.

Gransberg, D. D., Popescu, C. M., and Ryan, R. (2006). *Construction equipment management for engineers, estimators, and owners*. CRC Press.

Gruber, T. R. (1993). "A translation approach to portable ontology specifications." *Knowledge acquisition*, Citeseer, 5(2), 199–220.

Grüninger, M., and Fox, M. S. (1995). "Methodology for the design and evaluation of ontologies." *Proceedings Workshop on Basic Ontological Issues in Knowledge Sharing*, Citeseer Montreal, Canada.

Guarino, N. (1998). *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98),* IOS press. Trento, Italy.

Guizzardi, G. (2005). “Ontological foundations for structural conceptual models.” Doctoral dissertation, University of Twente, Enschede, The Netherlands.

Haghighi, P. D., Burstein, F., Zaslavsky, A., and Arbon, P. (2013). “Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings.” *Decision Support Systems*, Elsevier, 54(2), 1192–1204.

Halpin, D. W., Riggs, L. S., and Halpin, D. W. (1992). *Planning and analysis of construction operations*. Wiley, New York.

Hamdan, A.-H., and Scherer, R. J. (2020). “Integration of BIM-related bridge information in an ontological knowledgebase.” *Linked Data in Architecture and Construction.*

Hammad, A., Vahdatikhaki, F., and Zhang, C. (2013). “A novel integrated approach to project-level automated machine control/guidance systems in construction projects.” *Journal of Information Technology in Construction (ITcon)*, 18(9), 162–181.

Hammad, A., Vahdatikhaki, F., Zhang, C., Mawlana, M., and Doriani, A. (2012). “Towards the smart construction site: Improving productivity and safety of construction projects using multi-agent systems, real-time simulation and automated machine control.” *Proceedings of the 2012 Winter Simulation Conference (WSC)*. IEEE, 1–12.

Han, S., and Lee, S. (2013). “A vision-based motion capture and recognition framework for behavior-based safety management.” *Automation in Construction*, Elsevier, 35, 131–141.

Hare, W. L., Koch, V. R., and Lucet, Y. (2011). “Models and algorithms to improve earthwork operations in road design using mixed integer linear programming.” *European Journal of Operational Research*, Elsevier, 215(2), 470–480.

Heaton, J., Parlikad, A. K., and Schooling, J. (2019). “A Building Information Modelling approach to the alignment of organisational objectives to Asset Information Requirements.” *Automation in Construction*, Elsevier, 104, 14–26.

Hildreth, J., Vorster, M., and Martinez, J. (2005). “Reduction of short-interval GPS data for construction operations analysis.” *Journal of Construction Engineering and Management*, American Society of Civil Engineers, 131(8), 920–927.

Ho, S. P., Tserng, H. P., and Jan, S. H. (2013). “Enhancing knowledge sharing management using BIM technology in construction.” *The Scientific World Journal*, United States, 2013, 170498.

Holling, S., & Johanna, K. (2016). “New Cat 14M3 Motor Grader.” <https://www.cat.com/en_US/news/machine-press-releases/new-cat-14m3-motor-grader.html> (Jan. 27, 2021)

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

Hou, L., Tan, Y., Luo, W., Xu, S., Mao, C., and Moon, S. (2020). “Towards a more extensive application of off-site construction: a technological review.” *International Journal of Construction Management*, Taylor & Francis, 1–12.

Hu, B., Dasmahapatra, S., Lewis, P., and Shadbolt, N. (2003). “Ontology-based medical image annotation with description logics.” *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 77–82.

Ian, H., Peter, F. P.-S., Harold, B., Said, T., Benjamin, G., and Mike, D. (2004). “SWRL: A Semantic Web Rule Language Combining OWL and RuleML.” <www.w3.org/Submission/SWRL/> (Sep. 21, 2020).

Isikdag, U., Aouad, G., Underwood, J., and Wu, S. (2007). “Building information models: a review on storage and exchange mechanisms.” *Bringing ITC Knowledge to Work*.

JCB. (2021). “The JCB 3CX: A great value high performance backhoe loader.” <https://www.jcb.com/en-us/products/backhoe-loaders/3cx-new-us> (Feb. 14, 2021).

Katranuschkov, P., Rybenko, K., and Scherer, R. (2009). “Ontology-based dynamic process support on the example of defect management.” *Proceedings of the 26th CIB W078 Conference Managing IT in Construction*,1–3, Istanbul, Turkey.

Kaufmann, K., and Anderegg, R. (2008). “3D-construction applications III: GPS-based compaction technology.” *Proceedings of 1st International Conference on Machine Control & Guidance,* 1–10, ETH Zurich Switzerland.

Kim, S.-K., Seo, J., and Russell, J. S. (2012). “Intelligent navigation strategies for an automated earthwork system.” *Automation in Construction*, Elsevier, 21, 132–147.

Kiviniemi, M., Sulankivi, K., Kähkönen, K., Mäkelä, T., and Merivirta, M.-L. (2011). “BIM-based safety management and communication for building construction.” *VTT Technical Research Centre of Finland*.

Krötzsch, M., Simancik, F., and Horrocks, I. (2012). “A description logic primer.” *arXiv preprint arXiv:1201.4089*.

Kulhawy, F. H., and Chen, J.-R. (2009). “Identification and description of soils containing very coarse fractions.” *Journal of Geotechnical and Geoenvironmental Engineering*, American Society of Civil Engineers, 135(5), 635–646.

Laakso, M., and Kiviniemi, A. (2012). “The IFC standard: A review of history, development, and standardization, information technology.” *ITcon*, Open Publishing Services, 17(9), 134–161.

Labban, R., AbouRizk, S., Haddad, Z., and Elsersy, A. (2013). “A discrete event simulation model of asphalt paving operations.” *Proceedings of the 2013 Winter Simulations Conference (WSC)*. IEEE Press, 3215–3224.

Le, Q. T., Lee, D. Y., and Park, C. S. (2014). "A social network system for sharing construction safety and health knowledge." *Automation in Construction*, Elsevier, 46, 30–37.

Le, T., Hassan, F., Le, C., & Jeong, H. D. (2019). "Understanding dynamic data interaction between civil integrated management technologies: A review of use cases and enabling techniques." *International Journal of Construction Management*, 1–22.

Leacock, C., and Chodorow, M. (1998). "Combining local context and WordNet similarity for word sense identification." *WordNet: An electronic lexical database*, 49(2), 265–283.

Lee, Chi, H., Wang, J., Wang, X., and Park, C.-S. (2016). "A linked data system framework for sharing construction defect information using ontologies and BIM environments." *Automation in Construction*, Elsevier, 68, 102–113.

Lee, and Kim, B.-G. (2011). "IFC extension for road structures and digital modeling." *Procedia Engineering*, Elsevier, 14, 1037–1042.

Lee, Kim, K.-R., and Yu, J.-H. (2014). "BIM and ontology-based approach for building cost estimation." *Automation in Construction*, Elsevier, 41, 96–105.

Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. (1990). "Cyc: toward programs with common sense." *Communications of the ACM*, ACM, 33(8), 30–49.

Leukel, J., Hepp, M., Schmitz, V., and Tribowski, C. (2006). "Ontologizing b2b message specifications: Experiences from adopting the plib ontology for commercial product data." In *2006 IEEE International Conference on e-Business Engineering (ICEBE'06)*. IEEE, 146–153.

Leung, N. K., Lau, S. K., and Tsang, N. (2014). "Reuse existing ontologies in an ontology development process–an integration-oriented ontology development methodology." *International Journal of Web Science*, Inderscience Publishers, 2(3), 159–180.

Li, H., Chan, G., Wong, J. K. W., and Skitmore, M. (2016). "Real-time locating systems applications in construction." *Automation in Construction*, Elsevier, 63, 37–47.

Liu, H., Lu, M., and Al-Hussein, M. (2016). "Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry." *Advanced Engineering Informatics*, Elsevier, 30(2), 190–207.

Lu, Li, Q., Zhou, Z., and Deng, Y. (2015). "Ontology-based knowledge modeling for automated construction safety checking." *Safety Science*, Elsevier, 79, 11–18.

Lu, M., Chen, W., Shen, X., Lam, H.-C., and Liu, J. (2007). "Positioning and tracking construction vehicles in highly dense urban areas and building construction sites." *Automation in Construction*, Elsevier, 16(5), 647–656.

Lu, Y. (2017). "Industry 4.0: A survey on technologies, applications and open research issues." *Journal of Industrial Information Integration*, Elsevier, 6, 1–10.

Maalek, R., and Sadeghpour, F. (2013). “Accuracy assessment of Ultra-Wide Band technology in tracking static resources in indoor construction scenarios.” *Automation in Construction*, Elsevier, 30, 170–183.

Marković, M., and Gostojić, S. (2020). “A knowledge-based document assembly method to support semantic interoperability of enterprise information systems.” *Enterprise Information Systems*, Taylor & Francis, 1–20.

Martínez, J. C. (1998). “Earthmover-simulation tool for earthwork planning.” *Proceedings of the 1998 Winter Simulation Conference. Proceedings (Cat. No. 98CH36274)*. Vol. 2. IEEE Computer Society Press, 1263–1272.

Martinez-Rodriguez, J. L., Hogan, A., and Lopez-Arevalo, I. (2020). “Information extraction meets the semantic web: a survey.” *Semantic Web*, IOS Press, 1–81.

Mattila, M., Rantanen, E., and Hyttinen, M. (1994). “The quality of work environment, supervision and safety in building construction.” *Safety Science*, Elsevier, 17(4), 257–268.

McGuinness, D. L., and Van Harmelen, F. (2004). “OWL web ontology language overview.” *W3C Recommendation*, < http://www.w3.org/TR/owl-overview> (Sep. 5, 2020).

McLean, M. (2019). “How to Use Grade Control for Excavators—Volvo Dig Assist.” <https://volvoceblog.com/how-to-use-grade-control-for-excavators-volvo-dig-assist-part-/> (Jan. 27, 2021).

Meenachi, N. M., and Baba, M. S. (2012). “A survey on usage of ontology in different domains.” *International Journal of Applied Information Systems*, 4(2), 46–55.

Melenbrink, N., Werfel, J., and Menges, A. (2020). “On-site autonomous construction robots: Towards unsupervised building.” *Automation in Construction*, Elsevier, 119, 103312.

Mrugalska, B., and Wyrwicka, M. K. (2017). “Towards lean production in industry 4.0.” *Procedia Engineering*, Elsevier, 182, 466–473.

Musen, M. A. (2015). “The Protégé Project: A Look Back and a Look Forward.” *AI matters*, 1(4), 4–12.

NBS. (2020). “Unified classification system.” <https://www.thenbs.com/our-tools/uniclass-2015> (Oct. 20, 2020).

Neuhaus, H., and Compton, M. (2009). “The semantic sensor network ontology.” *In AGILE Workshop on Challenges in Geospatial Data Harmonisation*, 1–33.

Noran, O. (2004). “UML vs. IDEF: An Ontology-Oriented Comparative Study in View of Business Modelling.” *6th International Conference on Enterprise Information Systems (ICEIS 2004), 3, 674–682*. Porto, Portugal.

Noy, N. F., Griffith, N., and Musen, M. A. (2008). "Collecting community-based mappings in an ontology repository." *International Semantic Web Conference*, 371–386. Springer, Berlin, Germany.

Nunnally, S. W. (2004). *Construction methods and management*. Pearson Prentice Hall Upper Saddle River, NJ, USA.

OGC. (2020). "Sensor Model Language (SensorML)." *Sensor Model Language (SensorML)*, <https://www.ogc.org/standards/sensorml> (Sep. 10, 2020).

OmniClass. (2020). "OmniClass, a strategy for classifying the built environment." <www.omniclass.org/> (Sep. 5, 2020).

OSHA. (2020a). "Occupational Safety and Health Administration, Requirements for protective systems." <www.osha.gov/pls/oshaweb/owadisp.show_document?p_table=standards&p_id=10776> (Aug. 15, 2020).

OSHA. (2020b). "1926 Subpart P - Subpart P—Excavations." <https://www.osha.gov/laws-regs/regulations/standardnumber/1926/1926SubpartP> (Sep. 10, 2020).

OSHA. (2020c). "1926 Subpart P App B - Sloping and Benching." <https://www.osha.gov/laws-regs/regulations/standardnumber/1926/1926SubpartPAppB> (Sep. 10, 2020).

OSHA. (2020d). "1926 Subpart P App A - Soil Classification." <https://www.osha.gov/laws-regs/regulations/standardnumber/1926/1926SubpartPAppA> (Sep. 10, 2020).

Parente, M., Cortez, P., and Correia, A. G. (2015). "An evolutionary multi-objective optimization system for earthworks." *Expert Systems with Applications*, Elsevier, 42(19), 6674–6685.

Park, J., Marks, E., Cho, Y. K., and Suryanto, W. (2015). "Performance test of wireless technologies for personnel and equipment proximity sensing in work zones." *Journal of Construction Engineering and Management*, American Society of Civil Engineers, 142(1), 04015049.

Park, M., Lee, K., Lee, H., Jiayi, P., and Yu, J. (2013). "Ontology-based construction knowledge retrieval system." *KSCE Journal of Civil Engineering*, Springer, 17(7), 1654–1663.

Park, M., Makhmalbaf, A., and Brilakis, I. (2011). "Comparative study of vision tracking methods for tracking of construction site resources." *Automation in Construction*, Elsevier, 20(7), 905–915.

Pavement Interactive. (2010). "Hot Mix Asphalt (HMA) Pavement." <www.pavementinteractive.org/article/hma-pavement/> (Apr. 20, 2020).

Pedro, A., Lee, D. Y., Hussain, R., and Park, C. S. (2017). "Linked data system for sharing construction safety information." *Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC)*, IAARC Publications.

Peurifoy, L. R., Schexnayder, C. J., Shapira, A., and Schmitt, R. L. (2010). *Construction Planning, Equipment, and Methods*. McGraw-Hill Education, New York.

Pfaff, M., Neubig, S., and Krcmar, H. (2018). "Ontology for Semantic Data Integration in the Domain of IT Benchmarking." *Journal on Data Semantics*, Springer, 7(1), 29–46.

Pinto, A., Nunes, I. L., and Ribeiro, R. A. (2011). "Occupational risk assessment in construction industry–Overview and reflection." *Safety Science*, Elsevier, 49(5), 616–624.

Pinto, S., Gómez-Pérez, A., and Martins, J. P. (1999). "Some issues on ontology integration." *In Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI) and the Scandinavian AI Societies*, 1–7, Stockholm, Sweden.

Pinto, S., and Martins, J. P. (2001). "A methodology for ontology integration." *Proceedings of the 1st international Conference on Knowledge Capture*, 131–138.

Pratt, R. I., Carolina, S., Roush, N. H., Virginia, W., Ruff, W. T., Schneider, M. M. E., and Dakota, S. (2000). "American Association of State Highway and Transportation Officials."

Quinn, C., Shabestari, A. Z., Misic, T., Gilani, S., Litoiu, M., and McArthur, J. (2020). "Building automation system-BIM integration using a linked data structure." *Automation in Construction*, Elsevier, 118, 103257.

Radulovic, F., Poveda-Villalón, M., Vila-Suero, D., Rodríguez-Doncel, V., García-Castro, R., and Gómez-Pérez, A. (2015). "Guidelines for Linked Data Generation and Publication: An Example in Building Energy Consumption." *Automation in Construction*, Elsevier, 57, 178–187.

Razavi, S. N., and Haas, C. T. (2010). "Multisensor data fusion for on-site materials tracking in construction." *Automation in Construction*, Elsevier, 19(8), 1037–1046.

Razuri, C., Alarcón, L. F., & Diethelm, S. (2007). "Evaluating the effectiveness of safety management practices and strategies in construction projects." *In Proceedings of the 15th Conference of the IGLC,* 271–281. Michigan, USA.

Rebolj, D., Tibaut, A., Čuš-Babič, N., Magdič, A., and Podbreznik, P. (2008). "Development and application of a road product model." *Automation in construction*, Elsevier, 17(6), 719–728.

Rezazadeh Azar, E., and McCabe, B. (2011). "Automated visual recognition of dump trucks in construction videos." *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, 26(6), 769–781.

Russell, S. J., Norvig, P., and Davis, E. (2010). *Artificial intelligence: a modern approach*. Prentice Hall series in artificial intelligence, Prentice Hall, Upper Saddle River, N.J.

Russomanno, D. J., Kothari, C. R., and Thomas, O. A. (2005). "Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models." 637–643.

SAE. (2018). “Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.” <https://www.sae.org/standards/content/j3016_201806/> (Jan. 14, 2021).

Sakamoto, S., Kano, N., Igarashi, T., and Tomita, H. (2012). “Laser positioning system using rfid-tags.” Vilnius Gediminas Technical University, Department of Construction Economics & Property, 1.

Scania. (2019). “A new cabless concept – revealing Scania AXL.” <https://www.scania.com/uk /en/home/experience-scania/news-and-events/news/2019/09 /revealing-scania-axl.html> (Jan. 27, 2021).

Schreiber, G., Wielinga, B., and Jansweijer, W. (1995). “The KACTUS view on the ‘O’word.” *International Joint Conferences on Artificial Intelligence (IJCAI) workshop on basic ontological issues in knowledge sharing*, Citeseer, 159–168.

Siddiqui, H. (2014). “UWB RTLS for Construction Equipment Localization: Experimental Performance Analysis and Fusion with Video Data.” Master thesis, Concordia University, Department of Building, Civil, and Environmental Engineering.

Sifakis, J. (2019). “Autonomous systems–an architectural characterization.” *Models, Languages, and Tools for Concurrent and Distributed Programming*, Springer, 388–410.

Singh, J., and Sharan, A. (2014). “Lexical ontology-based computational model to find semantic similarity.” *Intelligent Computing, Networking, and Informatics*, Springer, 119–128.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). “Pellet: A practical owl-dl reasoner.” *Journal of Web Semantics*, Elsevier, 5(2), 51–53.

Skibniewski, M. J., and Jang, W. (2009). “Simulation of accuracy performance for wireless sensor-based construction asset tracking.” *Computer-Aided Civil and Infrastructure Engineering*, Wiley Online Library, 24(5), 335–345.

Skobelev, P., Zhilyaev, A., Larukhin, V. B., Grachev, S., and Simonova, E. V. (2020). “Ontology-based Open Multi-agent Systems for Adaptive Resource Management.” *In Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020)*, 127–135.

Smith, S., Osborne, J., and Forde, M. (1996). “The use of a discrete-event simulation model with Erlang probability distributions in the estimation of earthmoving production.” *Civil Engineering Systems*, Taylor & Francis Group, 13(1), 25–44.

Stephenson, H. L. (2007). “Scheduling Management: Classifications vs Levels.” *Proceedings of the 51st Annual Meeting of the Association for the Advancement of Cost Engineering.*, American Association of Cost Engineers, 1–10.

Swartout, B., Patil, R., Knight, K., and Russ, T. (1996). "Toward distributed use of large-scale ontologies." *In Proceedings of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems*, 138–148.

Synopsys. (2020). "The 6 Levels of Vehicle Autonomy Explained, Synopsys Automotive." <https://www.synopsys.com/automotive/autonomous-driving-levels.html> (Jan. 14, 2021).

Taher, A., Vahdatikhaki, F., and Hammad, A. (2017). "Towards Developing an Ontology for Earthwork Operations." *In Proceedings of International Workshop on Computing in Civil Engineering (ASCE)*, Proceedings, 101–108.

Taivalsaari, A. (1996). "Classes vs. Prototypes Some Philosophical and Historical Observations." *Journal of Object-Oriented Programming*, Citeseer, 10(7), 44–50.

Teizer, J., Caldas, C. H., and Haas, C. T. (2007). "Real-time three-dimensional occupancy grid modeling for the detection and tracking of construction resources." *Journal of Construction Engineering and Management*, American Society of Civil Engineers, 133(11), 880–888.

Thomas, E., Schott, P., Bowman, J., Snyder, J., and Spare, N. (2018). "Construction Disconnected." <https://www.plangrid.com/ebook/construction-disconnected/> (Nov. 23, 2020).

Thomopoulos, R., Destercke, S., Charnomordic, B., Johnson, I., & Abécassis, J. (2013). An iterative approach to build relevant ontology-aware data-driven models. *Information Sciences*, 221, 452–472.

UBIMET. (2020). "Weather Energy Data & Energy Forecasts." <https://www.ubimet.com/en/services/weather-energy-forecasts/> (Jul. 15, 2020).

Uschold, M., and Gruninger, M. (1996). *Ontologies: Principles, methods and applications*. Artificial Intelligence Applications Institute AIAI., University of Edinburgh, 11(2), 93–136.

Vahdatikhaki, F. (2015). "Towards Smart Earthwork Sites Guidance and Multi- Agent Systems." Doctoral dissertation, Department of Building, Civil, and Environmental Engineering, Concordia University.

Vahdatikhaki, F., Langari, S. M., Taher, A., El Ammari, K., and Hammad, A. (2017). "Enhancing coordination and safety of earthwork equipment operations using Multi-Agent System." *Automation in Construction*, Elsevier, 81, 267–285.

Valero, E., Adán, A., and Bosché, F. (2015). "Semantic 3D reconstruction of furnished interiors using laser scanning and RFID technology." *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, 30(4), 04015053.

Velardi, P. (2006). "Evaluation of ontolearn, a methodology for automatic population of domain ontologies." *Ontology Learning from Text: Methods, Applications and Evaluation, Frontiers in Artificial Intelligence and Applications*, IOS Press, 92–106.

Viljamaa, E., and Peltomaa, I. (2014). “Intensified construction process control using information integration.” *Automation in Construction*, Elsevier, 39, 126–133.

Wang, H. H., and Boukamp, F. (2011). “Ontology-based representation and reasoning framework for supporting job hazard analysis.” *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, 25(6), 442–456.

Wang, H. H., Boukamp, F., and Elghamrawy, T. (2010). “Ontology-based approach to context representation and reasoning for managing context-sensitive construction information.” *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, 25(5), 331–346.

Weber, J., Stolipin, J., König, M., and Wenzel, S. (2019). “Ontology for Logistics Requirements on a 4D BIM for Semi-Automatic Storage Space Planning.” IAARC Publications, 560–567.

Williams, O. S., Hamid, R. A., and Misnan, M. S. (2018). “Accident causal factors on the building construction sites: a review.” *International Journal of Built Environment and Sustainability*, 5(1).

Woo, S., Jeong, S., Mok, E., Xia, L., Choi, C., Pyeon, M., and Heo, J. (2011). “Application of WiFi-based indoor positioning system for labor tracking at construction sites: A case study in Guangzhou MTR.” *Automation in Construction*, Elsevier, 20(1), 3–13.

Wu, D. (2013). “Building knowledge modeling: Integrating knowledge in BIM.” *In Proceedings of the 30th International Conference of CIB W078,* Beijing.

Wu, W., Yang, H., Chew, D. A., Yang, S., Gibb, A. G., and Li, Q. (2010). “Towards an autonomous real-time tracking system of near-miss accidents on construction sites.” *Automation in Construction*, Elsevier, 19(2), 134–141.

Wu, Z., and Palmer, M. (1994). “Verbs semantics and lexical selection.” *In Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 133–138.

Xing, X., Zhong, B., Luo, H., Li, H., & Wu, H. (2019). “Ontology for safety risk identification in metro construction.” *Computers in Industry*, 109, 14–30.

Yu, J., Thom, J. A., and Tam, A. (2007). “Ontology evaluation using wikipedia categories for browsing.” *In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, ACM, 223–232.

Zeb, J., and Froese, T. (2012). “Transaction ontology in the domain of infrastructure management.” *Canadian Journal of Civil Engineering*, NRC Research Press, 39(9), 993–1004.

Zhang, C., and Hammad, A. (2011). "Multiagent approach for real-time collision avoidance and path replanning for cranes." *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, 26(6), 782–794.

Zhang, M., Cao, T., and Zhao, X. (2017). "Applying sensor-based technology to improve construction safety management." *Sensors*, Multidisciplinary Digital Publishing Institute, 17(8), 1841.

Zhang, S., Boukamp, F., and Teizer, J. (2015). "Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (JHA)." *Automation in Construction*, Elsevier, 52, 29–41.

Zhang, M., Shi, R., and Yang, Z. (2020). "A critical review of vision-based occupational health and safety monitoring of construction site workers." *Safety science*, Elsevier, 126, 104658.

Zhang, S., Teizer, J., Lee, J.-K., Eastman, C. M., and Venugopal, M. (2013). "Building information modeling (BIM) and safety: Automatic safety checking of construction models and schedules." *Automation in Construction*, Elsevier, 29, 183–195.

Zhao, L., Ichise, R., Liu, Z., Mita, S., and Sasaki, Y. (2017). "Ontology-based driving decision making: A feasibility study at uncontrolled intersections." *IEICE TRANSACTIONS on Information and Systems*, The Institute of Electronics, Information and Communication Engineers, 100(7), 1425–1439.

Zhong, B., Ding, L., Luo, H., Zhou, Y., Hu, Y., and Hu, H. (2012). "Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking." *Automation in Construction*, Elsevier, 28, 58–70.

Zhou, Z., Goh, Y. M., and Shen, L. (2016). "Overview and Analysis of Ontology Studies Supporting Development of the Construction Industry." *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, 04016026.

APPENDIX A THE RDF FILE FOR EW-ONTO

This RDF file includes parts of the object properties, data properties and the main classes in EW-Onto. The full-length RDF files for EW-Onto and IEW-Onto can be found at https://www.ew-onto.info/.

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#"
    xml:base="http://www.ew-onto.info/EarthworkOntologies/EW-Onto"
    xmlns:ew="http://www.semanticweb.org/umroot/ontologies/2018/5/EW#"
    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:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:skos="http://www.w3.org/2004/02/skos/core#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:swrla="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:EW-SKB="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#">
   <owl:Ontology rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto">
      <owl:versionIRI rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto"/>
      <About>This is the first ontology for earthwork domain. we called it (EW-Onto)</About>
      <Author>Alhusain Taher,PhD, Concordia University</Author>
      <Contact>Alhusain Taher</Contact>
      <Creation_Date>2019-06-11</Creation_Date>
      <Keywords>Earthwork, Operation, Process, Task, MicroTask</Keywords>
      <Name>Earthwork Ontology :EW-Onto</Name>
      <Short_name>EW-Onto</Short_name>
      <Syntax_Format>RDF</Syntax_Format>
      <Version_Number rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1.0</Version_Number>
   </owl:Ontology>

<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Object Properties
//
///////////////////////////////////////////////////////////////////////////////////////
-->
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ConnectedTo -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ConnectedTo">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
</owl:ObjectProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#coordinates -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#coordinates">
<owl:inverseOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isCoordinatedBy"/>
</owl:ObjectProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has">
<owl:inverseOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isPartOf"/>
</owl:ObjectProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasInstanceMicroTask -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasInstanceMicroTask">
```

```
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
</owl:ObjectProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasInstanceProcess -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasInstanceProcess">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
</owl:ObjectProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasInstanceTask -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasInstanceTask">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
</owl:ObjectProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasInvolved -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasInvolved">
<owl:inverseOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isInvolveIn"/>
</owl:ObjectProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isCoordinatedBy -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isCoordinatedBy"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isInvolveIn -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isInvolveIn"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isOperatedBy -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isOperatedBy">
<owl:inverseOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#operates"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthworkEquipment"/>
<rdfs:range rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Operator"/>
</owl:ObjectProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isPartOf -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isPartOf"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isPerformedBy -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#isPerformedBy">
<owl:inverseOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#performs"/>
</owl:ObjectProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#operates -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#operates">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#performs -->
<owl:ObjectProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#performs"/>
<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Data properties
//
///////////////////////////////////////////////////////////////////////////////////////
-->
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EqID -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EqID">
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Resource"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasAmount -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasAmount">
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Resource"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasBrand -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasBrand">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasColor -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasColor">
```

```
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasDepth -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasDepth">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasDistance -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasDistance">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasGeneralOperationDescription -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasGeneralOperationDescription">
<rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topDataProperty"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasIdNumber -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasIdNumber">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasInstanceID -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasInstanceID">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasLabel -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasLabel">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasLastName -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasLastName">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasLocation -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasLocation">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasMade -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasMade">
<rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topDataProperty"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasMicroTaskDescription -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasMicroTaskDescription">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasMicroTaskID -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasMicroTaskID">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasMicroTaskName -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasMicroTaskName">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasModel -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasModel">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasName -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasName">
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasNumberOfSuitableEquipment1 -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasNumberOfSuitableEquipment1">
<rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topDataProperty"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasNumberOfSuitableEquipment2 -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasNumberOfSuitableEquipment2">
<rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topDataProperty"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasOperationDescription -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasOperationDescription">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasOperationId -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasOperationId">
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Operation"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasOperationName -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasOperationName">
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Operation"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasPartNo -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasPartNo">
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Part"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasPlasticity -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasPlasticity">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasPriority -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasPriority">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasProcessDescription -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasProcessDescription">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasProcessID -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasProcessID">
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthworkProcess"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasProcessName -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasProcessName">
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthworkProcess"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasRecourceName -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasRecourceName">
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Resource"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasSize -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasSize">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasStructural -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasStructural">
<rdfs:comment>Granular
```

```
Cohesive   or
Granular Cohesionless</rdfs:comment>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasTaskDescription -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasTaskDescription">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasTaskID -->
<owl:DatatypeProperty rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#hasTaskID">
<rdfs:domain rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Task"/>
</owl:DatatypeProperty>
<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Classes
//
///////////////////////////////////////////////////////////////////////////////////////
-->
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor">
<owl:disjointWith rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Part"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Agent -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Agent">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Area -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Area">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Geographic-Information"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ArticulatedRearDump -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ArticulatedRearDump">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Truck"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Attachment -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Attachment">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Resource"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#AugerScraper -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#AugerScraper">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Scraper"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Backhoe -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Backhoe">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Hoe"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Blade -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Blade">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Part"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Boom -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Boom">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Part"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#BoomCylinder -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#BoomCylinder">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Part"/>
</owl:Class>
```

```xml
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#BottomDumpTrailer -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#BottomDumpTrailer">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Truck"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CollectedData -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CollectedData">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Data"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CompetentEngineering -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CompetentEngineering">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor"/>
<rdfs:comment>OSHA:1926.32(f)
&quot;Competent person&quot; means one who is capable of identifying existing and predictable hazards in the surroundings or working conditions which are unsanitary, hazardous, or dangerous to employees, and who has authorization to take prompt corrective measures to eliminate them.</rdfs:comment>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ComputerVision -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ComputerVision">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#RemoteSensing"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Image"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Drone">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#RemoteSensing"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Image"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#DumpingZone -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#DumpingZone">
<owl:equivalentClass rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#WorkZone"/>
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Zone"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthmovingOperation -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthmovingOperation">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Operation"/>
</owl:Class>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#performs"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Task"/>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:seeAlso>OSHA
Part Number:    1926
Part Number Title:    Safety and Health Regulations for Construction
Subpart:    1926 Subpart O
Subpart Title:    Motor Vehicles, Mechanized Equipment, and Marine Operations
Standard Number:    1926.601
Title:    Motor vehicles.
GPO Source:    e-CFR</rdfs:seeAlso>
<rdfs:seeAlso>https://www.osha.gov/laws-regs/regulations/standardnumber/1926/1926.601</rdfs:seeAlso>
</owl:Class>
```

```
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthworkProcess -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthworkProcess">
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor"/>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Geographic-Information"/>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformanceGuidline"/>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Product"/>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:seeAlso>Part Number:    1926
Part Number Title:    Safety and Health Regulations for Construction
Subpart:    1926 Subpart P
Subpart Title:    Excavations
Standard Number:    1926 Subpart P
Title:    Subpart P—Excavations
GPO Source:    e-CFR</rdfs:seeAlso>
<rdfs:seeAlso>https://www.osha.gov/laws-regs/regulations/standardnumber/1926/1926SubpartP</rdfs:seeAlso>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Execution -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Execution">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ManagementModality"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Expert -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Expert">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Extend -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Extend">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#TemporalModality"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#FirstMultiBenchExcavaion -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#FirstMultiBenchExcavaion">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#WorkZone"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Fleet -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Fleet">
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Location"/>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#FrontShovel -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#FrontShovel">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Shovel"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GeneralCoordinator -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GeneralCoordinator">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Geographic-Information -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Geographic-Information"/>
```

```xml
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GradallEquipment -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GradallEquipment">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Grader"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Grader -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Grader">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthworkEquipment"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GraderEquipment -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GraderEquipment">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Grader"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GradingMicroTask -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GradingMicroTask">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Micro-Task"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GradingOperation -->
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HaulingMicroTask -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HaulingMicroTask">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Micro-Task"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HaulingOperation -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HaulingOperation">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Operation"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HaulingProcess -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HaulingProcess">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthworkProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HaulingTask -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HaulingTask">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Task"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Hazard -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Hazard"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HazardWorkzone -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HazardWorkzone">
<rdfs:comment>The HazardWorkspace contains the places that need  to be protected from cave-ins by an adequate protective
system</rdfs:comment>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HeavyDutyBucket -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#HeavyDutyBucket">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#BucketAttachment"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Hoe -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Hoe">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Excavator"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#IdentifyAndTestExcavatedSoil -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#IdentifyAndTestExcavatedSoil">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ExcavationProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#IdentifyBaseCourseMaterialsandBoundaries -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#IdentifyBaseCourseMaterialsandBoundaries">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GradingProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#IdentifyExcavationandEmbankmentBoundaries -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#IdentifyExcavationandEmbankmentBoundaries">
```

```
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ExcavationProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#IdentifyExcavationorBorrowedMaterials -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#IdentifyExcavationorBorrowedMaterials">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CompactionProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#IdentifyTypeandAmountofSoil -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#IdentifyTypeandAmountofSoil">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ExcavationProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Image -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Image">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CollectedData"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InitiatingModality -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InitiatingModality">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ManagementModality"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspactionPersonnel -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspactionPersonnel">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionMethod -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionMethod">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Method"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CollectedData"/>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionTool"/>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Report"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionProcess -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionProcess">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthworkProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionReport -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionReport">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Data"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionTool -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionTool">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Tool"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Loader -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Loader">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Excavator"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Location -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Location">
```

```xml
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Geographic-Information"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ManagementModality -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ManagementModality">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Modality"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ManualMeasurement -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ManualMeasurement">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Measurement"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Material -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Material">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Resource"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Measurement -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Measurement">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionMethod"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#MeasurementResult -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#MeasurementResult">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CollectedData"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#MechanicalEngineering -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#MechanicalEngineering">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EngineeringModality"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Method -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Method"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Micro-Task -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Micro-Task">
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Schedule"/>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#has"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Technique"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#MiniHoe -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#MiniHoe">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Hoe"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Modality -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Modality"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#MultipleBenchExcavation -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#MultipleBenchExcavation">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#WorkZone"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#NonphysicalProduct -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#NonphysicalProduct">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Product"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#NotPlanned -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#NotPlanned">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SituationModality"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#NuclearDensityGauge -->
```

```xml
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#NuclearDensityGauge">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Non-DestructiveTesting"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Observation -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Observation"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#OntrackHoe -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#OnTrackHoe">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Hoe"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#OperationAgent -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#OperatorAgent">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Agent"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Operator -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Operator">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Operates"/>
<owl:someValuesFrom rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthworkEquipment"/>
</owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#WorkerOnFoot"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Owner -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Owner">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PadDrumVibrator -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PadDrumVibrator">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Compactor"/>
<rdfs:comment>same as ped foot roller</rdfs:comment>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Part -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Part">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Resource"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Path -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Path"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformClearingandGrubbing -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformClearingandGrubbing">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CleaningandGrubbingProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformCompaction -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformCompaction">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CompactionProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformExcavation -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformExcavation"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformGrading -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformGrading">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#GradingProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformHauling -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformHauling">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CleaningandGrubbingProcess"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformanceGuidline -->
```

```xml
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PerformanceGuidline">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#RuleandRegulation"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PhysicalProduct -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PhysicalProduct">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Product"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Planned -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Planned">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SituationModality"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Planning -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Planning">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ManagementModality"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Platform -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Platform"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PneumaticTire -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PneumaticTire">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Compactor"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PointCloud -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#PointCloud">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CollectedData"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#RegisteredProfessionalEngineer -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#RegisteredProfessionalEngineer">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Actor"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#RemoteSensing -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#RemoteSensing">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Measurement"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Report -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Report">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Data"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ResourceData -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ResourceData">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Data"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Rigid-FrameRearDump -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Rigid-FrameRearDump">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Truck"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Ripper -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Ripper">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Part"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#RuleandRegulation -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#RuleandRegulation"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Schedule -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Schedule"/>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Scraper -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Scraper">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#EarthworkEquipment"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SensorDevice -->
```

```xml
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SensorDevice">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#ElectronicEquipment"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SheepsFoot -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SheepsFoot">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Compactor"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Soil -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Soil">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Material"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilBrushingMachine -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilBrushingMachine">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SupportEquipment"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilClayLevel -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilClayLevel">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Soil"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilLayer -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilLayer">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Soil"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilMoisture -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilMoisture">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Soil"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilTypeA -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilTypeA">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilClassification"/>
<rdfs:isDefinedBy>OSHA:Type A Soils are cohesive soils with an unconfined compressive strength of 1.5 tons per square foot (tsf) (144 kPa) or greater. Examples of Type A cohesive soils are often: clay, silty clay, sandy clay, clay loam and, in some cases, silty clay loam and sandy clay loam. (No soil is Type A if it is fissured, is subject to vibration of any type, has previously been disturbed, is part of a sloped, layered system where the layers dip into the excavation on a slope of 4 horizontal to 1 vertical (4H:1V) or greater, or has seeping water</rdfs:isDefinedBy>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilTypeB -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilTypeB">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilClassification"/>
<rdfs:isDefinedBy>OSHA:Type B Soils are cohesive soils with an unconfined compressive strength greater than 0.5 tsf (48 kPa) but less than 1.5 tsf (144 kPa). Examples of other Type B soils are: angular gravel; silt; silt loam; previously disturbed soils unless otherwise classified as Type C; soils that meet the unconfined compressive strength or cementation requirements of Type A soils but are fissured or subject to vibration; dry unstable rock; and layered systems sloping into the trench at a slope less than 4H:1V (only if the material would be classified as a Type B soil).</rdfs:isDefinedBy>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilTypeC -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilTypeC">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilClassification"/>
<rdfs:isDefinedBy>OSHA:Type C Soils are cohesive soils with an unconfined compressive strength of 0.5 tsf (48 kPa) or less. Other Type C soils include granular soils such as gravel, sand and loamy sand, submerged soil, soil from which water is freely seeping, and submerged rock that is not stable. Also included in this classification is material in a sloped, layered system where the layers dip into the excavation or have a slope of four horizontal to one vertical (4H:1V) or greater.</rdfs:isDefinedBy>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilWaterLevel -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilWaterLevel">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Soil"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#StableRock -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#StableRock">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#SoilClassification"/>
```

```xml
<rdfs:isDefinedBy>OSHA:
Stable Rock is natural solid mineral matter that can be excavated with vertical sides and remain intact while exposed. It is usually identified by a rock name such as granite or sandstone. Determining whether a deposit is of this type may be difficult unless it is known whether cracks exist and whether or not the cracks run into or away from the excavation.</rdfs:isDefinedBy>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#TeamCoordinatorAgent -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#TeamCoordinatorAgent">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Agent"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#TestResult -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#TestResult">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#CollectedData"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Testing -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#Testing">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#InspectionMethod"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#TiltingDitchCleaningBucket -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#TiltingDitchCleaningBucket">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#BucketAttachment"/>
</owl:Class>
<!-- http://www.ew-onto.info/EarthworkOntologies/EW-Onto#TimePeriod -->
<owl:Class rdf:about="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#TimePeriod">
<rdfs:subClassOf rdf:resource="http://www.ew-onto.info/EarthworkOntologies/EW-Onto#TemporalModality"/>
</owl:Class>
</rdf:RDF>
```

APPENDIX B EW-ONTO EVALAUATION

The next figures show the visualization of EW-Onto evaluation results for each question.

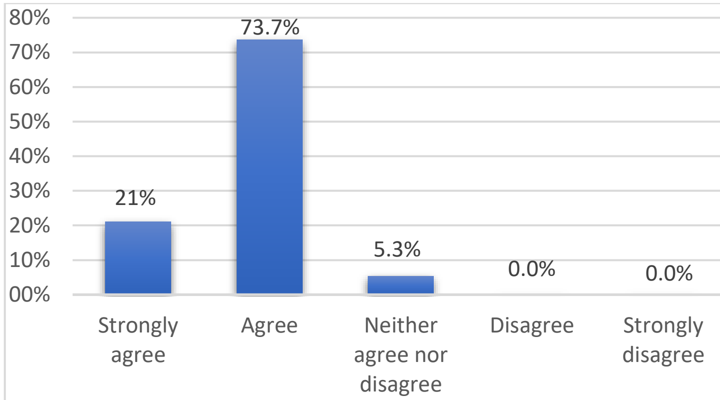


Figure B-1 Question 2 response

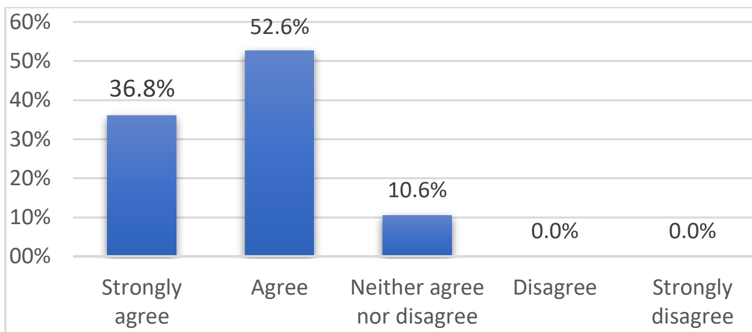


Figure B-2 Question 4 response

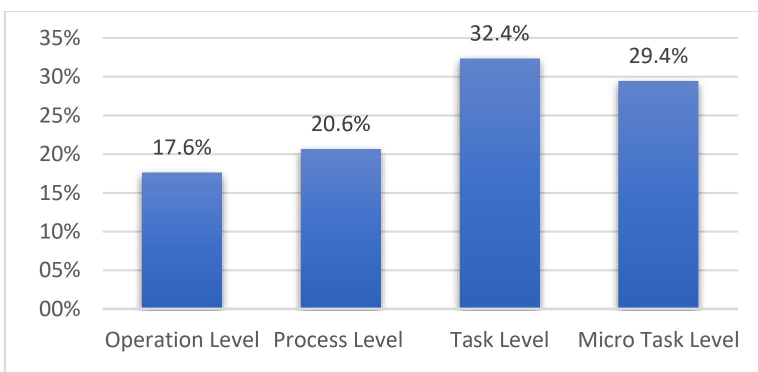


Figure B-3 Question 5 response

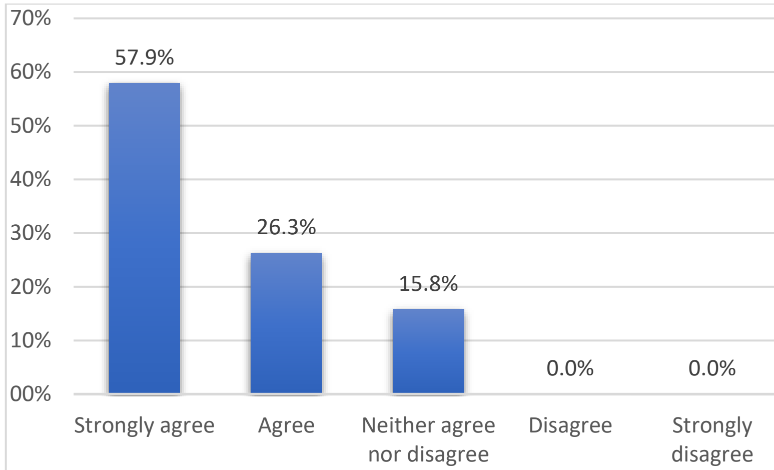


Figure B-4 Question 6 response

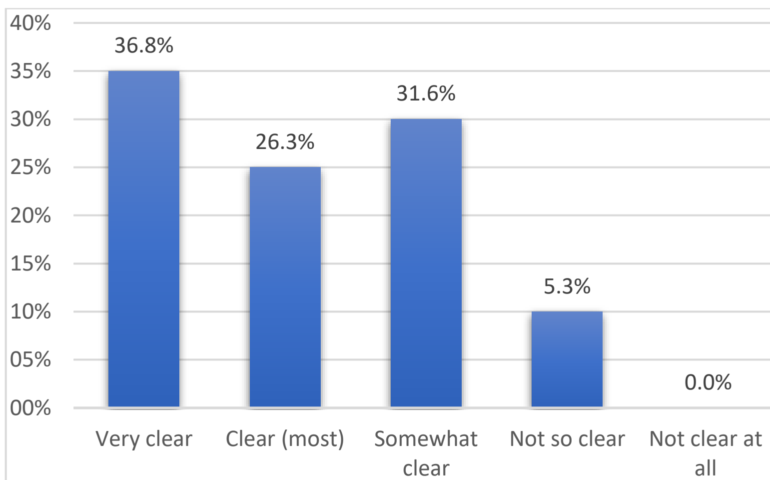


Figure B-5 Question 7 response

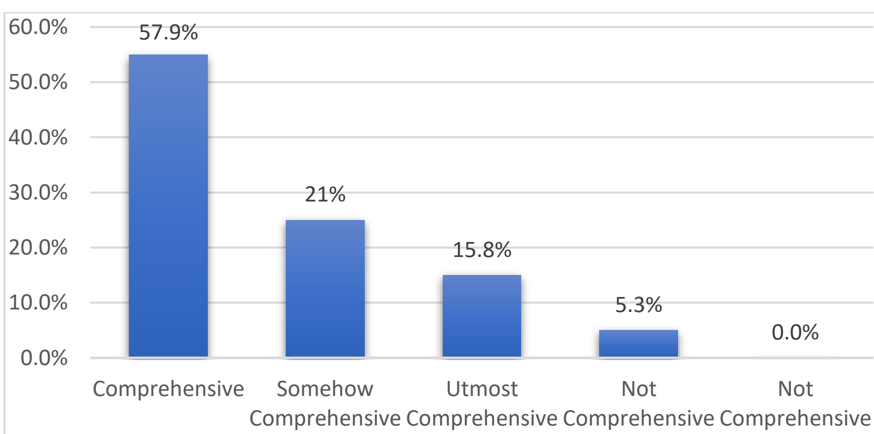


Figure B-6 Question 8 response

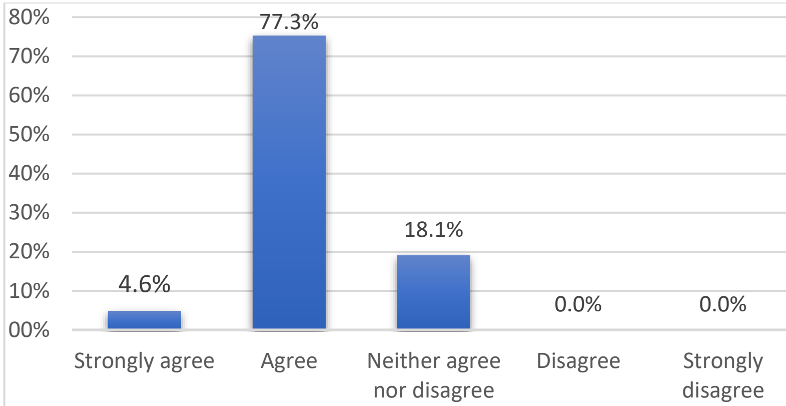


Figure B-7 Question 9 response

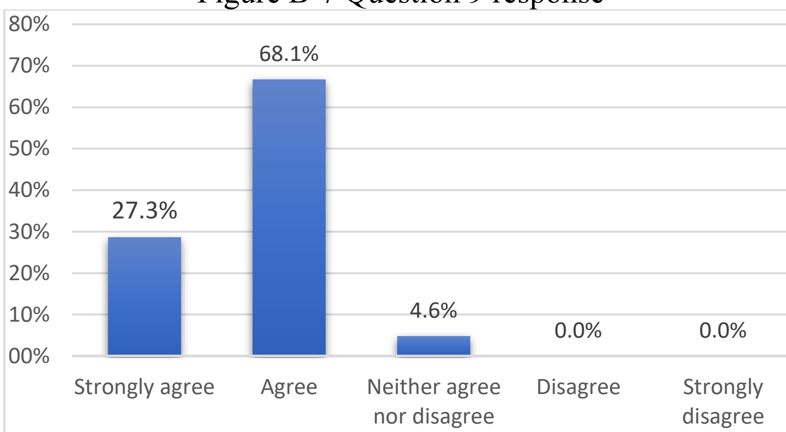


Figure B-8 Question 10 response

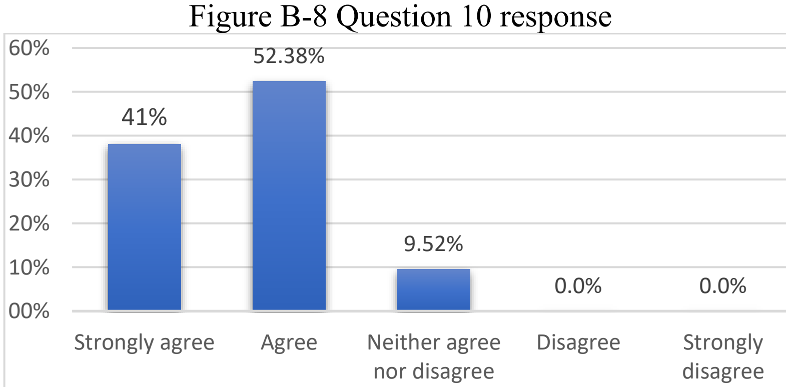


Figure B-9 Question 11 response

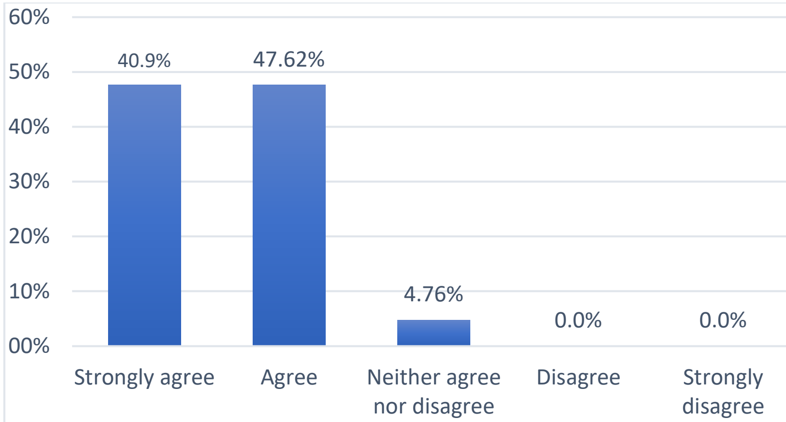


Figure B-10 Question 12 response

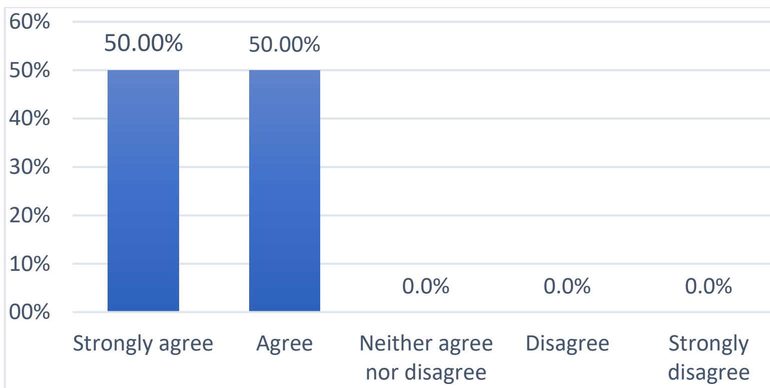


Figure B-11Question 13 response

APPENDIX C CODE FOR ONTOLOGIES COMPARISON

This code retrieves the main classes, the subclasses and their paths in the taxonomy for each ontology. Each class from the first ontology will be compared with all other classes in the second ontology.

```
/**************************/
   private List<string> GetAllClasses(string FilePath, string OntologyLabel, string OntologyUrl)
        {
            List<string> Classes = new List<string>();
            try
            {
                IGraph g = new Graph();
                g.LoadFromFile(FilePath);
                string GetAllClasses = @"
  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 " + OntologyLabel + @": <" + OntologyUrl + @">
SELECT  ?x
WHERE   {  ?x    rdf:type  owl:Class.
}
";
                //get all sub classes
                string Q2 = @"
  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 untitled-ontology-12: <http://www.semanticweb.org/alhusain/ontologies/2020/1/untitled-ontology-12#>
SELECT ?entity
WHERE {
  ?subclass rdfs:subClassOf <http://www.semanticweb.org/alhusain/ontologies/2020/1/untitled-ontology-12#Actor>.
  ?entity owl:type ?subclass.
}
";
                //get all sub classes
                string GetSuperClassesUntilRoot = @"
  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 untitled-ontology-12: <http://www.semanticweb.org/alhusain/ontologies/2020/1/untitled-ontology-12#>
  select ?superclass where {
  untitled-ontology-12:Student (rdfs:subClassOf|(owl:intersectionOf/rdf:rest*/rdf:first))* ?superclass .
}
";
                Object results = g.ExecuteQuery(GetAllClasses);
                if (results is SparqlResultSet)
                {
                    //SELECT/ASK queries give a SparqlResultSet
                    SparqlResultSet rset = (SparqlResultSet)results;
                    foreach (SparqlResult r in rset)
                    {
                        if (r["x"].ToString().StartsWith("http") ||
                            r["x"].ToString().StartsWith("https"))
                            Classes.Add(r["x"].ToString());
                        //TXT_TextToSearchIn.Text += r["x"].ToString() + Environment.NewLine;
                        //Do whatever you want with each Result
                    }
                }
                else if (results is IGraph)
                {
                    //CONSTRUCT/DESCRIBE queries give a IGraph
                    IGraph resGraph = (IGraph)results;
                    foreach (Triple t in resGraph.Triples)
                    {
                        //Do whatever you want with each Triple
                    }
                }
                else
                {
                    //If you don't get a SparqlResutlSet or IGraph something went wrong
                    //but didn't throw an exception so you should handle it here
                    MessageBox.Show("No Data Found.");
                }
                return Classes;
            }
```

```
            catch (Exception ex)
            {
                FRM_MSG f = new FRM_MSG();
                f.ShowDLG(AssemblyInfo.AssemblyTitle,
                ex.Message + "\n" + ex.StackTrace.ToString(),
                                    FRM_MSG.MSGIcon.Error,
                                    FRM_MSG.BTNS.One,
                                    new string[] { "Ok" });
                throw ex;
            }
        }
/*****************************************/
        private string GetLabel(string Class)
        {
            if (Class.LastIndexOf("#") <= 0)
            {
                return Class.Substring(Class.LastIndexOf("/") + 1);
            }
            else
            {
                return Class.Substring(Class.LastIndexOf("#") + 1);
            }
        }
        private string GetSuperClassesOfClassUntilRoot(string FilePath, string Class, string OntologyLabel,
string OntologyUrl)
        {
            try
            {
                string ClassPath = "";
                List<string> Classes = new List<string>();
                IGraph g = new Graph();
                g.LoadFromFile(FilePath);
                if (string.IsNullOrEmpty(Class) ||
                        string.IsNullOrWhiteSpace(Class))
                    throw new Exception("Empty Class");
                //get all sub classes
                string GetSuperClassesUntilRoot = @"
  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 : <" + OntologyUrl + @">
  select ?superclass where {
  <" + Class + @"> (rdfs:subClassOf|(owl:intersectionOf/rdf:rest*/rdf:first))* ?superclass .
  }
";
                string GetSuperClassesUntilRoot2 = @"
  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 : <" + OntologyUrl + @">
  SELECT ?superClass WHERE
{ <" + Class + @"> rdfs:subClassOf* ?superClass .
}
";
                // FILTER (!isBlank(rdfs:subClassOf))
                //FILTER(!isBlank(?superClass))
                Object results = g.ExecuteQuery(GetSuperClassesUntilRoot2);
                if (results is SparqlResultSet)
                {
                    //SELECT/ASK queries give a SparqlResultSet
                    SparqlResultSet rset = (SparqlResultSet)results;
                    foreach (SparqlResult r in rset)
                    {
                        if (IncludeSubclassOfAxiomValue)
                        {
                            Classes.Add(r["superClass"].ToString());
                        }
                        else
                        {
                            if (!r["superClass"].ToString().StartsWith("_:autos"))
                                Classes.Add(r["superClass"].ToString());
                        }
                        //Do whatever you want with each Result
                    }
                }
                else if (results is IGraph)
                {
                    //CONSTRUCT/DESCRIBE queries give a IGraph
                    IGraph resGraph = (IGraph)results;
                    foreach (Triple t in resGraph.Triples)
                    {
```

```
                    //Do whatever you want with each Triple
                }
            }
            else
            {
                //If you don't get a SparqlResutlSet or IGraph something went wrong
                //but didn't throw an exception so you should handle it here
                MessageBox.Show("No Data Found.");
            }
            // Classes.Reverse();
            foreach (string Cls in Classes)
            {
                string tmp = GetLabel(Cls);
                if (ClassPath == "")
                {
                    ClassPath = tmp;
                }
                else
                {
                    ClassPath += " > " + tmp;
                }
            }
            return ClassPath;
        }
        catch (Exception ex)
        {
            FRM_MSG f = new FRM_MSG();
            f.ShowDLG(AssemblyInfo.AssemblyTitle,
            ex.Message + "\n" + ex.StackTrace.ToString(),
                                FRM_MSG.MSGIcon.Error,
                                FRM_MSG.BTNS.One,
                                new string[] { "Ok" });
            throw ex;
        }
    }
             /****************************************/
              private void FillClasses( DoWorkEventArgs e)
    {
        try
        {
            UpdateCurrentOperation("Operation : Get First Onlogogy Name...");
            string OnlogogyAName = GetAllOntologies(FirstFilePath)[0];
            UpdateCurrentOperation("Operation : Get Second Onlogogy Name...");
            string OnlogogyBName = GetAllOntologies(SecondFilePath)[0];
            UpdateCurrentOperation("Operation : Get all first classes...");
            List<string> Classes = GetAllClasses(FirstFilePath, GetLabel(OnlogogyAName), OnlogogyAName);
            UpdateCurrentOperation("Operation : Get all second classes...");
            List<string> Classes2 = GetAllClasses(SecondFilePath, GetLabel(OnlogogyBName), OnlogogyBName);

            List<string> ClassesWithLabel = new List<string>();
            List<string> Classes2WithLabel = new List<string>();
            List<string> Similars = new List<string>();
            List<string> DifferencesInFirstAnology = new List<string>();
            List<string> DifferencesInSecondAnology = new List<string>();
            List<string> Differences = new List<string>();
            List<string> DifferencesOnlyInFirst = new List<string>();
            List<string> DifferencesOnlyInSecond = new List<string>();
            List<string> result = new List<string>();
            List<string> result2 = new List<string>();
            List<string> AllResults = new List<string>();
            UpdateCurrentOperation("Operation : Get RootPath of all first classes...");
            foreach (string Class in Classes)
            {
                if (worker.CancellationPending)
                {
                    e.Cancel = true;
                    //cancel backgroundworker
                    return;
                }
                //TXT_TextClassesResult.Text += GetLabel(Class) + " : " + Class + Environment.NewLine;
                string RootPath = GetSuperClassesOfClassUntilRoot(FirstFilePath, Class,
GetLabel(OnlogogyAName), OnlogogyAName);
                result.Add(GetLabel(Class) + " : " + RootPath);
                AllResults.Add(GetLabel(Class) + " : " + RootPath);

                ClassesWithLabel.Add(GetLabel(Class));
            }
            UpdateCurrentOperation("Operation : Get RootPath of all second classes...");
            foreach (string Class in Classes2)
            {
                if (worker.CancellationPending)
```

```
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    string RootPath = GetSuperClassesOfClassUntilRoot(SecondFilePath, Class,
GetLabel(OnlogogyBName), OnlogogyBName);
                    result2.Add(GetLabel(Class) + " : " + RootPath);
                    AllResults.Add(GetLabel(Class) + " : " + RootPath);

                    Classes2WithLabel.Add(GetLabel(Class));
                }
                DataGRD_ClassesInFirst_Ref.Invoke((MethodInvoker)(() =>
                {
                    DataGRD_ClassesInFirst_Ref.Sort(DataGRD_ClassesInFirst_Ref.Columns["ClassName"],
ListSortDirection.Ascending);

                }));
                DataGRD_ClassesInSecond_Ref.Invoke((MethodInvoker)(() =>
                {
                    DataGRD_ClassesInSecond_Ref.Sort(DataGRD_ClassesInSecond_Ref.Columns["ClassName2"],
ListSortDirection.Ascending);
                }));
                //Provider
                UpdateCurrentOperation("Operation : add classes in first");
                foreach (string Class in ClassesWithLabel)
                {
                    if (worker.CancellationPending)
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    DataGRD_ClassesInFirst_Ref.Invoke((MethodInvoker)(() =>
                    {
                        DataGRD_ClassesInFirst_Ref.Rows.Add((DataGRD_ClassesInFirst_Ref.Rows.Count + 1),
Class);
                    }));
                }
                UpdateCurrentOperation("Operation : add classes in second");
                foreach (string Class in Classes2WithLabel)
                {
                    if (worker.CancellationPending)
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    DataGRD_ClassesInSecond_Ref.Invoke((MethodInvoker)(() =>
                    {
                        DataGRD_ClassesInSecond_Ref.Rows.Add((DataGRD_ClassesInSecond_Ref.Rows.Count + 1),
Class);
                    }));
                }
                AllResults.Sort();
                /****************** begin************/
                UpdateCurrentOperation("Operation : Add subClasses in first...");
                foreach (String r in result)
                {
                    if (worker.CancellationPending)
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    string[] Tmp = r.Split(new string[] { ":" }, StringSplitOptions.RemoveEmptyEntries);
                    string ClassName = r.Substring(0, r.IndexOf(":") + 1).Trim();
                    if (Tmp.Length == 2 &&
                        Tmp[0].Trim().ToLower() == Tmp[1].Trim().ToLower())
                    {
                        int RowIndex = GetDataGRDRowIndex(Tmp[0].Trim(), DataGRD_ClassesInFirst_Ref,
"ClassName");
                        DataGRD_ClassesInFirst_Ref.Rows[RowIndex].Cells["SuperClass"].Value = "Yes";
                        int count = 0;
                        foreach (DataGridViewCell cell in DataGRD_ClassesInFirst_Ref.Rows[RowIndex].Cells)
                        {
                            if (count >= 3)
                            {
                                //cell.Value = " - ";
                            }
                            count++;
```

```
                    }
                }
                else
                {
                    string RootPath = r.Substring(r.IndexOf(":") + 1).Trim();
                    string[] SubClasses = RootPath.Split(new string[] { ">" },
StringSplitOptions.RemoveEmptyEntries);
                    int CurrentSubClassesColumnsCount = DataGRD_ClassesInFirst_Ref.Columns.Count - 3;
                    int SubClassesCount = SubClasses.Length - 1;
                    int CountOfMustAddedColumns = 0;
                    if (CurrentSubClassesColumnsCount < SubClassesCount)
                        CountOfMustAddedColumns = SubClassesCount - CurrentSubClassesColumnsCount;
                    if (CountOfMustAddedColumns > 0)
                    {
                        int ColumnCount = CurrentSubClassesColumnsCount;
                        for (int i = 0; i < CountOfMustAddedColumns; i++)
                        {
                            if (worker.CancellationPending)
                            {
                                e.Cancel = true;
                                //cancel backgroundworker
                                return;
                            }
                            DataGridViewTextBoxColumn NewColumn = new DataGridViewTextBoxColumn();
                            NewColumn.HeaderText = "SubClasseOf " + (ColumnCount + 1);
                            NewColumn.MinimumWidth = 50;
                            NewColumn.Name = "SubClasseOf " + (ColumnCount + 1);
                            NewColumn.ReadOnly = true;
                            NewColumn.Width = 50;
                            DataGRD_ClassesInFirst_Ref.Invoke((MethodInvoker)(() =>
                            {
                                DataGRD_ClassesInFirst_Ref.Columns.Add(NewColumn);
                            }));

                            ColumnCount++;
                        }
                    }
                    int count = 0;
                    string[] Tmp2 = r.Split(new string[] { ":" }, StringSplitOptions.RemoveEmptyEntries);
                    int RowIndex2 = GetDataGRDRowIndex(Tmp2[0].Trim(), DataGRD_ClassesInFirst_Ref,
"ClassName");
                    string RootPath2 = r.Substring(r.IndexOf(":") + 1).Trim();
                    string[] SubClasses2 = RootPath2.Split(new string[] { ">" },
StringSplitOptions.RemoveEmptyEntries);
                    int Index = 1;

                    foreach (DataGridViewCell cell in DataGRD_ClassesInFirst_Ref.Rows[RowIndex2].Cells)
                    {
                        if (worker.CancellationPending)
                        {
                            e.Cancel = true;
                            //cancel backgroundworker
                            return;
                        }
                        if (count >= 3)
                        {
                            if (Index < SubClasses2.Length)
                            {
                                DataGRD_ClassesInFirst_Ref.Invoke((MethodInvoker)(() =>
                                {
                                    cell.Value = SubClasses2[Index];
                                }));
                                Index++;
                            }
                            else
                            {
                                //cell.Value = " - ";
                            }
                        }
                        count++;
                    }

                }
            }
            /*************************/
            UpdateCurrentOperation("Operation : Add subClasses in second...");
            foreach (String r in result2)
            {
                if (worker.CancellationPending)
                {
                    e.Cancel = true;
```

```
                        //cancel backgroundworker
                        return;
                    }
                    string[] Tmp = r.Split(new string[] { ":" }, StringSplitOptions.RemoveEmptyEntries);
                    string ClassName = r.Substring(0, r.IndexOf(":") + 1).Trim();
                    if (Tmp.Length == 2 &&
                        Tmp[0].Trim().ToLower() == Tmp[1].Trim().ToLower())
                    {
                        int RowIndex = GetDataGRDRowIndex(Tmp[0].Trim(), DataGRD_ClassesInSecond_Ref,
"ClassName2");
                        DataGRD_ClassesInSecond_Ref.Rows[RowIndex].Cells["SuperClass2"].Value = "Yes";
                        int count = 0;
                        foreach (DataGridViewCell cell in DataGRD_ClassesInSecond_Ref.Rows[RowIndex].Cells)
                        {
                            if (count >= 3)
                            {
                                //cell.Value = " - ";
                            }
                            count++;
                        }
                    }
                    else
                    {
                        string RootPath = r.Substring(r.IndexOf(":") + 1).Trim();
                        string[] SubClasses = RootPath.Split(new string[] { ">" },
StringSplitOptions.RemoveEmptyEntries);
                        int CurrentSubClassesColumnsCount = DataGRD_ClassesInSecond_Ref.Columns.Count - 3;
                        int SubClassesCount = SubClasses.Length - 1;
                        int CountOfMustAddedColumns = 0;
                        if (CurrentSubClassesColumnsCount < SubClassesCount)
                            CountOfMustAddedColumns = SubClassesCount - CurrentSubClassesColumnsCount;
                        if (CountOfMustAddedColumns > 0)
                        {
                            int ColumnCount = CurrentSubClassesColumnsCount;
                            for (int i = 0; i < CountOfMustAddedColumns; i++)
                            {
                                if (worker.CancellationPending)
                                {
                                    e.Cancel = true;
                                    //cancel backgroundworker
                                    return;
                                }
                                DataGridViewTextBoxColumn NewColumn = new DataGridViewTextBoxColumn();
                                NewColumn.HeaderText = "SubClasseOf " + (ColumnCount + 1);
                                NewColumn.MinimumWidth = 50;
                                NewColumn.Name = "SubClasseOf " + (ColumnCount + 1);
                                NewColumn.ReadOnly = true;
                                NewColumn.Width = 50;
                                DataGRD_ClassesInSecond_Ref.Invoke((MethodInvoker)(() =>
                                {
                                    DataGRD_ClassesInSecond_Ref.Columns.Add(NewColumn);
                                }));

                                ColumnCount++;
                            }
                        }
                        int count = 0;
                        string[] Tmp2 = r.Split(new string[] { ":" }, StringSplitOptions.RemoveEmptyEntries);
                        int RowIndex2 = GetDataGRDRowIndex(Tmp2[0].Trim(), DataGRD_ClassesInSecond_Ref,
"ClassName2");
                        string RootPath2 = r.Substring(r.IndexOf(":") + 1).Trim();
                        string[] SubClasses2 = RootPath2.Split(new string[] { ">" },
StringSplitOptions.RemoveEmptyEntries);
                        int Index = 1;
                        foreach (DataGridViewCell cell in DataGRD_ClassesInSecond_Ref.Rows[RowIndex2].Cells)
                        {
                            if (worker.CancellationPending)
                            {
                                e.Cancel = true;
                                //cancel backgroundworker
                                return;
                            }
                            if (count >= 3)
                            {
                                if (Index < SubClasses2.Length)
                                {
                                    DataGRD_ClassesInSecond_Ref.Invoke((MethodInvoker)(() =>
                                    {
                                        cell.Value = SubClasses2[Index];
                                    }));
```

```
                        Index++;
                    }
                    else
                    {
                        //cell.Value = " - ";
                    }
                }
                count++;
            }
        }
    }
    /************************/
    Classes.Sort();
    UpdateCurrentOperation("Operation : Get extact similar classes...");
    foreach (string Class in result)
    {
        if (worker.CancellationPending)
        {
            e.Cancel = true;
            //cancel backgroundworker
            return;
        }
        bool Found = false;
        foreach (string Class2 in result2)
        {
            if (Class2.ToLower() == Class.ToLower())
            {
                Similars.Add(Class);
                Found = true;
            }
        }
        if (!Found)
            DifferencesInFirstAnology.Add(Class);
    }
    UpdateCurrentOperation("Operation : Get classes only in first/second...");
    foreach (string Class in result2)
    {
        if (worker.CancellationPending)
        {
            e.Cancel = true;
            //cancel backgroundworker
            return;
        }
        bool Found = false;
        foreach (string Class2 in result)
        {
            if (Class2.ToLower() == Class.ToLower())
            {
                Found = true;
            }
        }
        if (!Found)
            DifferencesInSecondAnology.Add(Class);
    }
    Similars.Sort();
    DifferencesInFirstAnology.Sort();
    DifferencesInSecondAnology.Sort();
    UpdateCurrentOperation("Operation : extract classes only in first...");
    //first only
    foreach (string Class in DifferencesInFirstAnology)
    {
        if (worker.CancellationPending)
        {
            e.Cancel = true;
            //cancel backgroundworker
            return;
        }
        bool Found = false;
        foreach (string Class2 in DifferencesInSecondAnology)
        {
            if (worker.CancellationPending)
            {
                e.Cancel = true;
                //cancel backgroundworker
                return;
            }
            if (Class.Substring(0, Class.IndexOf(":")).Trim().ToLower() == Class2.Substring(0,
Class2.IndexOf(":")).Trim().ToLower())
            {
                Differences.Add(Class);
                Differences.Add(Class2);
```

```
                        Found = true;
                    }
                }
                if (!Found)
                {
                    DifferencesOnlyInFirst.Add(Class);
                }
            }
            UpdateCurrentOperation("Operation : extract classes only in second...");
            //second only
            foreach (string Class in DifferencesInSecondAnology)
            {
                if (worker.CancellationPending)
                {
                    e.Cancel = true;
                    //cancel backgroundworker
                    return;
                }
                bool Found = false;
                foreach (string Class2 in DifferencesInFirstAnology)
                {
                    if (worker.CancellationPending)
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    if (Class.Substring(0, Class.IndexOf(":")).Trim().ToLower() == Class2.Substring(0,
Class2.IndexOf(":")).Trim().ToLower())
                    {
                        Found = true;
                    }
                }
                if (!Found)
                {
                    DifferencesOnlyInSecond.Add(Class);
                }
            }
            UpdateCurrentOperation("Operation : Add extact similar classes...");
            foreach (string r in Similars)
            {
                if (worker.CancellationPending)
                {
                    e.Cancel = true;
                    //cancel backgroundworker
                    return;
                }
                string[] tmp = r.Split(new string[] { ":" }, StringSplitOptions.RemoveEmptyEntries);
                DataGRD_ExtactSimilarClasses_Ref.Invoke((MethodInvoker)(() =>
                {
                    DataGRD_ExtactSimilarClasses_Ref.Rows.Add((DataGRD_ExtactSimilarClasses_Ref.Rows.Count
+ 1), tmp[0], tmp[1]);
                }));
            }
            UpdateCurrentOperation("Operation : add similar classes differnt position...");
            for (int i = 0; i < Differences.Count; i += 2)
            {
                if (worker.CancellationPending)
                {
                    e.Cancel = true;
                    //cancel backgroundworker
                    return;
                }
                string ClassName = Differences[i].Split(new string[] { ":" },
StringSplitOptions.RemoveEmptyEntries)[0];
                string PositionInFirstOnology = Differences[i].Split(new string[] { ":" },
StringSplitOptions.RemoveEmptyEntries)[1];
                string PositionInSecondOnology = Differences[i + 1].Split(new string[] { ":" },
StringSplitOptions.RemoveEmptyEntries)[1];
                DataGRD_SimilarClassesDifferentPostion_Ref.Invoke((MethodInvoker)(() =>
                {

DataGRD_SimilarClassesDifferentPostion_Ref.Rows.Add((DataGRD_SimilarClassesDifferentPostion_Ref.Rows.Count +
1), ClassName, PositionInFirstOnology, PositionInSecondOnology);
                }));

            }
            UpdateCurrentOperation("Operation : add classes only in first...");
            foreach (string r in DifferencesOnlyInFirst)
            {
                if (worker.CancellationPending)
```

```
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    string[] tmp = r.Split(new string[] { ":" }, StringSplitOptions.RemoveEmptyEntries);
                    DataGRD_ClassesInFirst_Ref.Invoke((MethodInvoker)(() =>
                    {
                        DataGRD_OnlyClassesInFirst_Ref.Rows.Add((DataGRD_OnlyClassesInFirst_Ref.Rows.Count +
1), tmp[0], tmp[1]);
                    }));

                }
                UpdateCurrentOperation("Operation : extract classes only in second...");
                foreach (string r in DifferencesOnlyInSecond)
                {
                    if (worker.CancellationPending)
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    string[] tmp = r.Split(new string[] { ":" }, StringSplitOptions.RemoveEmptyEntries);
                    DataGRD_OnlyClassesInSecond_Ref.Invoke((MethodInvoker)(() =>
                    {
                        DataGRD_OnlyClassesInSecond_Ref.Rows.Add((DataGRD_OnlyClassesInSecond_Ref.Rows.Count +
1), tmp[0], tmp[1]);
                    }));

                }
            }
            catch (Exception ex)
            {
                FRM_MSG f = new FRM_MSG();
                f.ShowDLG(AssemblyInfo.AssemblyTitle,
                ex.Message + "\n" + ex.StackTrace.ToString(),
                                    FRM_MSG.MSGIcon.Error,
                                    FRM_MSG.BTNS.One,
                                    new string[] { "Ok" });
            }
        }
```

The following code retrieves the Object properties from each ontology and compares each property in the first ontology with the all the properties in the second ontology

```
  /*********************/
  private List<string> GetAllObjectProperties(string FilePath, string OntologyLabel, string OntologyUrl)
        {
            List<string> Classes = new List<string>();
            try
            {
                IGraph g = new Graph();
                g.LoadFromFile(FilePath);
                string GetAllObjectProperties = @"
  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 " + OntologyLabel + @": <" + OntologyUrl + @">
SELECT  ?x ?subject
WHERE   {  ?x    rdf:type  owl:ObjectProperty
}
";

                Object results = g.ExecuteQuery(GetAllObjectProperties);
                if (results is SparqlResultSet)
                {
                    //SELECT/ASK queries give a SparqlResultSet
                    SparqlResultSet rset = (SparqlResultSet)results;
                    foreach (SparqlResult r in rset)
                    {
                            Classes.Add(r["x"].ToString());
                        //TXT_TextToSearchIn.Text += r["x"].ToString() + Environment.NewLine;
```

```
                    //Do whatever you want with each Result
                }
            }
            else if (results is IGraph)
            {
                //CONSTRUCT/DESCRIBE queries give a IGraph
                IGraph resGraph = (IGraph)results;
                foreach (Triple t in resGraph.Triples)
                {
                    //Do whatever you want with each Triple
                }
            }
            else
            {
                //If you don't get a SparqlResutlSet or IGraph something went wrong
                //but didn't throw an exception so you should handle it here
                MessageBox.Show("No Data Found.");
            }
            return Classes;
        }
        catch (Exception ex)
        {
            FRM_MSG f = new FRM_MSG();
            f.ShowDLG(AssemblyInfo.AssemblyTitle,
            ex.Message + "\n" + ex.StackTrace.ToString(),
                                FRM_MSG.MSGIcon.Error,
                                FRM_MSG.BTNS.One,
                                new string[] { "Ok" });
            throw ex;
        }
    }

             /*********************************/
    private void FillObjectProperties(DoWorkEventArgs e)
    {
        try
        {
            UpdateCurrentOperation("Operation : Get ontology name of first...");
            string OnlogogyAName = GetAllOntologies(FirstFilePath)[0];
            UpdateCurrentOperation("Operation : Get ontology name of second...");
            string OnlogogyBName = GetAllOntologies(SecondFilePath)[0];
            UpdateCurrentOperation("Operation : Get all object properties of first...");
            List<string> ObjectProperties = GetAllObjectProperties(FirstFilePath, GetLabel(OnlogogyAName),
OnlogogyAName);
            UpdateCurrentOperation("Operation : Get all object properties of second...");
            List<string> ObjectProperties2 = GetAllObjectProperties(SecondFilePath,
GetLabel(OnlogogyBName), OnlogogyBName);
            List<string> ExactSimilars = new List<string>();
            List<string> Similars = new List<string>();
            List<string> DifferencesInFirstAnology = new List<string>();
            List<string> DifferencesInSecondAnology = new List<string>();
            List<string> result = new List<string>();
            List<string> result2 = new List<string>();
            UpdateCurrentOperation("Operation : Get label of object properties in first...");
            foreach (string ObjectProperty in ObjectProperties)
            {
                if (worker.CancellationPending)
                {
                    e.Cancel = true;
                    //cancel backgroundworker
                    return;
                }
                if (GetLabel(ObjectProperty).Length>=3)
                    result.Add(GetLabel(ObjectProperty));
            }
            UpdateCurrentOperation("Operation : Get label of object properties in second...");
            foreach (string ObjectProperty in ObjectProperties2)
            {
                if (worker.CancellationPending)
                {
                    e.Cancel = true;
                    //cancel backgroundworker
                    return;
                }
                if (GetLabel(ObjectProperty).Length >= 3)
                    result2.Add(GetLabel(ObjectProperty));
            }
            /*************/
            UpdateCurrentOperation("Operation : Get exact similar/similar/OnlyInFirst object properties in
second...");
            foreach (string ObjectProperty in result)
```

```
{
    if (worker.CancellationPending)
    {
        e.Cancel = true;
        //cancel backgroundworker
        return;
    }
    bool Found = false;
    foreach (string ObjectProperty2 in result2)
    {
        if (worker.CancellationPending)
        {
            e.Cancel = true;
            //cancel backgroundworker
            return;
        }
        if (ObjectProperty.ToLower() == ObjectProperty2.ToLower())
        {
            Found = true;
        }
    }
    if (Found)
    {
        ExactSimilars.Add(ObjectProperty);
    }
    else
    {
        bool Found2 = false;
        foreach (string ObjectProperty2 in result2)
        {
            if (worker.CancellationPending)
            {
                e.Cancel = true;
                //cancel backgroundworker
                return;
            }
            //MessageBox.Show(ObjectProperty2 + "\n"+ObjectProperty);
            if (ObjectProperty2.ToLower().Contains(ObjectProperty.ToLower()) ||
                ObjectProperty.ToLower().Contains(ObjectProperty2.ToLower()))
            {
                Similars.Add(ObjectProperty + " : " + ObjectProperty2);
                Found2 = true;
            }
        }
        if (!Found2)
        {
            DifferencesInFirstAnology.Add(ObjectProperty);
        }
    }
}
/*****************/
UpdateCurrentOperation("Operation : Get only in second object properties in second...");
foreach (string ObjectProperty in result2)
{
    if (worker.CancellationPending)
    {
        e.Cancel = true;
        //cancel backgroundworker
        return;
    }
    bool Found = false;
    foreach (string ObjectProperty2 in result)
    {
        if (worker.CancellationPending)
        {
            e.Cancel = true;
            //cancel backgroundworker
            return;
        }
        if (ObjectProperty.ToLower() == ObjectProperty2.ToLower())
        {
            Found = true;
        }
    }
    if (Found)
    {
    }
    else
    {
        bool Found2 = false;
        foreach (string ObjectProperty2 in result)
```

```
                        {
                            if (worker.CancellationPending)
                            {
                                e.Cancel = true;
                                //cancel backgroundworker
                                return;
                            }
                            if (ObjectProperty2.ToLower().Contains(ObjectProperty.ToLower()) ||
                                ObjectProperty.ToLower().Contains(ObjectProperty2.ToLower()))
                            {
                                Found2 = true;
                            }
                        }
                        if (!Found2)
                        {
                            DifferencesInSecondAnology.Add(ObjectProperty);
                        }
                    }
                }
                DataGRD_SimilarObjectPropertiesDifferentTerminology_Ref.Invoke((MethodInvoker)(() =>
                {

DataGRD_SimilarObjectPropertiesDifferentTerminology_Ref.Sort(DataGRD_SimilarObjectPropertiesDifferentTerminolog
y_Ref.Columns["ObjectPropertyInFirstOntology"], ListSortDirection.Ascending);
                }));
                UpdateCurrentOperation("Operation : Add similar object properties different terminology...");
                foreach (string r in Similars)
                {
                    if (worker.CancellationPending)
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    string[] SimialrObjectProperty = r.Split(new string[] { ":" },
StringSplitOptions.RemoveEmptyEntries);
                    DataGRD_SimilarObjectPropertiesDifferentTerminology_Ref.Invoke((MethodInvoker)(() =>
                    {

DataGRD_SimilarObjectPropertiesDifferentTerminology_Ref.Rows.Add((DataGRD_SimilarObjectPropertiesDifferentTermi
nology_Ref.Rows.Count + 1), SimialrObjectProperty[0], SimialrObjectProperty[1]);
                    }));
                }
                DataGRD_ExactSameObjectProperties_Ref.Invoke((MethodInvoker)(() =>
                {

DataGRD_ExactSameObjectProperties_Ref.Sort(DataGRD_ExactSameObjectProperties_Ref.Columns["ExactSameObjectProper
ty"], ListSortDirection.Ascending);
                }));
                UpdateCurrentOperation("Operation : Add extact similar object properties...");
                foreach (string r in ExactSimilars)
                {
                    if (worker.CancellationPending)
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    DataGRD_ExactSameObjectProperties_Ref.Invoke((MethodInvoker)(() =>
                    {

DataGRD_ExactSameObjectProperties_Ref.Rows.Add((DataGRD_ExactSameObjectProperties_Ref.Rows.Count + 1), r);
                    }));

                }
                DataGRD_OnlyObjectPropertiesInFirst_Ref.Invoke((MethodInvoker)(() =>
                {

DataGRD_OnlyObjectPropertiesInFirst_Ref.Sort(DataGRD_OnlyObjectPropertiesInFirst_Ref.Columns["OnlyObjectPropert
iesInFirst"], ListSortDirection.Ascending);
                }));
                UpdateCurrentOperation("Operation : Add object properties only in first...");
                foreach (string r in DifferencesInFirstAnology)
                {
                    if (worker.CancellationPending)
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    DataGRD_OnlyObjectPropertiesInFirst_Ref.Invoke((MethodInvoker)(() =>
```

```
                    {
DataGRD_OnlyObjectPropertiesInFirst_Ref.Rows.Add((DataGRD_OnlyObjectPropertiesInFirst_Ref.Rows.Count + 1), r);
                    }));

                }
                DataGRD_OnlyObjectPropertiesInSecond_Ref.Invoke((MethodInvoker)(() =>
                {
DataGRD_OnlyObjectPropertiesInSecond_Ref.Sort(DataGRD_OnlyObjectPropertiesInSecond_Ref.Columns["OnlyObjectPrope
rtiesInSecond"], ListSortDirection.Ascending);
                }));
                UpdateCurrentOperation("Operation : Add object properties only in second...");
                foreach (string r in DifferencesInSecondAnology)
                {
                    if (worker.CancellationPending)
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    DataGRD_OnlyObjectPropertiesInSecond_Ref.Invoke((MethodInvoker)(() =>
                    {
DataGRD_OnlyObjectPropertiesInSecond_Ref.Rows.Add((DataGRD_OnlyObjectPropertiesInSecond_Ref.Rows.Count + 1),
r);
                    }));

                }

            }
            catch (Exception ex)
            {
                FRM_MSG f = new FRM_MSG();
                f.ShowDLG(AssemblyInfo.AssemblyTitle,
                ex.Message + "\n" + ex.StackTrace.ToString(),
                                            FRM_MSG.MSGIcon.Error,
                                            FRM_MSG.BTNS.One,
                                            new string[] { "Ok" });
            }
        }
```

APPENDIX D PYTHON CODE FOR SEMANTIC COMPARISON

This code executes Wu-Palmer Similarity (WUP) similarity approach. This code compares how similar two-word senses are. The code works based on the depth calculation of the two senses in the taxonomy and the their last common Subsumer.

```
public static string Execute_wup_similarity2(string Word1, string Word2)
        {
            try
            {
                string path = new
System.IO.FileInfo(Assembly.GetExecutingAssembly().Location).Directory.FullName;
                string path2 = path + "\\NLTKTestLast.py";
                path = "\"" + path + "\\NLTKTestLast.py\"";
                string PythonScript = @"from nltk.corpus import wordnet
syn1 = wordnet.synsets('" + Word1 + @"')
syn2 = wordnet.synsets('" + Word2 + @"')
if len(syn1) >0 and len(syn2) >0:
   r=syn1[0].wup_similarity2(syn2[0])
   if r is not None:
     print(round(r*100))";
                File.WriteAllText(path2, PythonScript);
                Process p = new Process();
                p.StartInfo = new ProcessStartInfo(Properties.Settings.Default.PythonPath, path)
                {
                    RedirectStandardOutput = true,
                    UseShellExecute = false,
                    CreateNoWindow = true
                };
                p.Start();

                string output = p.StandardOutput.ReadToEnd();
                p.WaitForExit();
                //Console.ReadLine();
                // MessageBox.Show(output);
                return output;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message + "\n" + ex.StackTrace.ToString());
                throw ex;
            }
        }

private void UpdateCurrentOperation(string text)

        {
            LBL_CurrentOperation_Ref.Invoke((MethodInvoker)(() =>
            {
                LBL_CurrentOperation_Ref.Text = text;
            }));
            Progress_ComparisonProgress_Ref.Invoke((MethodInvoker)(() =>
            {
                int progress=Progress_ComparisonProgress_Ref.Value;
                progress = (progress + 3);
                if (progress < 100)
                {

                    Progress_ComparisonProgress_Ref.Value = progress;
                    Progress_ComparisonProgress_Ref.Focus();
                    Progress_ComparisonProgress_Ref.Update();
                    Progress_ComparisonProgress_Ref.Refresh();
                }

            }));
            LBL_PrecentageComparesion_Ref.Invoke((MethodInvoker)(() =>
            {
                LBL_PrecentageComparesion_Ref.Text = Progress_ComparisonProgress_Ref.Value+ "%";
            }));
```

```
        }
                 /*********************/

        private string GetSmallPartOfSpeach(PartOfSpeech p)
        {
            switch (p)
            {
                case PartOfSpeech.Noun:
                    return "(n)";
                case PartOfSpeech.Adverb:
                    return "(av)";
                case PartOfSpeech.Verb:
                    return "(v)";
                case PartOfSpeech.Adjective:
                    return "(aj)";
                default:
                    return "";
            }
        }

                 /*********************/

        private bool IsResultFound(List<SematicResult> result, string Word)
        {
            try
            {
                foreach (SematicResult r in result)
                {
                    if (r.Word == Word)
                        return true;
                }
                return false;
            }
            catch
            {
                return false;
            }
        }
/**************************/
private string[] GetAllOntologies(string FilePath)
        {
            List<string> Ontologies = new List<string>();
            try
            {
                IGraph g = new Graph();
                g.LoadFromFile(FilePath);
                string GetAllOntologies = @"
PREFIX  owl:  <http://www.w3.org/2002/07/owl#>
PREFIX  rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?ontology
WHERE { ?ontology  rdf:type  owl:Ontology }
ORDER BY ?ontology
";
                Object results = g.ExecuteQuery(GetAllOntologies);
                if (results is SparqlResultSet)
                {
                    //SELECT/ASK queries give a SparqlResultSet
                    SparqlResultSet rset = (SparqlResultSet)results;
                    foreach (SparqlResult r in rset)
                    {
                        Ontologies.Add(r["ontology"].ToString());
                        //TXT_TextClassesResult.Text += r["x"].ToString() + Environment.NewLine;
                        //Do whatever you want with each Result
                    }
                }
                else if (results is IGraph)
                {
                    //CONSTRUCT/DESCRIBE queries give a IGraph
                    IGraph resGraph = (IGraph)results;
                    foreach (Triple t in resGraph.Triples)
                    {
                        //TXT_TextClassesResult.Text += t.Subject.ToString() + Environment.NewLine;
                        //Do whatever you want with each Triple
                    }
                }
                else
                {
                    //If you don't get a SparqlResutlSet or IGraph something went wrong
                    //but didn't throw an exception so you should handle it here
                    MessageBox.Show("No Data Found.");
```

```
                }
                return Ontologies.ToArray();
            }
            catch (Exception ex)
            {
                FRM_MSG f = new FRM_MSG();
                f.ShowDLG(AssemblyInfo.AssemblyTitle,
                ex.Message + "\n" + ex.StackTrace.ToString(),
                                      FRM_MSG.MSGIcon.Error,
                                      FRM_MSG.BTNS.One,
                                      new string[] { "Ok" });
                throw ex;
            }
        }
```

This code creates queries to retrieve the classes from the ontology and compares them with WordNet.

```
 private string GetLabel(string Class)
        {
            if (Class.LastIndexOf("#") <= 0)
            {
                return Class.Substring(Class.LastIndexOf("/") + 1);
            }
            else
            {
                return Class.Substring(Class.LastIndexOf("#") + 1);
            }
        }
                /*********************/
                private List<string> GetAllClasses(string FilePath, string OntologyLabel, string OntologyUrl)
        {
            List<string> Classes = new List<string>();
            try
            {
                IGraph g = new Graph();
                g.LoadFromFile(FilePath);
                string GetAllClasses = @"
                //get all sub classes
                string GetSuperClassesUntilRoot = @"
  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 untitled-ontology-12: <http://www.semanticweb.org/alhusain/ontologies/2020/1/untitled-ontology-12#>
  select ?superclass where {
  untitled-ontology-12:Student (rdfs:subClassOf|(owl:intersectionOf/rdf:rest*/rdf:first))* ?superclass .
}
";
                Object results = g.ExecuteQuery(GetAllClasses);
                if (results is SparqlResultSet)
                {
                    //SELECT/ASK queries give a SparqlResultSet
                    SparqlResultSet rset = (SparqlResultSet)results;
                    foreach (SparqlResult r in rset)
                    {
                        if (r["x"].ToString().StartsWith("http") ||
                            r["x"].ToString().StartsWith("https"))
                            Classes.Add(r["x"].ToString());
                        //TXT_TextToSearchIn.Text += r["x"].ToString() + Environment.NewLine;
                        //Do whatever you want with each Result
                    }
                }
                else if (results is IGraph)
                {
                    //CONSTRUCT/DESCRIBE queries give a IGraph
                    IGraph resGraph = (IGraph)results;
                    foreach (Triple t in resGraph.Triples)
                    {
                        //Do whatever you want with each Triple
                    }
                }
                else
                {
                    //If you don't get a SparqlResutlSet or IGraph something went wrong
                    //but didn't throw an exception so you should handle it here
                    MessageBox.Show("No Data Found.");
                }
                return Classes;
```

```
        }
        catch (Exception ex)
        {
            FRM_MSG f = new FRM_MSG();
            f.ShowDLG(AssemblyInfo.AssemblyTitle,
            ex.Message + "\n" + ex.StackTrace.ToString(),
                                FRM_MSG.MSGIcon.Error,
                                FRM_MSG.BTNS.One,
                                new string[] { "Ok" });
            throw ex;
        }
    }
         private void FillClasses( DoWorkEventArgs e,WordNetEngine wne)
    {
        try
        {
            UpdateCurrentOperation("Operation : Get First Onlogogy Name...");
            string OnlogogyAName = GetAllOntologies(FirstFilePath)[0];
            UpdateCurrentOperation("Operation : Get Second Onlogogy Name...");
            string OnlogogyBName = GetAllOntologies(SecondFilePath)[0];
            UpdateCurrentOperation("Operation : Get all first classes...");
            List<string> Classes = GetAllClasses(FirstFilePath, GetLabel(OnlogogyAName), OnlogogyAName);
            UpdateCurrentOperation("Operation : Get all second classes...");
            List<string> Classes2 = GetAllClasses(SecondFilePath, GetLabel(OnlogogyBName), OnlogogyBName);

            List<string> ClassesWithLabel = new List<string>();
            List<string> Classes2WithLabel = new List<string>();

            UpdateCurrentOperation("Operation : Get Class Label of all first classes...");
            foreach (string Class in Classes)
            {
                if (worker.CancellationPending)
                {
                    e.Cancel = true;
                    //cancel backgroundworker
                    return;
                }

                ClassesWithLabel.Add(GetLabel(Class));
            }
            UpdateCurrentOperation("Operation : Get class Label of all second classes...");
            foreach (string Class in Classes2)
            {
                if (worker.CancellationPending)
                {
                    e.Cancel = true;
                    //cancel backgroundworker
                    return;
                }

                Classes2WithLabel.Add(GetLabel(Class));
            }

            UpdateCurrentOperation("Operation : Fill class First VS WordsNet...");
            int ColumnCount = 1;
            foreach (string Class in ClassesWithLabel)
            {
                if (worker.CancellationPending)
                {
                    e.Cancel = true;
                    //cancel backgroundworker
                    return;
                }
                DataGridViewTextBoxColumn NewColumn = new DataGridViewTextBoxColumn();
                NewColumn.HeaderText = Class;
                NewColumn.MinimumWidth = 50;
                NewColumn.Name = Class + (ColumnCount + 1);
                NewColumn.ReadOnly = true;
                NewColumn.Width = 50;
                int NewColumnIndex = 0;
                DataGRD_Class_FirstVSWordNet_Ref.Invoke((MethodInvoker)(() =>
                {
                    NewColumnIndex = DataGRD_Class_FirstVSWordNet_Ref.Columns.Add(NewColumn);
                }));
                string InsteadWord = IsWordFound(Class);
                string TempClass = "";
                if (InsteadWord != "")
                    TempClass = InsteadWord;
                else
                    TempClass = Class;
```

```
                    string[] parts = Utiles.SpitByCapitalLetters(TempClass).Split(new string[] { " " },
StringSplitOptions.RemoveEmptyEntries);
                    bool isFoundResult = false;
                    foreach(string part in parts)
                    {
                        if (isFoundResult)
                            break;
                    List<SynSet> sets = wne.GetSynSets(part);
                    List<SematicResult> result = new List<SematicResult>();
                    foreach (SynSet s in sets)
                    {
                        if (worker.CancellationPending)
                        {
                            e.Cancel = true;
                            //cancel backgroundworker
                            return;
                        }
                        foreach (string w in s.Words)
                        {
                            if (worker.CancellationPending)
                            {
                                e.Cancel = true;
                                //cancel backgroundworker
                                return;
                            }
                            string PythonResult = ExecutePython.Execute2(part, w, AlgType);

                            if (!string.IsNullOrEmpty(PythonResult))
                            {
                                if (!IsResultFound(result, w + " " + GetSmallPartOfSpeach(s.PartOfSpeech)))
                                {
                                    result.Add(new SematicResult { Word = w + " " +
GetSmallPartOfSpeach(s.PartOfSpeech), Prectentage = Convert.ToInt32(PythonResult) });
                                        isFoundResult = true;
                                }
                            }
                        }
                    }
                    DataGRD_Class_FirstVSWordNet_Ref.Invoke((MethodInvoker)(() =>
                    {
                        //NewColumnIndex
                        if (result.Count > DataGRD_Class_FirstVSWordNet_Ref.Rows.Count)
                        {
                            int MustAddedRows = Math.Abs(DataGRD_Class_FirstVSWordNet_Ref.Rows.Count -
result.Count) + 1;
                            for (int i = 1; i <= MustAddedRows; i++)
                                DataGRD_Class_FirstVSWordNet_Ref.Rows.Add();
                        }
                    }));
                    int c = 0;
                    List<SematicResult> SortedResult = result.OrderByDescending(o => o.Prectentage).ToList();
                    foreach (SematicResult sr in SortedResult)
                    {
                        DataGRD_Class_FirstVSWordNet_Ref.Invoke((MethodInvoker)(() =>
                        {
                            DataGRD_Class_FirstVSWordNet_Ref.Rows[c++].Cells[NewColumnIndex].Value = sr.Word +
" " + sr.Prectentage + "%";
                        }));
                    }
                    result.Clear();
                }
                }


                UpdateCurrentOperation("Operation : Fill class Second VS WordsNet...");
                 ColumnCount = 1;
                foreach (string Class in Classes2WithLabel)
                {
                    if (worker.CancellationPending)
                    {
                        e.Cancel = true;
                        //cancel backgroundworker
                        return;
                    }
                    DataGridViewTextBoxColumn NewColumn = new DataGridViewTextBoxColumn();
                    NewColumn.HeaderText = Class;
                    NewColumn.MinimumWidth = 50;
                    NewColumn.Name = Class + (ColumnCount + 1);
                    NewColumn.ReadOnly = true;
                    NewColumn.Width = 50;
                    int NewColumnIndex = 0;
```

```
                    DataGRD_Class_SecondVSWordNet_Ref.Invoke((MethodInvoker)(() =>
                    {
                        NewColumnIndex = DataGRD_Class_SecondVSWordNet_Ref.Columns.Add(NewColumn);
                    }));
                    string InsteadWord = IsWordFound(Class);
                    string TempClass = "";
                    if (InsteadWord != "")
                        TempClass = InsteadWord;
                    else
                        TempClass = Class;
                    List<SynSet> sets = wne.GetSynSets(TempClass);
                    List<SematicResult> result = new List<SematicResult>();
                    foreach (SynSet s in sets)
                    {

                        if (worker.CancellationPending)
                        {
                            e.Cancel = true;
                            //cancel backgroundworker
                            return;
                        }
                        foreach (string w in s.Words)
                        {
                            if (worker.CancellationPending)
                            {
                                e.Cancel = true;
                                //cancel backgroundworker
                                return;
                            }
                            string PythonResult = ExecutePython.Execute2(TempClass, w, AlgType);
                            if (!string.IsNullOrEmpty(PythonResult))
                            {
                                if (!IsResultFound(result, w + " " + GetSmallPartOfSpeach(s.PartOfSpeech)))
                                {
                                    result.Add(new SematicResult { Word = w + " " +
GetSmallPartOfSpeach(s.PartOfSpeech), Prectentage = Convert.ToInt32(PythonResult) });
                                }
                            }
                        }
                    }
                    DataGRD_Class_SecondVSWordNet_Ref.Invoke((MethodInvoker)(() =>
                    {
                        if (result.Count > DataGRD_Class_SecondVSWordNet_Ref.Rows.Count)
                        {
                            int MustAddedRows = Math.Abs(DataGRD_Class_SecondVSWordNet_Ref.Rows.Count -
result.Count)+1;
                            for (int i = 1; i <= MustAddedRows; i++)
                                DataGRD_Class_SecondVSWordNet_Ref.Rows.Add();
                        }
                    }));
                    int c = 0;
                    List<SematicResult> SortedResult = result.OrderByDescending(o => o.Prectentage).ToList();
                    foreach (SematicResult sr in SortedResult)
                    {
                        DataGRD_Class_SecondVSWordNet_Ref.Invoke((MethodInvoker)(() =>
                        {
                            DataGRD_Class_SecondVSWordNet_Ref.Rows[c++].Cells[NewColumnIndex].Value = sr.Word +
" " + sr.Prectentage + "%";
                        }));
                    }
                    result.Clear();
                }
            }
            catch (Exception ex)
            {
                FRM_MSG f = new FRM_MSG();
                f.ShowDLG(AssemblyInfo.AssemblyTitle,
                ex.Message + "\n" + ex.StackTrace.ToString(),
                                    FRM_MSG.MSGIcon.Error,
                                    FRM_MSG.BTNS.One,
                                    new string[] { "Ok" });
            }
        }
                /******************/
```

APPENDIX E CODE FOR THE ONTOLOGY BROWSING TOOL

The following code retrieves the workzones in each operation.

```
public static List<string> GetOperation_Workzones(string FilePath, string OperationID)
        {
            List<string> OperationDetials = new List<string>();
            try
            {
                string OnlogogyAName = GetAllOntologies(FilePath)[0];
                string Label = GetLabel(OnlogogyAName);
                OperationDetials = GetOperation_Workzones(FilePath, Label, OnlogogyAName, OperationID);
                return OperationDetials;
            }
            catch (Exception ex)
            {
                FRM_MSG f = new FRM_MSG();
                f.ShowDLG(AssemblyInfo.AssemblyTitle,
                ex.Message + "\n" + ex.StackTrace.ToString(),
                                    FRM_MSG.MSGIcon.Error,
                                    FRM_MSG.BTNS.One,
                                    new string[] { "Ok" });
                throw ex;
            }
        }
                 /*********************/
                 private static List<string> GetOperation_Workzones(string FilePath, string OntologyLabel,
string OntologyUrl, string OperationID)
        {
            List<string> data = new List<string>();
            try
            {
                IGraph g = new Graph();
                g.LoadFromFile(FilePath);
                string GetAllClasses = @"
                string queryString = @"
  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 " + OntologyLabel + @": <" + OntologyUrl + @"#>"
                     + "SELECT  (str(?x) as ?EqID) (str(?a) as ?hasBrand) (str(?f) as ?hasColor) (str(?g) as
?hasModel) "
                     + "where { ?y EW:EqID ?x." + "?y EW:hasBrand ?a."
                     + "?y EW:hasColor ?f." + "?y EW:hasModel ?g." + " }";
                //MessageBox.Show(GetAllClasses);
                //get all sub classes
                string Q2 = @"
  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 untitled-ontology-12: <http://www.semanticweb.org/alhusain/ontologies/2020/1/untitled-ontology-12#>
SELECT ?entity
WHERE {
  ?subclass rdfs:subClassOf <http://www.semanticweb.org/alhusain/ontologies/2020/1/untitled-ontology-12#Actor>.
  ?entity owl:type ?subclass.
}
";
                //get all sub classes
                string GetSuperClassesUntilRoot = @"
  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 untitled-ontology-12: <http://www.semanticweb.org/alhusain/ontologies/2020/1/untitled-ontology-12#>
  select ?superclass where {
  untitled-ontology-12:Student (rdfs:subClassOf|(owl:intersectionOf/rdf:rest*/rdf:first))* ?superclass .
}
";
                string query = @"
  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 " + OntologyLabel + @": <" + OntologyUrl + @"#>" +
                 @"SELECT Distinct  ?object ?Workzone
                     where { ?subject " + OntologyLabel + @":hasOperationId ?object.
                            ?subject " + OntologyLabel + @":hasWorkzone ?Workzone.}";
                // "FILTER (?object=\"" + OperationID+"\")
                //MessageBox.Show(query);
                Object results = g.ExecuteQuery(query);
```

```
                if (results is SparqlResultSet)
                {
                    //SELECT/ASK queries give a SparqlResultSet
                    SparqlResultSet rset = (SparqlResultSet)results;
                    foreach (SparqlResult r in rset)
                    {
                        if (r["object"].ToString() == OperationID)
                        {
                            data.Add(r["Workzone"] == null ? "" : r["Workzone"].ToString().Replace(OntologyUrl
+ "#", ""));
                        }
                    }

                }
                else if (results is IGraph)
                {
                    //CONSTRUCT/DESCRIBE queries give a IGraph
                    IGraph resGraph = (IGraph)results;
                    foreach (Triple t in resGraph.Triples)
                    {
                    }
                }
                else
                {
                    //If you don't get a SparqlResutlSet or IGraph something went wrong
                    //but didn't throw an exception so you should handle it here
                    MessageBox.Show("No Data Found.");
                }
                return data;
            }
            catch (Exception ex)
            {
                FRM_MSG f = new FRM_MSG();
                f.ShowDLG(AssemblyInfo.AssemblyTitle,
                ex.Message + "\n" + ex.StackTrace.ToString(),
                                    FRM_MSG.MSGIcon.Error,
                                    FRM_MSG.BTNS.One,
                                    new string[] { "Ok" });
                throw ex;
            }
        }
```

APPENDIX F LIST OF PUBLICATIONS

Journal Papers

- **Taher, A.**, Vahdatikhaki, F., Hammad, A., (2021) "Formalizing Knowledge Representation in Earthwork Operations through Development of Domain Ontology", *Engineering, Construction and Architectural Management*, "In Print".
- **Taher, A.**, Vahdatikhaki, F., Hammad, A., (2021) "Extending Earthwork Ontology to Enhance Operation Safety", *Automation in Construction*, "Under Review".
- Vahdatikhaki, F., Lnagari, S. M., **Taher, A.**, El Ammari, K., Hammad, A., (2017) "Enhancing coordination and safety of earthwork equipment operations using Multi-Agent System", *Automation in Construction*, 81, pp. 267-285.
- Bahreini, F., Nasrollahi, M., **Taher, A.**, Hammad, A., (2021) "Ontology for BIM-Based Robotic Navigation and Inspection Tasks", *Advanced Engineering Informatic*, "Accepted with modifications".

Conference Papers

- **Taher, A.**, Vahdatikhaki, F., Hammad, A., (2019). "Integrating Earthwork Ontology and Safety Regulations to Enhance Operations Safety", *36th International Symposium on Automation and Robotics in Construction (ISARC 2019)*.
- **Taher, A.**, Vahdatikhaki, F., Hammad, A., (2017). "Towards Developing an Ontology for Earthwork Operations", *International Workshop on Computing in Civil Engineering*, pp. 101-108, Seattle. USA.
- Hammad, A., Motamedi, A., Yabuki, N., **Taher, A.**, Bahreini, F., (2017). "Towards an Ontology for Modeling Lifecycle Inspection and Repair Information of Civil Infrastructure Systems Focusing on Surface Defects", *17th International Conference on Computing in Civil and Building Engineering*, Tampere, Finland.
- Vahdatikhaki, F., Hammad, A., **Taher, A.**, (2015). "Multi-agent Approach for Automated Guidance and Control of Earthwork Equipment", *2nd International Conference on Civil Engineering Informatics*, Tokyo, Japan.

Conference Poster

- **Taher, A.**, Vahdatikhaki F., Hammad A., (2019). "Towards Developing a Framework for Earthwork Operations' Ontology", *6th Graduate Student Colloquium in Construction, CSCE 2019*, Montreal, Canada.