Extracting and Integrating Industrial Construction Steel Trade Data in ill-formed BIM Models

by

Mostafa A Abdelaleem Ali

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Construction Engineering and Management

Department of Civil and Environmental Engineering
University of Alberta

Abstract

Oil and gas are known for their huge-size and complex projects that consist of multiple trades' subprojects such as concrete, steel, and piping. These subprojects are executed within a confined area during a limited time frame. This requires careful planning and coordination between these different trades. Each trade creates a separate Building Information Modelling (BIM) model which are merged with others into one huge model. This model is used for coordinating work packages and detect any possible clashes.

From the contractor perspective, the BIM model has additional uses such as defining scope and obtaining a preliminary estimate during early stages of the project. The model's potential depends on its degree of completeness and time of availability. However, the current practice in the industry and the usage of specialized BIM solutions means that during early stages of the project the BIM model will immature, incomplete, and inconsistent. This means the model usability becomes limited and the contractor has to review the model manually to extract the required information including the scope of each trade, a preliminary estimate of quantities, etc.

The objective of this research is to investigate and provide a new methodology that can automatically fill the missing data in the BIM model and leverage its usage. This objective is achieved through three main steps. 1) automatically cluster the BIM objects based on their trade, 2) refine cluster results by identifying the shape and size of BIM objects automatically, and 3) leverage BIM model usage by merging its data with other data sources.

Accordingly, this research is subdivided into three sections. The first section focuses on clustering BIM models based on the trade (e.g. steel, piping, concrete) of BIM objects. The research provides four mathematical models that are able to automatically cluster the BIM models with a purity level up to 91%.

The second step focuses on obtaining a preliminary quantity take-off for the steel trade in BIM models using shape recognition techniques. This approach focuses on geometries rather than the incomplete descriptive attributes. The research reviews different shape recognition techniques to select the most suitable technique. Then, it introduces a method to estimate steel sections using the shape distribution technique. Finally, it optimizes method parameters and tests the method using three real-world industrial project models. Results indicate that the proposed method works best using around 50,000 random distances with an 8.8% margin of error at a 95% confidence level.

The third section demonstrates how the enhanced BIM data can be automatically merged with heterogeneous data sources using semantic web standards. This sections discusses developing an ontology which captures concepts related to the visualization process. Then, heterogeneous data sources that are commonly used in construction are fed into the ontology. The potential of this approach has been demonstrated by providing multiple visualization scenarios that cover different audiences, levels of detail, and time resolutions.

The methodology has been implemented and validated using three real-case projects. Results show that the proposed framework can automatically process ill-defined and incomplete BIM model to fill the missing data. It provides a quicker way than the manual one to provide a preliminary estimate of quantities. Additionally, the framework allows automatic merge of data between BIM models and other heterogeneous data sources that are commonly used in

the industry. Data merge has many usages; the research proves its usability by providing a way to automatically generate visualizations based on customized queries.

Dedication

To my Parents,

For their endless love and support

Acknowledgments

I would like to express my gratitude and appreciation to my supervisor Dr Yasser Mohamed for his continuous support and encouragement.

My gratitude also extends to my doctoral committee Dr Mohamed Al-Hussein, Dr Simaan AbouRizk, Dr Evan Davies, and Dr Tarek Zayed for their valuable comments and insights.

I would like to thank Rick Hermann from PCL for providing data that were essential for the thesis.

I would like to acknowledge the contribution of the Natural Science and Engineering Research Council of Canada (NSERC) for their funding of this research through the Collaborative Research and Development (CRD), Grant number: CRDPJ 416362 – 11.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Background and Problem Statement

Building Information Modelling (BIM) becomes modelling standard for construction projects; it provides a virtual environment for the Architecture, Engineering, and Construction (AEC) industry where digital models can be generated, exchanged, and merged to increase collaboration and productivity in different phases of projects. Many case studies have shown the benefits of BIM in construction projects [1]–[3]. These benefits are observed not only by private companies, but by government agencies around the world, which have started to mandate BIM models for public sector projects [4], [5]. BIM usage can be categorized into passive and active: passive usage encompasses engineering analyses like safety and scheduling, while active usage involves extracting embedded knowledge in BIM [6]. BIM has great potential but there are some unsolved issues in its modelling process. These issues can be grouped into two categories: contractual (e.g., model ownership) and technical (e.g. interoperability) [1]. The following section summarizes some of these issues in industrial projects.

1.1.1 Current Practice in Industrial Projects

Industrial projects were among the first to use BIM technology; this is largely due to projects complexity and size, which have increased exponentially in the last five years [7]. Studies have shown that complexity and size determine the degree of information technology usage in a project [8].

Because industrial projects comprise many trades and require careful planning and coordination, they use modelling tools, such as Navisworks® and SmartPlant®, which can merge multiple 3D models. Because of confusion around the definition of BIM, there is no consensus about whether or not these types of software should be considered BIM tools [3].

Some researchers argue that a model containing only 3D objects or with few attributes is not a BIM model [3]. According to this definition, industrial models in early stages of a project, which do not have sufficient attributes, are merely 3D models. Other researchers divided BIM tools to authoring tools which are capable of handling objects' classes and relationships and tools that do not provide full BIM capabilities such as Navisworks [9]. A third group of researchers consider models with few attributes as BIM models [10], [11]. In this research, we use the generic definition of BIM which includes models containing 3D objects with few attributes.

A typical industrial project BIM model consists of multiple sub-models (e.g., structural, mechanical, electrical, etc.). Each model is designed separately and then all models are compiled by an engineering firm into one model to be reviewed and have any clashes detected. After that, the engineering firm issues the model to the contractor as one model. Figure 1-1 shows an IDEF0 diagram for this process. This process is repeated multiple times for fast-tracked projects.

The above process results in the following concerns in fast-tracked industrial projects:

1. **BIM ownership (contractual level)**: The contractor receives only the compiled model for reviewing and visualization. However, the contractor cannot add to the

model and, therefore, any operational attributes have to be saved in a separate database.

2. **Lack of standards (contractual level)**: The same item might be labelled “I beam,” “I-Beam,” or “I Beam column” based on the engineering firm’s conventions. In addition, some objects do not have essential attributes such as object’s trade and type.
3. **Model limitation (technical level)**: The contractor cannot calculate quantity take-off accurately as labelling is not complete and some authoring tools do not provide access to BIM objects’ solids.
4. **Interoperability (technical level)**: Transferring data between systems or software is a tedious, error-prone task. For example, receiving software might drop unsupported classes and properties [12]. Even the use of the Industry Foundation Classes (IFC) format has many limitations in non-product information [13], and round-trip benchmarks show IFC limitations in exchanging both geometric and semantic information [14], [15].

1.1.2 Integrating Heterogeneous Data Sources

During a project life cycle, different parties (e.g. architects, engineers, contractors, etc.) generate a massive number of documents, CAD drawings, and BIM models [16]. Integrating these heterogeneous data sources involves many essential applications such as 4D visualization [17], and merging BIM and GIS data [18], etc.

However, these data come in different formats which are optimized for a specific application. This means that machines cannot automatically process and link them and human inputs are required to map the data between different sources, which as mentioned earlier, is a tedious and error-prone task.

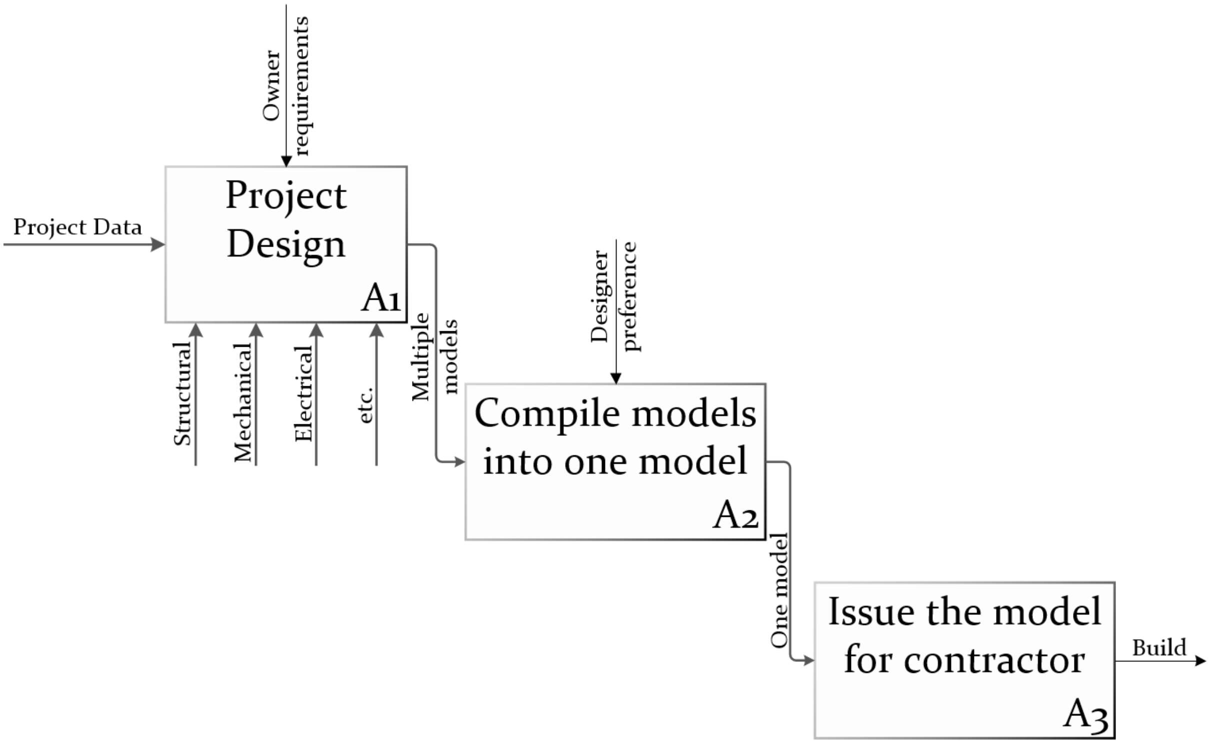


Figure 1-1: IDEF0 for issuing BIM model for the contractor.

A more universal and automatic way to merge different data sources is the semantic web [19]. The semantic web utilizes a simple data format known as a Resources Description Framework (RDF). RDF represents data as a set of triples and each triple has three parts: subject, predicate, and object. RDF provides a standard way to exchange data between different data sources [20].

In summary, data usage in fast-tracked industrial projects suffers from two limitations. First, the lack of data integrity and completeness in the BIM models which limits the model usability for preliminary analysis. Additionally, although the construction industry is information intensive [21], [22], there is no standard method for transferring and merging data between heterogeneous data sources, and these two tasks are usually performed on ad-hoc basis.

1.2 Research Objectives

The objectives of this study are based on the following hypothesis: During early stages of a fast-tracked industrial project, automated solutions can be developed to fill in missing data in BIM models and integrate these data with other heterogeneous data sources to increase information usage such as finding relations using data mining.

More specifically, the research aims to leverage the data usability for the steel trade in fast-tracked industrial projects. It focuses on data integration and processing from BIM models and other data sources during early stages of projects. During this stage, BIM models are incomplete yet contractors utilize these models to define the work scope for each trade, preliminary quantity estimate, and constructability analysis. This research aims to establish two steps to provide an automated solution to leverage data usage in BIM models at early project stages to 1) complete and validate inconsistent and missing data in the BIM models, and 2) link the BIM data with other data sources that may be used for project planning. The research objectives can be stated as follows.

1. Cluster BIM objects by trade based on attributes available in the BIM model even though they may be incomplete or inconsistent, and evaluate the quality of clusters under four alternative mathematical models.
2. Determine the shape and size of steel objects in BIM models based solely on their geometry by using a shape recognition algorithm; then quantify the total steel weight in the project and evaluate its accuracy.
3. Use a semantic web ontology approach to merge BIM data with other heterogeneous data sources that are commonly used by contractors. An ontology is the formal

definition of concepts that are used in a domain. In semantic web, it defines concepts that are captured in a set of triples.

4. Demonstrate the benefits of this framework by using automatically merged data to generate different levels of visualization for different activities in a project. The framework provides visualization with different levels of detail and a different time resolution.

1.3 Research Methodology

The research methodology, shown in Figure 1-2, consists of three key parts. The first part targets the clustering of preliminary 3D models which have minimal descriptive attributes and contain items from different trades. The research suggests multiple mathematical models that scan the 3D model and cluster items into separate clusters. The relevant mathematical models found in the literature are reviewed to select those that have potential for success in the BIM models' domain. The selected mathematical models are adapted and implemented. Then they are validated using real-case 3D models obtained from the industry, this step has been discussed in details in Chapter 2.

After separating the 3D model by trade, the second component of the research focuses on determining the shape and size of the steel sections of each item in the steel trade. Different shape recognition techniques, found in the literature, are reviewed and analyzed. Based on this analysis, a shape distribution algorithm [23] is selected. This technique is modified and adapted for the problem's domain and then used to compare the unidentified BIM object with a reference database that contains histograms for all steel sections found in Canadian standards [24]. This includes rolled solid sections (e.g., W, HP, C sections) and hollow sections (e.g., square and circular sections). The accuracy of the shape recognition techniques is estimated by drawing random samples from different real 3D models which are recognized manually and by the technique as explained in Chapter 3.

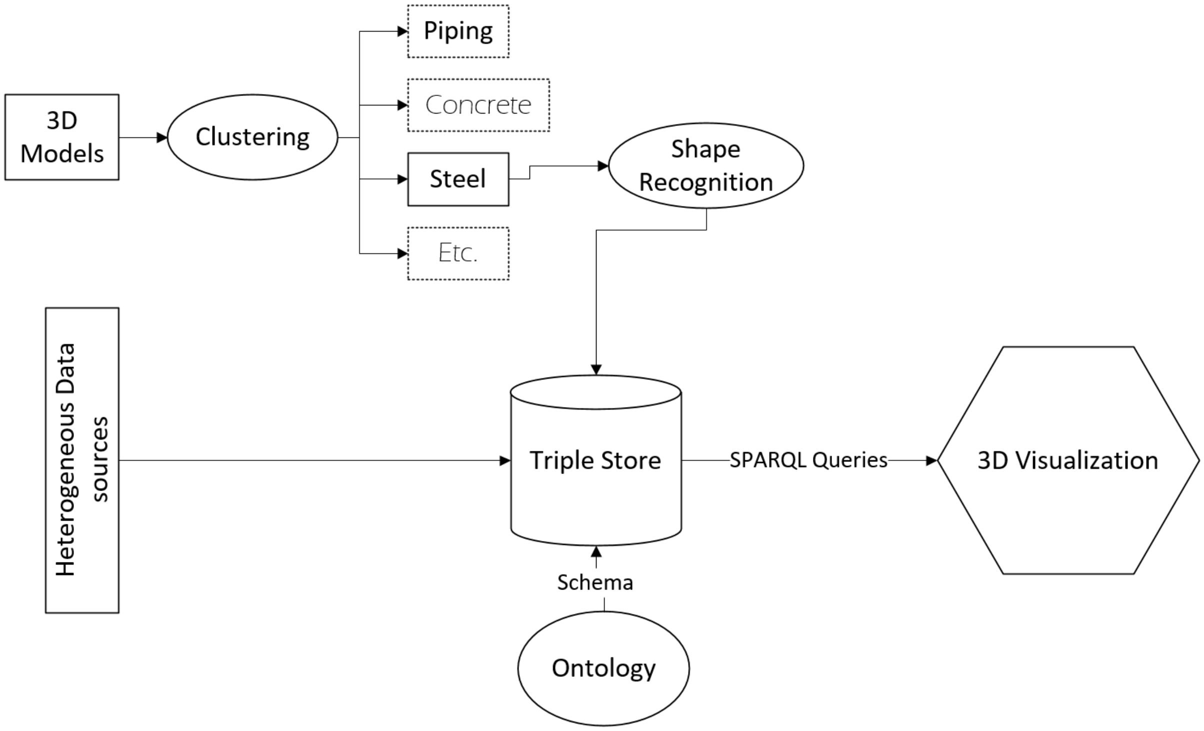


Figure 1-2 An overview of the research methodology.

The last part of the research, found in Chapter 4, focuses on merging the refined BIM data with other heterogeneous data sources commonly used in the industry to display visualization scenarios. This includes developing an ontology that conceptualizes spatial and temporal concepts. The ontology is backed by a triple store. Triple store is a specialized database that store RDF data and can process millions of generated triples (a triple represents one instance of data in RDF format). A set of custom connectors are introduced. These connectors use ontology to convert data to RDF triples. The data come from BIM models, scheduling applications, simulation engines, and spreadsheets. Afterwards, a custom code processes and reformats these triples to the selected visualization data-input format. The validity of this framework is tested by the framework's ability to automatically generate different visualization scenarios using different time resolutions and different levels of detail. Figure 1-3 shows the sequence of the work divided into separate chapters.

1.3.1 Validation

The proposed methodology is validated using three real-case oil and gas projects that have been executed in Alberta. Each step of the methodology is validated as follows:

- The mathematical models used for clustering the BIM models have been applied for the three projects. Then a random sample is drawn and investigated manually to determine the success rate and confidence interval.
- The shape recognition technique is applied for all steel objects in the three projects. Then an equal sample is drawn from each project to be identified manually to calculate success rate and confidence interval.
- Merging of heterogeneous data sources has been validated through providing different visualization scenarios using the proposed framework. The visualization scenarios include high-level and detailed visualization of different activities.

The details of each validation step have been discussed in details in the corresponding chapter.

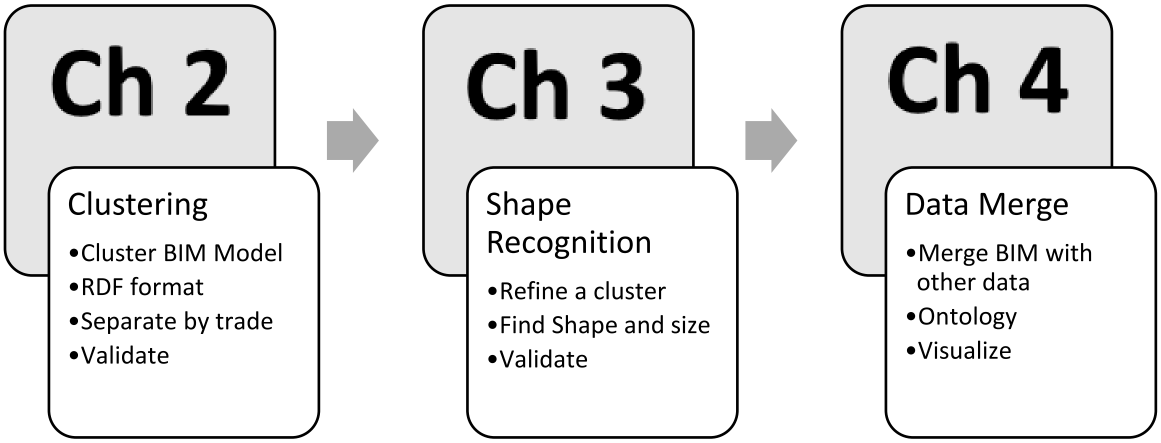


Figure 1-3 A summary of the conent of each chapter

1.4 Thesis Organization

The remainder of the thesis is organized as follows. Chapter 2 reviews the common BIM modelling process for fast-tracked projects. It shows that during early stages the models are incomplete, inconsistent, and missing attributes. These models cannot be automatically categorized into separate trades as most of the BIM models' objects lack an explicit declaration about their attribute. The chapter discusses our work using Shannon Entropy and TF-IDF methods to categorize BIM objects based on their trade. It shows four mathematical models that have been used to cluster the BIM objects. The suggested models have been tested with three real-case BIM models. Finally, the chapter summarizes the performance of each model, compares them and states their limitations.

Chapter 2 manages to separate BIM objects by trade, which means steel objects in the BIM model can be filtered out; however, more details about them (e.g., shape and size) are needed, which requires more detailed analysis. Chapter 3 discusses using shape recognition techniques to determine the shape and size of steel objects. It starts by reviewing shape recognition algorithms in terms of their applicability to our domain and their computation time. Afterwards, it implements the selected algorithm (i.e., Shape Distribution Algorithm) and tests it with three real-case BIM models to determine if it can find the shape and size of steel objects. Finally, the chapter summarizes the results and states the limitations of the proposed methodology.

Chapter 4 discusses a framework for visualizing heterogeneous construction data using semantic web standards. The chapter reviews the previous attempts to visualize construction projects and shows that these attempts suffer from two limitations: tightly-coupled data processing and targeting a specific level of details which limits the generality of the visualization process. Afterwards, the chapter outlines the proposed framework that breaks the visualization process into two separate parts: collecting the data, and formatting it for visualization. It discusses how the semantic web standard can be used to merge heterogeneous data sources into one data store and how SPARQL queries can generate the input files for a visualization application. The framework has been tested by being implemented for the common data sources used in construction projects. These data have been used to visualize multiple scenarios with different levels of details and time resolutions.

Finally, Chapter 5 concludes this research and summarizes the academic and industrial contributions, limitations, and future work. Appendix A and Appendix B show the code related to the clustering method. Appendix C shows the code related to the shape recognition technique. Appendix D shows the proposed ontology. Appendix E shows a sample of the generated triples. Appendix F and Appendix G show a sample of the visualization input files.

Chapter 2. A Method for Clustering Unlabeled BIM Objects Using Entropy and TF-IDF with RDF Encoding[1]

2.1 Introduction

The benefits of building information models have been identified in both Architecture Engineering and Construction (AEC) practice and research [1], [3], [25]. Industrial oil and gas projects are among the early adopters of the technology. In these projects, designs are produced by different engineering disciplines (i.e., structural, mechanical, electrical, etc.) and represented as separate BIM models. These models are merged into one model to check for conflicts or clashes. Afterwards, the merged model is issued to one or more contractors for construction. Large industrial projects are typically fast-tracked [26] and as such, BIM models of these projects are not finalized before construction starts. Consequently, they are issued periodically as designs progress in parallel to construction [1].

From a contractor perspective, although models may be incomplete and subject to change, they still represent a rich source of information that can be used for preliminary resource planning and estimation [3]. However, due to legal issues such as intellectual property and contractual requirements [27], and because most models are compilations of many incomplete smaller sub models, some data are lost, missing, or inconsistent. This hinders the ability to

[1] This chapter has been submitted to the Journal of Advanced Engineering and Informatics

automate different tasks in the model [28] and the contractor has to spend a significant number of man hours investigating and extracting relevant data.

A common requirement for a contractor is to classify BIM objects based on their trade (e.g., structural, mechanical, electrical, etc.). Traditionally, this involves reviewing the model manually and identifying custom rules that can be used to filter objects for each trade. A rule is a way to specify an attribute value that is consistent over the domain of one trade such that if the model is filtered based on this rule, only objects from this trade are retrieved. Identifying these rules is a tedious manual task that has to be executed for each project because rules inevitably change based on different modellers' conventions. Additionally, it is a repetitive task that has to be refined many times to make sure the right rules have been found. This task may also need to be repeated every time a new version of the model is released to the contractor.

This chapter proposes a method to automate this task. The proposed method relies on encoding a BIM model using the Resource Description Framework (RDF) [29], then applying an algorithm that utilizes Shannon entropy [30] and Term Frequency Inverse Document Frequency (TF-IDF) [31], [32] measures to group the objects into clusters that represent different trades.

In the following sections of this chapter we first explain the research objectives, then review literature related to the research problem, outline our proposed solution and, finally, present the results of testing the proposed solution on three real case studies for oil and gas projects that have been successfully completed in Alberta, Canada.

2.2 Objectives

This study is based on the assumption that the target BIM model is imprecise, inaccurate and made of several objects that have different attributes and different values for these attributes but none of these attributes explicitly identify the trade for a given object. It is also assumed that the set of attributes and their values change from one project to another. Ideally, a BIM model should contain enough data to sufficiently describe each object in the model. However, when merging models of different engineering systems that have been developed using different software applications into one BIM model (e.g., NavisWorks © [33] models, which are common in oil and gas projects), some data may be missing or inconsistent due to technical issues when merging different models, or incomplete design; or because of intellectual property, or contractual requirements. Data loss or inconsistency can severely limit the usability of the final model by the contractor who has to visually review and inspect objects in the model to extract required information about work items for different construction trades.

The objective of this research is to develop an automated solution to replace or support this visual inspection task. With minimum manual intervention from the user, this solution should assist in identifying distinct groups of objects in the model that belong mostly or fully to the same trade.

To achieve this objective, the following questions need to be addressed:

- Are the common attributes between objects from the same trade sufficient to distinguish them in a merged BIM model? A merged BIM model contains data from different sources which differ based on related attributes, authoring software, and modeller preferences. Hence, the proposed method has to find the attributes that are candidates for filtering objects based on a specific trade.

- What is a suitable method to encode and query a model to find these common attributes automatically? A trial and error method can usually be used to find these attributes manually. Instead, we examine the use of a semantic web data model (RDF) and query language (SPARQL) [34] to find these attributes automatically.
- How does one differentiate between common attributes that are used by more than one trade? After finding a common attribute within a trade, the attribute should not automatically be used to filter objects for that trade because the same attribute might be used in another trade as well. Here, Shannon entropy [30] and TF-IDF [31], [32] measures are used to test how much information a given attribute can provide to differentiate between trades.
- What is an appropriate measure of the success of the proposed method? Once clusters are identified, a measure of the purity of these clusters needs to be used to evaluate how much of the objects in a cluster truly belong to the same trade.

The following section reviews relevant solutions used in previous studies to address problems similar to the ones described above. It also provides justification for the choice of solutions used in this study.

2.3 Previous Work and Related Studies

The following sections discuss previous work related to information migration and merging and how semantic web technology is used as potential solution to improve BIM models. The discussion highlights why previously suggested approaches are not suitable for use in this study especially with the assumption of data completeness of BIM models, which is not always true in practice and is a key motivation of this research. Finally, some background about the different clustering and similarity measures used in this study is given.

2.3.1 Information Migration and RDF Format

Information is an essential asset for any business and the value of the information increases with the ability to link data from different sources [35]. Unfortunately, linking, transferring, and merging data from different sources is challenging as it requires creating a new data schema (a definition of a data's structure) that adapts data from different sources and transferring existing data to this new schema. Schema migration is not straightforward; it requires not only transferring data, but updating queries and business-logic layer [20].

As an alternative to rigid data schemas, RDF utilizes a flexible format [19]. The RDF statement consists of three parts: subject, predicate, and object, which is known as a "triple." Each part of the RDF triple (e.g., subject) is known as a Resource and it should be expressed uniquely using a Uniform Resource Identifier (URI). For example, if we want to use the RDF format to express that "Olivia sells antiques," each part (i.e., Olivia, sells, antiques) should have a unique URI. Using RDF triples can help with data transfer and merging but it requires a logic layer to describe the semantic relations between resources in the different triples.

2.3.2 Semantic Web

In order to add semantics to the stored triples, an ontology should be defined. Ontology is a precise explanation of terms and reasoning in a data domain that allows machines to act as if they understand the meaning and find relations in the domain. This is arguably one of the most powerful features of a semantic web framework. Along what was shown in the previous example, if the ontology states that “if ?x sells ?y then ?x is the seller,” then the machine can infer that Olivia is a seller. Ontology has been proven to enhance mapping data from different schemas [36].

The semantic web is considered to have evolved from the current web model [37], hence the name. The current web model is a set of web pages that contains plain text, figures, and tables, and is decipherable by humans but impossible for a machine to process in a semantic way. This is evident in the current search model where accuracy and relevance are closely related to the quality of keyword formations composed by humans. On the other hand, when the semantic format is utilized, search engines will actively help us retrieve all data related to the concept we are looking for as the search engines will not look for a simple keywords but they will retrieve a specific concept defined in an ontology; this can be demonstrated clearly by using semantic-based web sites such as DBpedia [38] and Wikidata [39], [40].

In addition to its popularity in the web world, the semantic web has many applications in other domains. For example, traditional database structures cannot sufficiently model relations in an object-oriented paradigm. A schema for a steel structure domain that models structural beams and columns cannot specify that the beams and columns have the same superclass (e.g., a structural element); therefore, it is the responsibility of the application side to capture this relationship. This is different in the semantic web perspective which can encourage capturing relationships between objects using statements like “*steel:Beam rdfs:subClassOf steel:StructuralElement*” and “*steel:Column rdfs:subClassOf steel:StructuralElement.*” This feature allows a complete decoupling between knowledge representation and the application layer, which means the data will be a stand-alone object with less dependence on software applications, which in turn means fewer problems when transferring data between applications. This problem of transferring data between different systems and applications has been discussed extensively in the literature [41]–[44].

2.3.3 Merging Information from Different Sources

The semantic web is an ideal approach to merge heterogeneous information sources. For example, in order to manage big and complex projects such as the Sydney Opera house, different systems have been used. These systems cover all data related to the facility including spatial data and benchmark databases. [35] suggests using the semantic web to merge these different data sources with the Industry Foundation Classes (IFC) model. They found that the semantic web provides a flexible platform for integrating data sources. Although they criticized the immaturity of software that implemented the semantic web technology, based on our experience, we can safely say that the semantic applications (both commercial and open-source) have evolved recently.

Semantic web technology has evolved in recent years from a pure research topic to a practical solution for domains, such as business [45] and vocabularies definitions and relations [46]. In the industrial construction domain, the upper ontology that appeared in ISO 15926 [47] helps promote the use of the semantic web and Web Ontology Language (OWL) in many industrial applications [48]–[50]. Studies have shown that ISO 15926 ontology overcomes the inability to capture object changes over time in the STEP data model [48]. The semantic web has been used by the Obama administration to provide transparency for government data [20]. Other governments have also adopted this approach [51], [52].

2.3.4 BIM and the Semantic Web

Semantic web has proliferated to the IFC file format. IFC is a data model for exchange data in AEC industry, and it is supported by most BIM suites. However, its current version has some drawbacks such as using a rigid schema that cannot be easily edited which hinders extensibility, geometric misrepresentation by software [53], and semantic data loss [54]. Also, studies have shown that it cannot be easily linked to other data sources [35], nor can it be used effectively to present management information [55], [56].

Previous studies suggested that semantic web technology can overcome many of the current IFC data model's limitations. For example, Jung and Joo emphasized using ontology with reasoning in BIM framework to automate spatial and temporal interrelationship [6]. Additionally, in their vision of BIM2.0, researchers suggested that information should be "*up-to-date and open for derivation of new information*" [57].They suggested that these various requirements will be addressed automatically if well-built ontologies are used with BIM models.

Many researchers introduced frameworks for migrating IFC files to an ontology. Ontology might be the preferable method to describe BIM data over IFC standards. This is because in addition to addressing the aforementioned issues, it adds a semantic layer over the synthetic data layer which improves queries and reasoning. For example, the IfcOWL [58] project converts EXPRESS schemas (which are used by IFC) to an ontology. It maintains the schemas' taxonomy by using the corresponding structure in OWL specifications. This project is currently embraced by building SMART® as a future development of the IFC standard [59].

Integrating BIM with semantic web technology provides a means to automate different tasks and applications. This includes: 1) generating partial models based on queries [60], 2) generating IFC models based on semantic web queries [61], 3) annotating BIM online resources using the semantic web to facilitate categorization and item retrieval [62], 4) integrating multiple data sources required during the facility management phase [63], and 5) merging BIM and GIS data [64].

Additionally, the semantic web leverages the capabilities of BIM. For example, one of the expected outcomes of BIM modelling is rule checking, which means the ability to automatically check that the model complies with regulations; historically, this is performed using IFC specifications [65]. However, with the increased complexity of BIM models, IFC fails to fulfill this role; it has been shown that this gap can be filled using semantic web technology [66], [67].

These researches used IFC files as a starting point for applying the semantic web technology on BIM models. They assume a precise well-defined BIM model as a starting point; however, this model might not be available in the real world. Hence, the research in this article takes a different approach by considering an imprecise, inaccurate, and uncertain BIM model that is typically circulated between different parties during early stages of construction projects. Based on these incomplete models, we used semantic web technology to cluster model objects based on their trades.

2.3.5 Clustering and Similarity Measures

One of the key components of semantic web mining is finding the similarity of different concepts based on their taxonomy as defined in ontologies [68]. Many methods have been proposed to measure semantic similarity [69], [70]. These methods form the basis of clustering semantic web data [71].

Clustering is a technique that subdivides a set of data into groups based on the similarity between data points. In most clustering algorithms, a distance between points is used as the similarity measure, which works very well with numerical data types. However with string attributes, it becomes more challenging to use geometrical distances [72].

Entropy measurement has been suggested as an alternative way to measure the similarity between different data points [73]–[75]. Entropy measures the uncertainty associated with an upcoming event [30]. For example, if we have a system with possible events $(E_1, E_2, \dots, E_n)$ and their probabilities $(p_1, p_2, \dots, p_n)$, the entropy of the system can be calculated as follows:

$$E = -\sum_{i=1}^{n} p_i \log_2 p_i \qquad [30] \qquad \textit{Eq. 2-1}$$

If there is only one event, E will be zero (i.e., there is no uncertainty). If we have two events, the maximum entropy happens if they have the same chance. Figure 2-1 shows the change of Shannon entropy value for a system of two possible events. It shows that a maximum value of one will happen if each event has a probability of 0.5 (the most uncertain case).

As we can see in the figure, entropy is minimized if there are a few different values in a system, while it increases steadily with more available options. As will be shown in Section 2.4, this feature is used to test if an attribute can be used to cluster the model. For example, an attribute that contains unique ID of each BIM object will have a high Entropy value and hence discarded.

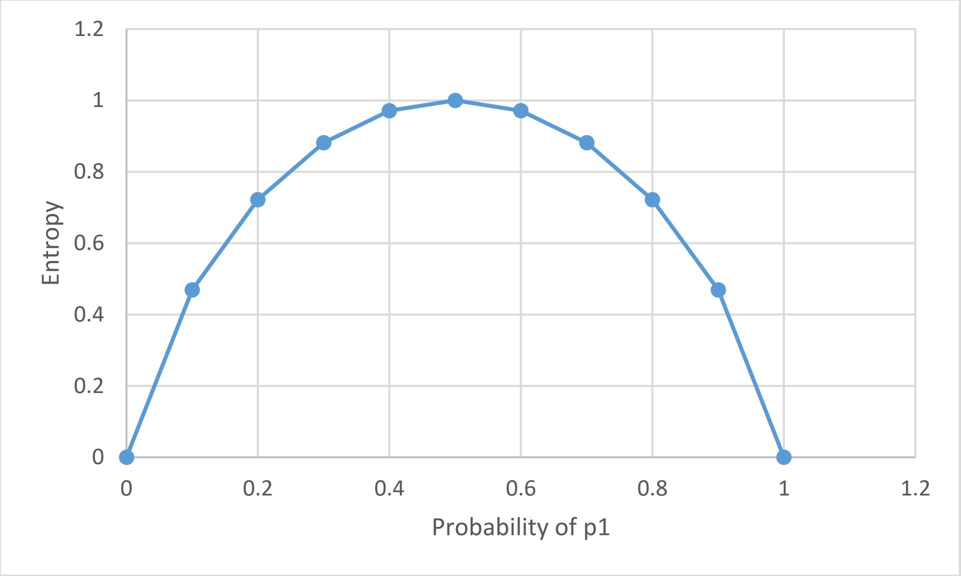


Figure 2-1 Shannon entropy is maximum when each event has a probability of 0.5 (the most uncertainty case) as in Eq. 2-1

Another measure commonly used for clustering text documents is Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF is a weighting function that has been used widely in text mining and information retrieval [76]. TF-IDF is a multiplication of two quantities, TF, which calculates the frequency of each term in a document, and IDF, which offsets the term weight by assuming that the importance of a term is inversely proportional to the frequency of the term in all documents in a corpus [77]. For example, if the word “safety” appears with high frequency in a set of documents, the TF-IDF measure ranks it low as a term to differentiate between these documents. Although TF-IDF was originally described as a heuristic method, some papers have related it to Shannon’s information theory [78].

TF-IDF is easy to compute and robust [78]; however, it depends on the concept of bag-of-words which ignores the relationship between terms and it does not consider synonyms. Also, extensive computation is required when dealing with large documents [79].

There is a strong analogy between using TF-IDF in document retrieval and the problem domain described in this study. Attribute values can be considered terms, while BIM objects represent a set of documents. Thus, we examine the use of the TF-IDF model to cluster instances based on relevant attribute values as shown in Section 2.4.

2.4 Summary of Proposed Method

The proposed method includes the following main steps: 1) encoding and querying BIM data using RDF and SPARQL to identify groups of objects with the same set of predicates, 2) identifying attributes that are common to a group but not common to the whole model using entropy and TF-IDF measures, and lastly 3) using identified attributes to merge groups of objects into clusters that represent work for the same construction trade. The following sections describe each of these main steps in more detail.

2.4.1 Model Encoding and Query

First we convert BIM model data to an RDF format. Unlike converting to traditional databases that will create high dimensional but sparsely occupied tables [80], RDF can be represented visually as a graph which makes it more efficient for representing BIM data. Many researchers outlined proposed methodologies to convert BIM model data to triples as described earlier. However, in our case, we use incomplete BIM models that do not have an explicit indication for each object’s trade. Therefore, the conversion from BIM to RDF results in a shallow ontology with few classes that cannot be used to infer trades based on semantics only.

Our conversion scheme utilizes the following standard classes: *rdf:type*, *rdfs:subClassOf*, *rdfs:Label*, *rdfs:range*, *owl:Ontology*, *owl:Class*, *owl:Thing*, *owl:DatatypeProperty*, and *xsd:string*. Along these standard classes, the ontology defines a custom class ":ModelItem," which represents any object in the BIM model. The conversion code can be found in Appendix A.

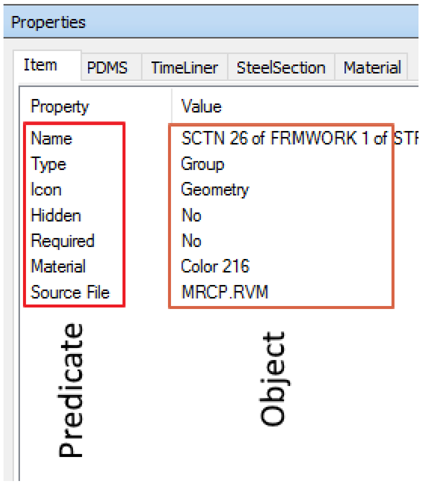
Attributes' names and values associated with each instance are converted to predicates and objects respectively as shown in Figure 2-2. Hence, a typical triple consists of a subject (a unique ID for each BIM object), a predicate (a property name), and an object (the property value). The number of triples associated with each BIM object is determined based on its associated properties as in Figure 2-2.

RDF representation does not consider a predicate merely as string value; instead it is an object (in the OOP realm) that supports instantiation, inheritance, etc. Therefore, each attribute name in a BIM model is converted to a custom class in the ontology.

Each attribute value is stored as an object – which can be derived from – in the ontology. Although RDF supports using classes and instances for the object part of the triple as well, we use only a literal node (a string) because the values are simple datatypes that do not encapsulate any relationships.

Based on this conversion, a set of triples is created. Any following step is independent of the BIM application because it deals with the generated triples set which can be extracted from different BIM applications.

Each subject in the triple set represents an object in the model; however, there are no relationships between instances from the same trade. In order to find those relationships, we start by querying the resulting triple store to select all instances that have the same set of predicates (regardless of the value of each predicate). This step generates a set of object groups where each group contains objects with the same predicate set. These ad-hoc groups only represent objects with the same attributes' set and each group contains items from different trades. The next is step is clustering these groups using Shannon Entropy and TF-IDF as follows.



Property	Value
Name	SCTN 26 of FRMWORK 1 of ST
Type	Group
Icon	Geometry
Hidden	No
Required	No
Material	Color 216
Source File	MRCP.RVM

Subject	Predicate	Object
:ModelItem1	:Type	“Group”
:ModelItem1	:Icon	“Geometry”
:ModelItem1	:Hidden	“No”

Figure 2-2 BIM attributes' name and value are converted to predicates and objects respectively.

2.4.2 Identify Candidate Attributes for Clustering

After the initial breakdown of the model into groups with similar predicates, and within each group, we want to select attributes that have a small number of values. We call these attributes “Dominant Attributes” (D_A). The goal here is to exclude all attributes with unique values for each instance, such as object IDs or unique names, as they cannot be used to cluster the model. In order to achieve this goal, two alternative methods are used (Shannon entropy and TF-IDF) to calculate a weight for each attribute within each group, and these weights are used to order attributes.

In the case of Shannon entropy, the measure guarantees that attributes with a unique value per instance are excluded. On the other hand, an attribute with a small entropy value does not necessarily mean it is usable because it might have the same value over different groups; hence, it also cannot be used to cluster trades. Therefore, entropy for each attribute of the D_A is recalculated based on the whole domain (versus per group), and the attributes with the highest value - we call them Clustering Attribute C_A - are selected.

2.4.3 Clustering Using Selected Attributes

Finally, instances with the same C_A value are merged into one cluster. This cluster represents objects from the same trade. The merging is executed using a standard SPARQL query.

2.5 Detailed Algorithm and Example

2.5.1 Mathematical Representation and Pseudo Code

For more clarity and for coding purposes, the steps of the proposed method described above are represented using mathematical notations and pseudo code (List 2-1) in the following sections. The developed code can be found in Appendix B.

This proposed methodology can be formulated mathematically as follows:

Given a set of instances (In this context, an instance represents a geometrical BIM object):

$$I = \{I_1, I_2, \dots, I_n\} \qquad \text{Eq. 2-2}$$

Where each instance contains a set of predicates P and values V

$$\forall\, I_i \in I = \{(P_{i1}, V_{i1}), (P_{i2}, V_{i2}), \dots, (P_{ij}, V_{ij})\} \qquad \text{Eq. 2-3}$$

Instances are placed into groups G where each group has the same set of predicates regardless of their values

$$G = \big\{x \in I \mid x = \{P_1, P_2, \dots, P_k\}\big\} \qquad \text{Eq. 2-4}$$

To determine a D_A that has a few number of values within a group but many values within the whole domain, the following two alternative weighting models are used to evaluate attributes within each group:

Model 1: Shannon Entropy

Shannon entropy value E_g is calculated within a group G_i using Eq. 2-1.

$$\forall\, P_i \in G_i \rightarrow E_g(P_i) = -\sum_i p(v_i)\, log(p(v_i)) \qquad \text{Eq. 2-5}$$

$$where\ p(v_i)\ is\ the\ probability\ of\ the\ value\ v_i\ within\ the\ group$$

Then, another entropy value E_d is calculated for each D_A for the whole domain.

$$\forall\, P_i \in G_i \rightarrow E_d(P_i) = -\sum_i p_{all}(v_i)\log(p_{all}(v_i)) \qquad \textit{Eq. 2-6}$$

$$\textit{where } p_{all}(v_i) \textit{ represents the probability of the value } v_i \textit{ for the whole domain}$$

Finally, E_g and E_d are used to calculate one weight measure (W_i). All attributes are ordered in descending order based on W_i. Two alternative equations are used to calculate this weight measure. The first one (**Model 1A**) is:

$$\forall\, P_i \in G_i \rightarrow W_i = \frac{E_d(P_i) - E_g(P_i)}{E_d(P_i) + E_g(P_i)} \subseteq [0,1] \qquad \textit{Eq. 2-7}$$

If the value of W_i is equal or close to 0, it means that E_d equals or is close to E_g and this attribute has many different values and has to be excluded. On the other hand, if W_i equals 1, $E_g = 0$ and this attribute has one and only one value in the group. Another model (**Model 1B**) takes only weights less than one which guarantees that each initial group is clustered into at least two clusters.

$$\forall\, P_i \in G_i \rightarrow W_i = \frac{E_d(P_i) - E_g(P_i)}{E_d(P_i) + E_g(P_i)} < 1 \qquad \textit{Eq. 2-8}$$

<u>Model 2: term frequency–inverse document frequency</u>

Based on the TD-IDF concept, this model uses a predicate and its values to promote D_A as follows (**Model 2A**):

For each value for each predicate in the group calculates tf and idf values

$$\forall\, P_i \;\in\; G_i \;\rightarrow \forall\, v_i \;\in\; P_i \;\rightarrow tf(v_i) = \frac{count\ of\ the\ value\ in\ the\ group}{count\ of\ items\ in\ the\ group}$$

$$\forall\, P_i \;\in\; G_i \;\rightarrow \forall\, v_i \;\in\; P_i \;\rightarrow idf(v_i) = \log_{10} \frac{count\ of\ groups}{count\ of\ groups\ containing\ the\ value} \quad [77] \quad Eq.\ 2\text{-}9$$

$$tf - idf(v_i) = tf(v_i) * idf(v_i)$$

Then calculate the weight of the predicate and order by this value in descending order

$$\forall\, P_i \;\in\; G_i \;\rightarrow W_i = average(tf - idf(v_i)) \quad Eq.\ 2\text{-}10$$

Model 2B uses the same equations as Model 2A, but only includes weight less than 1, for the same reason stated in Model 1B.

$$\forall\, P_i \;\in\; G_i \;\rightarrow W_i = average\big(tf - idf(v_i)\big) < 1 \quad Eq.\ 2\text{-}11$$

Each model provides a list of D_A. Based on the weight, the top three attributes are considered C_A and have been tested with real case scenarios as shown in the following section.

List 2-1 A pseudo code for the proposed method.

```
• Select all instances with the same predicate set.
• For each group:
    o For each predicate in the group:
        ▪ Calculate predicate weights based on models 1A, 1B, 2A, 2B.
```

    - *Select the attribute with the highest weight (D_A) for each model and store them in separate lists.*
- *For each weight calculation model:*
    - *Select the top 3 attributes (C_A) in its list.*
    - *Use each attribute to cluster the BIM objects.*
    - *Evaluate each weighting model to determine its validity.*

2.5.2 A Numerical Example

This section provides a small example to illustrate how the weight can be calculated by each mathematical model. In this example, the model is divided into 20 groups and Table 2-1 shows a group that contains five BIM objects and three attributes.

For Shannon Entropy, we first calculate the probability of each unique value for each attribute. Attribute 1 has five unique values with probability equals 0.2 for each. Attribute 2 has two unique values with probability 0.8 and 0.2 respectively while the last attribute has one unique value with probability of 1.

Accordingly, the Shannon values E_g for the three attributes are 2.32, 0.72, 0 respectively. This means that "Attribute 1" differs significantly within the group while "Attribute 2" and "Attribute 3" has few values. However, this is not enough to select D_A as the same steps have to be repeated for these attributes taking into consideration the whole model – instead of one group. We will assume that the Shannon value E_d is $\{2.5, 6, 0.2\}$ which means the weight is $\{0.04, 0.79, 1.00\}$. Hence, Model 1A will select "Attribute 3" as a top candidate while Model 2A will select "Attribute 2".

The TF-IDF method will first calculate the TF term which is equivalent to the probabilities mentioned earlier. The second term (IDF) gives more weight for values that appear in fewer documents. For example, if we assume "Attribute 1" contains unique IDs then each value IDF will equal $\log_{10} {}^{20}/_{1} = 1.3$. On the other hand, if "Attribute 3" value appears in 19 groups, then its IDF will be 0.02. We will assume that "Attribute 2" IDF is $\{0.82, 1.00\}$. Then multiplying the terms and averaging values for each attribute, the weight will be {0.26, 0.43, 0.02}. Hence, both Model 2A & 2B will select "Attribute 2" as a top clustering candidate.

Table 2-1 Objects in one of the groups and their corresponding properties.

BIM Object	Attribute 1	Attribute 2	Attribute 3
Ob1	AB1	T1	True
Ob2	AB2	T1	True
Ob3	AB3	T1	True
Ob4	AB4	T1	True
Ob5	AB5	T2	True

2.6 Testing and Evaluation of the Proposed Method

The proposed method has been tested with three real projects – named P1, P2, and P3 here – that have been successfully completed in Alberta, Canada in previous years. These projects represent three huge facilities for the oil and gas industry. The average budget was around $750 million per project. Each project includes construction activities for a different trade (e.g., civil, mechanical, electrical) that had to be executed in a tight time frame.

A BIM model for each project has been retrieved; these models contain data for all engineering disciplines (e.g., civil, mechanical, electrical) and there is no explicit attribute that states the discipline for each object in the BIM models. As shown in Table 2-2, the number of objects in each model varies between 750,000 to 2,500,000 objects. Object attributes have been encoded – using developed code and dotNetRDF library [81] – to triples. We tested two triple stores, the Fuseki version 2.4.1 [82] and Stardog 4.2.1 [83], to store generated triples. Table 2-2 shows the number of generated triples along with the processing time for each triple store. Triples have been processed in batches of 10,000 triples using a machine with an Intel® Xeon® CPU E5-1650 3.20 GHz and 32.0 GB RAM memory.

Table 2-2 Number of objects and corresponding triples for each project along with processing time in milliseconds and hours.

Project	No. of Objects	No. of triples	Processing Time in milliseconds (hrs)		Database size (GB)	
			Fuseki	Stardog	Fuseki	Stardog
P1	752,093	18,386,176	9,292,573 (2.58)	4,893,103 (1.36)	2.85	1.09
P2	922,038	11,936,925	6,270,449 (1.74)	2,986,134 (0.83)	1.91	0.73
P3	2,459,939	49,915,158	38,818,936 (10.78)	17,102,270 (4.75)	7.75	3.67

Table 2-3 shows the number of unique attributes in each project as retrieved using a standard SPARQL query. Interestingly, the project with the fewest number of triples has the largest number of unique attributes, which indicates that each BIM model has a different modelling style in terms of geometric and descriptive detailing.

The first step in categorizing the data is grouping triples based on their predicates (i.e., find objects that have the same set of predicates regardless of the objects' values – refer to Figure 2-2). Table 2-3 shows the number of initial groups for each project and the average number of objects in each group.

Table 2-3 Number of unique attributes and initial groups for each project.

	P1	P2	P3
Number of objects	752,093	922,038	2,459,939
Number of attributes	215	50	75
Initial groups	286	128	29
Avg. objects per group	2,629.70	7,203.422	84,825.48
Standard deviation	27,034.62	62,902.36	163,198.3

After creating initial groups, a program that processes each group using the proposed weighting models described earlier is used to identify the D_A for each group. The same D_A might be promoted by a weighting model for multiple groups. We take the top three D_A (i.e., the ones that appear in the largest number of groups) for each weighting model and use each one of them individually to cluster the BIM model.

The method used for testing and evaluation is summarized in Figure 2-3. It starts by clustering the BIM model based on a cluster attribute C_A, then determines the trade of each cluster by visually inspecting the BIM model, and finally calculates the purity level – an evaluation of clustering results – of each cluster.

Because of the large number of instances, we resort to a sampling technique to validate our method and calculate the purity value. For each cluster set, different random samples have been drawn. Each sample is collected from all generated clusters in proportion to the cluster size but not less than 10 instances from each cluster.

$$Cluster\ sample\ size = {}^{Cluster\ Size}/_{Domain\ Size} * Total\ Sample\ Size \geq 10 \qquad Eq.\ 2\text{-}12$$

The total sample size for each case varies slightly based on the actual number of clusters. The average is 410 instances per cluster set. Every instance in the sample set is reviewed manually to determine its discipline based on visual inspection. Figure 2-4 shows a screenshot of one of the projects and some of the trades generated by the model. By adding a label for each instance we calculated the purity measure based on the following equation.

$$purity(\mathbb{C}, T) = \frac{1}{N}\sum_{k} \max_{j} \left| c_k \cap t_j \right| \qquad [32] \qquad \text{Eq. 2-13}$$

where N is the total sample size, $\mathbb{C} = \{c_1, c_2, \dots, c_k\}$ is the set of clusters, and $T = \{t_1, t_2, \dots, t_j\}$ is the set of disciplines in the BIM model.

Purity values for all scenarios are illustrated in Table 2-4 and Figure 2-5 - Figure 2-7. Because the purity measure is calculated based on a sample rather than the whole population, we calculated a confidence interval around these values based on a sample size using the following equation.

$$P = \left[f + \frac{Z^2}{2N} \pm Z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{Z^2}{4N^2}} \right] \Big/ \left[1 + \frac{Z^2}{N} \right] \qquad [84] \qquad \text{Eq. 2-14}$$

where f is the purity measure, N is the sample size, and Z equals 1.65 (confidence level equals 95%).

Table 2-4 shows the calculated confidence interval for all scenarios. The purity level varies between [50% and 95%] with an overall weighted average of 91%. The results indicate that Model 1A performs poorly compared to other models and models 1B and 2B are slightly better than Model 2A. Comparing results between different projects shows that the performance of Project 2 is slightly worse than other projects for all weighting models as it contains fewer attributes comparing to the other two projects, which shows the effect of the modeling style and the level of details embedded in the BIM models, this will be discussed further in section 2.7. However, in all cases, the proposed weighting models are able to capture the C_A with the highest purity level but not necessarily as the first choice.

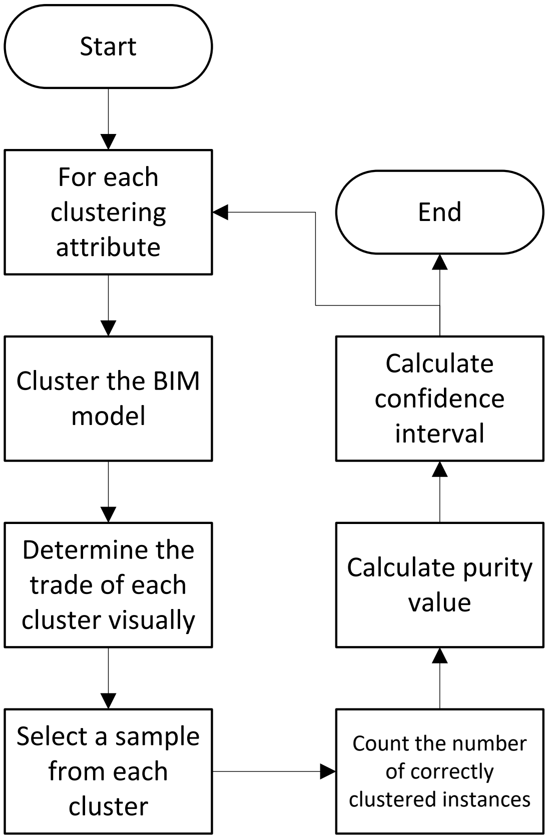

Start
For each clustering attribute
Cluster the BIM model
Determine the trade of each cluster visually
Select a sample from each cluster
Count the number of correctly clustered instances
Calculate purity value
Calculate confidence interval
End

Figure 2-3 A flow chart for the validation process for each weighting model.

The proposed method can save significant time for practitioners as it automatically tests all attributes in the BIM model (113 attributes on average) and suggests three attributes only. Practitioners can quickly cluster the model by each attribute, examine it visually, and determine which one is best for the model.

Table 2-4 Purity measure and confidence interval for different weighting models in all projects.

Project	Weighting model	Dominant attribute	# of clusters	Sample size	Purity measure	Confidence interval
P1	Model 1A	1	10	165	58%	[%51, %64]
		2	31	317	82%	[%78, %86]
	Model 1B	1	31	317	82%	[%78, %86]
		2	58	525	95%	[%93, %96]
		3	47	450	93%	[%91, %95]
	Model 2A	1	47	450	93%	[%91, %95]
		2	58	525	95%	[%93, %96]
		3	10	165	58%	[%51, %64]
	Model 2B	1	47	450	93%	[%91, %95]
		2	58	525	95%	[%93, %96]
		3	31	317	82%	[%78, %86]
P2	Model 1A	1	19	104	50%	[%43, %59]
		2	101	206	87%	[%83, %91]
		3	62	983	75%	[%73, %77]
	Model 1B	1	101	206	87%	[%83, %91]
		2	19	104	50%	[%43, %59]
		3	62	983	75%	[%73, %77]
	Model 2A	1	101	206	87%	[%83, %91]
		2	19	104	50%	[%43, %59]
	Model 2B	1	101	206	87%	[%83, %91]
		2	19	104	50%	[%43, %59]
		3	62	983	75%	[%73, %77]
P3	Model 1A	1	19	234	74%	[%70, %79]
		2	61	623	90%	[%88, %92]
		3	32	329	85%	[%82, %88]
	Model 1B	1	61	623	90%	[%88, %92]
		2	32	329	85%	[%82, %88]
		3	19	234	74%	[%70, %79]
	Model 2A	1	19	234	74%	[%70, %79]
		2	61	623	90%	[%88, %92]
	Model 2B	1	19	234	74%	[%70, %79]
		2	61	623	90%	[%88, %92]
		3	32	329	85%	[%82, %88]

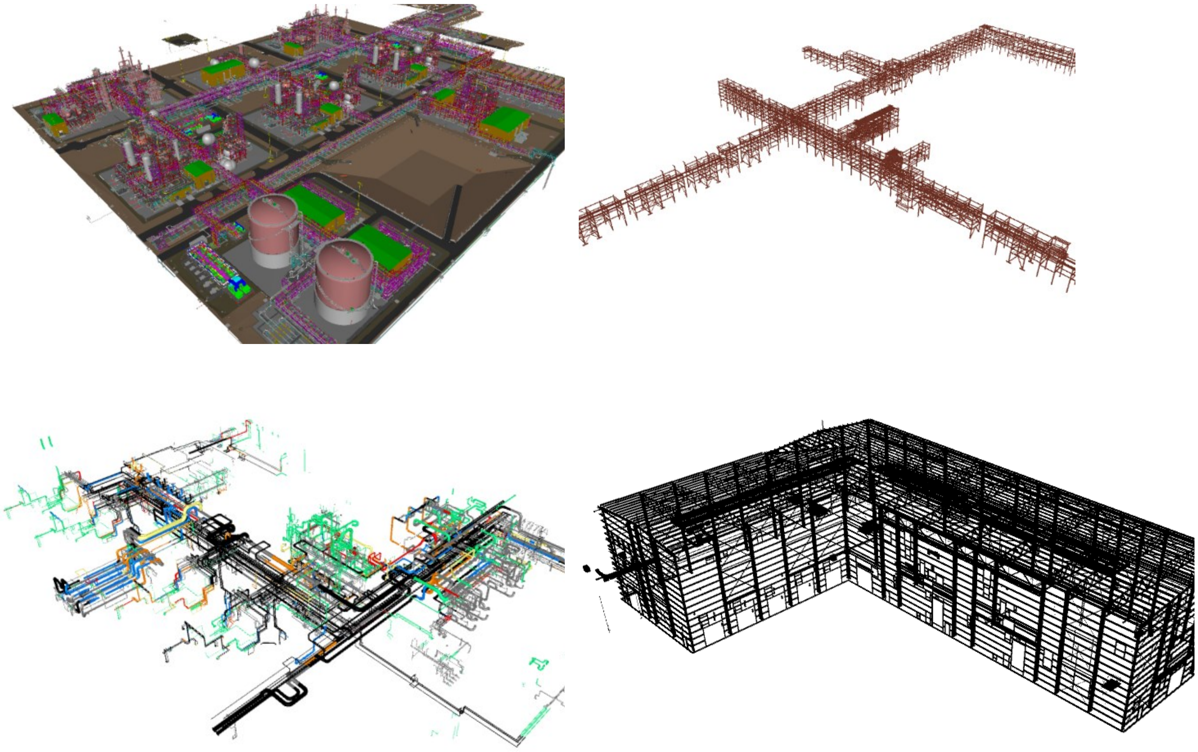


Figure 2-4 Extracted trades for project P3.

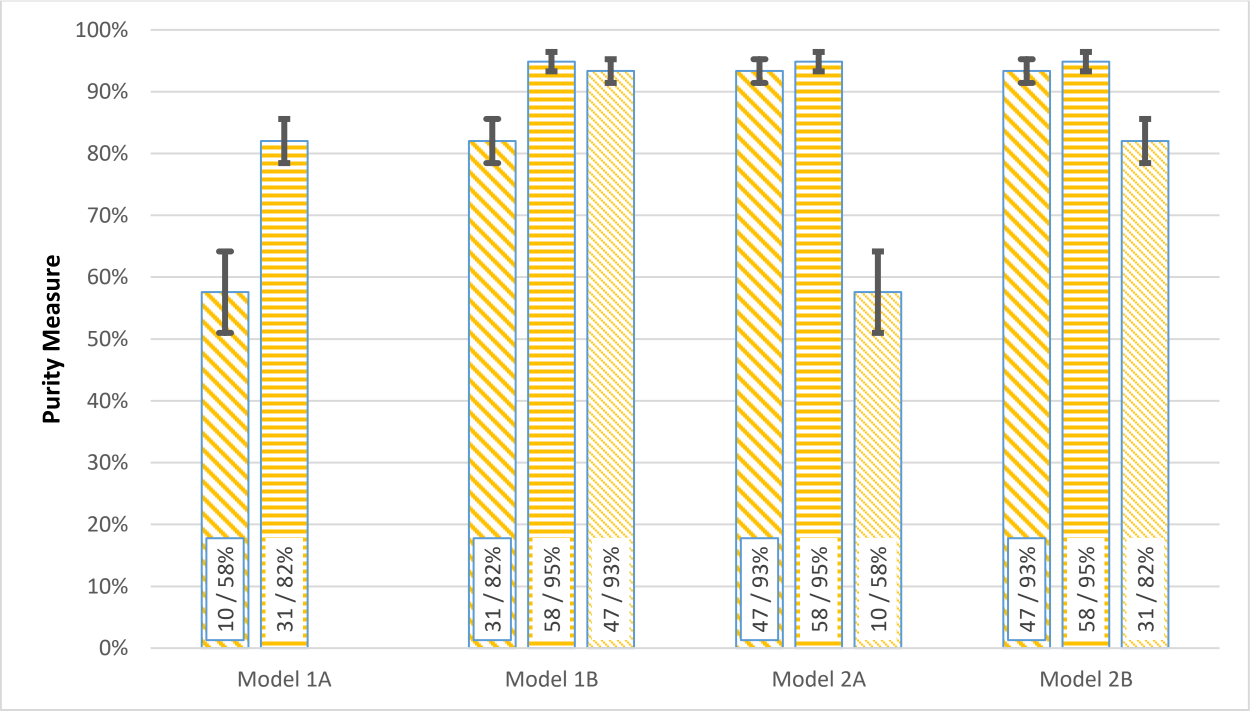


Figure 2-5 Performance of each model for Project P1; numbers in each column represent the number of clusters and purity measures respectively.

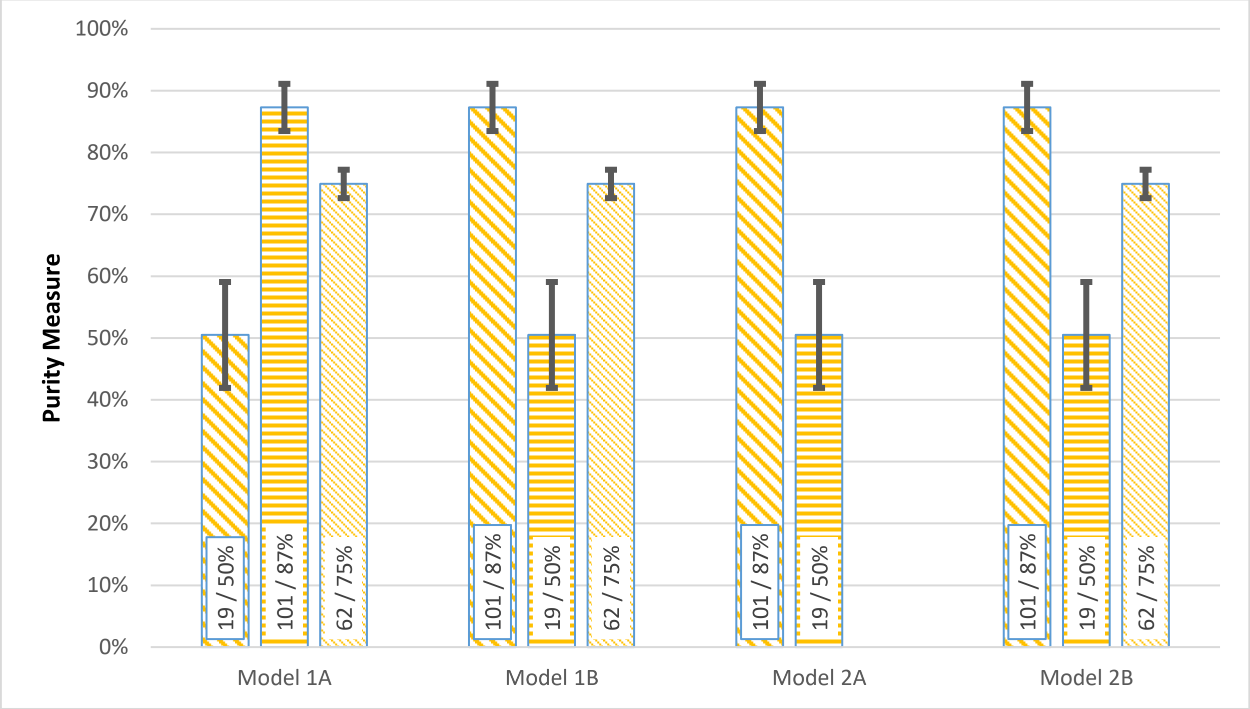


Figure 2-6 Performance of each model for Project P2; numbers in each column represent the number of clusters and purity measures respectively.

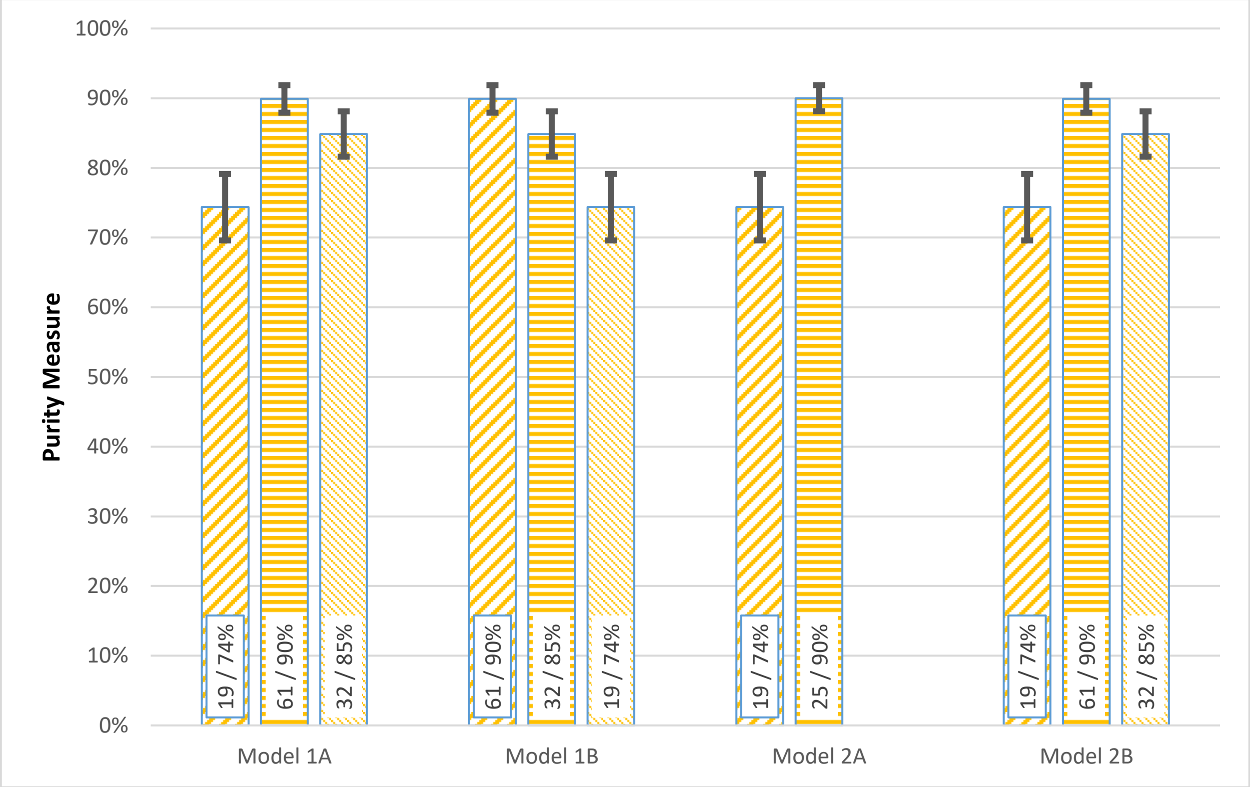


Figure 2-7 Performance of each model for Project P3; numbers in each column represent the number of clusters and purity measures respectively.

2.7 Limitations and Future Work

The proposed method does not guarantee complete purity of clustered items. Test results show an error (around 10%) associated with the generated clusters. This error might be significant in the case of massive BIM models. Therefore, a manual investigation of the results may still be required. However, this investigation is significantly quicker and simpler task as it is done within each cluster in comparison to performing it manually on the whole model as shown in Figure 2-4.

Another limitation is associated with the level of details of the BIM model. The results indicate that all four models performed poorly in Project 2 compared to other projects. Upon closer investigation, we found that Project 2 has very few attributes and most of them either possess the same value for all BIM objects or different for each item. Because the proposed mathematical models rely on descriptive attributes rather than geometry, it requires a certain level of details before it can be used.

The mathematical models work better for Project 1 & 3 because they are more mature compared to Project 2. Model 1B, 2A, and 2B are better than model 1A, but there is no decisive advantage of one of these mathematical models over others. Hence, a user will need to run the three models and judge the results visually based on clusters as shown in Figure 2-4.

Future work for this study may consider semi-supervised learning which utilizes both data points and labels in the clustering process [85]. It has been proven that semi-supervised learning increases the purity value of the generated clusters by labelling them [86]. Our current model can be considered an unsupervised learning as it provides an unlabelled cluster set; however, the future work will include enhancing the purity level by using statistical analysis to find common labels for each cluster. Although the selected labels will not necessarily be an explicit trades' labels, the generated labels can be linked to different concepts in the ontology similar to methodology proposed in [87].

Future studies may also consider integrating the analysis in this study with information related to the geometrical and spatial relations between 3D items to have more accurate classification and/or clustering results.

2.8 Conclusion

The nature of fast-tracked projects in the oil and gas industry leads to the use of incomplete BIM models, especially during early stages of a project. Contractors resort to manual methods to add missing data (such as the trade of each element) to the model. This research proposes a novel method to automatically cluster objects by trades in an unlabeled BIM model. The proposed method requires minimal human input to give a label for each cluster rather than have to check each object individually. The method depends on converting BIM models to an RDF format and utilizes different attribute-weighting models to perform the clustering. Results show that both entropy and TF-IDF can be used to group objects by trade based on their predicates and values with a high purity measure. Testing the method on three real projects with a total of 4.1 million objects shows an average success rate of 91% in terms of cluster purity. Models 1B and 2B (using entropy or TF-IDF with weight less than 1) showed higher accuracy in the tests when compared to models 1A and 2A (unrestricted entropy and TF-IDF).

Chapter 3. Identifying Unlabeled Steel Structure Items in BIM for Industrial Projects Using the Shape Distribution Method[2]

3.1 Introduction

Industrial projects such as oil processing and petrochemical plants utilize Building Information Modelling (BIM) to model and maintain complex designs. A typical industrial project consists of multiple disciplines (e.g., structural, mechanical, and electrical) that are merged into one complex BIM model. This model is essential for planning and coordinating the project execution.

A BIM model usually evolves through contributions from different parties during the project lifecycle [1]. However, the level of each party's contribution varies significantly based on the contract type. For example, compared to the Integrated Project Delivery (IPD) approach, the typical Design-Bid-Build approach limits the contractor's contribution in the early stages [3], [88], [89]. Industrial projects are usually fast tracked with an IPD approach as they generate revenue faster by cutting 50% of construction time [26].

[2] This chapter has been submitted to Automation in Construction Journal

In these fast-tracked industrial projects, the contractor is involved in the early stages by contributing knowledge and experience during the design phase [3]. In addition, it is crucial for the contractor at this time to do a preliminary estimate before the design is complete. According to [3] "*[t]here are many types of estimates that can be developed during the design process. These range from approximate values early in the design to more precise values after the design is complete. Clearly, it is undesirable to wait until the end of the design phase to develop a cost estimate.*"

We observed that industrial domains use specialized BIM platforms such as Navisworks® and SmartPlant®. These platforms are powerful for clash detection and coordination between different trades. Nonetheless, they have been criticized as mere 3D modeling tools rather than true BIM platforms, as will be shown later. Because the industrial BIM models are incomplete and lack many attributes that are needed in early stages, it is not possible to achieve automating quantity take-off. Hence, the contractor has to parse these models manually, which is a tedious, ad-hoc, error-prone, and expensive task.

In this context, using an IFC file format will not overcome the problem. Because the source models lack an explicit indication of each object's type and size, any IFC exporter will use a high-level IFC class such as IfcElement to describe BIM objects which in turn prevents automated quantity take-off.

On the other hand, using enough attributes does not necessarily mean a straightforward automated data extraction from the BIM model. As pointed out by [90], the use of concepts, terms, and definitions is inconsistent and unstandardized, as each party uses its internal naming convention and synonyms to describe the same object.

Previous chapter illustrates a methodology to subdivide an ill-defined model into separate clusters. Each cluster contains a set of objects from the same trade. However, there is no explicit indication of each item shape and size.

In this chapter, we used shape recognition techniques to analyze items' geometry and find their classes and sizes for one of the clustered trade. This will allow obtaining an automatic preliminary estimate of the quantity for ill-defined BIM models.

This chapter is structured as follows: 1) outline the research objectives, 2) provide a brief introduction of shape recognition techniques, 3) compare these techniques and select the most suitable one, 4) describe the utilization of the shape recognition method, and 5) test and optimize the proposed method using real-world examples.

3.2 Objectives

Quantifying material in a BIM model is an essential task, and should be straightforward and automated because BIM supports attributes and Object Oriented Programming (OOP) capabilities. Nonetheless, due to a lack of standard labeling conventions and premature models, contractors rely on manual inspections to estimate quantities, especially in a project's early stages.

One way to automate quantity take-off is using shape recognition techniques. Shape recognition techniques have been used extensively in robotics and point cloud domains to automatically identify shapes. In this research, we apply these techniques to recognize and identify unlabeled items in BIM models.

Identifying BIM objects using shape recognition techniques does not deal with some of the common problems of recognizing real objects such as lighting, shadows, reflections, and view angle, as we are working with a virtual digitized world where each object can be easily isolated. On the other hand, we face different problems including distortions, inaccuracies, different modeling styles and different proprietary BIM formats.

Despite these differences, we believe that most of shape recognition techniques can be applied in our context as both problems have the same inputs (i.e., unidentified objects and known objects to compare with). Therefore, we developed a framework to identify unlabeled BIM items using shape recognition techniques; bearing in mind that we are looking for a quick preliminary estimate using premature models in early stages of projects, we gave weight to algorithm speed over accuracy.

More specifically, the study answers the following questions:

1. **Can existing shape recognition techniques recognize unlabeled BIM items?** These techniques have been applied in many fields such as robot vision; this study applies the techniques in a different field.
2. **Which technique is the most suitable one for this scenario?** As we will show later, all shape techniques follow the same paradigm, but each technique is optimized to a specific context and conditions. This chapter discusses different techniques and suggest the most suitable one for proposed scenario.
3. **Will these techniques be accurate enough for practitioners?** As we focus on performing a rough analysis during early stages of a project, the technique should provide sufficient accuracy for this stage.

4. **Can a practical solution be built using these techniques?** Besides acceptable accuracy, the solution must be fast enough to recognize a large number of BIM items in a relatively short time.

In order to answer these questions, we applied the following methodology:

1. Review most common shape recognition techniques and select the most suitable one in terms of applicability, limitations, and computation time.
2. Apply selected technique on a large sample of different industrial BIM models.
3. Evaluate success rate and computation time of the technique.
4. Modify the selected technique to enhance accuracy and decrease computation time.

3.3 Previous Work and Related Studies

3.3.1 BIM in Industrial Projects

Industrial projects are larger and more complex than building projects and embrace design and information technology more than other types of projects [8]. This can be attributed to the complexity and the fact that many parties are involved in these types of projects. As a result, careful planning and coordination are required.

Industrial projects are executed utilizing fast-track contracts to reach markets faster [91]. In these types of projects, the construction usually starts before the design is finalized, which requires the contractor to consider new designs in the construction plans [41]. In addition, a prefabrication paradigm is usually utilized in these projects by manufacturing modules off-site before shipping them for final installation [41], [91]. These unique characteristics of fast-track projects require careful planning and coordination, which can be achieved using BIM.

Because industrial projects consist of many trades and require careful planning and coordination, they use modelling tools, such as Navisworks® and SmartPlant®, which can merge multiple 3D models. The confusion around the definition of BIM means that there is no consensus about whether these types of software should be considered BIM tools [3].

Some researchers argue that a model containing only 3D objects, or one with no or few object attributes, is not a BIM model [3]. According to this definition, industrial models in early stages of the project, which do not have sufficient attributes, are merely 3D models. Other researchers divided BIM tools to authoring tools which are capable of handling objects' classes and relationships and BIM-related tools such as Navisworks [9]. A third group of researchers consider models with few attributes to be BIM models [10], [11].

In this research, we opted to use the loose definition of BIM to include models that contain 3D objects with few attributes. We observed that these models are common during early stages of fast-tracked projects and we are trying to compensate for missing data by automatically finding the type of steel objects type and their sections by generating points on the BIM object surfaces and using shape recognition techniques.

This technique is similar to obtaining as-built models using laser scanning. According to [92], creating an as-built model using laser scanning technology requires three main steps: 1) Data collection: surveying techniques will obtain a dense cloud of points that accurately measures the physical facility; 2) Data preprocessing: as multiple laser scanners must be used to capture different faces of the facility, the collected points must be registered in a single coordinate system; and 3) Modelling in BIM: using the collected point cloud, different objects should be identified and categorized in the BIM model.

Although many algorithms try to construct 3D objects from point clouds [93]–[97], using point clouds to model different objects in BIM is a manual task that consumes most of the time required to create an as-built model [92], [94].

In addition to creating as-built models, the point cloud has other applications such as detecting a household environment [98], and reverse engineering [99], [100].

There is a similarity between detecting objects from the point cloud process and our case; both try to identify a 3D object (typically a physical object in the point cloud but a digitized virtual object in our case) by collecting points from the objects' surfaces. Then, the data has to be preprocessed by being cleaned, smoothed, and having its outliers removed. Afterwards, a surface model can be generated using a curve-net-based method or by a polygon-based modeling method [101]. Finally, these surfaces can be used to construct the 3D surface object [102]–[105].

We believed that reconstructing the 3D objects was not required for our context as we can directly use some of the shape recognition techniques to find classes and sizes based on the collected set of points. The only limitation to using shape recognition techniques over a point cloud is that these techniques can work only with standardized objects that have predefined shapes and classes (e.g., steel and piping) and will fail in case of arbitrary shapes such as concrete structures.

3.3.2 Shape Recognition

Since the mid-1970s, Computer Aided Design (CAD) has replaced traditional paper drawing [106] as it provides more quality, facilitates editing quickly and accurately, and increases productivity. Since its inception, CAD has revolutionized the design process not only in engineering but in academia [107], [108].

However, CAD systems lack an objects' attributes concept, which limits their ability to share data between different systems [109]. Thus, BIM quickly superseded CAD systems as BIM seem to address CAD's limitations (e.g., attributes). BIM provides a massive information source with search and analysis capabilities [106].

The proliferation of 3D objects required new methods to search and query these objects (3D objects retrieval) as a traditional text search is insufficient [110]. There are many studies regarding 3D object retrievals (see [111], [112] and [113]) that have been used in several applications, including cost estimation in engineering mechanics, which compared current and previous models [111].

The basic idea behind 3D object retrieval is finding the shape signature (also called descriptor and shape representation in some references) and comparing it to a previously stored signatures database. The similarity between two 3D objects is measured by the distance between the two signatures (zero means the two objects are identical). This process of shape recognition (or shape retrieval) imitates brain functionality, in which neurons in the inferior temporal cortex respond to recognized objects [114], [115].

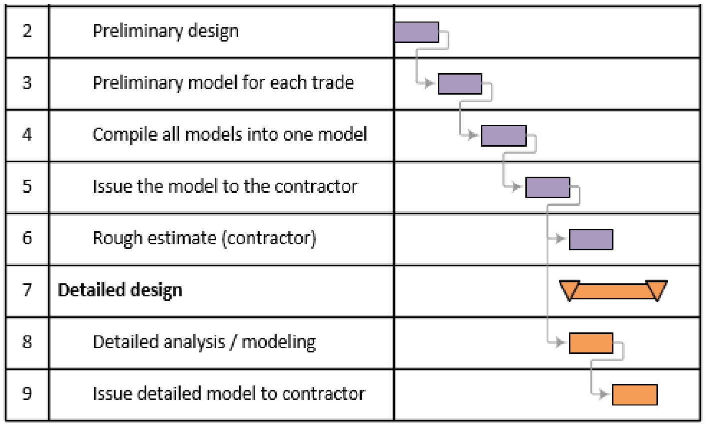
There are six fundamental approaches to calculate shape signature; they vary in their efficiencies and computational power based on the type of 3D object (e.g., 3D solid or 3D surface) and the degree of intricate details in the object. [111] listed the following techniques to calculate shape signature for 3D solids:

1. **Feature extraction methods**: These methods try to extract the most relevant data from the object or picture and use it to create the shape signature. The extracted features should be distortion-free [116]. Nonetheless, there is no consensus on what should be "feature" or "non-feature" [117].
2. **Spatial Function**: These techniques register the 3D object into different surface variations (e.g., spherical and tensor representation). These variations work as the shape signature for matching [118]–[120].
3. **Shape histograms**: By sampling points on the 3D object surface, a histogram can be generated for any characteristics of these points (say, distance between two random points). This histogram works as a shape signature for comparison and retrieval [23].
4. **Section images**: As the name implies, these techniques capture images of the 3D object. The images are then used to analyze and index the object.
5. **Topological graphs**: These techniques abstract the 3D object into a set of nodes and edges that can be indexed and compared to other objects. The comparison for these techniques is usually NP-complete problems that require a robust algorithm to find a solution in suitable time [121]–[123].
6. **Shape statistics**: These techniques depend on intrinsic properties of the 3D object such as volume, circularity, and moment invariants to describe and index the shape. Despite their speed, these methods can be used only for preliminary filtering; after that, more sophisticated techniques are needed [124].

3.4 Proposed Method

We are trying to quantify steel structure quantities in BIM models at early stages of fast-tracked industrial projects. These early-stage BIM models are premature and incomplete and lack a sufficient number of attributes to automate the task. Therefore, we have to rely on model items' geometry to estimate quantities.

An industrial BIM model might contain more than a half-million items that cover all trades involved in the project. As shown in Figure 3-1, an engineering firm performs a preliminary design for each trade based on owner requirements; afterwards it compiles all trades' models into one model and issues the model to the contractor. This model is usually used for collision detection and alignment.

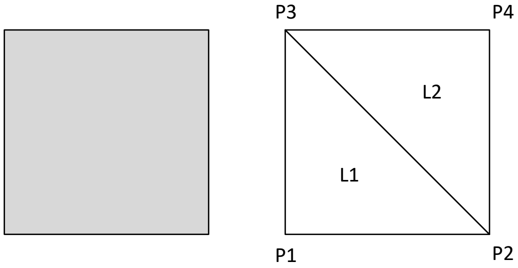


2	Preliminary design
3	Preliminary model for each trade
4	Compile all models into one model
5	Issue the model to the contractor
6	Rough estimate (contractor)
7	**Detailed design**
8	Detailed analysis / modeling
9	Issue detailed model to contractor

Figure 3-1 A preliminary BIM model is issued to the contractor before the detailed design stage; the contractor uses this model to perform a preliminary analysis.

Chapter 2 discussed our work regarding isolating items based on their trade. This chapter illustrates filtering items based on common attributes or geometric features between trade items. This method enables us to retrieve all steel items in a BIM model. However, it cannot find the type and section size of each item.

3D objects in the model are exported as a surface mesh. The mesh is stored into two arrays: the first contains a list of the 3D points' coordinates, while the second contains a list of triangles or quadrants that connect three or four points from the first array as in Figure 3-2. In this figure, a square is meshed using four points and two loops.



$$Points = \{P1, P2, P3, P4\};\ Loops = \begin{vmatrix} L1: P1, P2, P3 \\ L2: P4, P2, P3 \end{vmatrix}$$

Figure 3-2 A solid shape is converted to a set of triangles and exported using two arrays: coordinates, and triangles' vertices.

This simple and limited output format complicates the process of identifying steel objects' sections as we can only retrieve surface mesh instead of the objects' solid. Moreover, when we tried to reconstruct the solid surface using this mesh, we noticed many distortions on the reconstructed solid surface.

Accordingly, for our first attempt, we estimated the solid volume using the volume of tetrahedron technique [125]. This technique calculates the signed volume of the object based on surface triangulations; however, it requires all triangles' normal to face the same direction (i.e., all triangles' points are clockwise or counter clockwise). Unfortunately, the exported set of triangles was not in compliance with this rule; hence, it could not be used.

3.4.1 Algorithm Selection

In our second attempt, we used shape recognition techniques to identify objects; there are many techniques, as discussed previously, with different characteristics. We compared different algorithms to find the most suitable one for our context.

First, we gave more weight to speediness over accuracy. In addition, we needed an orientation-independent algorithm as an item's inclination angle cannot be determined or inferred from the output format.

In summary, we learned that we need a fast shape recognition technique that can identify objects defined by a set of points and/or triangles. The technique should be orientation-independent. Finally, it has to be insensitive to small distortions because such distortions are relatively common in the exported mesh.

We investigated three techniques to determine their feasibility for our scenario: the first technique, attributed graph [122], [126], converted boundary representation (B-Rep) to a graph $G = \{N, E,\ \psi\}$ where N is the set of nodes, E is the edges, and ψ is the relationships. This graph can be compared with other graphs by measuring the distance between corresponding points using the following formula:

$$d(a,b) = \frac{|a-b|}{\max(a,b)} \qquad [126] \qquad \text{Eq. 3-1}$$

Correct points pairing between two graphs is essential for the algorithm's success rate; the following mathematical model was introduced [126]:

If we have two graphs:

$$\begin{aligned} G_0 &= \{N_0,\ E_0, \psi_0\} \\ G_1 &= \{N_1,\ E_1, \psi_1\} \end{aligned} \qquad [126] \qquad \text{Eq. 3-2}$$

And if we have the following nodes:

$$a, b \in N_0, \qquad ab \in E_0$$

$$m, n \in N_1, \qquad mn \in E_1 \qquad \text{[126]} \qquad \text{Eq. 3-3}$$

Then:

$$max: \sum_{a,b}^{A} \sum_{m,n}^{M} \alpha_{ab}^{mn} x_{ab}^{mn} + \sum_{a}^{A} \sum_{m}^{M} \beta_{a}^{m} y_{a}^{m} \qquad \text{[126]} \qquad \text{Eq. 3-4}$$

The second group of techniques that we investigated is shape invariant; invariant descriptors provide a representation that does not change under transformation such as projection. For example, if we have object S, and $s = \mathcal{T}(S)$, where $\mathcal{T}$ is a transformation function, then the invariant descriptor $I(S) = I(s)$ [127]. Invariant descriptors might be geometrical like distance ratios, algebraic-like eigenvalues, or differential functions [124].

The third group of techniques that we investigated is shape histogram techniques, which try to abstract a 3D object into a histogram based on specific criteria. Then they use the distance between histograms as a dissimilarity measure. One of the common shape histograms techniques is the shape distribution algorithm [128]. Shape distribution generates a histogram by randomly selecting points on the solid surface and constructs the histogram using any spatial function (e.g., the distance between two random points).

Based on the available data and literature investigation, we believe that the shape distribution algorithm is a good candidate for our case. Table 3-1 shows a comparison between the three algorithms; all algorithms are orientation-independent. Orientation-dependency is effective in this context to correctly identify inclined steel members. Low sensitivity to small changes is also required to ignore distortion from modelling inaccuracy and any bolt-holes in the steel members. A short computation and comparison time is crucial for huge BIM models which are common in industrial projects.

Table 3-1 Comparison between different algorithm applicability in our context.

	Time (computation & comparison)	**Orientation**	**Sensitivity to small changes**	**Notes**
Attributed graph [122], [126]	NP-complete or $O(N_oN_j \times L_oL_g)$ where N & L are number of nodes and edges respectively	Independent with high computation cost	Low	Similar models with different sizes might have similar graphs. Number of nodes and edges of shapes must be very close to work correctly.
Shape invariants [124]	NP-complete [129]	Independent	Very sensitive	Sensitive to boundary errors and requires effective algorithm to find acceptable solution in reasonable time.
Shape distribution [23]	$O(N^3B) + O(B^2)$ [112]	Independent	Low	The most suitable one for our case as shown in section 3.4.2.

3.4.2 Shape Distribution Algorithm

The shape distribution algorithm is usually used as a preliminary classifier prior to introducing a more accurate algorithm [23]. However, we found it the most suitable algorithm for our case for the following reasons:

1. It works on points instead of the solid surface.

2. It is orientation-independent; it detects sections regardless of their orientations, and orientations are not possible to determine in our case.
3. Because of its stochastic nature, this algorithm is insensitive to small changes and distortions on the solid surface. This means it overcomes distortions we noticed in the exported sections.
4. This technique is fast, albeit less accurate, which matches our requirements as we only need a preliminary estimate for the early stages of a project.

Shape recognition algorithms encompass two essential parts: shape function, and dissimilarity measure. These are implemented in different ways. Shape function abstracts the unrecognized shape to a mathematical representation that can be manipulated and compared with other models. Selection of the shape function usually depends on the domain of the problem and the format of the unrecognized shapes. Shape function should also sufficiently describe the unrecognized shape [130] in order to fully differentiate between different shapes. Sampling points from contours [131], Fourier descriptors [132], and neural networks [133] are examples of shape functions that have been used in shape recognition.

The shape distribution algorithm uses distance distribution as a shape function. By measuring the distance or angle between random points, a unique histogram for each object can be constructed. Figure 3-3 shows a sample of the generated histogram for different steel sections. It shows that sections with different shapes and/or sizes will have a unique Probability Density Function (PDF).

There are four variations of this shape function:

1. The distance between random points and a fixed point (**D1**).

2. The distance between two random points (**D2**).
3. The square root of the area of a three-random-points triangle (**D3**).
4. The cube root of the volume of a four-random-points tetrahedron (**D4**).
5. The angle between three points (**A3**).

After abstracting shapes into histograms, mathematical models can be constructed using piecewise functions. Consequently, dissimilarity measure methods are used to compare an unidentified object and a set of reference objects. Then, based on a predefined threshold, a match can be found.

A selection of dissimilarity measure depends on the shape function format (i.e., continuous vs. discrete); if we have two histograms $f(x) = x_1 + x_2 + \cdots + x_m$ and $g(y) = y_1 + y_2 + \cdots + y_m$, then the distance can be calculated using one of the following:

- Minkowski L_N norms

$$L_p(x,y) = \left[\sum_{i=0}^{N} |x_i - y_i|^p\right]^{1/p} \qquad \textit{Eq. 3-5}$$

- Kolmogorov-Smirnov distance

$$D_{mn} = \sqrt{\left(\frac{mn}{m+n}\right)} \sup_x |f_m(x) - g_n(x)| \qquad \textit{Eq. 3-6}$$

- Bhattacharyya distance

$$D(f,g) = 1 - \int \sqrt{fg} \qquad \textit{Eq. 3-7}$$

- Earth movers' distance

$$EMD = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} c_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}} \qquad [134] \qquad \textit{Eq. 3-8}$$

In short, the fundamental idea behind this algorithm is to convert a 3D object into a distribution that can be compared to another generated distribution using a dissimilarity measure as in Figure 3-4. The histogram is constructed using one of the aforementioned methods.

Shape function **D2** was chosen in our model because of its robustness and accuracy [111]. The accuracy of the results will depend on the number of distances and the distribution of points on the 3D object surface; however, increasing the number of distances will also increase the computation time as will be shown later, in sections 3.4.3 and 3.5.1.

The number of generated points will determine how many times the point will be repeatedly used in measuring distances. The following combination formula can be used to obtain the minimum number of points required to generate distances without having to use any point two times.

$$number\ of\ unique\ distances = \binom{number\ of\ points}{2} \qquad \textit{Eq. 3-9}$$

Although the algorithm does not require unique distances, we used the aforementioned formula to make sure we have enough points to generate unique distances to avoid local matching. This equation will not ensure that each point is used one time only, but with a large number of points and a robust randomness algorithm, the replication occurrence can be minimized. The code can be found in Appendix C.

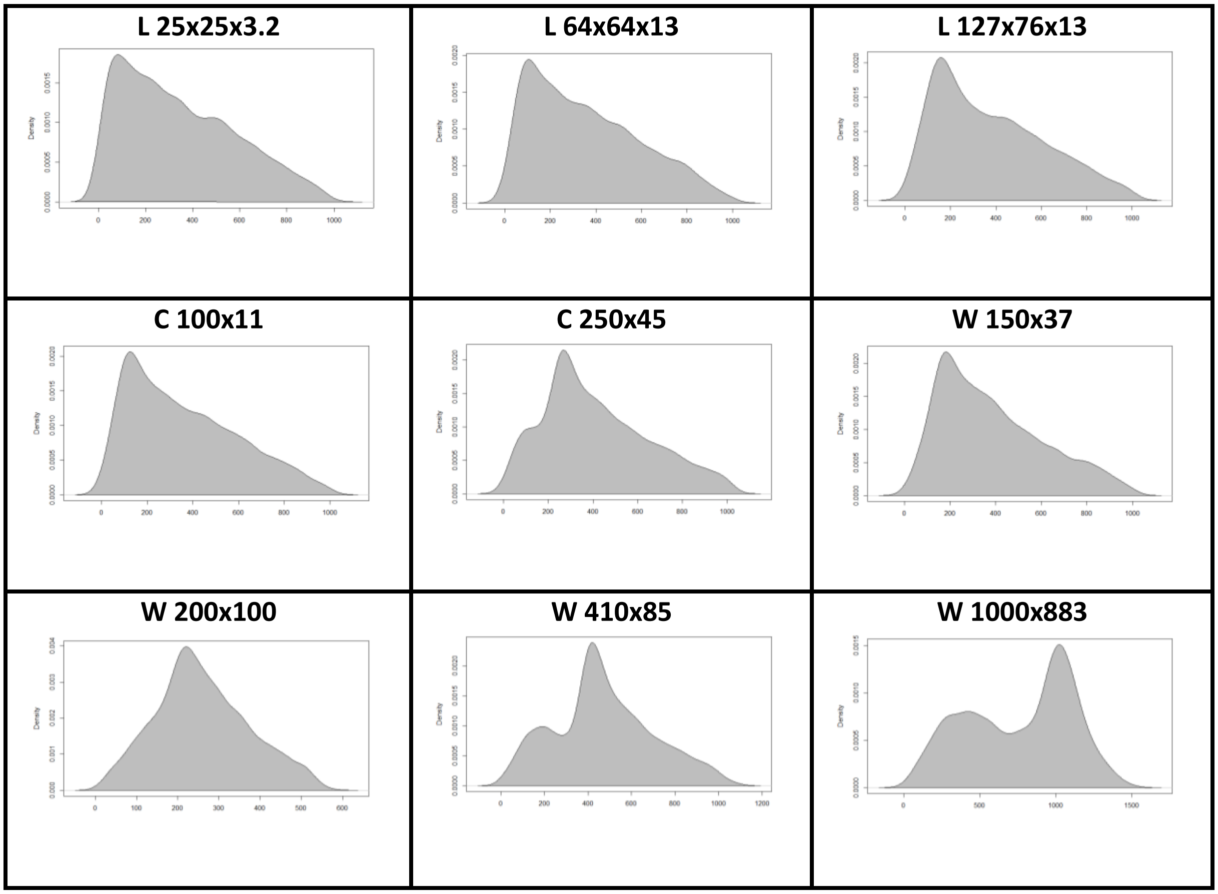


Figure 3-3 Histograms for different steel sections generated by measuring 5000 random distances on the objects' surface for 2D sections.

3.4.3 Computation Time

In computer science, algorithms' computation time is usually measured using "Big O notation" [135], which estimates change in the computation time based on the input size. For example, if Big-O for an algorithm is *O*(1), it means the computation time is constant regardless of the input size; on the other hand, O(n!) or O(2^n) algorithms' computation time will increase exponentially based on the input size.

Because a typical industrial BIM model may have over a half-million items, and each item consists of a significant number of points, Big O notation is a pivotal factor in determining the algorithm applicability in our case. According to [112], [136], a shape distribution algorithm will take $O(N^3B)$ for computing and constructing histograms and $O(B^2)$ for comparison with a standard set of histograms, where N is the number of voxels along each axis and B is the number of bins; and because the number of bins is determined based on the number of points, we can see how increasing the number of points affects the processing time. Section 3.5.1 will demonstrate how the number of points exponentially increases the computation time; nonetheless, we modified the algorithm to decrease the computation time while preserving the same margin of error by projecting objects into a 2D plan, as will be explained in the next section.

3.4.4 Algorithm

The first step in applying the algorithm was preparing the reference sections that will be used for comparison with the unidentified sections. In our case, we used sections in the Canadian Institute of Steel Construction (CISC) handbook [24]; however, any other standardized section, such as pipes, can be used.

Each steel section in the Canadian standard has been mapped to a unique histogram, Figure 3-3 shows a sample of the histograms generated for the reference sections. These histograms have been compiled into a SQLite database with a table for each steel section type. This database works as a reference for comparison with any unidentified steel section.

The second part of the algorithm is related to exporting the unidentified sections from the BIM software as a set of points, generating the histogram for each section – this step is independent of the BIM software, and comparing it with the reference database to find the correct section using a dissimilarity measure as shown in Figure 3-4 and List 3-1.

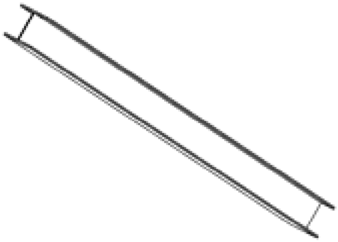
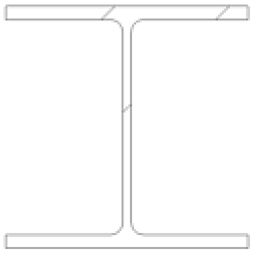
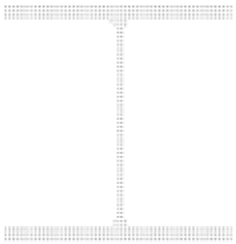
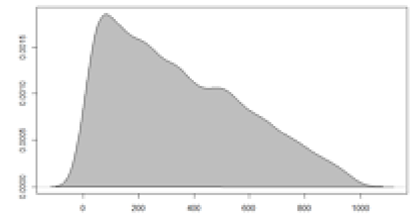
1	**Unidentified section**	**2**	**2D Projection**
			
3	**Section Points**	**4**	**Shape Distribution**
			
5	**Comparison**	**6**	**Identified Section**
	$L_p(x,y) = \left[\sum_{i=0}^{N} \lvert x_i - y_i \rvert^p\right]^{1/p}$		*UC305x305x97*

Figure 3-4 A schematic illustration for the main steps of the shape distribution algorithm.

List 3-1 A pseudo code for identifying unlabeled BIM items.

```
Load reference sections' distribution
For each unidentified section:
        Compute the section shape distribution
        Compute section area A
        Select reference sections where area = A ± 20%
        Calculate the dissimilarity measure
        Select the reference section with the minimum dissimilarity measure
```

Initial implementation of the original algorithm was relatively slow with a high error margin; therefore, we introduced some modifications to the algorithm to be more applicable in our context and increase accuracy. These modifications encompass two parts as follows:

1. **2D instead of 3D**: our first model deals with the 3D section of the unidentified section, which requires equating the length for the unidentified and reference sections before performing the comparison. In addition, working with the 3D section will require more points than working with the 2D section, which in turn increases computation time significantly. We decided to project steel items into a plan and work with the 2D section. There are two reasons for this: 1) by considering only a plane section we minimized the distortion effect found in some sections and, more importantly, 2) we significantly decreased computation time while maintaining the same accuracy due to the prismatic nature of the steel sections. This pays off in time required to extract the 2D section from the 3D one. Based on experiments, we found that using a 2D scenario outperformed the 3D one as it achieved the same accuracy with a smaller number of points. For that reason, we used the 2D scenario for all prismatic sections and resorted to the 3D scenario only for the non-prismatic section.
2. **Limiting the scope:** Because of its random nature, shape distribution was not considered an accurate algorithm and was mainly used for pre-classification. Therefore, we tried to enhance the accuracy by limiting the comparison set. Instead of comparing the unidentified item with all the sections in the standards, we first calculated the item’s area approximately. Then we selected only the standard sections within an allowable range. Consequently, instead of making comparisons with thousands of sections, we ended up comparing only around 30 sections; this enhanced

the accuracy while decreasing comparing time. Another enhancement was limiting the reference set based on the project, as designers usually use a limited set of sections per project to reduce waste.

3.4.4.1 Dissimilarity Measure

After constructing the histogram for the unidentified item, we calculated the dissimilarity between its histogram and the reference set histograms and chose the section with the smallest value. The dissimilarity between histograms can be calculated using the L_p (Minkowski) distance (Eq. 3-5) [112] by aggregating the distance between each two corresponding points. This equation requires using the same number of bins for the two histograms as shown in Figure 3-5. More details about measuring distance and the dissimilarity between histograms can be found in [137].

Changing values of p did not significantly impact the accuracy of the output, but it did slightly increase the computation time. Hence, we arbitrary selected the Euclidean distance (p=2). After calculating the dissimilarity measure, the reference sections were ordered based on their distance from the unidentified section, with the smallest on the top. We decided to show the top four sections to the user, and he/she could automatically select the top section, or reorder them by introducing preference weight based on the project data. For example, the user can exclude a section that is not used in the project or add more weight for a frequently used section.

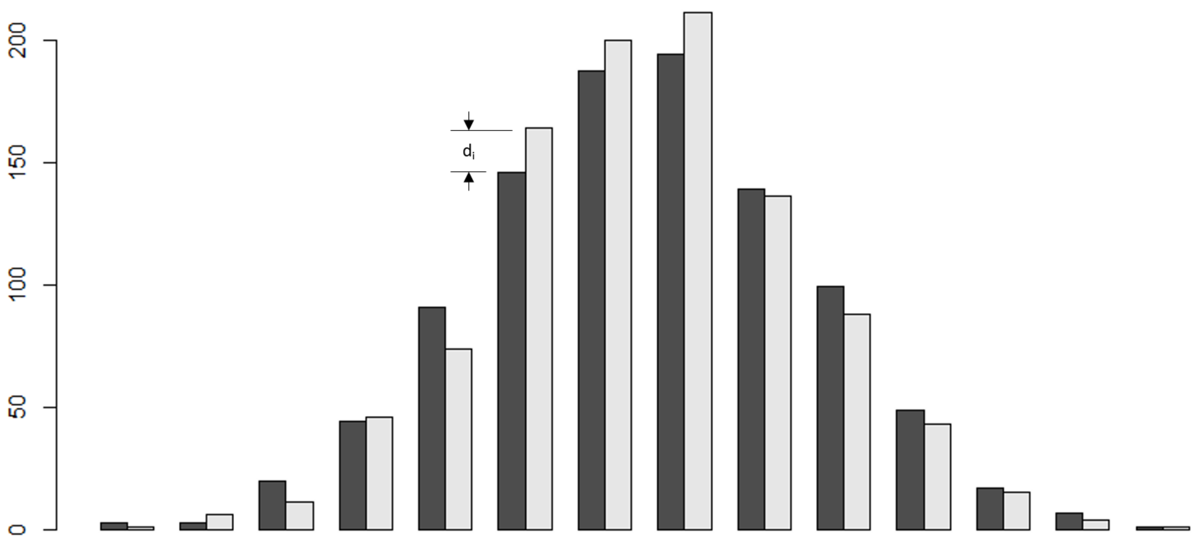
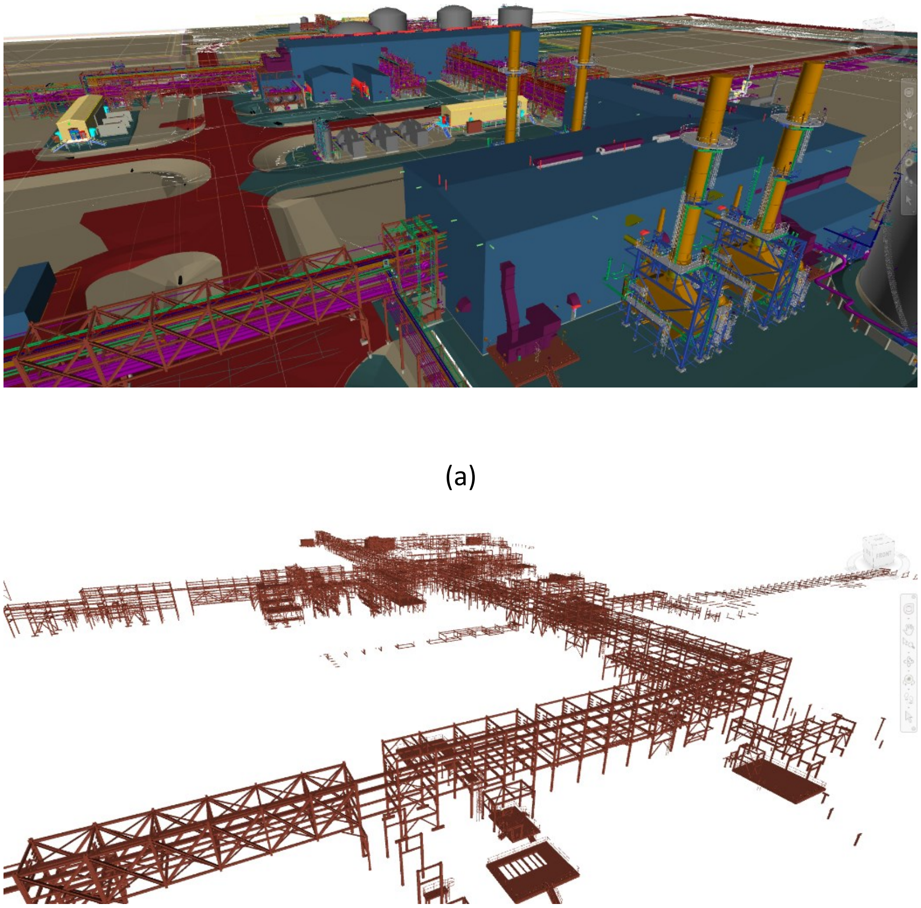


Figure 3-5 Dissimilarity measure represents the sum of the difference d_i between each two corresponding columns

3.5 Method Testing and Evaluation

This section contains a summary of our analysis of the algorithm performance using three real BIM models obtained from a major contractor in Canada. These models have been created by three different engineering firms around the world. Consequently, each model has different labels and color-coding. The contractor performed a manual preliminary estimate for these projects during the design stage.

The number of items in each model varies from a half-million to three-quarters of a million items; this includes items from different trades such as mechanical, electrical, and structural. Hence, we used the grouping technique [138] to isolate structural steel items. We found that each model contains around 60,000 unlabeled structural steel items. A visual inspection showed that steel items included angles for trusses and bracing, and H sections for columns and beams, beside hollow, square, and circular sections. We also noticed that length and orientation changed significantly from one item to the next. In addition, by measuring cross-section dimensions for a randomly selected sample, we found that the sections had been loosely drawn with ±2 cm error. Figure 3-6a shows a sample full model for one of the projects and Figure 3-6b shows steel items only in the model.



(a)

(b)

Figure 3-6 Sample model of all items in one project (a) and of steel items only (b).

Figure 3-7 shows a flowchart for the validation procedures followed in this study. We drew a random sample of sections and identified them manually and then ran the algorithm independently using the same sections and compared the results. Bearing in mind that the algorithm provides the closest four sections to the unidentified section and the user can give different weights, we decided to use a more conservative approach and only take the top section. Moreover, although we could increase the accuracy by reducing the reference set based on the project data as mentioned in previous sections, we decided to use a reference database that contains all Canadian standard steel sections, which gives the highest possible error.

Because each model contains around 60,000 items, we need to draw a sample from this population. The sample should be big enough to truly represent the population, yet small enough to process feasibly. The sample size determines the confidence level and margin of error according to Eq. 2-14.

Therefore, for a confidence level of 95% (Z=1.96) and a confidence interval of 10%, we needed a sample size of 96 items. We drew the sample from the three models (equal number from each model) using a simple random method in which all items had an equal chance to be drawn. The sample contained steel sections with different shapes and sizes.

After drawing the sample, we manually identified each item by measuring its dimension, and looking it up in the standard sections' tables. Then we identified the sample by the algorithm several times using different numbers of random distances. For each scenario, we recorded the success rate and the average computation time.

In order to calculate the average computation time, we calculated the total time required to identify the total sample, then divided that by the number of sections in the sample; this was more accurate than calculating the time required to identify one section because of the fixed time required to load the reference database into the memory.

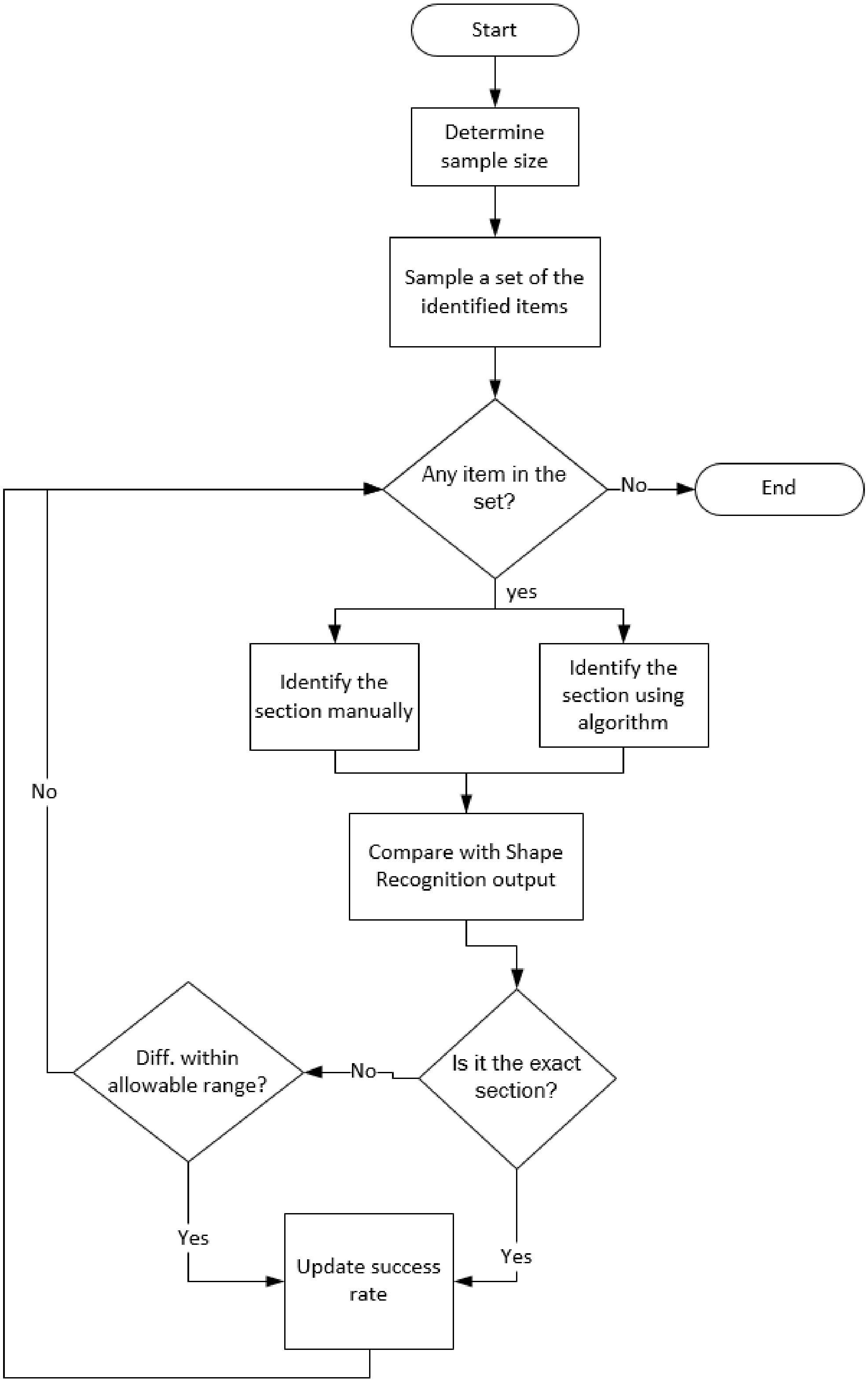

Start
Determine sample size
Sample a set of the identified items
Any item in the set?
No
End
yes
Identify the section manually
Identify the section using algorithm
No
Compare with Shape Recognition output
Diff. within allowable range?
No
Is it the exact section?
Yes
Update success rate
Yes

Figure 3-7 Validation flowchart.

The standard steel sections usually have multiple variations of one section with very small changes in the geometry (around ± 2 mm); for example, Table 3-2 presents an excerpt from the Canadian standard sections [24] that shows three sections with very close dimensions. This difference cannot be identified by the algorithm (a side effect of its noise insensitivity), nor it affects the early-stage estimate. We consider a case a success if the identified section is the same class as the actual section with a weight difference within 20%. Practitioners in the partner company have suggested this percentage based on the required accuracy for early stages of projects. We should emphasize here that the identified section is rejected if it is a different class (e.g., W section vs C section), even if it is within the allowable weight range.

Table 3-2 A sample of three W sections in the Canadian standard that share similar dimensions.

Designation	Dead load kN/m	Depth mm	Flange width mm	Flange thickness mm	Web thickness mm
W310x74	0.726	310	205	16.3	9.4
W310x67	0.651	306	204	14.6	8.5
W310x60	0.580	303	203	13.1	7.5

3.5.1 Results

The accuracy of the algorithm is dependent on the number of measured distances. We present five scenarios for the following numbers of distances: 1,000,000; 500,000; 50,000; 1,000; and 100. Table 3-3 summarizes the results for the five scenarios; it shows that the computation time spikes from 0.04 seconds for a 100-distance scenario to 15 seconds per item for 1,000,000--distances scenario. Nonetheless, the success rate does not increase at the same rate.

We found that the success rate increases sharply from the100-distance to the 50,000-distance scenario; after this, the success rate is almost constant regardless of the number of distances.

This was expected because of the random nature of the algorithm and the closeness in dimensions between standard steel sections. Figure 3-8 shows that a success rate between 80% and 90% can be achieved if more than 50,000 distances have been used.

This error can be minimized by reducing the number of sections in the reference database; this can be easily achieved in most of the projects as designers usually use a small set of sections per project to eliminate waste and streamline the construction phase. Moreover, because the algorithm orders all reference sections based on their distance from the unidentified section, different weights can be introduced to prioritize sections.

In order to quantify the error for the base case (i.e., using all reference sections and without introducing weights), the average difference in total weight between the actual sections and identified sections for the sample is 8.8%, which is within the acceptable tolerance in the industry (±10%) [139]. Based on this analysis, we suggest using the 50,000-distance scenario because it gives acceptable results in a manageable computation time.

Table 3-3 Summary of the success rate for different number of measured distances along the average computation time.

	Avg. Time per section (seconds)	Success Rate
1,000,000 distances	15	91%
500,000 distances	9.8	84%
50,000 distances	1	82%
1,000 distances	0.05	49%
100 distances	0.04	24%

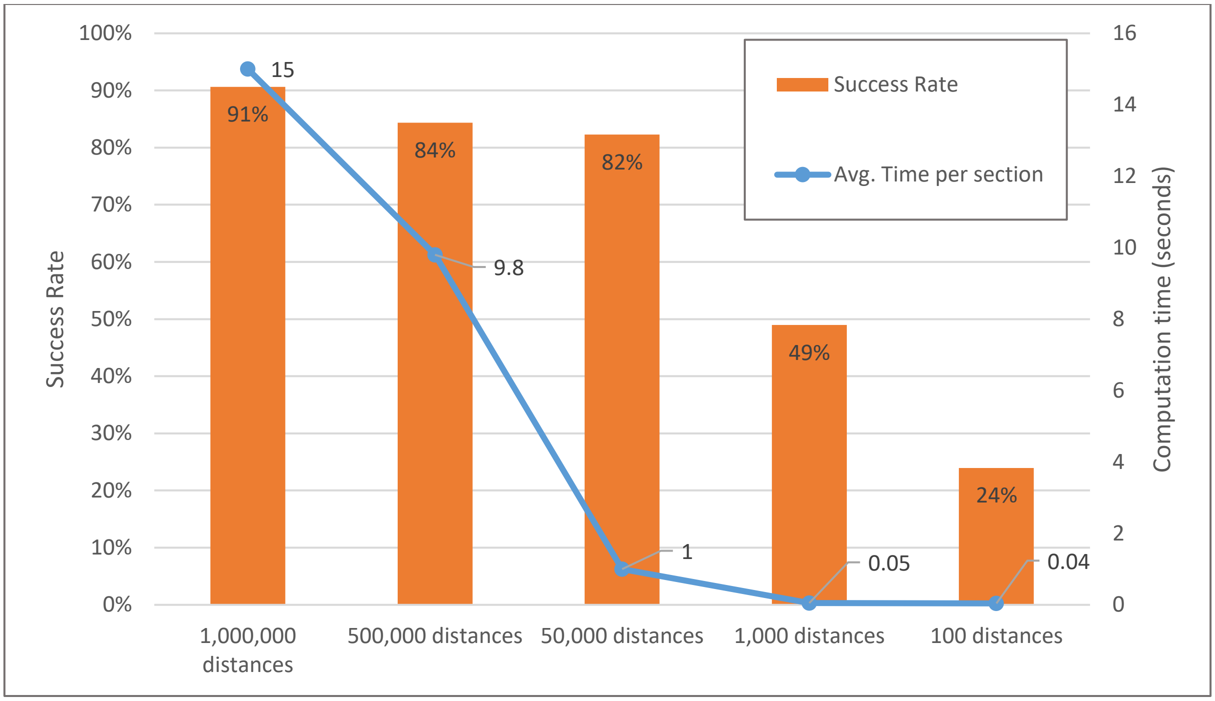


Figure 3-8 Success rate for the five scenarios.

3.6 Limitations and Future Work

Although this research manages to provide an alternative to the manual process with acceptable accuracy during early stages of the projects, there are limitations that can be summarized as follows:

1. The proposed technique will fail to detect arbitrary shapes as it only compares unidentified objects to a reference set of objects.
2. A large number of distances is required to achieve acceptable results, which in turn increases the computation time.
3. The algorithm is not inclusive and it is usually used as a pre-classifier algorithm; therefore, it is limited to preliminary estimates and cannot be used, for example, to formulate the bill of materials.

This research can be extended in two different directions. One direction would be to apply the same methodology to other trades, especially piping. Piping is more challenging than steel structures, as it has more classes and groups, and not all classes are prismatic (e.g., valves, elbows, and tee). Applying the methodology to piping would provide an interesting opportunity to compare how the algorithm performs with different trades.

Another direction for this research is using more sophisticated shape recognition techniques and measuring the difference in accuracy and processing time. Additionally, more enhancement can be achieved by considering the existing attributes, if any, along with the 3D geometry in the comparison. This can be achieved by creating an ontology that represents the solid with its attributes. That ontology could then be used in the comparison [36], [140].

3.7 Conclusion

Estimating quantities from BIM models in industrial fast-tracked projects can be a tedious process for contractors at early project stages. We propose a method to automate the task of visually inspecting items and labeling them, which is usually done manually by a coordinator. Our proposed method uses a shape distribution algorithm to compare the unidentified item's histogram to a reference set to find the closest section. We enhanced the accuracy of the algorithm by projecting prismatic sections into 2D plans. This method has been validated using data from BIM models for three major projects with over a half-million 3D items in each model. Results show that the approximate estimate with a ± 8.8% difference in weight at a 95% confidence level can be achieved using around 50,000 random distances. The proposed approach can reduce the amount of effort required to complete this task manually from a couple of weeks to only a few hours.

Chapter 4. A Framework for Visualizing Heterogeneous Construction Data Using Semantic Web Standards

4.1 Introduction

A typical visualization paradigm in construction utilizes Building Information Modeling (BIM) data along with the project schedule to depict the construction progress. This approach, known as 4D, focuses on the high-level details only, in which each element simply appears in its final position when the corresponding task in the schedule is complete. This type of visualization provides an appealing interface to illustrate the sequence of activities in the schedule and helps align objectives between different stakeholders [141]. Although 4D visualization helps users to understand the schedule and identify potential problems [142], it fails to give more details about site conditions (e.g., congestions) and the interaction between personnel and equipment on site [143].

A more detailed approach is operational visualization, which focuses on intricate details rather than the big picture. This type of visualization depicts the material movement, interaction with cranes, etc. [144]. A successful implementation of operational visualization requires more data than the typical visualization approach. Unfortunately, these data come from different parties in heterogeneous formats and merging these data is usually performed on an ad-hoc basis per specific case study.

The construction industry is characterized by large volumes of data that come from heterogeneous data sources [145]. Heesom and Mahdjoubi [146] have stated that the flow of data is one of the most critical issues in the development of visualization tools. They argued that most visualization applications require manual input from different data sources, which is a potential reason that it is not widely used in the construction industry.

In this chapter, we provide a framework for generating visualizations of different construction activities with different levels of detail and minimal human input. Our approach is data-centric and focuses on merging and processing data rather than the visualization application. The data source will be the refined BIM models processed in Chapter 2 and Chapter 3 along any additional operational data used in a project life cycle.

In order to focus on merging and processing data, we first developed an ontology [19] that conceptualizes information related to the visualization process. This ontology formulates and automates data flow from different data sources that need to be visualized.

Afterwards, we used SPARQL queries [34] to retrieve and manipulate the stored data, and automatically generate input files for the visualization application. Separating the data from the visualization application allowed us to change the visualization application without having to edit the ontology or the input data. Additionally, it made it possible to add new data sources to an existing visualization process without breaking compatibility with the visualization application, as will be shown later.

The remainder of the chapter is structured as follows: First we outline the research objectives, then we discuss our previous experience with visualization, and finally discuss in detail our proposed framework and test it with real case scenarios that cover different applications and different ranges of activities commonly used in the construction domain.

4.2 Objectives

Both high-level and operational visualization require mapping data from different sources (e.g., 3D models, CPM schedules, simulation data) to a format that the visualizer can comprehend. This mapping is usually done on a case-by-case basis, which limits the usability of the process in different contexts. Moreover, introducing an additional data source to an existing visualization process requires extensive work, as this additional data source has to be merged.

In this chapter, we are trying to design, develop, and test an automated framework for generating visualizations at different levels of time resolutions from heterogeneous construction data. The following steps will be followed:

- Evaluate previous work related to construction visualization. This also includes the authors' previous experience.
- List the capabilities required for the framework and how they can be achieved.
- Propose the framework (Figure 4-1 which will be discussed in details in section 4.5).
- Develop the framework components.
- Test the framework with real-case scenarios.

To achieve these objectives, the following questions will also need to be addressed:

- What are the main concepts to be included in the ontology? Because we are trying to develop a general ontology that can be utilized with different visualization applications, the key concepts and taxonomy in visualization—such as time, position, orientation—must be carefully investigated.

- Will the semantic web be able to merge different data sources and reformat them for visualization? The construction domain uses different applications, including schedule engines, relational databases, BIM, spreadsheets and, to lesser extent, simulation engines. We attempt to develop specialized connectors for some common applications that can retrieve the data and convert it to the proposed ontology schema.
- Can the stored Resource Description Framework (RDF) data be formatted to the visualizer format? For now, the stream of data coming from different sources has been converted to the RDF format. Another connector is required to process this data and feed it to the visualizer to be shown.
- Can the RDF format and its query engine enhance the visualization process by providing a way to query and display objects based on the required level of detail? Different audiences require different visualization levels of detail. For example, a project owner or an engineering firm might be interested in 4D or 5D visualization, while a contractor might be more interested in a more detailed visualization that shows operational activities such as crane movement and scaffold erection. These different scenarios require different time resolutions and objects. Our ontology should be able to store and show all these scenarios using a querying engine.

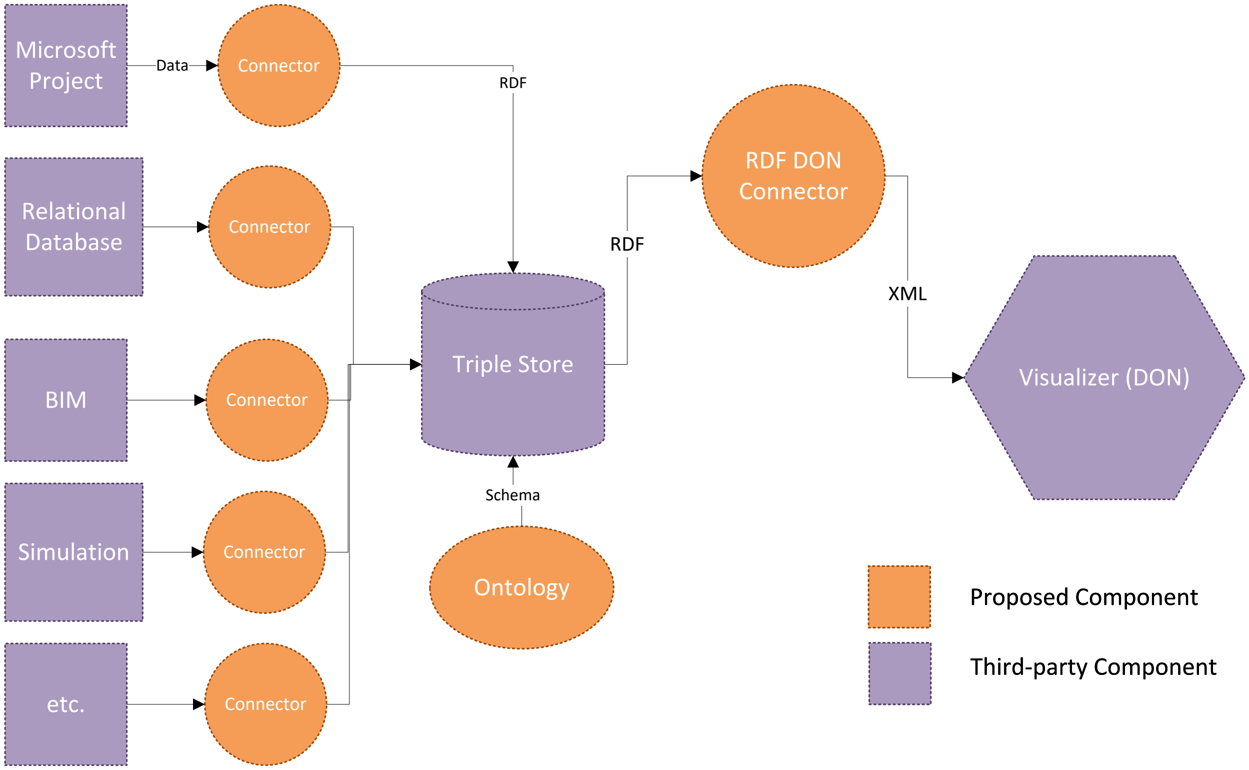


Figure 4-1 The proposed framework for visualizing heterogeneous data.

4.3 Literature Review

4.3.1 Visualization in Construction

3D visualization of construction projects is challenging due to its complexity and unpredictable nature [147], which leads to many customized applications that are applicable only to a certain type of construction project or even to a specific project.

Visualization plays a critical role in many construction domain applications, such as simulation; it has been stated that a typical construction simulation consists of eight smaller components – known as federates –[148]. An essential federate of these eight federates is a visualization federate, which shows the simulation progress and results to the end user. There are many applications for using visualization in simulation models. These applications include tunneling [149], training [150], [151], and crane operations [152], [153].

Outside the simulation realm, visualization applications can be found in virtual reality [154], [155], safety [156], [157], and transportation [158], [159]. The variety of applications shows how important visualization is in the construction domain. However, we argue that most of the visualization applications demonstrated earlier focus on a specific case study which limits the visualization applications' usage in different contexts. Therefore, here we will not focus on a specific application and instead try to provide a visualization framework that can be used with different applications and case studies.

4.3.2 ifcOWL

BIM is prevalent in the Architecture, Engineering, and Construction (AEC) industry; it works as a data store for attributes along geometries. Merging geometries and attributes provides industry with many benefits such as the ability to visualize the model and coordinate different trades involved in the project [3].

Different software vendors use an internal closed-source format to store BIM models, which hinders the exchange of information between different BIM applications. Consequently, Industry Foundation Classes (IFC) has been initiated to create an intermediate open BIM format and all software vendors are expected to implement importing and exporting this format. IFC is maintained by buildingSMART (formerly known as IAI) and the current version is IFC4 [160]. IFC became the de facto standard of the BIM industry [161] and is supported by most BIM applications.

An IFC data file is modelled in EXPRESS data specification language [162]. The International Organisation for Standardisation (ISO) defined EXPRESS language as follows: “EXPRESS is a data specification language as defined in ISO 10303-1. It consists of language elements that allow an unambiguous data definition and specification of constraints on the data defined.” [163].

EXPRESS language excels in defining detailed data types, relations between elements, restrictions, and lists [164]. However, it has been criticized for lacking semantic data interpretation [165], [166].

Researchers have relied on the semantic web to add semantic layers for data represented in the EXPRESS language format. The semantic web can be used to convert heterogeneous data sources to the IFC format [61]. Alternatively, work has been done to express IFC data using ontologies. For example, an ontology has been developed to improve EXPRESS files by semantically capturing geometrical constraints in CAD systems [167]. Similarly, a mapping between IFC EXPRESS schema and ontologies has been suggested [168]. This work has led to the development of an ontology using Web Ontology Language (OWL) for IFC known as ifcOWL [58], [161], [162].

Additional ontologies such as ifcWOD [169] and SimpleBIM [170] have tried to improve ifcOWL by making it less verbose and focusing on semantic information rather than the rigid conversion from EXPRESS language. We work with ifcOWL, as it has been endorsed by W3C[3] and buildingSmart[4] [164].

[3] https://www.w3.org/community/lbd/

[4] http://www.buildingsmart-tech.org/future/linked-data

We decided to build on ifcOWL instead of developing a new ontology to support interoperability and integrate more with other ontologies which is encouraged in the semantic web world [19], [161]. In our ontology, we created custom properties and specified their ranges and domains from the imported ifcOWL ontology[5], as will be shown later.

4.3.3 Modelling Time in RDF

RDF captures knowledge through triples (subject, predicate, and object); this format is the basis for reasoning engines. However, there are some challenges to model additional information about triples. These challenges include:

- **The source of the statement**, such as an historian claimed the great pyramid was built in 2570 BC.
- **A ternary relationship**, such as the relationship between a donor, a receiver, and an origin.
- **The severity of a relationship**, such as Max will perform a surgery with an 80% success rate.
- **Dates and locations**, such as the soccer game between France and Germany is next Friday in the “Stade de France,”

The last example encompasses location “Spatial information” and time “Temporal information”, which are common in most real-world cases [171]; as usual, whenever data are collected, they are associated with a location along the recording time. Capturing these previous examples along spatial and temporal information is debated by researchers and practitioners [172]–[174]. It has been suggested to use N-ary technique for describing

[5] http://ifcowl.openbimstandards.org/IFC4

relations [175]. Reification was suggested to capture additional information about statements (such as the source) [176] Reification has been used to generate the knowledge base from Wikipedia [177]. Another approach to store spatial and temporal information, known as named graph, has been proposed to store statements' annotations in a separate data graph [178], [179]. All the aforementioned approaches conform to RDF standards; hence, they can implemented, queried by SPARQL, and reasoned by existing reasoning engines, but they tend to obfuscate the model [180].

On the other hand, more revolutionary approaches that do not necessarily conform to RDF standards have been proposed. For example, because triples can capture unary and binary relationships, quads have been recommended instead of triples [181]. A similar approach suggested adding annotation capabilities to the standard RDF [180].

In our domain, time is a critical component to capture key frames for animations, so we decided to use reification to add time stamps for statements as follows:

```
m:s1 rdf:type rdf:statement;
m:s1 rdf:subject m:ModelItem1.
      rdf:predicate m:Location.
       rdf:object [x y z];
m:s1 :timestamp 2;
Which can be retrieved with the following query:
SELECT ?item ?coordinates
WHERE {
          ?s rdf:type rdf:statement
          ?s rdf:subject ?item
          ?s rdf:predicate m:Location
          ?s rdf:object ?coordinates
Filter (?s :timestamp = 2)
}
```

In order to facilitate compatibility with other ontologies, we decided to use “Time ontology in OWL” which is adopted by W3C to model temporal concepts [182]. This ontology contains time-related concepts such as “time interval,” “before,” and “after.”

4.4 Previous Experience

In this section, we briefly discuss three previous projects that are closely related to the visualization process. These projects were built upon High-level Architecture (HLA)-distributed simulation [183], [184]. In HLA simulation, the simulated problem is broken into federates that interact with each other during the simulation. Each of the mentioned project here has a visualizer federate that shows progress made by other federates.

Here, we focus on the visualization federate in each case by discussing the challenges and lessons learned to show the necessity of a generic visualization process that can accept data from different sources – whether from simulation components or stand-alone applications – and show them visually.

4.4.1 Pipe Manufacturing Visualizer

The first project simulates the construction of oil refineries and petrochemical plants, which follow modular construction paradigms. Different modules are fabricated by assembling components from different trades (e.g., pipes, structural steel, and equipment) in an off-site module yard; then they are shipped to the project site and installed using heavy-lift cranes. This is a complex process that involves multiple parties and requires careful planning and coordination.

An HLA-distributed simulation has been developed to model this operation [185], [186]. The simulation focuses on the piping manufacturing process; it simulates constructing piping modules from the module yard, transportation to the construction site, and the final installation. In addition, it tracks the associated schedule and ensures that predecessors have been fulfilled before installation.

This distributed simulation contains five federates:

1. **Simulation Federate**: This simulates all fabrication operations, and produces related statistics like the production rate.
2. **Resource Allocation Federate**: This allocates available cranes to ready-to-install modules. Crane selection depends on availability and ability to handle the module.
3. **Site Construction Federate**: This is responsible for preparing the construction site topography based on a topography data file.
4. **Yard Viewer Federate**: This is a 3D visualizer that displays simulation activities in the module yard.
5. **Site Viewer Federate**: This is another 3D visualizer that displays simulation activities in the construction site.

The visualization federates enhance a result’s readability, give more insight on the piping manufacturing process, and provide an easy way to validate the simulation. The visualizer has been used to provide the following:

- Visualize the logical sequence of the schedule as it displays the installation sequence according to the provided schedule.

- Help to display the utilization of the module yard, which is divided into bays where different pipe modules can be assembled in parallel. The visualizer also helps to determine whether the bays are over- or under-crowded.
- Combine the schedule logic with the spatial data to show the congestion in the construction site.

4.4.2 Earthmoving Visualizer

The second case tackles the earthmoving process; due to its repetitive nature, an earthmoving operation is a good candidate for simulation. Many simulation models have been built to study the effect of different factors on earthmoving operations. This includes fleet optimization [187]–[189]; decision support [190]–[192]; and utilizing real data [193]–[195].

Most of these simulation models considered an earthmoving operation as one model, which limits scalability and extensibility (i.e., adding new functionality in next developing cycles); we used distributed simulation to overcome these limitations by breaking the earthmoving operations into six federates: Controller, Loader, Mover, Breakdown, Weather, and Visualizer- [196].

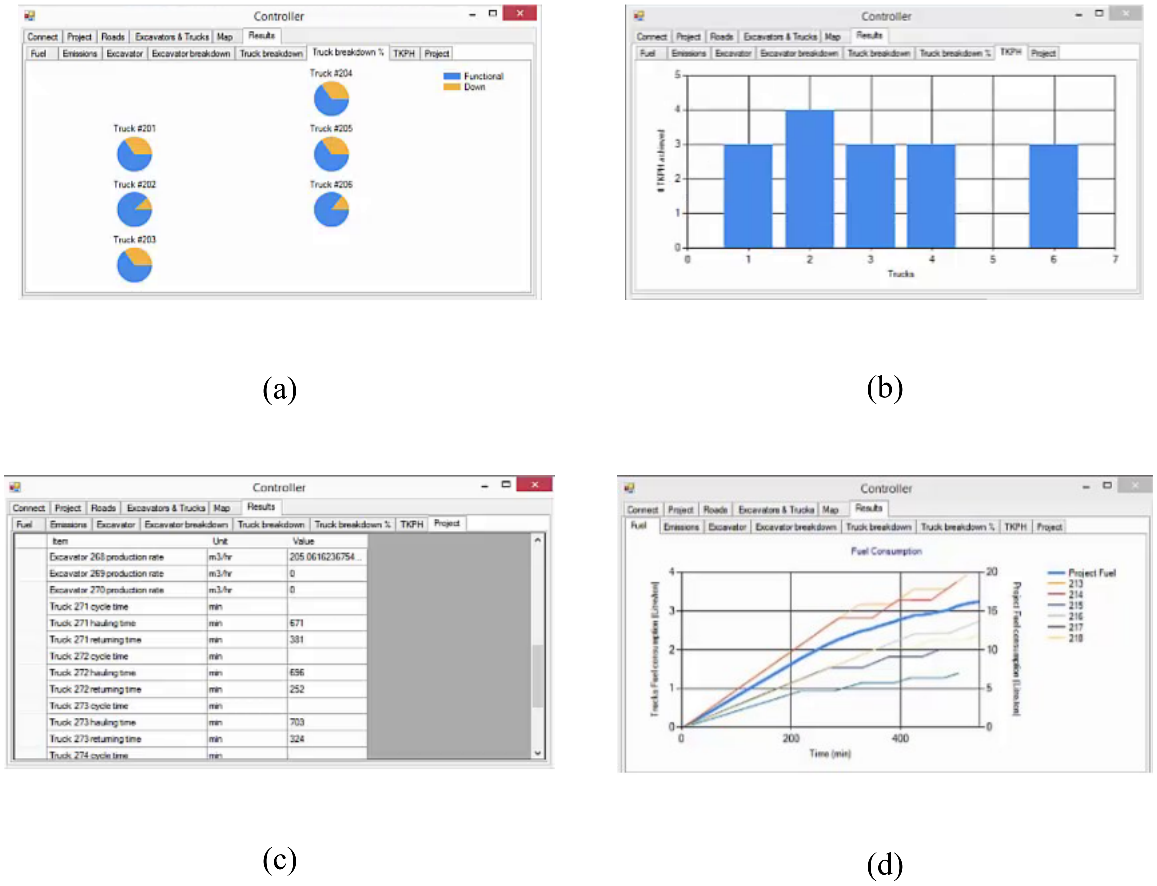
The Controller federate, as the name implies, is responsible for creating a federation, defining a scenario (e.g., fleet composition, road length, and hauling material), and displaying statistical results (e.g., production rate and utilization). The **Loader and Mover** deals with equipment movements in the mine and road respectively; this separation allows different teams to focus on different conditions; for example, while mover focused on tire wearing due to rolling resistance, the loader considered queuing trucks in the mine. The **Breakdown federate** simulates the breakdown effect on the production rate by breaking down trucks and excavators based on distributions drawn from historical data. The **Weather federate** studies the effect of weather conditions (e.g., precipitation, wind speed, and snow depth) on the earthmoving process; based on the earthmoving location, the Weather federate sends weather conditions to all other federates.

The first prototype consisted of five federates as we relied on the controller federate to display outputs as graphs such as excavator and truck utilization, fuel consumption, and breakdown percentage as shown in Figure 4-2. Afterwards, we decided to introduce a new federate, “Visualizer,” to extend our federation so it could display the dynamic real-time simulation behavior to the end user.

The Visualizer federate was built using Windows Presentation Foundation® (WPF) technology and a Helix 3D toolkit®. The WPF utilizes a Model-View-View Model (MVVM) design pattern [197]. It provides solid capabilities for 3D applications [198], which is greatly enhanced using the Helix toolkit. These technologies can be used to load and transform different 3D assets that have provided a real-time 3D output for the earthmoving operation.

At the simulation inception, the Visualizer federate loads terrain and roads from a 3ds file format. Afterwards, it loads trucks and excavators when they are registered by other federates; the visualizer is equipped with 3D assets for many trucks’ and excavators’ models and it loads the required model based on other federates’ requests. During the simulation, the trucks’ positions and states (i.e., loaded vs empty, working vs broken down) are interpreted based on other federates’ updates.

The Visualizer federate provides insight into the earthmoving operation; it shows the truck movement and state as seen in Figure 4-3. The user can manipulate (i.e., zooming, panning, rotating) the model. He/she can also hover over each piece of equipment and thus show its condition. This project shows how important and necessary a visualizer is to introduce simulation results to a non-expert in a more intuitive way.



Item	Unit	Value
Excavator 268 production rate	m3/hr	205.0616236754...
Excavator 269 production rate	m3/hr	0
Excavator 270 production rate	m3/hr	0
Truck 271 cycle time	min	
Truck 271 hauling time	min	671
Truck 271 returning time	min	381
Truck 272 cycle time	min	
Truck 272 hauling time	min	696
Truck 272 returning time	min	252
Truck 273 cycle time	min	
Truck 273 hauling time	min	703
Truck 273 returning time	min	324
Truck 274 cycle time	min	

(a)

(b)

(c)

(d)

Figure 4-2 Earthmoving simulation results as displayed in the Controller federate, (a) Breakdown percentage, (b) #Tons Kilometers Per Hour (TKPH) achieved, (c) Project statistics (e.g., Truck cycle time), and (d) Fuel consumption.

4.4.3 Distributed Observer Network (DON)

The Simulation Exploration Experience (SEE) [199] is an annual event organized by The National Aeronautics and Space Administration (NASA) and The Society for Modeling & Simulation International. SEE invites students from different universities around the world along with industry and professional associations to develop a distributed simulation for a space mission. The SEE challenge, with a time frame of around six months, gives students an inspiring way to learn and apply HLA standards while collaborating with each other around the world.

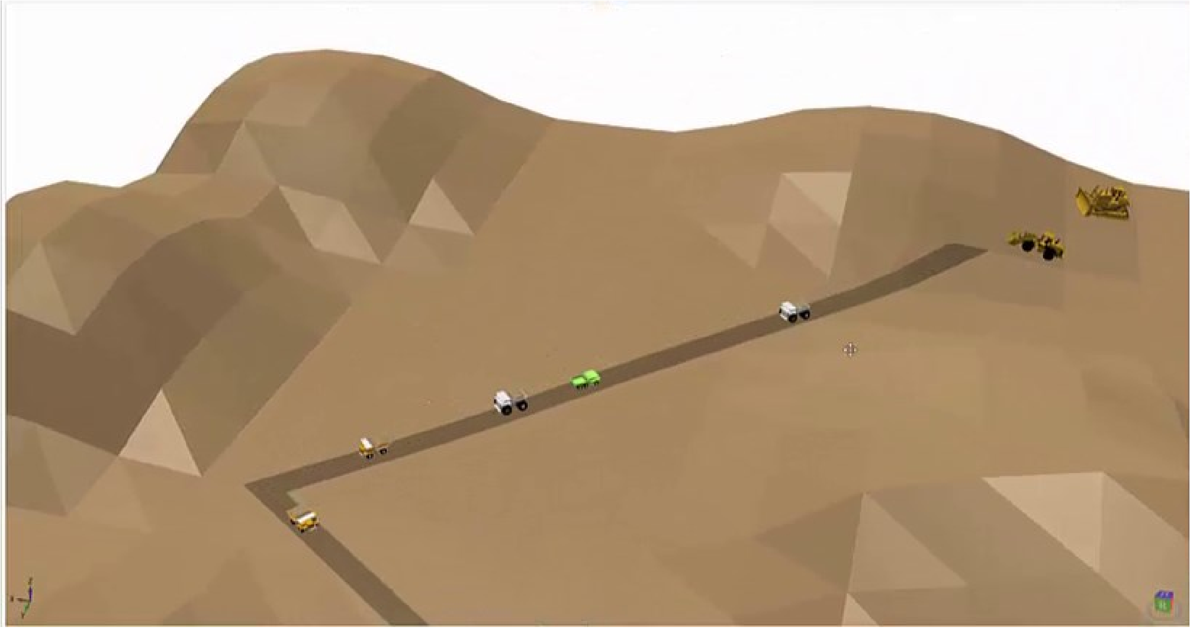


Figure 4-3 Earthmoving operation as displayed in the Visualizer.

I had the chance to join the SEE challenge last year (SEE 2015); the challenge was to develop a distributed simulation of a lunar mission. Teams from eight universities (University of Alberta, University of Bordeaux, University of Brunel, University of Calabria, University of Genoa, University of Liverpool, University of Munich, and University of Nebraska) participated. The teams developed 18 federates that simulated different tasks on the moon surface. The tasks included mining, an asteroid warning system, and transportation using rovers.

Our team (University of Alberta) developed a federate that simulates erecting a facility on the moon surface. The scenario was as follows: by utilizing a modular construction paradigm, a set of modules would be erected on earth and shipped to the moon. These modules would be moved to the construction site by the rover (a federate developed by another team). At the construction site, we had two cranes that were controlled from the earth, and were to be assembled based on a provided schedule and spatial data as seen in Figure 4-4 [200].

NASA teams contributed two federates to the simulation mission: The Environment and Distributed Observer Network (DON). The Environment federate set up the simulation scene and provided time management details such as the federation execution epoch and physical time representation. In addition, it provided a spatial position and orientation for different reference frames such as the sun-centered inertial, earth-centered inertial, and earth-centric fixed [201].

The DON federate provided 3D visualization capabilities for distributed simulation. DON was developed by NASA using a commercial game engine (Torque) and was released in 2008 [202]; we should mention here that since Torque 1.2, it has become an open source software.

Simulation plays an important role in each cycle of NASA exploration missions [203]. NASA uses DON to visualize simulation data. DON follows the client-server model by providing three classes: a master server which accepts/rejects credentials from the user, a dedicated server to run the simulation, and a client component which displays the simulation results for the end-user [203].

4.4.4 Lesson Learned

Different visualizers were developed for each project, and using them interchangeably is impossible as each was customized for a specific scenario. For example, in the earthmoving case, to empty/fill or move a truck, the visualizer assumed a specific format that might not be used in other simulation models.

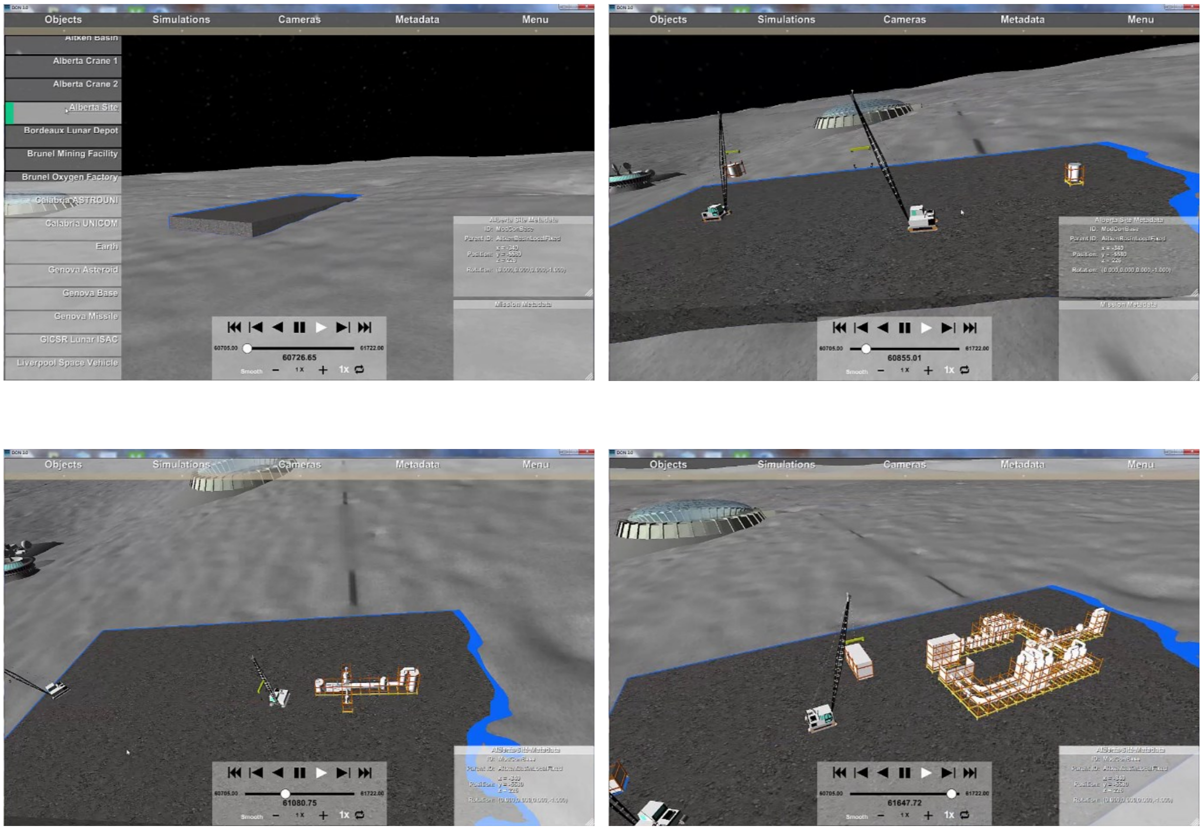


Figure 4-4 Erecting water treatment facility on the moon surface as rendered by DON.

Additionally, the visualizer shows all data generated by these scenarios and they cannot be filtered by focusing on a specific work area, and changing time resolution or level of detail requires rerunning the scenarios with a new configuration.

These limitations can be removed by using semantic web framework as an extra layer between data and a visualizer application. Semantic web framework will allow writing queries to filter data passed to the visualizer and hence control the visualization scenario. This leads to a generic visualizer that is not coupled with a specific scenario. The following section discusses our work and shows different scenarios shown by the same visualizer.

4.5 Proposed Framework

In this chapter, we propose a general framework (Figure 4-1) for visualizing heterogeneous data coming from different data sources. These data are merged using semantic web technology. We introduce an ontology that receives data from different sources and processes it according to the visualizer schema. Our scope of work includes developing the ontology—backed by a triple store—preparing connectors that take raw data from different applications that are widely used in the construction domain and converting it to an RDF format, and developing a hub that retrieves data from the triple store and converts it to the visualizer format as shown in Figure 4-1. We should emphasis here that although we utilize only one visualization application, the same ontology can be used with different visualization applications with minimal effort to create a new connector.

The framework consists of two main parts: 1) existing components that are widely used in the construction domain, and 2) developed components that stream data between different sources and the visualization applications. In this section, we discuss the developed components and their relationships with the existing components.

4.5.1 Ontology

Ontology is the key component in this framework as it formulates the relationship between data sources and the visualization. It should be generic enough to capture even unforeseen data sources, but at the same time it should be structured to be able to export temporal and positional information.

Researchers suggested defining ontology requirements in the form of questions. These questions, which are known as “competency questions” determine the scope of the ontology. [204]. Our competency questions are:

- Will the ontology be generic enough to receive data from different sources?
- Will it be able to convert the data to the visualization format?
- Will it be able to filter by item type?
- Will it be able to filter by levels?
- Will it be able to filter by different time resolutions?
- Will it be able to filter by time intervals?
- Will it be able to show secondary items (such as a scaffold)?

First, to ensure interoperability, we built upon existing ontologies by importing ifcOWL [160] and W3C time ontology [182]. ifcOWL contains concepts and definitions related to BIM (e.g., walls, doors), while time ontology defines time concepts such as time positions, before and after.

Next, we defined a new class, “Model Object,” for any object that has to be shown in the visualizer. This class encompasses definitions for position coordinates and units, 3D orientation in a quaternion format, object scale, and file path for the 3D asset as shown in Figure 4-5. Now, any object from any data source has to inherit this class to be shown in the visualizer. For example, we asserted that “IFC4:IfcElement” (from ifcOWl) is a subclass of “Model Object.” This means that if any instance in ifcOWL provides positional input and 3D asset’s file path, the instance will be displayed in the visualizer.

The previous step will show a static 3D scene; to add animation, different positional properties should be provided at key time frames. Hence, we add another class, "Object Time Stamp," as shown in Figure 4-6. This class captures the relationship between the model object and the time instance as defined in the W3C time ontology. In addition, this class contains information about object orientation and position at a specific time instant. Appendix D contains a full version of the proposed ontology in the turtle file format.

The ontology is backed by a triple store to handle the expected huge number of triples. We used Fuseki version 2.4.1 [82] which can handle millions of triples and also provides a web-based interface and SPARQL endpoints which will be used to process the data.

4.5.2 Data Mapping (Data Sources to RDF)

This section describes our work exporting data from different data sources to an RDF format based on the proposed ontology; this process is known as data integration [205]. We demonstrate the integration of some of the common applications in the construction domain. However, we should emphasize here that this list is not inclusive and additional data sources can be easily added. We only provide the data sources that have been used in the visualization scenarios shown later.

4.5.2.1 BIM Models

BIM models are widely used by engineering firms and in the construction domain to improve project management and collaboration [3]. In our context, BIM models provide rich information for visualization. This includes 3D assets and positional data. To export 3D assets from the 3D model, we used a customized plug-in that creates a separate OBJ file for each object in the model. Afterwards, we used the visual programming tool (Dynamo) [206]–[208]

to export: 1) the location, 2) the orientation, and 3) OBJ file path for each object in the model. Figure 4-7 shows a sample data flow that has been used to export columns and beams in a BIM model. The data are exported in spreadsheet format which can be converted to RDF triples as will be shown later. The BIM models connector has been tested with a steel structure frame, shown in Figure 4-8, which will be animated based on the associated schedule as will be shown later.

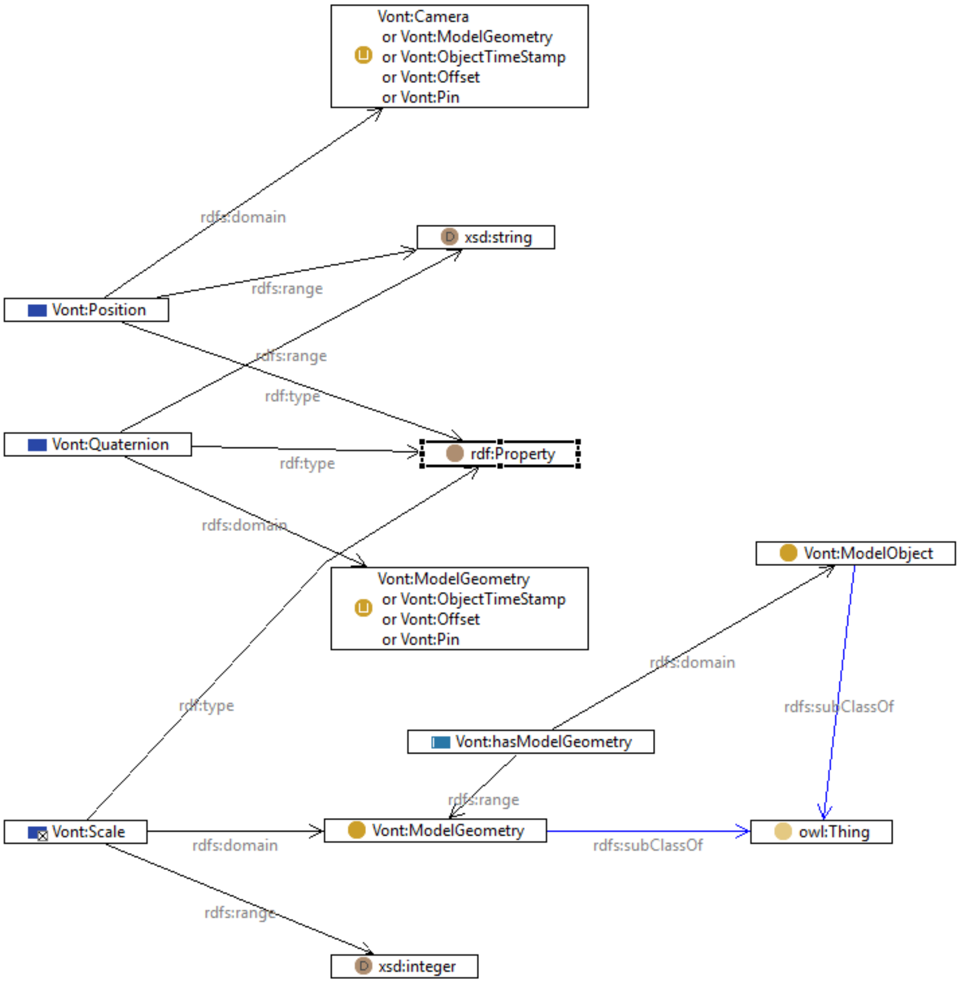


Figure 4-5 A sample of the proposed ontology that shows the relationship between "Model Object" and position, quaternion, and scale.

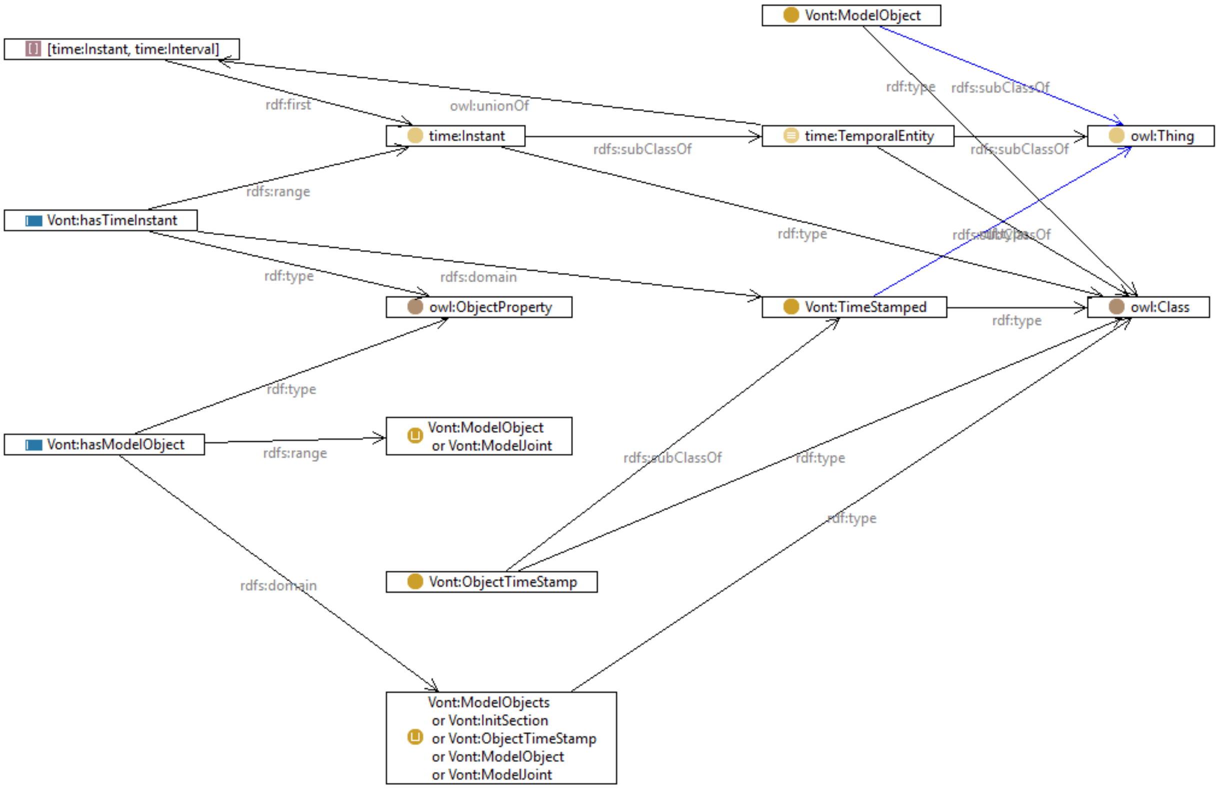
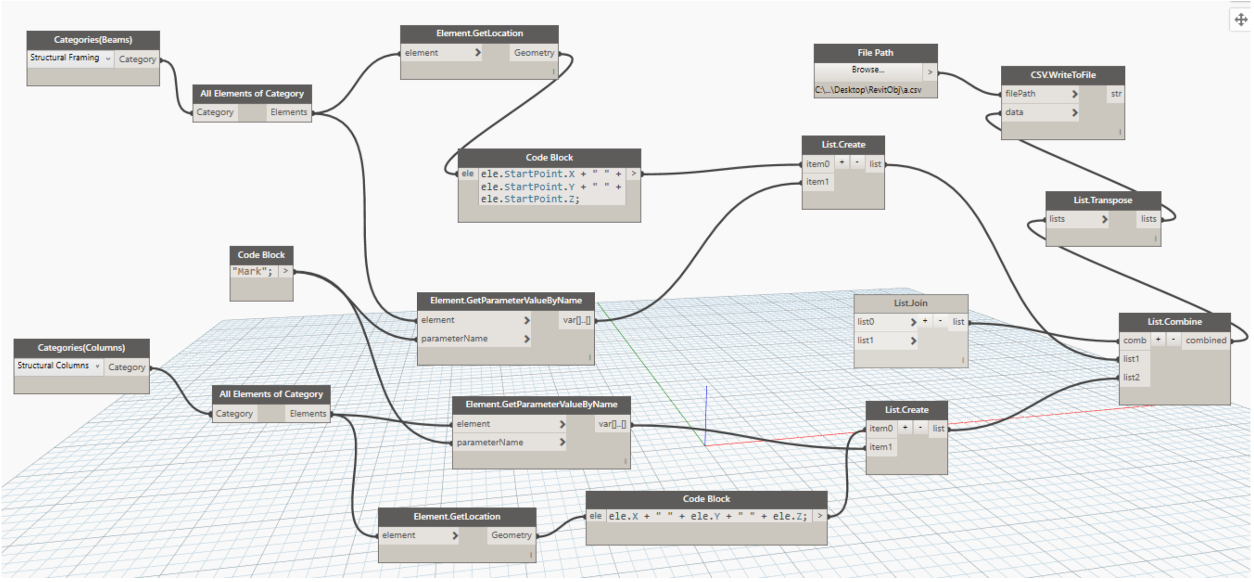


Figure 4-6 A sample of the ontology that shows the relationship between "Object Time Stamp" and "Model Object."



```
ele.StartPoint.X + " " +
ele.StartPoint.Y + " " +
ele.StartPoint.Z;
```

```
"Mark";
```

```
ele.X + " " + ele.Y + " " + ele.Z;
```

Figure 4-7 Exporting beams and columns using the visual programming tool.

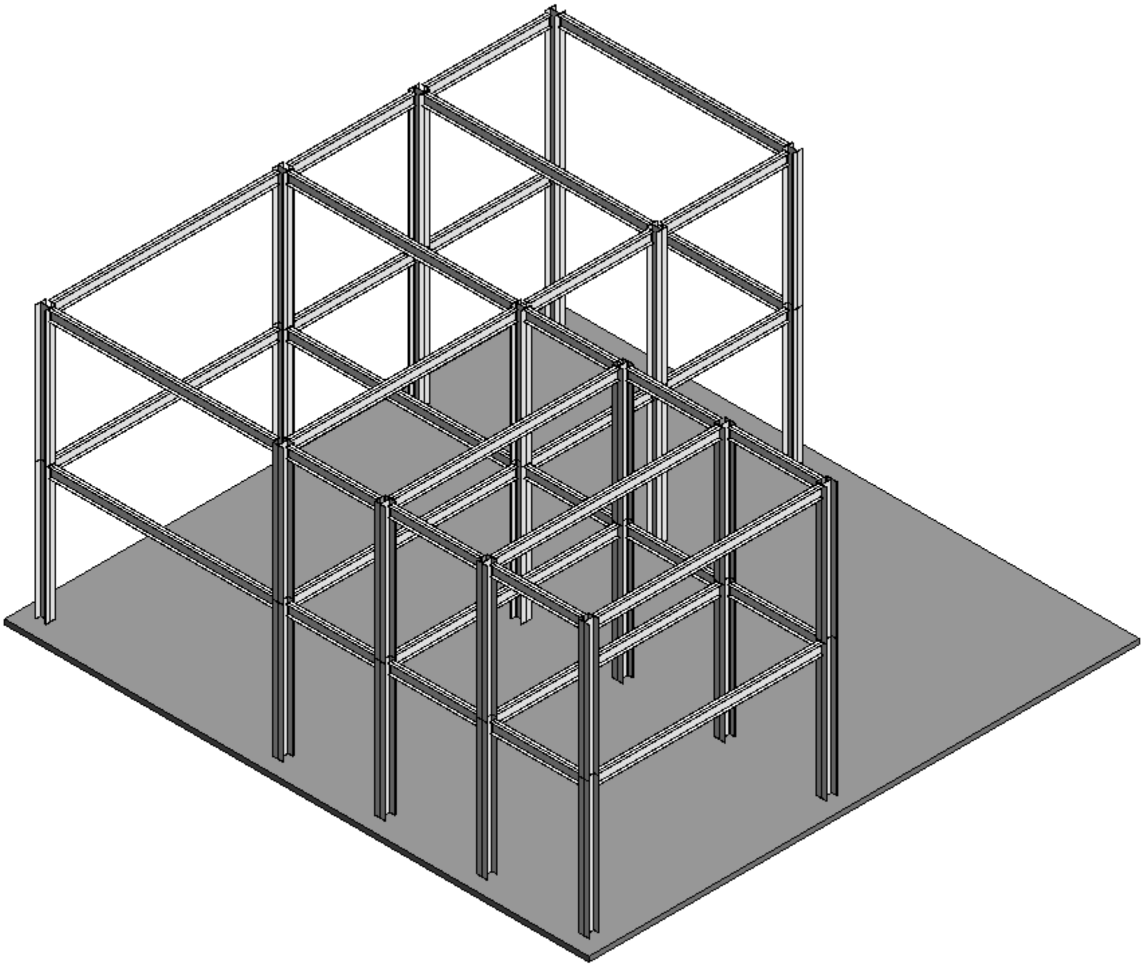
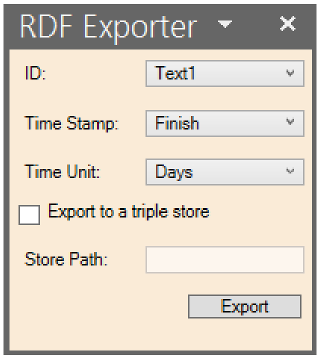


Figure 4-8 A sample BIM model for a steel structure frame.

4.5.2.2 Schedules

Another key component of a construction project is the schedule. The schedule contains temporal information that can be used to animate objects from other sources. Additionally, we used it to capture the resources data. A customized plug-in for Microsoft Project® (Figure 4-9) has been used to convert tasks finish/actual finish to the RDF format; this plug-in can write the RDF triples to a local file or a triple store through an HTTP connection. Clearly, an ID that links the task with items from other sources is required.



ID: Text1

Time Stamp: Finish

Time Unit: Days

□ Export to a triple store

Store Path:

Figure 4-9 The interface of the schedule connectors.

4.5.2.3 Simulation Models

Simulation is a powerful tool to capture and model dynamic systems with a large number of variables that are hard to model using mathematical models. It provides an experimental frame for testing a real world system effectively and cheaply [209].

A construction project is a good example of a dynamic, random, and heterogeneous system [210] with many variables, such as weather conditions, that severely affect progress. Simulation has been applied in many construction fields such as dams [211], dispute resolution [212], tunneling [149], and bridges [213].

Most of these applications were initiated and developed by researchers, not practitioners, who are still reluctant to use simulation in the industrial world [214]. This gap might be filled by providing easy-to-understand results with minimum training in simulation [214].

Visualization is arguably one of the most suitable ways not only to interpret results but to validate and accredit the simulation model [215]. Attempts have been made to provide a visualization interface along the simulation. For example, a visualization engine for tunneling has been developed [149]. Additionally, Visualization has been used extensively in training

[150], [151], and [153]. However, we noticed that instead of using a generic visualizer, each simulation runs on a specialized one.

In our case, we use simulation models as another data source to give more details about a process modelled using RDF triples. For example, merging a BIM model with its associated schedule creates a stand-alone 4D visualization in which objects appear in their final location when the associated tasks are complete. However, a simulation model allows for the addition of more details about object hauling from the storage area and crane lifting and swinging.

We created a special simulation template in Simphony [216], [217] that simulates the interactions between modules, trailers, and cranes, a screenshot of a sample model is shown in Figure 4-10. This template exports the results as RDF triples. A sample of generated RDF triples can be found in Appendix E.

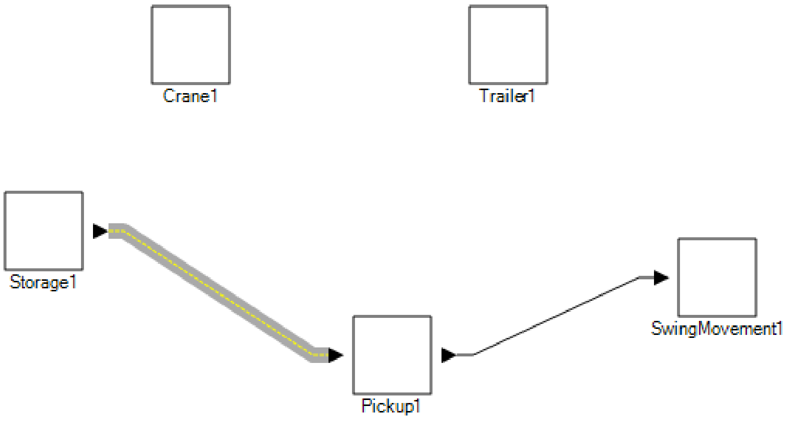


Figure 4-10 A simulation model developed in Simphony that captures the module and crane swing movement.

4.5.2.4 Spreadsheets

Many data sources come in a spreadsheet format or at least can be converted to this format – such as data from BIM models as shown earlier. This section describes how the data in the format have been mapped to RDF triples.

RDF123 [218], [219] is an open-source tool that exports tabular data to an RDF format through a mapping graph. The mapping graph should be structured based on the spreadsheet structure and the ontology. Figure 4-11 shows a sample for a mapping graph which converts csv file format to RDF format.

As an example of this conversion, we obtained a scaffold requests log for an oil and gas project in Alberta. The log is in a tabular data format with the following relevant columns: Request ID, Location, Required Elevation, Erection Date, and Dismantle Date.

We converted this data to RDF triples which were added to an existing 4D visualization to show scaffolds during the visualization process, as will be shown later. We used a semi-transparent box to model the scaffold but a more realistic representation can be used as well.

4.5.3 RDF to Visualizer Connector

In the previous section, we discussed our work regarding converting different data sources to the RDF format. Now, we have a collection of RDF triples that has to be converted to the visualizer application data format as shown in Figure 4-1. This requires a customized connector that converts from RDF to the visualizer format. In this section, we will describe how we converted the RDF triples to visualize them in DON.

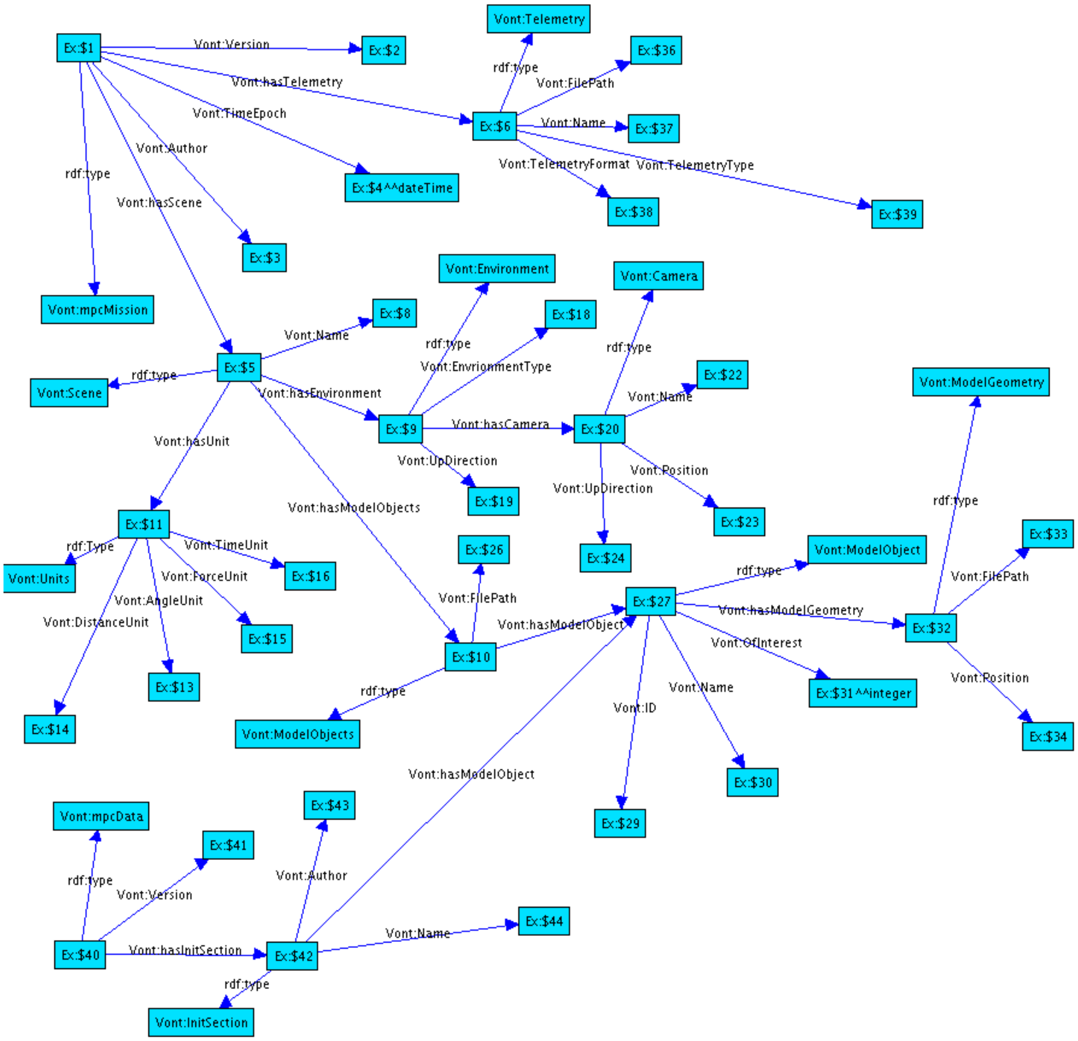


Figure 4-11 A sample map graph used to export spreadsheet data.

DON is a virtual environment for visualization [220] that accepts XML files as an input [221]. In general, two XML files are required to visualize a process in DON. The first file is a "Mission File" which constructs the visualization scene by providing information about environment, cameras, lights, object hierarchy, and a reference for the second XML file [221]. The second XML file, known as the "Data File," contains two main sections: 1) initialization, and 2) time section.

The initialization section contains metadata definitions and a list of objects that will be referenced in the time section. The time section captures the time steps in chronological order. Each time step might specify a new position or orientation for any object defined in the initialization section. The visualizer interpolates the animation between each two consecutive time steps. The current XML schema of the input file is “MPC3” and it is documented in [221]. Appendix F and Appendix G show the sample mission and data files respectively.

We developed a connector that takes RDF triples and converts them to XML files according to DON schema. The connector utilizes dotNetRDF [81] to execute remote SPARQL [34] queries and create the two XML files based on the query’s results. This structure allows us to filter data based on customized queries as shown in the following section.

4.6 Testing Scenarios

After describing different data sources that we have used, this section illustrates how we merge data from different sources to get an animation for the whole process with different levels of detail. The following sections describe three scenarios. The first is a merger of a schedule with a BIM model to display 4D animation. The second adds scaffold erection and dismantling times and locations, while the third focuses on a finer time resolution and visualizes the handling of a module using a crane. Figure 4-12 shows the components that the framework used to produce these scenarios. This includes CPM schedules, csv scaffold requests, BIM models, and simulation models. These data have been converted to RDF and stored in a triple store which is then queried to generate different visualization scnearios.

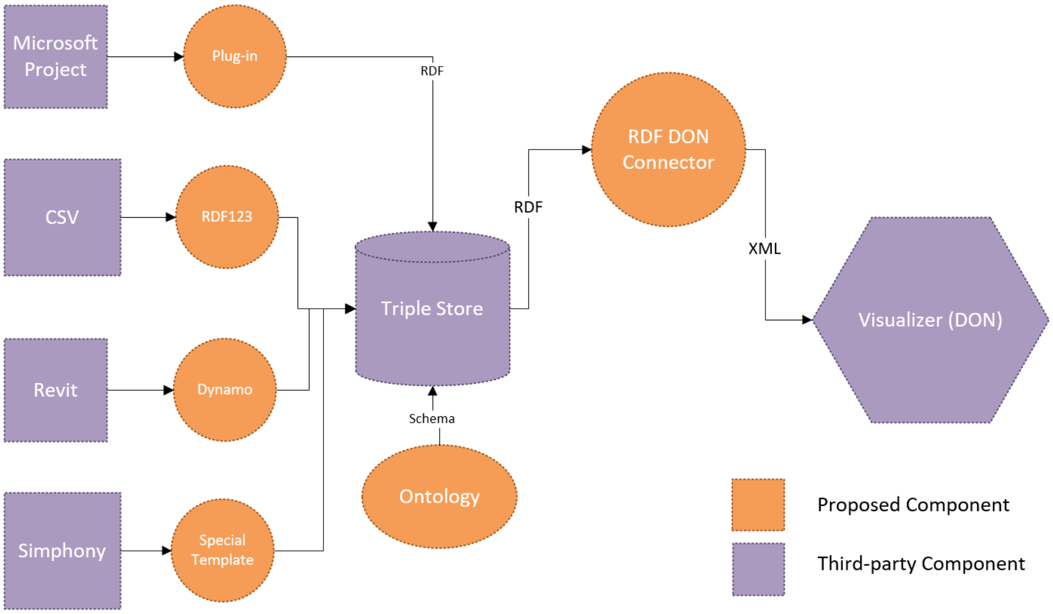


Figure 4-12 The components used to generate the visualization scenarios.

4.6.1 4D

4D animation is a high-level visualization that focuses on the big picture by showing the erection of a facility based on the actual or planned progress. Figure 4-13 shows different timeframes for steel frame erection. Figure 4-14 shows modules installation for an oil and gas project.

In both cases, we took the following steps: first we exported the 3D assets, from Revit in the first case and Blender in the second. Then we converted spatial information into an RDF format. Afterwards, we mapped each item to the corresponding task in the schedule and converted the schedule to an RDF format. Finally, we used the developed connector to create the XML files which are visualized in DON.

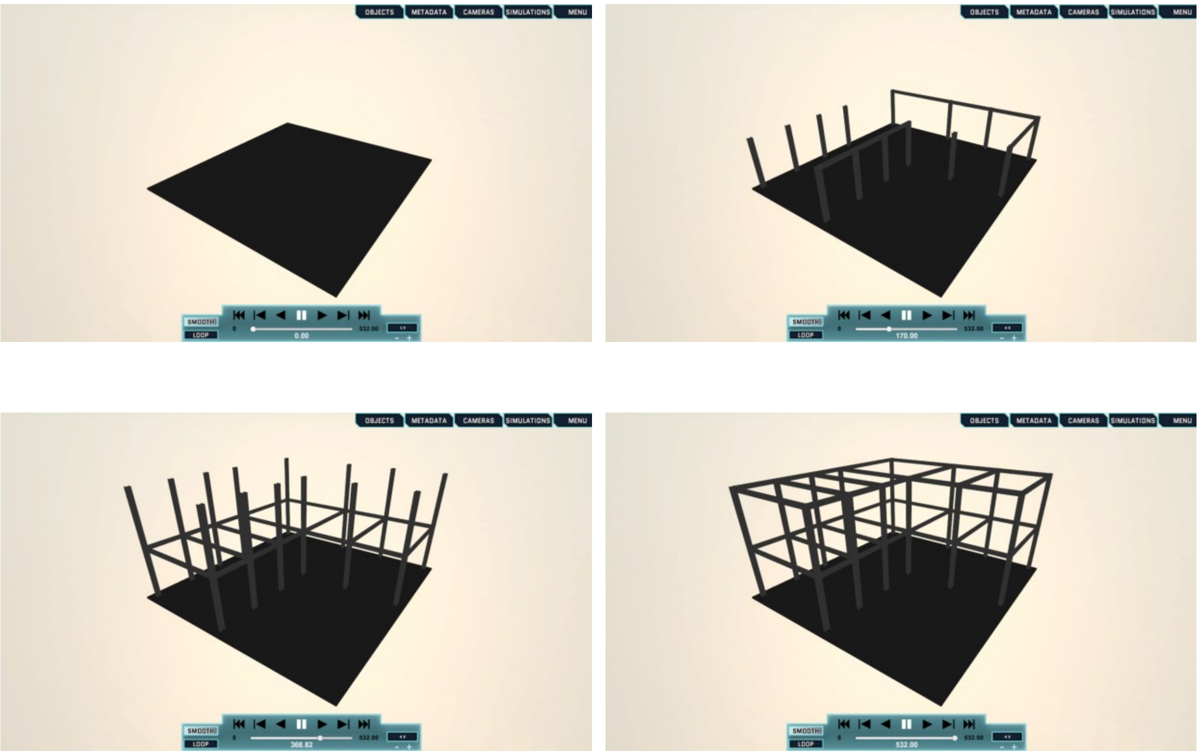


Figure 4-13 A sequence of screenshots that shows steel frame erection.

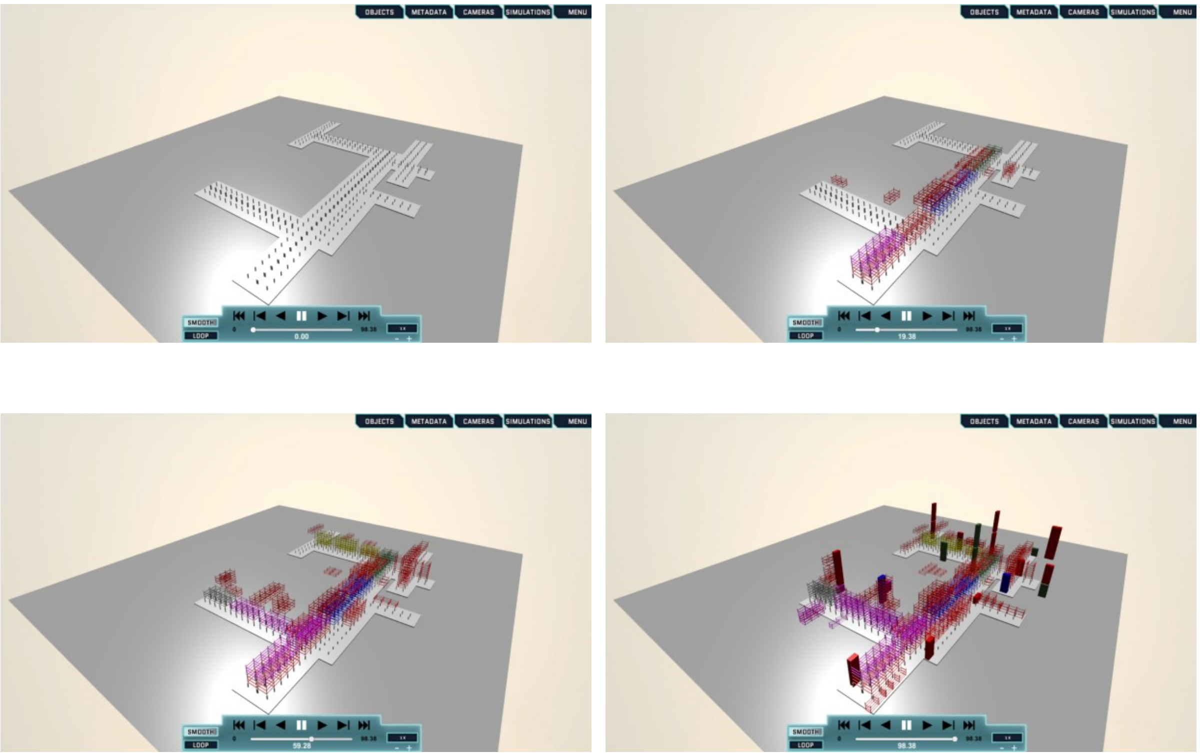


Figure 4-14 A sequence of screenshots that shows module installation in an oil and gas project.

4.6.2 4D with Scaffold

This scenario demonstrates the advantages of the proposed framework as it automatically adds another data source (spreadsheet) to an existing visualization process without any required editing. We obtained the scaffold log for the same oil and gas project. The log is maintained by general foremen in the site. These are not the same employees who maintained the 3D model. Merging these heterogeneous data sources went smoothly. Figure 4-15 shows that scaffolds were in the right place at the right time. Because neither we nor the project owner possess 3D assets for the scaffold, we used transparent boxes that changed the dimension based on the scaffold's dimensions and heights.

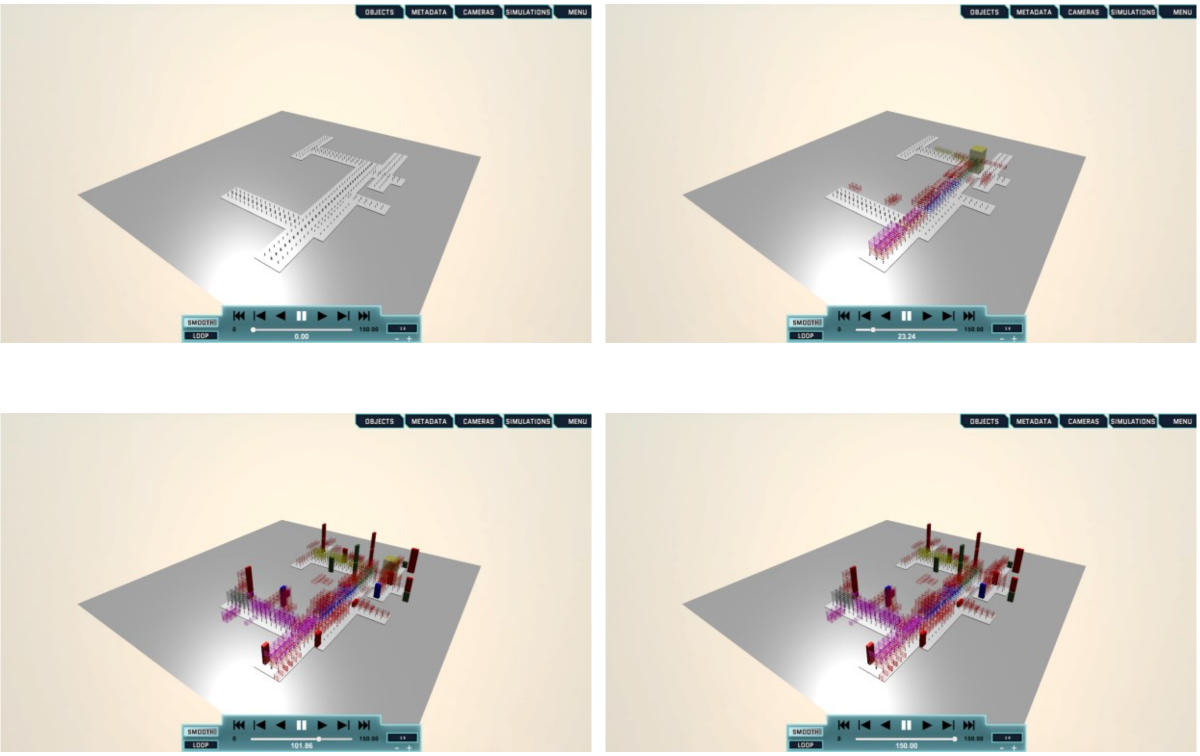


Figure 4-15 This is the same project shown in Figure 4-14 but we added the scaffold (the transparent objects in the second and third frame).

4.6.3 Crane Movement

The same framework can be used to show a more detailed visualization using simulation data as shown in this scenario. This scenario captures the module lifecycle starting from the storage area. A trailer moved the module to the pickup point. Afterwards, the crane lifted the module, swung it, moved to the drop point, and then dropped the module in the final location as shown in Figure 4-16. Each movable part of the crane had to be exported as a separate 3D asset to capture the relative movement between parts.

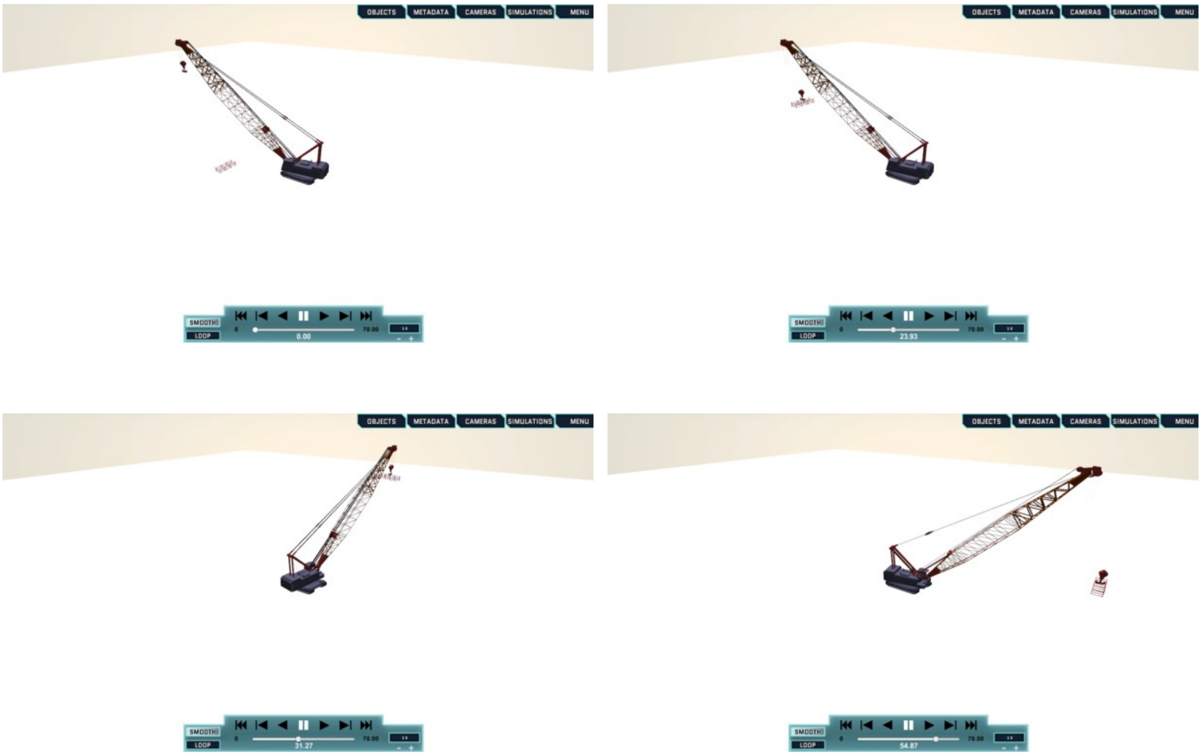


Figure 4-16 Handling of a module using a crane that shows lifting, swinging, hauling, and dropping.

4.7 Conclusion

Visualization of construction activities is a complex process that requires significant effort to select the visualizer, prepare 3D assets, retrieve data, and transform it according to the visualizer specification. This ad-hoc methodology led to the creation of many visualization models that are only suitable for one or two application cases. This chapter presented a new framework for visualizing construction projects and activities. The framework focuses on the data rather than the visualization engine. By using an RDF data format as a data hub, data from different sources and formats can be merged into one triple store. This reduces the problem of visualization to the selection of a visualization engine and development of data bridges between the triple store and the visualizer and between the data sources and the triple store. Using the proposed framework, we tested our methodology with different scenarios that demonstrated the ability to visualize different ranges of activities with different levels of detail.

Chapter 5. Conclusion

5.1 Conclusion

Building Information Modeling (BIM) has changed our way of dealing with construction projects. It serves as a data store that can capture attributes other than geometrical objects. This allows engineers and contractors to work mainly with one source of the data that is expected to provide all information related to a project.

However, the current data flow practice between engineering firms and contractors and the usage of customized BIM solutions in industrial projects limits the potentials of BIM, as there is no consensus on the naming convention, and the meta-data are not fully described in BIM models, especially during the early stages of projects. This leads to what is known as a "Dump Model," which can be inspected visually but is hard or impossible to utilize for repetitive tasks, such as quantity take-off, that are needed for planning.

This research aims to leverage information usage in BIM models through two steps: 1) automatically complete and validate missing data in the BIM models and 2) develop a semantic web ontology that can automatically merge BIM data with other data sources commonly used in early stages of a project.

Our work included proposing a methodology to automatically categorize objects by their trade, taking into consideration the inconsistent and missing attributes. We proposed two mathematical models that scan attributes in the model and promote clustering attributes. This will generate a set of clusters that separate BIM objects by trade. Afterwards, we used shape recognition techniques to find the correct shape and size of each steel object in the model. Finally, we used semantic web technology to store and merge calculated data and with any additional data sources. This led to one data store – know as a triple store – that captured BIM data along operational data such as scaffolds and crane positions. We demonstrated the capabilities of this triple store by using these heterogeneous data to provide an animated visualization of the project with different level of detail and time resolutions.

In order to validate this framework, we started with real BIM models for oil and gas projects that have been executed in Alberta, Canada during the last decade. The average budget for each project is C$750 million. Each model is a typical "Dump Model" that contains 3D objects but without enough attributes to provide an accurate description. These early-stage models – which are commonly used in fast-tracked projects- cannot be easily categorized by trade, let alone categorized by their classes. The results show that the proposed clustering technique is able to achieve 91% purity level on average and shape recognition technique can provide an acceptable preliminary estimate. The work was structured as follows.

Chapter 2 discusses our work regarding automatically categorizing ill-defined BIM objects. Our proposed methodology utilized two models: 1) the Shannon Entropy and 2) TF-IDF to analyze the BIM data and find relationships between BIM objects.

In order to apply these two models, we converted the BIM data to the RDF format (triples) and then we used the two models to promote candidate attributes. Candidate attributes are the attributes that can be used to partition the data into groups.

We tested the models with three real-case projects. Each project generates millions of triples, yet the models clustered them with a purity measure up to 91%. This enabled us to easily extract steel trade objects from the BIM models. These steel objects were then processed further.

Chapter 3 discusses our work regarding determining the shape and size of each steel object using shape recognition techniques.

Our work here focuses on the geometrical properties rather than the descriptive attributes. After reviewing shape recognition algorithms in the literature, we decided to use the shape distribution algorithm. Although this algorithm has been criticized for low accuracy, we believe it is suitable for preliminary analysis because of its smaller computation time and low sensitivity.

The fundamental idea of this algorithm is to select enough random points on the object surface and measure the distances between randomly selected points' pairs. Each shape generates an unique histogram which can be considered a "shape signature" and can be used to determine if two objects are similar.

We generated the shape signature of all standard sections in the Canadian standards. Then we generated the shape signature of the steel objects and compared them to determine the closest section for each steel object in the BIM model. The results show that by using 50,000 random distances—requiring on average one second to process each object—we get an 82% success rate. A higher success rate might be obtained with more distances but the computation time will be higher.

The previous chapters focus on finding automated ways to fill the missing and inconsistent data in the BIM model. Chapter 4 includes our proposed method to handle heterogeneous data from different data sources.

We propose using RDF triples to store these data. The RDF format allows heterogenous data to merge automatically—provided that related data has the same unique ID. – The RDF format has many applications such as using BIM models to relate scaffold man-hours from scaffold logs to an object's size and height. We focused on visualization as an application for data merging.

Our methodology includes developing a set of connectors that convert data from different applications to the RDF format. These applications include BIM applications, simulation models, spreadsheets, and scheduling engines. After developing the connectors that convert data from different applications to RDF format, we selected a visualizer application and created another connector that converts RDF data to the visualization input files. The flexibility of the RDF format means that with minimal human interaction, we can show many visualization scenarios with different levels of detail and time resolutions.

5.2 Research Contribution

We have categorized our work contributions into academic and industrial.

5.2.1 Academic Contribution

The main academic contributions are:

1. The study presented a clustering technique that works with a non-tabular data format. It demonstrates a novel utilization of Shannon Entropy and TF-IDF for clustering BIM

objects. It also provided a comparison between Shannon Entropy and TF-IDF performances.

2. The study showed that the shape distribution algorithm can be used to give a preliminary estimate of steel quantity in unlabelled BIM models based on geometrical properties. The results showed that the average difference in total weight between the actual sections and identified sections for the sample is 8.8%.
3. The study demonstrated that RDF format can be used to merge heterogeneous data from different data sources and can do so automatically. This merging has potential applications such as visualizing data with different levels of detail and time resolutions.

5.2.2 Industrial Contribution

These industrial contributions are:

1. The study provided a framework for categorizing ill-defined BIM objects based on their trade. This simplifies the preliminary analysis for contractors as it subdivides the BIM models into smaller, manageable divisions.
2. The study introduced a tool that can perform shape recognition techniques on BIM objects during early stages of a project. The tool estimates the quantity take-off with a ±8.8% difference in weight, which is within the acceptable tolerance in the industry (±10%) [139].
3. The study created a visualization technique which is capable of automatically capturing data from common applications used in the industry and visualizing them. This allows practitioners to easily provide an animated visualization to different audiences.

5.3 Limitations and Future Work

The limitations of and areas of improvement for this research are:

1. The clustering technique results show that Project 2 performs worse than the other project which means that the techniques requires the BIM model to have sufficient level of attributes to be successfully used. An extend to the work will consider geometric properties along the attributes.
2. The proposed clustering technique can be considered an unsupervised learning as it provides an unlabelled cluster set; however, a future study will enhance the purity level by using statistical analysis to find common labels for each cluster.
3. The clustering technique includes four mathematical models (1A, 1B, 2A, and 2B) and the results show that 2A and 2B are comparable and we are not able to promote one of them over the other. More validation is required to differentiate between these two models.
4. The shape recognition technique will fail to detect arbitrary shapes as it only compares an unidentified object to a reference set of objects. Hence, it cannot be used with other trades such as concrete.
5. The shape recognition algorithm requires huge number distances to achieve acceptable results, which in turn increases computation time. We found that 50,000 distances will give acceptable results but it takes one second to process each section, which is relatively long, especially for huge BIM models.
6. The shape recognition algorithm is not inclusive and it is usually used as a pre-classifier algorithm; therefore, it is limited to use only for preliminary estimates and cannot be used, for example, to formulate the bill of materials.

7. The visualizer application (DON) requires OBJ files for 3D assets which might not be available from some data sources.
8. The proposed ontology does not contain enough constraints to prevent adding invalid or discrepant data.

Bibliography

[1] S. Azhar, "Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry," *Leadership and Management in Engineering*, vol. 11, no. 3, pp. 241–252, 2011.

[2] D. Bryde, M. Broquetas, and J. M. Volm, "The project benefits of Building Information Modelling (BIM)," *International Journal of Project Management*, vol. 31, no. 7, pp. 971–980, Oct. 2013.

[3] C. Eastman, P. Teicholz, R. Sacks, and K. Liston, *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, 2nd edition. Hoboken, NJ: Wiley, 2011.

[4] Andy K.D. Wong, Francis K.W. Wong, and Abid Nadeem, "Government roles in implementing building information modelling systems," *Construction Innovation*, vol. 11, no. 1, pp. 61–76, Jan. 2011.

[5] A. Porwal and K. N. Hewage, "Building Information Modeling (BIM) partnering framework for public construction projects," *Automation in Construction*, vol. 31, pp. 204–214, May 2013.

[6] Y. Jung and M. Joo, "Building information modelling (BIM) framework for practical implementation," *Automation in Construction*, vol. 20, no. 2, pp. 126–133, Mar. 2011.

[7] H. Tanaka, "Toward Project and Program Management Paradigm in the Space of Complexity: A Case Study of Mega and Complex Oil and Gas Development and Infrastructure Projects," *Procedia - Social and Behavioral Sciences*, vol. 119, pp. 65–74, Mar. 2014.

[8] S. R. Thomas, C. L. Macken, and S.-H. Lee, "Impacts of design/information technology on building and industrial projects," *A Report submitted to NIST, Construction Industry Institute, University of Texas, Austin, TX*, 2001.

[9] Guillermo Aranda-Mena, John Crawford, Agustin Chevez, and Thomas Froese, "Building information modelling demystified: does it make business sense to adopt BIM?," *Int J Managing Projects in Bus*, vol. 2, no. 3, pp. 419–434, Jun. 2009.

[10] N. Han, Z. F. Yue, and Y. F. Lu, "Collision Detection of Building Facility Pipes and Ducts Based on BIM Technology," *Advanced Materials Research*, vol. 346, pp. 312–317, 2012.

[11] A. A. Latiffi, S. Mohd, N. Kasim, and M. S. Fathi, "Building Information Modeling (BIM) Application in Malaysian Construction Industry," *International Journal of Construction Engineering and Management*, vol. 2, no. A, pp. 1–6, 2013.

[12] R. Amor and H. Ma, "Preservation of meaning in mapped IFCs," *Proceedings of EC-PPM*, pp. 233–236, 2006.

[13] T. Froese, "Future directions for IFC-based interoperability," *Journal of Information Technology in Construction (ITcon)*, vol. 8, no. 17, pp. 231–246, Jul. 2003.

[14] Y.-S. Jeong, C. M. Eastman, R. Sacks, and I. Kaner, "Benchmark tests for BIM data exchanges of precast concrete," *Automation in Construction*, vol. 18, no. 4, pp. 469–484, Jul. 2009.

[15] T. Pazlar and Ž. Turk, "Interoperability in practice: geometric data exchange using the IFC standard," *Journal of Information Technology in Construction (ITcon)*, vol. 13, no. 24, pp. 362–380, Jun. 2008.

[16] T. Rujirayanyong and J. J. Shi, "A project-oriented data warehouse for construction," *Automation in Construction*, vol. 15, no. 6, pp. 800–807, Nov. 2006.

[17] K. W. Chau, M. Anson, and J. P. Zhang, "4D dynamic construction management and visualization software: 1. Development," *Automation in Construction*, vol. 14, no. 4, pp. 512–524, Aug. 2005.

[18] T. W. Kang and C. H. Hong, "A study on software architecture for effective BIM/GIS-based facility management data integration," *Automation in Construction*, vol. 54, pp. 25–38, Jun. 2015.

[19] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities," *Scientific American*, vol. 284, no. 5, pp. 35–43, May-2001.

[20] T. Segaran, C. Evans, and J. Taylor, *Programming the Semantic Web*, 1st edition. O'Reilly Media, 2009.

[21] M. Betts, P. S. Brandon, and M. B. Nfa, *Integrated Construction Information*, 1 edition. Routledge, 1995.

[22] Y. Chen and J. M. Kamara, "A framework for using mobile computing for information management on construction sites," *Automation in Construction*, vol. 20, no. 7, pp. 776–788, Nov. 2011.

[23] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape Distributions," *ACM Trans. Graph.*, vol. 21, no. 4, pp. 807–832, Oct. 2002.

[24] CISC/ICCA, *Handbook of Steel Construction - Tenth Edition*. Canadian Institute of Steel Construction, 2012.

[25] A. Khanzode, M. Fischer, and D. Reed, "Benefits and Lessons Learned of Implementing Building Virtual Design and Construction (vdc) Technologies for Coordination of Mechanical, Electrical, and Plumbing," *ITcon*, vol. 13, pp. 324–342, Jun. 2008.

[26] G. Williams, "Fast Track Pros and Cons: Considerations for Industrial Projects," *J. Manage. Eng.*, vol. 11, no. 5, pp. 24–32, Sep. 1995.

[27] K. Davies, D. J. McMeel, and S. Wilkinson, "Making friends with Frankenstein: hybrid practice in BIM," *Eng, Const and Arch Man*, vol. 24, no. 1, pp. 78–93, Jan. 2017.

[28] S.-K. Lee, K.-R. Kim, and J.-H. Yu, "BIM and ontology-based approach for building cost estimation," *Automation in Construction*, vol. 41, pp. 96–105, May 2014.

[29] "RDF 1.1 XML Syntax." [Online]. Available: https://www.w3.org/TR/rdf-syntax-grammar/. [Accessed: 19-Jan-2017].

[30] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[31] H. P. Luhn, "The Automatic Creation of Literature Abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, Apr. 1958.

[32] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 1st edition. New York: Cambridge University Press, 2008.

[33] "Navisworks | Project Review Software | Autodesk." [Online]. Available: http://www.autodesk.com/products/navisworks/overview. [Accessed: 18-Jan-2017].

[34] The W3C SPARQL Working Group, "SPARQL 1.1 Overview." [Online]. Available: https://www.w3.org/TR/sparql11-overview/. [Accessed: 18-Jan-2017].

[35] H. Schevers *et al.*, "Towards digital facility modelling for Sydney opera house using IFC and semantic web technology," *ITcon*, vol. 12, pp. 347–362, 2007.

[36] M. Ehrig and Y. Sure, "Ontology Mapping – An Integrated Approach," in *The Semantic Web: Research and Applications*, C. J. Bussler, J. Davies, D. Fensel, and R. Studer, Eds. Springer Berlin Heidelberg, 2004, pp. 76–91.

[37] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data the story so far," in *Semantic Services, Interoperability and Web Applications: Emerging Concepts: Emerging Concepts*, IGI Global, 2011.

[38] J. Lehmann *et al.*, "DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia," *Semantic Web – Interoperability, Usability, Applicability Journal*, vol. 2012, no. 1, pp. 1–5, 2012.

[39] D. Vrandečić, "Wikidata: A New Platform for Collaborative Data Collection," in *Proceedings of the 21st International Conference on World Wide Web*, New York, NY, USA, 2012, pp. 1063–1064.

[40] D. Vrandečić and M. Krötzsch, "Wikidata: A Free Collaborative Knowledgebase," *Commun. ACM*, vol. 57, no. 10, pp. 78–85, Sep. 2014.

[41] T. Hartmann, J. Gao, and M. Fischer, "Areas of Application for 3D and 4D Models on Construction Projects," *J. Constr. Eng. Manage.*, vol. 134, no. 10, pp. 776–785, Oct. 2008.

[42] O. López-Ortega and R. Moramay, "A STEP-based manufacturing information system to share flexible manufacturing resources data," *J Intell Manuf*, vol. 16, no. 3, pp. 287–301, Jun. 2005.

[43] S. A. Mamrak, M. S. Kaelbling, C. K. Nicholas, and M. Share, "Chameleon: a system for solving the data-translation problem," *IEEE Transactions on Software Engineering*, vol. 15, no. 9, pp. 1090–1108, Sep. 1989.

[44] R. J. Scherer and S.-E. Schapke, "A distributed multi-model-based Management Information System for simulation and decision-making on construction projects," *Advanced Engineering Informatics*, vol. 25, no. 4, pp. 582–599, Oct. 2011.

[45] "EDM COUNCIL: Financial Industry Business Ontology™," *FINANCIAL INDUSTRY BUSINESS ONTOLOGY™*, 25-Feb-2016. [Online]. Available: http://www.edmcouncil.org/financialbusiness. [Accessed: 25-Feb-2016].

[46] "Schema.org," *schema.org*, 25-Feb-2016. [Online]. Available: http://schema.org/. [Accessed: 25-Feb-2016].

[47] "15926.org," Jul-2016. [Online]. Available: http://15926.org/. [Accessed: 29-Jul-2016].

[48] R. Batres *et al.*, "An upper ontology based on ISO 15926," *Computers & Chemical Engineering*, vol. 31, no. 5–6, pp. 519–534, May 2007.

[49] J. W. Klüwer, M. G. Skjæveland, and M. Valen-Sendstad, "ISO 15926 templates and the Semantic Web," in *Position paper for W3C Workshop on Semantic Web in Energy Industries; Part I: Oil and Gas*, 2008.

[50] M. West, "Some industrial experiences in the development and use of ontologies," in *EKAW 2004 Workshop on Core Ontologies in Ontology Engineering*, 2004, vol. 8, pp. 1–14.

[51] I. Herman, "UK Government Moves to Put Data on the Web | Semantic Web Activity News," *W3C Semantic Web*, 10-Jun-2009. .

[52] T. Vitvar, A. Mocan, and V. Peristeras, "Pan-european e-government services on the semantic web services," *ResearchGate*, Jan. 2006.

[53] M. Fischer and C. Kam, "PM4D Final Report," Stanford University, Stanford, CA, Technical Report 143, Oct. 2002.

[54] E. Alreshidi, M. Mourshed, and Y. Rezgui, "Factors for effective BIM governance," *Journal of Building Engineering*, vol. 10, pp. 89–101, Mar. 2017.

[55] Z. Ma, N. Lu, and S. Wu, "Identification and representation of information resources for construction firms," *Advanced Engineering Informatics*, vol. 25, no. 4, pp. 612–624, Oct. 2011.

[56] W. Tizani and M. J. Mawdesley, "Advances and challenges in computing in civil and building engineering," *Advanced Engineering Informatics*, vol. 25, no. 4, pp. 569–572, Oct. 2011.

[57] Jason Underwood and Umit Isikdag, "Emerging technologies for BIM 2.0," *Construction Innovation*, vol. 11, no. 3, pp. 252–258, Jul. 2011.

[58] J. Beetz, J. van Leeuwen, and B. de Vries, "IfcOWL: A case of transforming EXPRESS schemas into ontologies," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 23, no. 01, p. 89, Feb. 2009.

[59] "ifcOWL — Welcome to buildingSMART-Tech.org," *Building Smart Future*, 30-Jan-2017. [Online]. Available: http://www.buildingsmart-tech.org/future/linked-data/linked-data. [Accessed: 30-Jan-2017].

[60] L. Zhang and R. R. Issa, "Ontology-Based Partial Building Information Model Extraction," *Journal of Computing in Civil Engineering*, vol. 27, no. 6, pp. 576–584, 2012.

[61] Ebrahim Karan, Javier Irizarry, and John Haymaker, "Generating IFC models from heterogeneous data using semantic web," *Construction Innovation*, vol. 15, no. 2, pp. 219–235, Mar. 2015.

[62] G. Gao, Y.-S. Liu, P. Lin, M. Wang, M. Gu, and J.-H. Yong, "BIMTag: Concept-based automatic semantic annotation of online BIM product resources," *Advanced Engineering Informatics*, no. (In Press), 2015.

[63] M. Asfand-e-yar, A. Kučera, and T. Pitner, "Semantic Web technology for Building Information Model," in *2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA)*, 2014, pp. 109–116.

[64] C. Mignard and C. Nicolle, "Merging BIM and GIS using ontologies application to urban facility management in ACTIVe3D," *Computers in Industry*, vol. 65, no. 9, pp. 1276–1290, Dec. 2014.

[65] C. Eastman, J. Lee, Y. Jeong, and J. Lee, "Automatic rule-based checking of building designs," *Automation in Construction*, vol. 18, no. 8, pp. 1011–1033, Dec. 2009.

[66] D.-Y. Cheng, T.-C. Chao, C.-C. Lo, and C.-H. Chen, "Research of Ontology and Semantic Web Apply for Building Information Model," in *Proceedings of the 2013 IEEE 10th International Conference on e-Business Engineering*, Washington, DC, USA, 2013, pp. 358–363.

[67] P. Pauwels *et al.*, "A semantic rule checking environment for building performance checking," *Automation in Construction*, vol. 20, no. 5, pp. 506–518, Aug. 2011.

[68] S. Harispe, D. Sánchez, S. Ranwez, S. Janaqi, and J. Montmain, "A framework for unifying ontology-based semantic similarity measures: A study in the biomedical domain," *Journal of Biomedical Informatics*, vol. 48, pp. 38–53, Apr. 2014.

[69] G. Aa. Grimnes, P. Edwards, and A. Preece, "Instance Based Clustering of Semantic Web Resources," in *The Semantic Web: Research and Applications*, S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, Eds. Springer Berlin Heidelberg, 2008, pp. 303–317.

[70] T. Pedersen, S. V. S. Pakhomov, S. Patwardhan, and C. G. Chute, "Measures of semantic similarity and relatedness in the biomedical domain," *Journal of Biomedical Informatics*, vol. 40, no. 3, pp. 288–299, Jun. 2007.

[71] A. Maedche and V. Zacharias, "Clustering Ontology-Based Metadata in the Semantic Web," in *Principles of Data Mining and Knowledge Discovery*, T. Elomaa, H. Mannila, and H. Toivonen, Eds. Springer Berlin Heidelberg, 2002, pp. 348–360.

[72] T. Li, S. Ma, and M. Ogihara, "Entropy-based Criterion in Categorical Clustering," in *Proceedings of the Twenty-First International Conference on Machine Learning*, New York, NY, USA, 2004, p. 68–.

[73] J. M. Santos and F. Morais, "Evaluating Entropic Based Clustering Algorithms on Biomedical Data," in *2013 12th Mexican International Conference on Artificial Intelligence (MICAI)*, 2013, pp. 194–199.

[74] D. Barbará, Y. Li, and J. Couto, "COOLCAT: An Entropy-based Algorithm for Categorical Clustering," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, New York, NY, USA, 2002, pp. 582–589.

[75] J. G. Cleary and L. E. Trigg, "K*: An Instance-based Learner Using an Entropic Distance Measure," in *Proceedings of the 12th International Conference on Machine Learning*, 1995, pp. 108–114.

[76] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, Jan. 2003.

[77] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF*IDF, LSI and multi-words for text classification," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758–2765, Mar. 2011.

[78] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," *Journal of Documentation*, vol. 60, no. 5, pp. 503–520, Oct. 2004.

[79] J. Ramos, "Using TF-IDF to determine word relevance in document queries," *ResearchGate*, Jan. 2003.

[80] D. Pullwitt, “Integrating Contextual Information to Enhance SOM-based Text Document Clustering,” *Neural Netw.*, vol. 15, no. 8–9, pp. 1099–1106, Oct. 2002.

[81] dotNetRDF Project, “dotNetRDF.” [Online]. Available: https://dotnetrdf.github.io/. [Accessed: 23-Jan-2017].

[82] “Apache Jena - Fuseki: serving RDF data over HTTP.” [Online]. Available: https://jena.apache.org/documentation/serving_data/. [Accessed: 23-Jan-2017].

[83] “Stardog: Enterprise Data Unification with Smart Graphs.” [Online]. Available: http://stardog.com/. [Accessed: 23-Jan-2017].

[84] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd edition. Burlington, MA: Morgan Kaufmann, 2011.

[85] S. Basu and R. J. Mooney, “Comparing and Unifying Search-Based and Similarity-Based Approaches to Semi-Supervised Clustering,” in *Proceedings of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003, pp. 42–49.

[86] B. Peralta, A. Caro, and A. Soto, “A proposal for supervised clustering with Dirichlet Process using labels,” *Pattern Recognition Letters*, vol. 80, pp. 52–57, Sep. 2016.

[87] Z. S. Syed, T. Finin, and A. Joshi, “Wikipedia as an Ontology for Describing Documents.,” in *ICWSM*, 2008.

[88] S. Azhar, M. Khalfan, and T. Maqsood, “Building information modelling (BIM): now and beyond,” *Australasian Journal of Construction Economics and Building*, vol. 12, no. 4, p. 15, Dec. 2012.

[89] D. Kent and B. Becerik-Gerber, “Understanding Construction Industry Experience and Attitudes toward Integrated Project Delivery,” *J. Constr. Eng. Manage.*, vol. 136, no. 8, pp. 815–825, Jan. 2010.

[90] S. Keenliside, “Comparative analysis of existing building information modelling (BIM) guides,” Jun. 2015.

[91] J. Song, C. T. Haas, C. Caldas, E. Ergen, and B. Akinci, “Automating the task of tracking the delivery and receipt of fabricated pipe spools in industrial projects,” *Automation in Construction*, vol. 15, no. 2, pp. 166–177, Mar. 2006.

[92] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, “Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques,” *Automation in Construction*, vol. 19, no. 7, pp. 829–843, Nov. 2010.

[93] P. Axelsson, “Processing of laser scanner data—algorithms and applications,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 2–3, pp. 138–147, Jul. 1999.

[94] I. Brilakis *et al.*, “Toward automated generation of parametric BIMs based on hybrid video and laser scanning data,” *Advanced Engineering Informatics*, vol. 24, no. 4, pp. 456–465, Nov. 2010.

[95] L. Linsen, *Point cloud representation*. Univ., Fak. für Informatik, Bibliothek, 2001.

[96] H.-G. Maas and G. Vosselman, “Two algorithms for extracting building models from raw laser altimetry data,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 2–3, pp. 153–163, Jul. 1999.

[97] X. Xiong, A. Adan, B. Akinci, and D. Huber, "Automatic creation of semantically rich 3D building models from laser scanner data," *Automation in Construction*, vol. 31, pp. 325–337, May 2013.

[98] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D Point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, Nov. 2008.

[99] P. Benkő, R. R. Martin, and T. Várady, "Algorithms for reverse engineering boundary representation models," *Computer-Aided Design*, vol. 33, no. 11, pp. 839–851, Sep. 2001.

[100] T. Várady, R. R. Martin, and J. Cox, "Reverse engineering of geometric models—an introduction," *Computer-Aided Design*, vol. 29, no. 4, pp. 255–268, Apr. 1997.

[101] H. Woo, E. Kang, S. Wang, and K. H. Lee, "A new segmentation method for point cloud data," *International Journal of Machine Tools and Manufacture*, vol. 42, no. 2, pp. 167–178, Jan. 2002.

[102] J. C. Carr *et al.*, "Reconstruction and Representation of 3D Objects with Radial Basis Functions," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2001, pp. 67–76.

[103] R. Fabio and others, "From point cloud to surface: the modeling and visualization problem," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, no. 5, p. W10, 2003.

[104] A. Gruen and D. Akca, "Least squares 3D surface and curve matching," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 3, pp. 151–174, May 2005.

[105] M. Pauly, R. Keiser, L. P. Kobbelt, and M. Gross, "Shape Modeling with Point-sampled Geometry," in *ACM SIGGRAPH 2003 Papers*, New York, NY, USA, 2003, pp. 641–650.

[106] Y.-F. Chang and S.-G. Shih, "BIM-based Computer-Aided Architectural Design," *Computer-Aided Design and Applications*, vol. 10, no. 1, pp. 97–109, Jan. 2013.

[107] P. Brown, "CAD: Do Computers Aid the Design Process After All?," *Intersect: The Stanford Journal of Science, Technology and Society*, vol. 2, no. 1, pp. 52–66, 2009.

[108] X. Ye, W. Peng, Z. Chen, and Y.-Y. Cai, "Today's students, tomorrow's engineers: an industrial perspective on CAD education," *Computer-Aided Design*, vol. 36, no. 14, pp. 1451–1460, Dec. 2004.

[109] J. Pan and C. J. Anumba, "Semantic-discovery of construction project files," *Tsinghua Science and Technology*, vol. 13, no. S1, pp. 305–310, Oct. 2008.

[110] T. Funkhouser *et al.*, "A Search Engine for 3D Models," *ACM Trans. Graph.*, vol. 22, no. 1, pp. 83–105, Jan. 2003.

[111] A. Cardone, S. K. Gupta, and M. Karnik, "A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications," *Journal of Computing and Information Science in Engineering*, vol. 3, no. 2, p. 109, 2003.

[112] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani, "Three-dimensional shape searching: state-of-the-art review and future trends," *Computer-Aided Design*, vol. 37, no. 5, pp. 509–530, Apr. 2005.

[113] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools and Applications*, vol. 39, no. 3, pp. 441–471, Sep. 2008.

[114] M. C. Booth and E. T. Rolls, "View-invariant representations of familiar objects by neurons in the inferior temporal visual cortex.," *Cereb. Cortex*, vol. 8, no. 6, pp. 510–523, Sep. 1998.

[115] J. H. R. Maunsell, "The Brain's Visual World: Representation of Visual Targets in Cerebral Cortex," *Science*, vol. 270, no. 5237, pp. 764–769, Nov. 1995.

[116] Ø. Due Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition-A survey," *Pattern Recognition*, vol. 29, no. 4, pp. 641–662, Apr. 1996.

[117] K. Ishii and R. A. Miller, "Design Representation for Manufacturability Evaluation in CAD: Beyond Feature-based Design," *Computers in Engineering*, no. 1, pp. 337–43, 1992.

[118] M. Hebert, K. Ikeuchi, and H. Delingette, "A spherical representation for recognition of free-form surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 681–690, Jul. 1995.

[119] W. Li, Z. Yin, Y. Huang, and Y. Xiong, "Automatic registration for 3D shapes using hybrid dimensionality-reduction shape descriptions," *Pattern Recognition*, vol. 44, no. 12, pp. 2926–2943, Dec. 2011.

[120] S. M. Yamany and A. A. Farag, "Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1105–1120, Aug. 2002.

[121] C. Di Ruberto, "Recognition of shapes by attributed skeletal graphs," *Pattern Recognition*, vol. 37, no. 1, pp. 21–31, Jan. 2004.

[122] M. El-Mehalawi and R. Allen Miller, "A database system of mechanical components based on geometric and topological similarity. Part I: representation," *Computer-Aided Design*, vol. 35, no. 1, pp. 83–94, Jan. 2003.

[123] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2001, pp. 203–212.

[124] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, Jan. 2004.

[125] C. Zhang and T. Chen, "Efficient feature extraction for 2D/3D objects in mesh representation," presented at the 2001 International Conference on Image Processing, 2001. Proceedings, 2001, vol. 3, pp. 935–938 vol.3.

[126] M. El-Mehalawi and R. Allen Miller, "A database system of mechanical components based on geometric and topological similarity. Part II: indexing, retrieval, matching, and similarity assessment," *Computer-Aided Design*, vol. 35, no. 1, pp. 95–105, Jan. 2003.

[127] S. Z. Li, "Shape matching based on invariants," 1998.

[128] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3D models with shape distributions," in *Shape Modeling and Applications, SMI 2001 International Conference on.*, 2001, pp. 154–166.

[129] D. M. Squire and T. M. Caelli, "Invariance Signatures: Characterizing Contours by Their Departures from Invariance," *Computer Vision and Image Understanding*, vol. 77, no. 3, pp. 284–316, Mar. 2000.

[130] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer Science & Business Media, 2013.

[131] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, 2005, vol. 1, pp. 26–33 vol. 1.

[132] C. T. Zahn and R. Z. Roskies, "Fourier Descriptors for Plane Closed Curves," *IEEE Transactions on Computers*, vol. C-21, no. 3, pp. 269–281, Mar. 1972.

[133] S. KAPARTHI and N. C. SURESH, "A neural network system for shape-based classification and coding of rotational parts," *International Journal of Production Research*, vol. 29, no. 9, pp. 1771–1784, Sep. 1991.

[134] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Sixth International Conference on Computer Vision, 1998*, 1998, pp. 59–66.

[135] S. K. Chang, *Data Structures and Algorithms*. World Scientific, 2003.

[136] C. Y. Ip, D. Lapadat, L. Sieger, and W. C. Regli, "Using Shape Distributions to Compare Solid Models," in *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications*, New York, NY, USA, 2002, pp. 273–280.

[137] S.-H. Cha, "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions," *INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES*, vol. 1, no. 4, pp. 300–307, 2007.

[138] M. Ali, Y. Mohamed, H. Taghaddos, and R. Hermann, "BIM obstacles in industrial projects: a contractor perspective," in *Proceedings of ICSC15: The Canadian Society for Civil Engineering 5th International/11th Construction Specialty Conference*, Vancouver, Canada, 2015.

[139] A. A. Aibinu and T. Pasco, "The accuracy of pre-tender building cost estimates in Australia," *Construction Management and Economics*, vol. 26, no. 12, pp. 1257–1269, Dec. 2008.

[140] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Semantic Matching of Web Services Capabilities," in *The Semantic Web — ISWC 2002*, I. Horrocks and J. Hendler, Eds. Springer Berlin Heidelberg, 2002, pp. 333–347.

[141] K. W. Chau, M. Anson, and J. P. Zhang, "Four-Dimensional Visualization of Construction Scheduling and Site Utilization," *Journal of Construction Engineering and Management*, pp. 598–606, 2004.

[142] B. Koo and M. Fischer, "Feasibility Study of 4D CAD in Commercial Construction," *Journal of Construction Engineering and Management*, vol. 126, no. 4, pp. 251–260, Jul. 2000.

[143] K. W. Chau, M. Anson, and J. P. Zhang, "Implementation of visualization as planning and scheduling tool in construction," *Building and Environment*, vol. 38, no. 5, pp. 713–719, May 2003.

[144] V. R. Kamat, J. C. Martinez, M. Fischer, M. Golparvar-Fard, and S. Savarese, "Research in Visualization Techniques for Field Construction," *Journal of*

Construction Engineering and Management, vol. 137, no. 10, pp. 853–862, Oct. 2011.

[145] M. Bilal *et al.*, "Big Data in the construction industry: A review of present status, opportunities, and future trends," *Advanced Engineering Informatics*, vol. 30, no. 3, pp. 500–521, Aug. 2016.

[146] D. Heesom and L. Mahdjoubi, "Trends of 4D CAD applications for construction planning," *Construction Management and Economics*, vol. 22, no. 2, pp. 171–182, Feb. 2004.

[147] V. R. Kamat and J. Martinez, "Visualizing Simulated Construction Operations in 3D," *Journal of Computing in Civil Engineering*, vol. 15, no. 4, pp. 329–337, 2001.

[148] S. M. AbouRizk, "Role of Simulation in Construction Engineering and Management," *Journal of Construction Engineering and Management*, vol. 136, no. 10, pp. 1140–1153, 2010.

[149] Y. Zhang, S. M. AbouRizk, H. Xie, and E. Moghani, "Design and Implementation of Loose-Coupling Visualization Components in a Distributed Construction Simulation Environment with HLA," *Journal of Computing in Civil Engineering*, vol. 26, no. 2, pp. 248–258, 2012.

[150] S. Lee, D. Nikolic, J. I. Messner, and C. J. Anumba, "The Development of the Virtual Construction Simulator 3: An Interactive Simulation Environment for Construction Management Education," in *Computing in Civil Engineering (2011)*, American Society of Civil Engineers, 2011, pp. 454–461.

[151] S. Jain and C. R. McLean, "Integrated simulation and gaming architecture for incident management training," in *Simulation Conference, 2005 Proceedings of the Winter*, 2005, pp. 904–913.

[152] M. Al-Hussein, M. Athar Niaz, H. Yu, and H. Kim, "Integrating 3D visualization and simulation for tower crane operations on construction sites," *Automation in Construction*, vol. 15, no. 5, pp. 554–562, Sep. 2006.

[153] S.-C. Kang, H.-L. chi, and E. Miranda, "Three-Dimensional Simulation and Visualization of Crane Assisted Construction Erection Processes," *Journal of Computing in Civil Engineering*, vol. 23, no. 6, pp. 363–371, 2009.

[154] W. Ribarsky, J. Bolter, A. O. den Bosch, and R. van Teylingen, "Visualization and analysis using virtual reality," *IEEE Computer Graphics and Applications*, vol. 14, no. 1, pp. 10–12, Jan. 1994.

[155] A. H. Behzadan and V. R. Kamat, "Visualization of Construction Graphics in Outdoor Augmented Reality," in *Proceedings of the 37th Conference on Winter Simulation*, Orlando, Florida, 2005, pp. 1914–1920.

[156] Y. Zhou, L. Y. Ding, and L. J. Chen, "Application of 4D visualization technology for safety management in metro construction," *Automation in Construction*, vol. 34, pp. 25–36, Sep. 2013.

[157] T. Cheng and J. Teizer, "Real-time resource location data collection and visualization technology for construction safety and activity monitoring applications," *Automation in Construction*, vol. 34, pp. 3–15, Sep. 2013.

[158] K. A. Liapi, "4D visualization of highway construction projects," in *Proceedings on Seventh International Conference on Information Visualization, 2003. IV 2003.*, 2003, pp. 639–644.

[159] M. L. Pack, “Visualization in Transportation: Challenges and Opportunities for Everyone,” *IEEE Computer Graphics and Applications*, vol. 30, no. 4, pp. 90–96, Jul. 2010.

[160] “buildingSMART,” *buildingSMART*, 2016. [Online]. Available: http://buildingsmart.org/. [Accessed: 23-Nov-2016].

[161] W. Terkaj and A. Šojić, “Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology,” *Automation in Construction*, vol. 57, pp. 188–201, Sep. 2015.

[162] P. Pauwels and W. Terkaj, “EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology,” *Automation in Construction*, vol. 63, pp. 100–133, Mar. 2016.

[163] International Organisation for Standardisation (ISO), *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*, Second edition. International Organisation for Standardisation (ISO), 2004.

[164] P. Pauwels, W. Terkaj, T. Krijnen, and J. Beetz, “Coping with lists in the ifcOWL ontology,” in *22nd EG-ICE International Workshop, Proceedings*, 2015, pp. 113–122.

[165] R. Barbau *et al.*, “OntoSTEP: Enriching product model data using ontologies,” *Computer-Aided Design*, vol. 44, no. 6, pp. 575–590, Jun. 2012.

[166] M. I. Sarigecili, U. Roy, and S. Rachuri, “Interpreting the semantics of GD&T specifications of a product for tolerance analysis,” *Computer-Aided Design*, vol. 47, pp. 72–84, Feb. 2014.

[167] W. Lu, Y. Qin, X. Liu, M. Huang, L. Zhou, and X. Jiang, “Enriching the semantics of variational geometric constraint data with ontology,” *Computer-Aided Design*, vol. 63, pp. 72–85, Jun. 2015.

[168] H. Schevers and R. Drogemuller, “Converting the Industry Foundation Classes to the Web Ontology Language,” in *2005 First International Conference on Semantics, Knowledge and Grid*, 2005, pp. 73–73.

[169] T. M. de Farias, A. Roxin, and C. Nicolle, “IfcWoD, Semantically Adapting IFC Model Relations into OWL Properties,” *arXiv:1511.03897 [cs]*, Nov. 2015.

[170] P. Pauwels and A. Roxin, “SimpleBIM: from full ifcOWL graphs to simplified building graphs,” in *11th European Conference on Product and Process Modelling*, 2016, pp. 11–18.

[171] A. Sheth and M. Perry, “Traveling the semantic web through space, time, and theme,” *IEEE Internet Computing*, vol. 12, no. 2, pp. 81–86, 2008.

[172] I. Davis, “Representing Time in RDF Part 1,” *Internet Alchemy*, 2009. [Online]. Available: http://blog.iandavis.com/2009/08/representing-time-in-rdf-part-1/. [Accessed: 17-Nov-2016].

[173] C. Gutierrez, C. A. Hurtado, and A. Vaisman, “Introducing Time into RDF,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 2, pp. 207–218, Feb. 2007.

[174] C. Gutierrez, C. Hurtado, and A. Vaisman, “Temporal RDF,” in *SpringerLink*, Springer Berlin Heidelberg, 2005, pp. 93–107.

[175] N. Noy and A. Rector, "Defining N-ary Relations on the Semantic Web," 2006. [Online]. Available: https://www.w3.org/TR/swbp-n-aryRelations/. [Accessed: 17-Nov-2016].

[176] J. Hebeler, M. Fisher, R. Blace, A. Perez-Lopez, and M. Dean, *Semantic Web Programming*, 1 edition. Wiley, 2009.

[177] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia," *Artificial Intelligence*, vol. 194, pp. 28–61, Jan. 2013.

[178] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler, "Named graphs," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 3, no. 4, pp. 247–267, Dec. 2005.

[179] J. Tappolet and A. Bernstein, "Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL," in *SpringerLink*, Springer Berlin Heidelberg, 2009, pp. 308–322.

[180] O. Udrea, D. R. Recupero, and V. S. Subrahmanian, "Annotated RDF," *ACM Trans. Comput. Logic*, vol. 11, no. 2, p. 10:1–10:41, Jan. 2010.

[181] C. Welty and R. Fikes, "A Reusable Ontology for Fluents in OWL," in *Proceedings of the 2006 Conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, Amsterdam, The Netherlands, The Netherlands, 2006, pp. 226–236.

[182] S. Cox and C. Little, "Time Ontology in OWL," 2016. [Online]. Available: https://www.w3.org/TR/owl-time/. [Accessed: 17-Nov-2016].

[183] "IEEE Standard for Modeling and Simulation (M\&S) High Level Architecture (HLA)-- Federate Interface Specification," *IEEE Std 1516.1-2010 (Revision of IEEE Std 1516.1-2000)*, p. 1,378, Aug. 2010.

[184] "IEEE Standard for Modeling and Simulation (M\&S) High Level Architecture (HLA)– Object Model Template (OMT) Specification," *IEEE Std 1516.2-2010 (Revision of IEEE Std 1516.2-2000)*, pp. 1–110, Aug. 2010.

[185] A. ElNimr and Y. Mohamed, "Loosely coupled visualization of industrial construction simulation using a gaming engine," in *Simulation Conference (WSC), Proceedings of the 2011 Winter*, 2011, pp. 3577–3587.

[186] A. ElNimr and Y. Mohammed, "A Simulation Driven Visualization Framework for Construction Operations: Development and Application," in *Construction Research Congress 2010*, American Society of Civil Engineers, 2010, pp. 257–266.

[187] S. Alkass, K. El-Moslmani, and M. AlHussein, "A computer model for selecting equipment for earthmoving operations using queuing theory," *CIB REPORT*, vol. 284, p. 1, 2003.

[188] M. Marzouk and O. Moselhi, "Object-oriented Simulation Model for Earthmoving Operations," *Journal of Construction Engineering and Management*, vol. 129, no. 2, pp. 173–181, 2003.

[189] Y. Mohamed and M. Ali, "A Simplified Online Solution for Simulation-Based Optimization of Earthmoving Operations," in *The 30th International Symposium on Automation and Robotics in Construction (ISARC 2013)*, Montreal, Canada, 2013.

[190] M. Ali and Y. Mohamed, “Development of a model-based DSS for earth moving operations,” in *Al-Azhar Engineering twelfth international conference (AEIC 2012)*, Cairo, Egypt, 2012.

[191] G. Kannan, L. Schmitz, and C. Larsen, “An Industry Perspective on the Role of Equipment-based Earthmoving Simulation,” in *Proceedings of the 32Nd Conference on Winter Simulation*, San Diego, CA, USA, 2000, pp. 1945–1952.

[192] S. Smith, J. Osborne, and M. Forde, “Analysis of Earth-Moving Systems Using Discrete-Event Simulation,” *Journal of Construction Engineering and Management*, vol. 121, no. 4, pp. 388–396, 1995.

[193] A. Montaser, M. Ibrahim, and O. Moselhi, “Adaptive Forecasting in Earthmoving Operation Using DES and Site Captured Data,” *Procedia Engineering*, vol. 85, pp. 377–384, 2014.

[194] S. M. AbouRizk and K. Mather, “Simplifying Simulation Modeling through Integration with 3D CAD,” *Journal of Construction Engineering and Management*, vol. 126, no. 6, pp. 475–483, 2000.

[195] F. Vahdatikhaki and A. Hammad, “Framework for near real-time simulation of earthmoving projects using location tracking technologies,” *Automation in Construction*, vol. 42, pp. 50–67, Jun. 2014.

[196] M. Ali, M. Fagiar, Y. Mohamed, and S. M. AbouRizk, “Beyond classic models—design and development of a comprehensive earthmoving simulator,” in *14th International Conference on Construction Applications of Virtual Reality in Construction and conference on Islamic Architecture*, Sharjah, UAE, 2014.

[197] E. Sorensen and M. I. Mihailesc, “Model-View-ViewModel (MVVM) Design Pattern using Windows Presentation Foundation (WPF) Technology,” *MegaByte Journal*, 2010.

[198] A. Kozminski, “Windows Presentation Foundation (WPF) technology meets the challenges of operator interface design in automatic test systems,” in *AUTOTESTCON, 2012 IEEE*, 2012, pp. 80–83.

[199] “SEE • Simulation Exploration Experience,” *Simulation Exploration Experience (SEE)*. [Online]. Available: http://www.exploresim.com. [Accessed: 09-Mar-2017].

[200] M. Fagiar, M. Ali, and Y. Mohamed, “Capitalizing on visualizing complex systems using distributed simulation approach,” in *15th International Conference on Construction Applications of Virtual Reality in Construction*, Banff, Canada, 2015.

[201] E. Crues, “Simulation Smackdown Environment Federate.” NASA Johnson Space Center, Apr-2013.

[202] M. Conroy *et al.*, “Distributed Observer Network,” NASA Tech Brief KSC-13081, Jun. 2010.

[203] W. Little and R. Mazzone, “The NASA Distributed Observer Network,” in *Proceedings of the 2008 Summer Computer Simulation Conference*, Vista, CA, 2008, p. 56:1–56:6.

[204] M. Grüninger and M. S. Fox, “Methodology for the Design and Evaluation of Ontologies,” 1995.

[205] H. Kondylakis and D. Plexousakis, “Ontology evolution without tears,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 19, pp. 42–58, Mar. 2013.

[206] "Dynamo BIM." [Online]. Available: http://dynamobim.org/. [Accessed: 15-Mar-2017].

[207] P. Janssen and K. W. Chen, "Visual Dataflow Modelling," in *Proceedings of CAAD Futures*, 2011, pp. 801–816.

[208] M. R. Asl, M. Bergin, A. Menter, and W. Yan, "BIM-based parametric building energy performance multi-objective optimization," *Education and Research in Computer Aided Architectural Design in Europe*, vol. 32, pp. 1–10, 2014.

[209] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, 2000.

[210] J. J. Shi, "Activity-Based Construction (ABC) Modeling and Simulation Method," *Journal of Construction Engineering and Management*, vol. 125, no. 5, pp. 354–360, 1999.

[211] D. Zhong, J. Li, H. Zhu, and L. Song, "Geographic Information System-Based Visual Simulation Methodology and Its Application in Concrete Dam Construction Processes," *Journal of Construction Engineering and Management*, vol. 130, no. 5, pp. 742–750, 2004.

[212] S. M. AbouRizk and D. S. Peter, "Application of Computer Simulation in Resolving Construction Disputes," *Journal of Construction Engineering and Management*, vol. 119, no. 2, pp. 355–373, 1993.

[213] P. Reddy, J. Ghaboussi, and H. Neil M., "Simulation of Construction of Cable-Stayed Bridges," *Journal of Bridge Engineering*, vol. 4, no. 4, pp. 249–257, 1999.

[214] D. F. McCahill and B. Leonhard E., "Resource-Oriented Modeling and Simulation in Construction," *Journal of Construction Engineering and Management*, vol. 119, no. 3, pp. 590–606, 1993.

[215] V. R. Kamat and J. C. Martinez, "Validating Complex Construction Simulation Models Using 3D Visualization," *Systems Analysis Modelling Simulation*, vol. 43, no. 4, pp. 455–467, Apr. 2003.

[216] S. AbouRizk and Y. Mohamed, "Simphony-an integrated environment for construction simulation," in *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, 2000, vol. 2, pp. 1907–1914 vol.2.

[217] D. Hajjar and S. AbouRizk, "Simphony: An Environment for Building Special Purpose Construction Simulation Tools," in *Proceedings of the 31st Conference on Winter Simulation: Simulation—a Bridge to the Future - Volume 2*, New York, NY, USA, 1999, pp. 998–1006.

[218] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi, "RDF123: From Spreadsheets to RDF," in *The Semantic Web - ISWC 2008*, 2008, pp. 451–466.

[219] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi, "RDF123: a mechanism to transform spreadsheets to RDF," in *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI 2006). AAAI Press, Menlo Park*, 2006.

[220] "NASA | DON." [Online]. Available: https://public.ksc.nasa.gov/DON. [Accessed: 15-Mar-2017].

[221] R. A. Mazzone and M. P. Conroy, "Data Presentation and Visualization (DPV) Interface Control Document," Apr. 2015.

Appendix A

This code converts BIM model to RDF format

```
using Autodesk.Navisworks.Api;
using Autodesk.Navisworks.Api.Plugins;
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Linq;
using System.Reflection;
using System.Text;
using System.Windows.Forms;
using log4net;
using log4net.Appender;
using log4net.Config;
using log4net.Layout;
using VDS.RDF;
using VDS.RDF.Ontology;
using VDS.RDF.Storage;
using VDS.RDF.Writing;

namespace Navis2Rdf
{
    [Autodesk.Navisworks.Api.Plugins.Plugin("Navis2RDF", "MAli", ToolTip =
        "Save the selected model items properties to RDF triples, " +
        "use the config file along the dll to configure the location",
        DisplayName = "Export RDF")]
    public class Navis2Rdf : AddInPlugin
    {
        private static readonly ILog Log = LogManager.GetLogger(MethodBase.GetCurrentMethod().DeclaringType);
        public override int Execute(params string[] parameters)
        {
            try
            {
                if (!LogManager.GetRepository().Configured)
                {
                    var layout = new PatternLayout("%-4timestamp %date [%thread] %-5level %logger - %message%newline");
    var appender = new RollingFileAppender
    {
        File = @"D:\BIM_RDF_LOG\logger.log",
        Layout = layout,
        RollingStyle = RollingFileAppender.RollingMode.Size,
        MaxSizeRollBackups = 10,
        MaximumFileSize = "50000KB",
        LockingModel = new FileAppender.MinimalLock(),
        AppendToFile = false
    };
    layout.ActivateOptions();
    appender.ActivateOptions();
    BasicConfigurator.Configure(appender);
}
Log.Info("Retrieving items...");
var selectionModelItems = new ModelItemCollection(
    Autodesk.Navisworks.Api.Application.ActiveDocument.CurrentSelection.SelectedItems);
Log.Info($"Processing {selectionModelItems.Count} item(s) in total");
if (selectionModelItems.Count == 0)
{
    MessageBox.Show("Please select items first", "Empty selection",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return 0;
}
Log.Info("Creating the graph");
var g = new OntologyGraph();
g.NamespaceMap.AddNamespace("", new Uri("http://www.mali.ca#"));
var baseNode = g.CreateUriNode(UriFactory.Create("http://www.mali.ca"));
var modelItemClass = g.CreateUriNode(":ModelItem");
var signatureNode = g.CreateUriNode(":predicateSignature");
var rdfType = g.CreateUriNode("rdf:type");
var rdfsSubClass = g.CreateUriNode("rdfs:subClassOf");
var rdfsLabel = g.CreateUriNode("rdfs:Label");
var rdfsRange = g.CreateUriNode("rdfs:range");
var owlOntology = g.CreateUriNode("owl:Ontology");
var owlVersionInfo = g.CreateUriNode("owl:versionInfo");
var owlClass = g.CreateUriNode("owl:Class");
var owlThing = g.CreateUriNode("owl:Thing");
var owlDataTypeProperty = g.CreateUriNode("owl:DatatypeProperty");

var xsdString = g.CreateUriNode("xsd:string");
var versionValue = g.CreateLiteralNode(
    "V1.0 Created by Mostafa Ali <engabdomostafa@gmail.com>");

g.Assert(new Triple(baseNode, rdfType, owlOntology));
g.Assert(new Triple(baseNode, owlVersionInfo, versionValue));
g.Assert(new Triple(modelItemClass, rdfType, owlClass));
g.Assert(new Triple(modelItemClass, rdfsSubClass, owlThing));
var appConfig = ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
var filePath = appConfig.AppSettings.Settings["RDF_File_Path"].Value;
var fusekiServerPath = appConfig.AppSettings.Settings["Fuseki_Server_Address"].Value;
var stardogServerPath = appConfig.AppSettings.Settings["StarDog_Server_Address"].Value;
var stardogServerDb = appConfig.AppSettings.Settings["StarDog_Server_DB"].Value;
var stardogServerUser = appConfig.AppSettings.Settings["StarDog_Server_UserName"].Value;
var stardogServerPassword = appConfig.AppSettings.Settings["StarDog_Server_Password"].Value;
FusekiConnector fuseki = null;
StardogConnector stardog = null;

if (string.IsNullOrEmpty(filePath) && string.IsNullOrEmpty(fusekiServerPath)
    && string.IsNullOrEmpty(stardogServerPath))
{
    Log.Warn("No output has been specified, please add output to config file");
    return 0;
}

if (!string.IsNullOrEmpty(filePath))
{
    Log.Info("Writing the graph to the file...");
    var writer = new CompressingTurtleWriter();
    writer.Save(g, filePath);
}

if (!string.IsNullOrEmpty(fusekiServerPath))
{
    Log.Info("Writing the graph to Fuseki Server...");
    fuseki = new FusekiConnector(fusekiServerPath);
    fuseki.SaveGraph(g);
}

if (!string.IsNullOrEmpty(stardogServerPath))
{
    Log.Info("Writing the graph to StarDog Server...");
    stardog = new StardogConnector(stardogServerPath, stardogServerDb,
        stardogServerUser, stardogServerPassword);
    stardog.SaveGraph(g);
}

Log.Info("Iterating items...");
long i = 0;
var triples = new List<Triple>();

foreach (var item in selectionModelItems.DescendantsAndSelf.Where(x => x.HasGeometry))
{
    try
    {
        Log.Info($"Processing item #{++i}, name: {item.DisplayName}");

        var modelItemNode = g.CreateUriNode($":{Guid.NewGuid()}");
        triples.Add(new Triple(modelItemNode, rdfType, modelItemClass));
        var predicateSet = new SortedSet<string>();
        foreach (var oPc in item.GetUserFilteredPropertyCategories())
        {
            if (oPc.DisplayName.ToLower() == "material" || oPc.DisplayName.ToLower() == "timeliner") continue;
            foreach (var property in oPc.Properties)
            {
                var propertyName = RemoveSpecialCharacters(oPc.Name + property.Name + property.DisplayName);
                var predicate = g.CreateUriNode($":{propertyName}");
                if (predicateSet.Add(propertyName))
                {
                    triples.Add(new Triple(predicate, rdfType, owlDataTypeProperty));
                    triples.Add(new Triple(predicate, rdfsLabel,
                        g.CreateLiteralNode(property.DisplayName)));
                    triples.Add(new Triple(predicate, rdfsRange, xsdString));
                }
                var value = g.CreateLiteralNode(
                    RemoveSpecialCharacters(property.Value.ToString().Substring(
                        property.Value.ToString().LastIndexOf(':') + 1)));
                triples.Add(new Triple(modelItemNode, predicate, value));
            }
        }
        var predicateSignature = string.Join("", predicateSet).GetHashCode();
        triples.Add(new Triple(modelItemNode, signatureNode, predicateSignature.ToLiteral(g)));

        if (triples.Count > 10000)
        {
```

```
                            Log.Info($"Writing {triples.Count} triples to the triplestore...");
                            fuseki?.UpdateGraph(g.BaseUri, triples, null);
                            stardog?.UpdateGraph(g.BaseUri, triples, null);
                            triples.Clear();
                        }
                    }
                    catch (Exception ex)
                    {
                        Log.Error($"Exception within item source {ex.Source}, message {ex.Message}, inner exception {ex.InnerException} ");
                    }
                }
                Log.Info($"Writing {triples.Count} triples to the triplestore...");
                fuseki?.UpdateGraph(g.BaseUri, triples, null);
                stardog?.UpdateGraph(g.BaseUri, triples, null);
                Log.Info("Process complete");
            }
            catch (Exception ex)
            {
                Log.Error(
                    $"Exception source {ex.Source}, message {ex.Message}, inner exception {ex.InnerException} ");
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            return 0;
        }

        private static string RemoveSpecialCharacters(string str)
        {
            var sb = new StringBuilder(str.Length);
            foreach (var c in str.Where(c => (c >= '0' && c <= '9') || (c >= 'A' && c <= 'Z') ||
            (c >= 'a' && c <= 'z') || c == '.' || c == '_' || c == '-'))
            {
                sb.Append(c);
            }
            return sb.ToString();
        }
    }
}
```

Appendix B

The clustering code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Xml;
using VDS.RDF.Nodes;
using VDS.RDF.Query;

namespace Grouping
{
  class Program
  {
    private delegate IEnumerable<PredicateEval> CalculateWeight(List<PredicateEval> predicateEvals, int count);
    //Define the SPARQL endpoint URL
    private const string QueryUrl = "http://localhost:3030/Demo1/query";
    private static readonly NetworkCredential Credentials = new NetworkCredential
    {
      UserName = "admin",
      Password = "admin"
    };
    private static readonly SparqlRemoteEndpoint Endpoint = new SparqlRemoteEndpoint(new Uri(QueryUrl))
    {
      Credentials = Credentials,
      Timeout = int.MaxValue
    };
    private static int _groupCount;
    static void Main(string[] args)
    {
      var queryString = "PREFIX : <http://www.mali.ca/#>\r\n" +
                "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\r\n\r\n" +
                "SELECT ?object (COUNT(?object) AS ?count)\r\n" +
                "WHERE {\r\n" +
                "  ?subject rdf:type :ModelItem.\r\n" +
                  "  ?subject :predicateSignature ?object\r\n" +
                  "}\r\n" +
                  "Group by ?object\r\n" +
                  "Order by DESC(?count)";
var groupResults = Endpoint.QueryWithResultSet(queryString);
_groupCount = groupResults.Count;
if (_groupCount <= 0)
{
    Console.WriteLine("No groups found...");
    Console.WriteLine("Press any key to exit...");
    Console.ReadKey();
    return;
}
Console.WriteLine($"We have {groupResults.Count} groups.");
var dominantPred = new List<PredicateEval>();
var groupNo = 0;
var ignoredAttributes = new List<string> { };
foreach (var groupResult in groupResults)
{
    var predicateEvals = new List<PredicateEval>();
    queryString = "PREFIX : <http://www.mali.ca/#>\r\n" +
                  "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\r\n\r\n" +
                  "SELECT DISTINCT ?predicate\r\n" +
                  "WHERE {\r\n" +
                  "  ?subject rdf:type :ModelItem.\r\n" +
                  "  ?subject ?predicate ?object.\r\n" +
                  $" ?subject :predicateSignature {groupResult[0].AsValuedNode().AsString()}\r\n" +
                  "FILTER (?predicate != <http://www.mali.ca/#predicateSignature>). \r\n" +
                  "FILTER (?predicate != <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>)\r\n}";
    var attributeResults = Endpoint.QueryWithResultSet(queryString);
    Console.WriteLine($"Processing group {++groupNo} out of {groupResults.Count}, it has {attributeResults.Count - 1} attributes");
    foreach (var attributeResult in attributeResults)
    {
        try
        {
            if (ignoredAttributes.Contains(attributeResult[0].AsValuedNode().AsString())) continue;
            Console.WriteLine($"Processing attribute {attributeResult[0]}");
            //If the attribute has many unique values, skip it
            queryString = "PREFIX : <http://www.mali.ca/#>\r\n" +
                          "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\r\n\r\n"+
                          "SELECT (COUNT(DISTINCT ?value) AS ?count)\r\n" +
                          "WHERE {\r\n  ?subject rdf:type :ModelItem.\r\n" +
                                $"  ?subject <{attributeResult[0]}> ?value\r\n" +
                                "}";
                    var entropyResult = Endpoint.QueryWithResultSet(queryString);
                    var valuesCount = entropyResult.Results.First()[0].AsValuedNode().AsInteger();
                    if (valuesCount > 5000)
                    {
                        ignoredAttributes.Add(attributeResult[0].AsValuedNode().AsString());
                        Console.WriteLine($"\tThis attribute has {valuesCount} unique values, it will be ignored.");
                        continue;
                    }
                    queryString = "PREFIX : <http://www.mali.ca/#>\r\n" +
                                "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\r\n\r\n" +
                                "SELECT (COUNT(?value) AS ?count)\r\n" +
                                "WHERE {\r\n" +
                                "  ?subject rdf:type :ModelItem.\r\n" +
                                $" ?subject :predicateSignature  {groupResult[0].AsValuedNode().AsString()} .\r\n" +
                                $" ?subject <{attributeResult[0]}> ?value\r\n}}\r\n" +
                                "Group by ?value";
                    entropyResult = Endpoint.QueryWithResultSet(queryString);
                    var entropyG = CalculateEntropy(entropyResult.Results.Select(a => a[0].AsValuedNode().AsInteger()).ToList());
                    queryString = "PREFIX : <http://www.mali.ca/#>\r\n" +
                                "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\r\n\r\n" +
                                "SELECT (COUNT(?value) AS ?count)\r\n" +
                                "WHERE {\r\n" +
                                "  ?subject rdf:type :ModelItem.\r\n" +
                                $" ?subject <{attributeResult[0]}> ?value\r\n}}\r\n" +
                                "Group by ?value";
                    entropyResult = Endpoint.QueryWithResultSet(queryString);
                    var entropyD = CalculateEntropy(entropyResult.Results.Select(a => a[0].AsValuedNode().AsInteger()).ToList());
                    predicateEvals.Add(new PredicateEval(attributeResult[0].AsValuedNode().AsString(),
groupResult[0].AsValuedNode().AsString(), entropyG, entropyD));
                }
                catch (Exception ex)
                {
                    Console.WriteLine(ex.Message);
                }
            }
            Console.WriteLine();
            //CalculateWeight cw = TfIdfEvalLessThanOne; //Trial 4
            //CalculateWeight cw = TfIdfEval;  //Trial 3
            //CalculateWeight cw = MaxEntropyDiffEval;  //Trial 1
            CalculateWeight cw = MaxEntropyDiffLessThanOneEval;  //Trial 2
```

```
    dominantPred.AddRange(cw(predicateEvals, 1));
}
var y =
    dominantPred.GroupBy(p => p.PredicateUri)
        .Select(group => new {PredicateName = group.Key, Count = group.Count()})
        .OrderByDescending(x => x.Count)
        .ToArray();
Console.WriteLine($"There are {y.Length} unique dominant attribute(s)");
foreach (var result in y)
{
    Console.WriteLine($"{result.PredicateName} appears {result.Count}");
}
Console.WriteLine("Do you want to write Navisworks selection files [Y/N]?");
var consoleKeyInfo = Console.ReadKey();
Console.WriteLine();
if (consoleKeyInfo.KeyChar == 'y' || consoleKeyInfo.KeyChar == 'Y')
{
    //Generate Naviswork xml files
    var xmlWriterSettings = new XmlWriterSettings
    {
        Indent = true,
        IndentChars = "\t"
    };
    Console.WriteLine("Do you want to include the initial group as a condition [Y/N]?");
    consoleKeyInfo = Console.ReadKey();
    Console.WriteLine();
    var addGroupCondition = consoleKeyInfo.KeyChar == 'y' || consoleKeyInfo.KeyChar == 'Y';
    foreach (IGrouping<string, PredicateEval> predicateEvals in dominantPred.GroupBy(p => p.GroupSignature))
    {
        foreach (PredicateEval predicateEval in predicateEvals)
        {
            //Retrieve all possible values
            queryString = "PREFIX : <http://www.mali.ca/#>\r\n\r\n" +
                          "SELECT DISTINCT ?value\r\n" +
                          "WHERE {" +
                          $"\r\n  ?subject <{predicateEval.PredicateUri}> ?value .\r\n" +
                          "}";
            foreach (var predicateValue in Endpoint.QueryWithResultSet(queryString))
            {
                var sb = new StringBuilder("PREFIX : <http://www.mali.ca/#>\r\n\r\n" +
                              "SELECT ?value\r\n" +
                              "WHERE {\r\n" +
                       "  \t?subject a :ModelItem.\r\n" +
                       "\t?subject :LcOaPropOverrideCatPCLPCLID ?value .\r\n");
                if (addGroupCondition)
                {
                  sb.Append($"  \t?subject :predicateSignature {predicateEvals.Key} .\r\n");
                }
                sb.Append($"  \t?subject <{predicateEval.PredicateUri}> \"{predicateValue[0].AsValuedNode().AsString()}\" .\r\n");
                sb.Append("}");
                var queryWithResultSet = Endpoint.QueryWithResultSet(sb.ToString());
                var xmlWriter = XmlWriter.Create($"G_{predicateEvals.Key}_{predicateValue[0].AsValuedNode().AsString()}.xml",
xmlWriterSettings);
                xmlWriter.WriteStartDocument();
                xmlWriter.WriteStartElement("exchange");
                xmlWriter.WriteAttributeString("xmlns", "xsi", null, "http://www.w3.org/2001/XMLSchema-instance");
                xmlWriter.WriteAttributeString("xsi", "noNamespaceSchemaLocation", null,
"http://download.autodesk.com/us/navisworks/schemas/nw-exchange-12.0.xsd");
                xmlWriter.WriteStartElement("findspec");
                xmlWriter.WriteAttributeString("mode", "all");
                xmlWriter.WriteAttributeString("disjoint", "0");
                xmlWriter.WriteStartElement("conditions");
                foreach (var result in queryWithResultSet)
                {
                  xmlWriter.WriteStartElement("condition");
                  xmlWriter.WriteAttributeString("test", "equals");
                  xmlWriter.WriteAttributeString("flags", "74");
                  xmlWriter.WriteStartElement("category");
                  xmlWriter.WriteStartElement("name");
                  xmlWriter.WriteAttributeString("internal", "LcOaPropOverrideCat");
                  xmlWriter.WriteValue("PCL");
                  xmlWriter.WriteEndElement();
                  xmlWriter.WriteEndElement();
                  xmlWriter.WriteStartElement("property");
                  xmlWriter.WriteStartElement("name");
                  xmlWriter.WriteAttributeString("internal", "PCL");
                  xmlWriter.WriteValue("PCL ID");
                  xmlWriter.WriteEndElement();
                  xmlWriter.WriteEndElement();
                  xmlWriter.WriteStartElement("value");
                  xmlWriter.WriteStartElement("data");
                  xmlWriter.WriteAttributeString("type", "wstring");
                  xmlWriter.WriteValue(result[0].AsValuedNode().AsString());
                  xmlWriter.WriteEndElement();
```

```
                        xmlWriter.WriteEndElement();
                        xmlWriter.WriteEndElement();
                    }
                    xmlWriter.WriteEndDocument();
                    xmlWriter.Close();
                }
            }
        }
    }
    Console.WriteLine("Press any key to exit...");
    Console.ReadKey();
}
private static double CalculateEntropy(List<long> values)
{
    double sumValue = values.Sum();
    var probabilities = new List<double>(values.Count);
    values.ForEach(v => probabilities.Add(v / sumValue));
    var entropy = -probabilities.Sum(p => p * Math.Log(p, 2));
    return entropy;
}
private static IEnumerable<PredicateEval> MaxEntropyDiffEval(List<PredicateEval> predicateEvals, int count)
{
    foreach (var predicateEval in predicateEvals)
    {
        predicateEval.Weight = CalculateEntropyDiff(predicateEval.EntropyGroup, predicateEval.EntropyDomain);
    }
    return predicateEvals.OrderByDescending(p => p.Weight).Take(count);
}
private static IEnumerable<PredicateEval> MaxEntropyDiffLessThanOneEval(List<PredicateEval> predicateEvals, int count)
{
    foreach (var predicateEval in predicateEvals)
    {
        predicateEval.Weight = CalculateEntropyDiff(predicateEval.EntropyGroup, predicateEval.EntropyDomain);
    }
    return predicateEvals.Where(p => p.Weight < 1).OrderByDescending(p => p.Weight).Take(count);
}
private static double CalculateEntropyDiff(double entropyGroup, double entropyDomain)
{
    if (entropyDomain < 0.01 || entropyGroup > entropyDomain)
    {
        return 0;
    }
    return (entropyDomain - entropyGroup) /
        (entropyDomain + entropyGroup);
}
private static IEnumerable<PredicateEval> TfIdfEval(List<PredicateEval> predicateEvals, int count)
{
    foreach (var predicateEval in predicateEvals)
    {
        predicateEval.Weight = CalculateTfIdf(predicateEval);
    }
    return predicateEvals.OrderByDescending(p => p.Weight).Take(count);
}
private static IEnumerable<PredicateEval> TfIdfEvalLessThanOne(List<PredicateEval> predicateEvals, int count)
{
    foreach (var predicateEval in predicateEvals)
    {
        predicateEval.Weight = CalculateTfIdf(predicateEval);
    }
    return predicateEvals.Where(p => p.Weight < 1).OrderByDescending(p => p.Weight).Take(count);
}
private static double CalculateTfIdf(PredicateEval predicateEval)
{
    //Retrieve all possible values and their count in the group
    var queryString = "PREFIX : <http://www.mali.ca/#>\r\n" +
                        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\r\n\r\n" +
                        "SELECT DISTINCT ?value (COUNT(?value) as ?count) \r\n" +
                        "WHERE {\r\n" +
                        $"  ?subject <{predicateEval.PredicateUri}> ?value.\r\n" +
                        $"  ?subject :predicateSignature {predicateEval.GroupSignature} .\r\n" +
                        "}\r\n" +
                        "GROUP BY ?value";
    var entropyResults = Endpoint.QueryWithResultSet(queryString);
    double tfidfSum = 0;
    long tfidfCount = 0;
    //Count of the value in group
    queryString = "PREFIX : <http://www.mali.ca/#>\r\n" +
                    "SELECT (COUNT(?subject) as ?count) \r\n" +
                    "WHERE {\r\n" +
                    $"  ?subject :predicateSignature {predicateEval.GroupSignature} .\r\n" +
                    "}\r\n";
    var countTerm = Endpoint.QueryWithResultSet(queryString);
    var groupItemCount = countTerm.Results.First()[0].AsValuedNode().AsInteger();
    foreach (var entropyResult in entropyResults)
```

```
            {
                try
                {
                    //Count of the value in group
                    queryString = "PREFIX : <http://www.mali.ca/#>\r\n" +
                                  "SELECT (COUNT(?subject) as ?count) \r\n" +
                                  "WHERE {\r\n" +
                                  $"  ?subject <{predicateEval.PredicateUri}> \"{entropyResult[0]}\".\r\n" +
                                  $"  ?subject :predicateSignature {predicateEval.GroupSignature} .\r\n" +
                                  "}\r\n";
                    countTerm = Endpoint.QueryWithResultSet(queryString);
                    var tf = (double)countTerm.Results.First()[0].AsValuedNode().AsInteger() / groupItemCount;
                    //Count of the groups contain the value
                    queryString = "PREFIX : <http://www.mali.ca/#>\r\n" +
                                  "SELECT (COUNT(DISTINCT ?value) as ?count) \r\n" +
                                  "WHERE {\r\n" +
                                  $"  ?subject <{predicateEval.PredicateUri}> \"{entropyResult[0]}\".\r\n" +
                                  "?subject :predicateSignature ?value .\r\n" +
                                  "}\r\n";
                    countTerm = Endpoint.QueryWithResultSet(queryString);
                    var idf = Math.Log10((double)_groupCount / countTerm.Results.First()[0].AsValuedNode().AsInteger());
                    tfidfSum += tf * idf;
                    tfidfCount++;
                }
                catch (Exception ex)
                {
                    Console.WriteLine(ex.Message);
                }
            }
            return tfidfSum / tfidfCount;
        }
    }
}
```

Appendix C

The shape recognition code

```
using Autodesk.Navisworks.Api;
using COMApi = Autodesk.Navisworks.Api.Interop.ComApi;
using ComBridge = Autodesk.Navisworks.Api.ComApi.ComApiBridge;
using Autodesk.Navisworks.Api.Plugins;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Windows.Forms;
using System.Runtime.Serialization.Formatters.Binary;
using System.Diagnostics;
using Autodesk.Navisworks.Api.Interop.ComApi;
using Autodesk.Navisworks.Api.ComApi;
using System.Reflection;
using System.Data;

namespace NavisWorks_Geometry
{
    //Using any .NET class inside the following class throws an exception UNLESS it is INSTALLED in GAC
    class CallbackGeomListener : InwSimplePrimitivesCB
    {
        public List<Point3d> points = new List<Point3d>();
        public List<Polyline> triangles = new List<Polyline>();
        public void Line(InwSimpleVertex v1, InwSimpleVertex v2)
        {
            var Nv1 = (Array)(object)v1.coord;
            var f11 = (float)(Nv1.GetValue(1));
            var f21 = (float)(Nv1.GetValue(2));
            var f31 = (float)(Nv1.GetValue(3));
            points.Add(new Point3d(f11, f21, f31));

            var Nv2 = (Array)(object)v2.coord;
            var f12 = (float)(Nv2.GetValue(1));
            var f22 = (float)(Nv2.GetValue(2));
    var f32 = (float)(Nv2.GetValue(3));
    points.Add(new Point3d(f12, f22, f32));
}

public void Point(InwSimpleVertex v1)
{
    var Nv1 = (Array)(object)v1.coord;
    var f11 = (float)(Nv1.GetValue(1));
    var f21 = (float)(Nv1.GetValue(2));
    var f31 = (float)(Nv1.GetValue(3));
    points.Add(new Point3d(f11, f21, f31));
}

public void SnapPoint(InwSimpleVertex v1)
{
    var Nv1 = (Array)(object)v1.coord;
    var f11 = (float)(Nv1.GetValue(1));
    var f21 = (float)(Nv1.GetValue(2));
    var f31 = (float)(Nv1.GetValue(3));
    points.Add(new Point3d(f11, f21, f31));
}

public void Triangle(InwSimpleVertex v1, InwSimpleVertex v2, InwSimpleVertex v3)
{
    var triangle = new Polyline(3);
    var Nv1 = (Array)(object)v1.coord;
    var f11 = (float)(Nv1.GetValue(1));
    var f21 = (float)(Nv1.GetValue(2));
    var f31 = (float)(Nv1.GetValue(3));
    triangle.AddVertex(new Point3d(f11, f21, f31));

    var Nv2 = (Array)(object)v2.coord;
    var f12 = (float)(Nv2.GetValue(1));
    var f22 = (float)(Nv2.GetValue(2));
    var f32 = (float)(Nv2.GetValue(3));
    triangle.AddVertex(new Point3d(f12, f22, f32));

    var Nv3 = (Array)(object)v3.coord;
    var f13 = (float)(Nv3.GetValue(1));
    var f23 = (float)(Nv3.GetValue(2));
    var f33 = (float)(Nv3.GetValue(3));
    triangle.AddVertex(new Point3d(f13, f23, f33));
```

```
            triangles.Add(triangle);
        }
    }

    [PluginAttribute("GeometryExtractor.MALI", "MALI", ToolTip = "Extract model item geometry", DisplayName = "Geometry Extractor")]
    [AddInPluginAttribute(AddInLocation.AddIn)]
    public class GeometryExtractor : AddInPlugin
    {
        private static readonly log4net.ILog log = log4net.LogManager.GetLogger(MethodBase.GetCurrentMethod().DeclaringType);
        public override int Execute(params string[] parameters)
        {
            try
            {
                var ds1 = new DataSet();
                DataTable t1;
                SQLiteDataAdapter dataAdapter;
                var recordsNo = 0;

                {
                    Hierarchy hierarchy = (Hierarchy)LogManager.GetRepository();
                    PatternLayout patternLayout = new PatternLayout();
                    patternLayout.ConversionPattern = "%date [%thread] %-5level %logger - %message%newline";
                    patternLayout.ActivateOptions();

                    FileAppender fileAppender = new FileAppender();
                    fileAppender.AppendToFile = false;
                    fileAppender.Layout = patternLayout;
                    fileAppender.File = Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location) + @"\log.txt";
                    fileAppender.ActivateOptions();
                    hierarchy.Root.AddAppender(fileAppender);
                    hierarchy.Root.Level = Level.Info;
                    hierarchy.Configured = true;
                }
                var selectedItems = new
ModelItemCollection(Autodesk.Navisworks.Api.Application.ActiveDocument.CurrentSelection.SelectedItems);

                if (selectedItems.Count < 1)
                {
                    MessageBox.Show("Please select model items first", "Selection error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
                    return 0;
                }
```

```
SQLiteConnection conn = new SQLiteConnection("data source=D:\\Mostafa\\Dropbox\\PhD\\ShapeRecognition\\FHSE.db3");
var commandText = "SELECT * FROM SteelItems";
dataAdapter = new SQLiteDataAdapter(commandText, conn);
dataAdapter.Fill(ds1);
t1 = ds1.Tables[0];
log.InfoFormat("Number of sections to be processed: {0}", selectedItems.Count);
var oFD = new OpenFileDialog
{
    Filter = "Section file (*.sec)| *.sec",
    Title = "Select sections file"
};
if (oFD.ShowDialog() != DialogResult.OK)
{
    return 0;
}

var stopWatch = new Stopwatch();
stopWatch.Start();

List<SteelSection> sections;
using (var fs = new FileStream(oFD.FileName, FileMode.Open, FileAccess.Read))
{
    var b = new BinaryFormatter();
    sections = (List<SteelSection>)b.Deserialize(fs);
}
var iterator = 0;
foreach (var selectedItem in selectedItems)
{
    try
    {
        log.InfoFormat("Processing Section #{0}", ++iterator);
        var collection = new ModelItemCollection {selectedItem};
        var oSel = ComBridge.ToInwOpSelection(collection);

        var callbkListener = new CallbackGeomListener();
        foreach (InwOaPath3 path in oSel.Paths())
        {
            foreach (InwOaFragment3 frag in path.Fragments())
            {
                frag.GenerateSimplePrimitives(nwEVertexProperty.eNORMAL, callbkListener);
            }
        }
```

```
List<Polyline> triangles = callbkListener.triangles;
double maxDist = 0;
double maxDiff = 0;
double minCoord = 0;
NormalAxis orientation = NormalAxis.Inclined;
foreach (var item in triangles)
{
    if (item.GetMaxLength() > maxDist)
    {
        maxDist = item.GetMaxLength();
        double diffX = item.MaxX - item.MinX;
        double diffY = item.MaxY - item.MinY;
        double diffZ = item.MaxZ - item.MinZ;
        if (diffX > diffY && diffX > diffZ)
        {
            orientation = NormalAxis.X;
            minCoord = item.MinX;
            maxDiff = diffX;
        }
        else if (diffY > diffX && diffY > diffZ)
        {
            orientation = NormalAxis.Y;
            minCoord = item.MinY;
            maxDiff = diffY;
        }
        else
        {
            orientation = NormalAxis.Z;
            minCoord = item.MinZ;
            maxDiff = diffZ;
        }

    }
}

var stepIncrement = .35;
var currentStep = .001;
var isContinue = true;
IEnumerable<Polyline> selectedTriangles = new List<Polyline>();
switch (orientation)
{
    case NormalAxis.Inclined:
        break;
    case NormalAxis.X:
        selectedTriangles = triangles.Where(t => Math.Abs(t.MaxX - minCoord) < currentStep);
        break;
    case NormalAxis.Y:
        selectedTriangles = triangles.Where(t => Math.Abs(t.MaxY - minCoord) < currentStep);
        break;
    case NormalAxis.Z:
        selectedTriangles = triangles.Where(t => Math.Abs(t.MaxZ - minCoord) < currentStep);
        break;
    default:
        break;
}

var previousCount = selectedTriangles.Count();

while (isContinue)
{
    currentStep += stepIncrement;
    if (currentStep > 1)
    {
        break;
    }
    switch (orientation)
    {
        case NormalAxis.Inclined:
            break;
        case NormalAxis.X:
            selectedTriangles = triangles.Where(t => Math.Abs(t.MaxX - minCoord) < currentStep);
            break;
        case NormalAxis.Y:
            selectedTriangles = triangles.Where(t => Math.Abs(t.MaxY - minCoord) < currentStep);
            break;
        case NormalAxis.Z:
            selectedTriangles = triangles.Where(t => Math.Abs(t.MaxZ - minCoord) < currentStep);
            break;
        default:
            break;
    }
    if (selectedTriangles.Count() == previousCount && previousCount > 0)
    {
        break;
    }
    previousCount = selectedTriangles.Count();
}

double area = selectedTriangles.Sum(t => t.GetArea());
area *= maxDiff / maxDist;
var distances = GetRandomDistances(selectedTriangles);
Histogram h1 = new Histogram(distances, 80);



var results = new Dictionary<string, double>();

IEnumerable<SteelSection> selectedSections = from t in sections
                                             where t.Area < area * 1.2 && t.Area > area * .8
                                             select t;

foreach (SteelSection item in selectedSections)
{
    var compDistances = GetRandomDistances(item.Triangles);
    Histogram h2 = new Histogram(compDistances, 80);
    var a1 = new double[h1.BucketCount];
    var a2 = new double[h1.BucketCount];
    for (var i = 0; i < h1.BucketCount; i++)
    {
        a1[i] = h1[i].Count;
        a2[i] = h2[i].Count;
    }
    results.Add(item.SectionName, Distance.Minkowski(2, a1, a2));
}

var sortedDict = (from entry in results orderby entry.Value select entry).Take(4);
var j = 1;
log.Info("\tWriting data property for the section");
foreach (var entry in sortedDict)
{
    AddDataProperty(selectedItem, "SteelSection", "Section" + j++, entry.Key);
}
var nameValue = "";
var sectionProp = "";
var pc = selectedItem.PropertyCategories.FindCategoryByName("LcOaNode"); if (pc != null)
{
```

```
                    var dp = pc.Properties.FindPropertyByName("LcOaSceneBaseUserName"); if (dp != null)
                    {
                        nameValue = dp.Value.ToString();
                        nameValue = nameValue.Substring(nameValue.LastIndexOf(':') + 1);
                    }
                }

                pc = selectedItem.PropertyCategories.FindCategoryByName("lcldrvm_props"); if (pc != null)
                {
                    var dp = pc.Properties.FindPropertyByName("lcldrvm_prop_spec_reference"); if (dp != null)
                    {
                        sectionProp = dp.Value.ToString();
                        sectionProp = sectionProp.Substring(sectionProp.LastIndexOf(':') + 1);
                    }
                }

                t1.Rows.Add(1, nameValue, selectedItem.BoundingBox().Min.X, selectedItem.BoundingBox().Min.Y,
selectedItem.BoundingBox().Min.Z, selectedItem.BoundingBox().Max.X, selectedItem.BoundingBox().Max.Y,
selectedItem.BoundingBox().Max.Z, sortedDict.ElementAt(0).Key, sectionProp);

                if (recordsNo++ > 50000)
                {
                    SQLiteCommandBuilder builder1 = new SQLiteCommandBuilder(dataAdapter);
                    builder1.GetInsertCommand();
                    dataAdapter.Update(t1);
                    recordsNo = 0;
                }
            }

            catch (Exception ex)
            {
                log.Error("Error while processing the section", ex);
            }
        }

        SQLiteCommandBuilder builder = new SQLiteCommandBuilder(dataAdapter);
        builder.GetInsertCommand();
        dataAdapter.Update(t1);
        stopWatch.Stop();
        log.InfoFormat("Time required to process all sections: {0}", stopWatch.Elapsed);

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    return 0;
}

public IEnumerable<double> GetRandomDistances(IEnumerable<Polyline> triangles)
{
    //Calculate the total surface area
    double totalarea = triangles.Sum(x => x.GetArea());
    List<Point3d> points = new List<Point3d>(2500);
    //We need 1,000,000 distance therefore I will generate 2000 points (2000 C 2 = 1999000)
    foreach (Polyline triangle in triangles)
    {
        var n = (int)(triangle.GetArea() / totalarea * 350);
        points.AddRange(triangle.GetPointsInside(n));
    }
    var distancesList = GenerateDistances(points, 50000); //1048576 = 1024^2
    return distancesList;
}

static readonly Random rnd = new Random();

public static IEnumerable<double> GenerateDistances(IList<Point3d> pointList, int count)
{
    var distances = new List<double>(count);
    for (var i = 0; i < count; i++)
    {
        var p1Index = rnd.Next(pointList.Count);
        var p2Index = rnd.Next(pointList.Count);
        distances.Add(pointList[p1Index].DistanceTo(pointList[p2Index]));
    }
    return distances;
}

internal static void AddDataProperty(ModelItem item, string tabName, string propertyName, string propertyValue)
{
    try
    {
        var state = ComApiBridge.State;
```

```
                InwOaProperty newProperty = state.ObjectFactory(nwEObjectType.eObjectType_nwOaProperty, null, null);
                newProperty.name = propertyName;
                newProperty.value = propertyValue;
                InwOaPropertyVec newPropertyCategory = state.ObjectFactory(nwEObjectType.eObjectType_nwOaPropertyVec, null, null);
                newPropertyCategory.Properties().Add(newProperty);

                var miPath = ComApiBridge.ToInwOaPath(item);
                var PropertiesCategories = (InwGUIPropertyNode2)state.GetGUIPropertyNode(miPath, true);
                int index = 0, i = 0;

                foreach (InwGUIAttribute2 propertyCategory in PropertiesCategories.GUIAttributes())
                {
                    if (propertyCategory.UserDefined)
                    {
                        index += 1;
                        if (propertyCategory.ClassUserName == tabName)
                        {
                            i = index;
                            foreach (InwOaProperty property in propertyCategory.Properties())
                            {
                                InwOaProperty tempProperty = state.ObjectFactory(nwEObjectType.eObjectType_nwOaProperty, null, null);
                                tempProperty.name = property.name;
                                tempProperty.value = property.value;
                                newPropertyCategory.Properties().Add(tempProperty);
                            }
                        }
                    }
                }
                PropertiesCategories.SetUserDefined(i, tabName, tabName, newPropertyCategory);
            }
            catch (Exception ex)
            {
                log.Error("Error while writing data property", ex);
            }
        }
    }
}
```

Appendix D

The proposed Ontology (turtle format)

```
# baseURI: http://visualization.mali.ca/VisualizationOnt
# imports: http://ifcowl.openbimstandards.org/IFC4
# imports: http://www.w3.org/2006/time#
# prefix: Vont

@prefix IFC4: <http://ifcowl.openbimstandards.org/IFC4#> .
@prefix IFC4_ADD1: <http://www.buildingsmart-tech.org/ifcOWL/IFC4_ADD1#> .
@prefix Vont: <http://visualization.mali.ca/VisualizationOnt#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix spin: <http://spinrdf.org/spin#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

IFC4:IfcElement
  rdfs:subClassOf Vont:ModelObject ;
.
<http://visualization.mali.ca/VisualizationOnt>
  rdf:type owl:Ontology ;
  spin:imports <http://topbraid.org/spin/owlrl-all> ;
  owl:imports <http://ifcowl.openbimstandards.org/IFC4> ;
  owl:imports time: ;
  owl:versionInfo "Created with TopBraid Composer" ;
.
Vont:AngleUnit
  rdf:type rdf:Property ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:Units
          Vont:Camera
          Vont:ModelObject
          Vont:ModelJoint
```

```
        ) ;
    ] ;
  rdfs:label "Specify angle unit (degree, radian)" ;
  rdfs:range [
      rdf:type rdfs:Datatype ;
      owl:oneOf (
          "degree"
          "radian"
        ) ;
    ] ;
.
Vont:AngularVelocity
  rdf:type rdf:Property ;
  rdfs:comment "Defines the angular velocity of the object relative to the reference frame used to capture the data in the format \"Vx Vy
Vz\"" ;
  rdfs:domain Vont:ObjectTimeStamp ;
  rdfs:range xsd:string ;
.
Vont:Author
  rdf:type rdf:Property ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:mpcMission
          Vont:InitSection
        ) ;
    ] ;
  rdfs:label "The author of the object" ;
  rdfs:range xsd:string ;
.
Vont:CG
  rdf:type rdf:Property ;
  rdfs:comment "Defines the location of the center of gravity of the object relative to the design reference frame of the object as a function
of time in the format \"X Y Z\"" ;
  rdfs:domain Vont:ObjectTimeStamp ;
  rdfs:range xsd:string ;
.
Vont:Camera
  rdf:type owl:Class ;
  rdfs:label "Define camera for the environment" ;
.
Vont:CameraFieldView
```

```
  rdf:type rdf:Property ;
  rdfs:domain Vont:Camera ;
  rdfs:label "Define field view of a camera (0, 180) in degrees or (0, PI) in radians" ;
  rdfs:range xsd:string ;
.
Vont:DistanceUnit
  rdf:type rdf:Property ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:Units
          Vont:Camera
          Vont:ModelGeometry
          Vont:ModelObject
          Vont:ModelJoint
          Vont:Offset
          Vont:Pin
        ) ;
    ] ;
  rdfs:label "Unit of the distance" ;
  rdfs:range [
      rdf:type rdfs:Datatype ;
      owl:oneOf (
          "millimeter"
          "centimeter"
          "meter"
          "kilometer"
          "AU"
          "inch"
          "foot"
          "yard"
          "mile"
        ) ;
    ] ;
.
Vont:Environment
  rdf:type owl:Class ;
  rdfs:label "Environment of the scene" ;
.
Vont:EnvrionmentType
  rdf:type rdf:Property ;
  rdfs:domain Vont:Environment ;
  rdfs:range [
    rdf:type rdfs:Datatype ;
    owl:oneOf (
      "Earth"
      "Moon"
      "Mars"
      "Space"
      "Custom"
     ) ;
   ] ;
.
Vont:EventTimeStamp
  rdf:type owl:Class ;
  rdfs:subClassOf Vont:TimeStamped ;
.
Vont:FilePath
  rdf:type rdf:Property ;
  rdfs:domain [
    rdf:type owl:Class ;
    owl:unionOf (
      Vont:Telemetry
      Vont:Environment
      Vont:ModelObjects
      Vont:ModelGeometry
     ) ;
   ] ;
  rdfs:label "This property might represent the file or the base path" ;
  rdfs:range xsd:string ;
.
Vont:ForceUnit
  rdf:type rdf:Property ;
  rdfs:domain Vont:Units ;
  rdfs:label "Specify force unit (newton, pound)" ;
  rdfs:range [
    rdf:type rdfs:Datatype ;
    owl:oneOf (
      "newton"
      "pound"
     ) ;
   ] ;
.
Vont:ForwardDirection
```

```
  rdf:type rdf:Property ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:Environment
          Vont:Camera
        ) ;
    ] ;
  rdfs:label "Specify the forward direction in the format \"x y z\"" ;
  rdfs:range xsd:string ;
.
Vont:ID
  rdf:type rdf:Property ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:ModelObject
          Vont:metaData
          Vont:ModelJoint
        ) ;
    ] ;
  rdfs:label "A unique ID" ;
  rdfs:range xsd:string ;
.
Vont:InitSection
  rdf:type owl:Class ;
  rdfs:label "The init section in data file" ;
.
Vont:JointType
  rdf:type rdf:Property ;
  rdfs:domain Vont:ModelJoint ;
  rdfs:label "Unit of the distance" ;
  rdfs:range [
      rdf:type rdfs:Datatype ;
      owl:oneOf (
          "rotational"
          "translational"
        ) ;
    ] ;
.
Vont:JointValue
  rdf:type rdf:Property ;
  rdfs:comment "Defines the joint value at specific time stamp" ;
  rdfs:domain Vont:ObjectTimeStamp ;
  rdfs:range xsd:double ;
.
Vont:MetaType
  rdf:type rdf:Property ;
  rdfs:domain Vont:metaData ;
  rdfs:range [
      rdf:type rdfs:Datatype ;
      owl:oneOf (
          "string"
          "double"
        ) ;
    ] ;
.
Vont:ModelGeometry
  rdf:type owl:Class ;
  rdfs:label "A geometry of the ModelObject" ;
.
Vont:ModelJoint
  rdf:type owl:Class ;
  rdfs:comment "Define a movement relation between two objects or joints" ;
.
Vont:ModelObject
  rdf:type owl:Class ;
  rdfs:label "An object that will be shown in the visualizer" ;
.
Vont:ModelObjects
  rdf:type owl:Class ;
  rdfs:label "A container for the objects to be modelled" ;
.
Vont:Name
  rdf:type rdf:Property ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:Scene
          Vont:Telemetry
          Vont:Camera
          Vont:ModelObject
          Vont:InitSection
          Vont:ModelJoint
        ) ;
    ] ;
  rdfs:label "Name of the item" ;
  rdfs:range xsd:string ;
.
Vont:ObjectTimeStamp
  rdf:type owl:Class ;
  rdfs:subClassOf Vont:TimeStamped ;
.
Vont:OfInterest
  rdf:type rdf:Property ;
  rdfs:comment "Identification of the object as an object of interest 0 (False), 1 (True)" ;
  rdfs:domain Vont:ModelObject ;
  rdfs:range xsd:integer ;
.
Vont:Offset
  rdf:type owl:Class ;
  rdfs:comment "An object offset" ;
.
Vont:Pin
  rdf:type owl:Class ;
  rdfs:comment "A joint pin" ;
.
Vont:Position
  rdf:type rdf:Property ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:Camera
          Vont:ModelGeometry
          Vont:ObjectTimeStamp
          Vont:Offset
          Vont:Pin
        ) ;
    ] ;
  rdfs:label "Specify the coordinates in the format \"x y z\"" ;
  rdfs:range xsd:string ;
.
Vont:Quaternion
  rdf:type rdf:Property ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:ModelGeometry
          Vont:ObjectTimeStamp
          Vont:Offset
          Vont:Pin
        ) ;
    ] ;
  rdfs:label "Specify the quaternion in the format \"x y z w\"" ;
  rdfs:range xsd:string ;
.
Vont:Scale
  rdf:type rdf:Property ;
  rdfs:domain Vont:ModelGeometry ;
  rdfs:label "Scale of the item" ;
  rdfs:range xsd:integer ;
.
Vont:Scale3Dir
  rdf:type rdf:Property ;
  rdfs:comment "Defines a scale value to be applied to the object at this time in the format \"Sx Sy Sz\"" ;
  rdfs:domain Vont:ObjectTimeStamp ;
  rdfs:range xsd:string ;
.
Vont:Scene
  rdf:type owl:Class ;
  rdfs:label "A scene of a mission" ;
.
Vont:Telemetry
  rdf:type owl:Class ;
  rdfs:label "Telemetries in mission file" ;
.
Vont:TelemetryFormat
  rdf:type rdf:Property ;
  rdfs:domain Vont:Telemetry ;
  rdfs:label "Defines the format of the data set" ;
  rdfs:range [
      rdf:type rdfs:Datatype ;
      owl:oneOf (
          "MPC"
          "MPC2"
          "MPC3"
        ) ;
    ] ;
```

```
.
Vont:TelemetryType
 rdf:type rdf:Property ;
 rdfs:domain Vont:Telemetry ;
 rdfs:label "Specify if the telemetry is file or network" ;
 rdfs:range [
   rdf:type rdfs:Datatype ;
   owl:oneOf (
     "file"
     "network"
    ) ;
  ] ;
.
Vont:TimeEpoch
 rdf:type rdf:Property ;
 rdfs:domain [
   rdf:type owl:Class ;
   owl:unionOf (
     Vont:mpcMission
     Vont:Telemetry
     Vont:InitSection
    ) ;
  ] ;
 rdfs:label "The time epoch for the mission" ;
 rdfs:range xsd:dateTime ;
.
Vont:TimeStamped
 rdf:type owl:Class ;
 rdfs:comment "A parent class for all objects with time stamp" ;
.
Vont:TimeUnit
 rdf:type rdf:Property ;
 rdfs:domain Vont:Units ;
 rdfs:label "Unit of the time" ;
 rdfs:range [
   rdf:type rdfs:Datatype ;
   owl:oneOf (
     "millisecond"
     "second"
     "minute"
     "hour"
     "day"
```

```
        ) ;
    ] ;
.
Vont:UnitLabel
  rdf:type rdf:Property ;
  rdfs:domain Vont:metaData ;
  rdfs:range xsd:string ;
.
Vont:Units
  rdf:type owl:Class ;
  rdfs:label "Units of the scene" ;
.
Vont:UpDirection
  rdf:type rdf:Property ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:Environment
          Vont:Camera
        ) ;
    ] ;
  rdfs:label "Specify the up direction in the format \"x y z\"" ;
  rdfs:range xsd:string ;
.
Vont:Value
  rdf:type rdf:Property ;
  rdfs:comment "A string value for an object" ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:metaDataTimeStamp
          Vont:EventTimeStamp
        ) ;
    ] ;
  rdfs:range xsd:string ;
.
Vont:Velocity
  rdf:type rdf:Property ;
  rdfs:comment "Defines the velocity of the object relative to the reference frame used to capture the data in the format \"Vx Vy Vz\"" ;
  rdfs:domain Vont:ObjectTimeStamp ;
  rdfs:range xsd:string ;
.
```

```
Vont:Version
  rdf:type rdf:Property ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:mpcMission
          Vont:mpcData
        ) ;
    ] ;
  rdfs:label "the version of the object" ;
  rdfs:range xsd:string ;
.
Vont:Visibility
  rdf:type rdf:Property ;
  rdfs:comment "Provides an ability to hide or show the object, 0 (hide), 1 (show)" ;
  rdfs:domain Vont:ObjectTimeStamp ;
  rdfs:range xsd:int ;
.
Vont:hasCamera
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:Environment ;
  rdfs:label "State cameras for the item" ;
  rdfs:range Vont:Camera ;
.
Vont:hasEnvironment
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:Scene ;
  rdfs:label "Used to state the environment of the scene" ;
  rdfs:range Vont:Environment ;
.
Vont:hasInitSection
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:mpcData ;
  rdfs:label "Captures the relation between mpcData and InitSection" ;
  rdfs:range Vont:InitSection ;
.
Vont:hasMetaData
  rdf:type owl:ObjectProperty ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:InitSection
          Vont:ModelObject
          Vont:metaDataTimeStamp
        ) ;
    ] ;
  rdfs:range Vont:metaData ;
.
Vont:hasMetaDataTimeStamp
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:ObjectTimeStamp ;
  rdfs:range Vont:metaDataTimeStamp ;
.
Vont:hasModelGeometry
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:ModelObject ;
  rdfs:label "State the geometry of the ModelObject" ;
  rdfs:range Vont:ModelGeometry ;
.
Vont:hasModelObject
  rdf:type owl:ObjectProperty ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:ModelObjects
          Vont:InitSection
          Vont:ObjectTimeStamp
          Vont:ModelObject
          Vont:ModelJoint
        ) ;
    ] ;
  rdfs:label "Relate Model objects and joints to a container" ;
  rdfs:range [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:ModelObject
          Vont:ModelJoint
        ) ;
    ] ;
.
Vont:hasModelObjects
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:Scene ;
  rdfs:label "State the model objects of the scene" ;
  rdfs:range Vont:ModelObjects ;
.
Vont:hasOffset
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:ModelObject ;
  rdfs:label "Captures the relation betwen object and offset" ;
  rdfs:range Vont:Offset ;
.
Vont:hasParent
  rdf:type rdf:Property ;
  rdfs:domain Vont:ObjectTimeStamp ;
  rdfs:range xsd:string ;
.
Vont:hasPin
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:ModelJoint ;
  rdfs:label "Captures the relation betwen object and pin" ;
  rdfs:range Vont:Pin ;
.
Vont:hasScene
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:mpcMission ;
  rdfs:label "Captures the relation betwen mision and scene" ;
  rdfs:range Vont:Scene ;
.
Vont:hasTelemetry
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:mpcMission ;
  rdfs:range Vont:Telemetry ;
.
Vont:hasTimeInstant
  rdf:type owl:ObjectProperty ;
  rdfs:domain Vont:TimeStamped ;
  rdfs:range time:Instant ;
.
Vont:hasUnit
  rdf:type owl:ObjectProperty ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          Vont:Scene
          Vont:InitSection
      ) ;
    ] ;
  rdfs:label "Used to state if item has a unit" ;
  rdfs:range Vont:Units ;
.
Vont:isStatic
  rdf:type rdf:Property ;
  rdfs:domain Vont:ModelObject ;
  rdfs:label "0 (False), 1 (True)" ;
  rdfs:range xsd:integer ;
.
Vont:metaData
  rdf:type owl:Class ;
.
Vont:metaDataTimeStamp
  rdf:type owl:Class ;
  rdfs:subClassOf Vont:TimeStamped ;
.
Vont:mpcData
  rdf:type owl:Class ;
  rdfs:label "The parent class of the data file" ;
.
Vont:mpcMission
  rdf:type owl:Class ;
  rdfs:label "A DON mission" ;



.
```

Appendix E

A sample of generated triples by the simulation model

```
# imports: http://visualization.mali.ca/VisualizationOnt

@prefix : <http://visualization.mali.ca/VisualizationInstances#> .
@prefix Vont: <http://visualization.mali.ca/VisualizationOnt#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix spin: <http://spinrdf.org/spin#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://visualization.mali.ca/Simulation>
  rdf:type owl:Ontology ;
  owl:imports <http://visualization.mali.ca/VisualizationOnt> ;

:0bcd812097e14a08ac8817cc1c9ecc03 Vont:Visibility 0 ;
  Vont:hasModelObject :qwe;
  Vont:hasTimeInstant :0bcd812097e14a08ac8817cc1c9ecc03_TimeInstant;
  a Vont:ObjectTimeStamp.

:0bcd812097e14a08ac8817cc1c9ecc03_TimeInstant a time:Instant;
  time:inTimePosition :0bcd812097e14a08ac8817cc1c9ecc03_TimePosition.

:0bcd812097e14a08ac8817cc1c9ecc03_TimePosition a time:TimePosition;
  time:numericPosition "0"^^time:Number.

:e9992566ea494417b631ae9ade4e3c7f Vont:Visibility 1 ;
  Vont:hasModelObject :qwe;
  Vont:hasTimeInstant :e9992566ea494417b631ae9ade4e3c7f_TimeInstant;
  a Vont:ObjectTimeStamp.
```

```
:e9992566ea494417b631ae9ade4e3c7f_TimeInstant a time:Instant;
  time:inTimePosition :e9992566ea494417b631ae9ade4e3c7f_TimePosition.

:e9992566ea494417b631ae9ade4e3c7f_TimePosition a time:TimePosition;
  time:numericPosition "259200"^^time:Number.
.
:Instant_qwe_0
  rdf:type time:Instant ;
  time:inTimePosition :TimePosition_qwe_0 ;
.
:EventTimeStamp_qwe_0
  rdf:type Vont:EventTimeStamp ;
  Vont:hasTimeInstant :Instant_qwe_0 ;
.
:TimePosition_qwe_0
  rdf:type time:TimePosition ;
  time:numericPosition "0"^^time:Number ;
.
:ObjectTimeStamp_qwe_0
  rdf:type Vont:ObjectTimeStamp ;
  Vont:Position "0 0 0" ;
  Vont:hasModelObject :qwe ;
  Vont:hasTimeInstant :Instant_qwe_0 ;
.
:Instant_qwe_86.4
  rdf:type time:Instant ;
  time:inTimePosition :TimePosition_qwe_86.4 ;
.
:EventTimeStamp_qwe_86.4
  rdf:type Vont:EventTimeStamp ;
  Vont:hasTimeInstant :Instant_qwe_86.4 ;
.
:TimePosition_qwe_86.4
  rdf:type time:TimePosition ;
  time:numericPosition "86.4"^^time:Number ;
.
:ObjectTimeStamp_qwe_86.4
  rdf:type Vont:ObjectTimeStamp ;
  Vont:Position "12 0 0" ;
  Vont:hasModelObject :qwe ;
  Vont:hasTimeInstant :Instant_qwe_86.4 ;
```

Appendix F

Sample Mission File

```
<?xml version="1.0" encoding="utf-8"?>
<mpcMission version="3.0" epoch="2016-09-20T15:10:03.00Z" author="Mostafa Ali">
    <note>This is a sample mission file to test DON3.1</note>
    <scene name="InitialScene">
        <units time="minute" distance="meter" angle="radian" force="newton"/>
        <environment type="Earth" up = "1 0 0" forward = "0 1 0">
        </environment>
        <objects baseFilePath="data/models/">
            <object id="Crane" name = "Crane" isStatic="0">
                <geometry>
                    <model pos = "0 0 0" filePath = "Base.obj"/>
                </geometry>
                <joint id="BasePivot" type="rotational" name="Base Pivot" >
                    <pin unitDistance="meter" pos="0 0 0" quat="1 0 0 1"/>
                    <object id="MovableBase" name = "Movable Base" isStatic="1">
                        <offset unitDistance="meter" pos="0 0 0" quat="-1 0 0 1"/>
                        <geometry>
                            <model pos = "0.0 0.0 6.49" quat = "0 0 0 1" filePath = "BaseMovable.obj"/>
                        </geometry>
                    </object>
                    <joint id = "HookPivot" type = "translational" name = "Hook Pivot">
                        <pin unitDistance="meter" pos="0 0 0" quat="0 0 0 1"/>
                        <object id = "Hook" name = "Hook" isStatic = "0">
                            <offset unitDistance="meter" pos="0 0 0" quat="-1 0 0 1"/>
                            <geometry>
                                <model pos = "0 0 0" quat = "0 0 0 1" filePath = "Hook2.obj"/>
                            </geometry>
                        </object>
                    </joint>
                </joint>
            </object>
            <object id="Module" name = "Module" isStatic="0">
```

```
                <geometry>
                    <model pos = "0 0 0" filePath = "Module.obj"/>
                </geometry>
            </object>
            <object id="Plane1" name = "Plane1" isStatic="1">
                <geometry>
                    <model pos = "0 0 0" scale = "1000" filePath = "Plane.obj"/>
                </geometry>
            </object>
        </objects>
    </scene>
    <telemetries>
        <telemetry name ="InitialTelemetry" type="file" format="MPC3" epoch="2016-09-20T15:10:03.00Z"
source="telemetry/DataFileSample2.mpc3"/>
    </telemetries>
</mpcMission>
```

Appendix G

Sample Data File

```
<?xml version="1.0" encoding="utf-8"?>
<mpcData version="3.0">
        <init name="InitialInit1" author="Mostafa Ali">
                <units time="seconds" distance="meter" angle="degree" force="newton"/>
                <object id="Crane" ofInterest="1"/>
                <object id="Module" ofInterest="1"/>
                <object id="Plane1" ofInterest="1"/>
                <object id="Hook" />
                <joint unitAngle="degree" id="BasePivot" />
                <joint unitAngle="meter" id="HookPivot" />
        </init>
        <time value="0">
                <event>Mission Start</event>
                <object id="Crane" pos = "0 0 0" />
                <object id="Hook" pos = "0 0 0" />
                <object id="Module" pos = "0 -700 -75" />
        </time>
        <time value="10">
                <event>Move crane to pickup position</event>
                <object id="Crane" pos = "0 -750 0" />
        </time>
        <time value="20">
                <event>Lower the hook</event>
                <joint id="HookPivot" value ="-40" />
        </time>
        <time value="21">
                <event>hook the hook</event>
                <object id = "Module" parent = "Hook"  pos = "0 0 0" />
        </time>
        <time value="25">
                <event>wait</event>
        </time>
        <time value="35">
                <event>Lift the load</event>
                <joint id="HookPivot" value ="0" />
        </time>
        <time value="45">
                <event>Swing and start moving</event>
                <joint id="BasePivot" value ="180" />
                <object id = "Crane" pos = "0 -400 0" />
        </time>
        <time value="60">
                <event>Move to drop point</event>
                <object id = "Module" parent = "Hook"  />
                <object id="Crane" pos = "0 400 0" />
        </time>
        <time value="70">
                <event>Drop the load</event>
                <joint id="HookPivot" value ="-40" />
        </time>
        <time value="71">
                <event>Drop the load</event>
                <object id = "Module" parent = "" />
        </time>
        <time value="75">
                <event>Unhook the load</event>
                <object id = "Module" parent = "" pos = "2 440 -75"/>
        </time>
        <time value="80">
                <event>Lift the hook</event>
                <joint id="HookPivot" value ="0" />
        </time>
</mpcData>
```